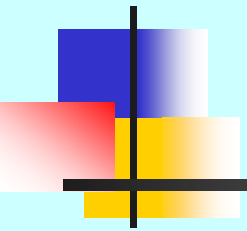
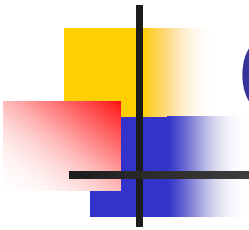


Chapter 4: Cryptography



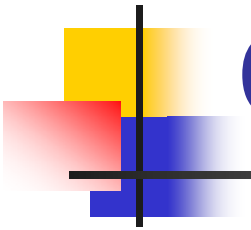
OBJECTIVES

- Categories of Cryptography
- Cryptography terminology
- Cryptographic algorithm types
- Symmetric key cryptosystems
- Asymmetric key cryptosystems
- Cryptography protocols

Categories of Cryptography

- Physical : refers to any method that doesn't alter the value using a mathematical process
- Mathematical :deals with using mathematical processes on characters or messages
- Quantum (classified): makes use of the fact that measuring quantum properties can change the state of a system. This implies that someone eavesdropping on a secret communication will destroy the message making it possible to devise a secure communication protocol.

Approaches to Secure Communications

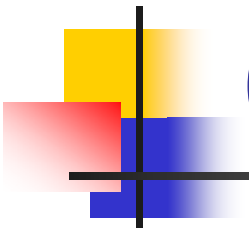


- Cryptography :
  - “Hiddenwriting”
  - Hides the meaning of the message
- Steganography:
  - “Covered writing”
  - Hides the existence of a message

Cryptography Terminology

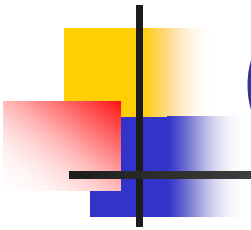
- Cryptography: science of transforming information so it is secure while being transmitted or stored
- Cryptology : the mathematics behind cryptography
- Steganography: attempts to hide existence of data
- Encryption: changing the original text to a secret message using cryptography

Cryptography Terminology (cont.)



- Decryption: reverse process of encryption
- Algorithm: process of encrypting and decrypting information based on a mathematical procedure
- Key: value used by an algorithm to encrypt or decrypt a message, and only known to sender/receiver

Cryptography Terminology (cont.)



- Weak key: mathematical key that creates a detectable pattern or structure
- Plaintext: original unencrypted information (also known as clear text)
- Cipher: encryption or decryption algorithm tool used to create encrypted or decrypted text
- Ciphertext: data that has been encrypted by an encryption algorithm

Cryptography Terminology (cont.)

- Cryptanalysis
  - The process of trying to break a cryptosystem
  - Finding the meaning of an encrypted message without being given the key.
  - Most cryptosystems are breakable, some just cost more than others. The job of a cryptosystem is to make the cost infeasible.

Cryptography Terminology (continued)

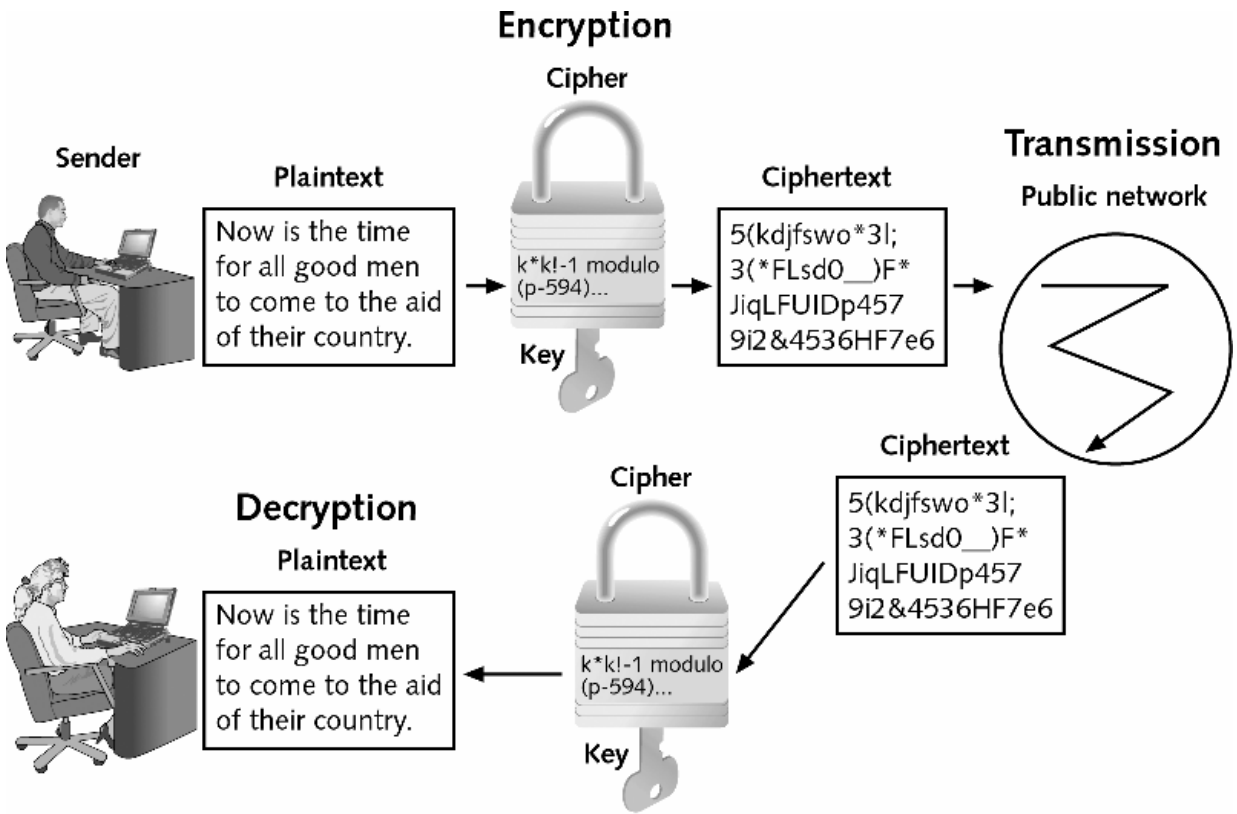
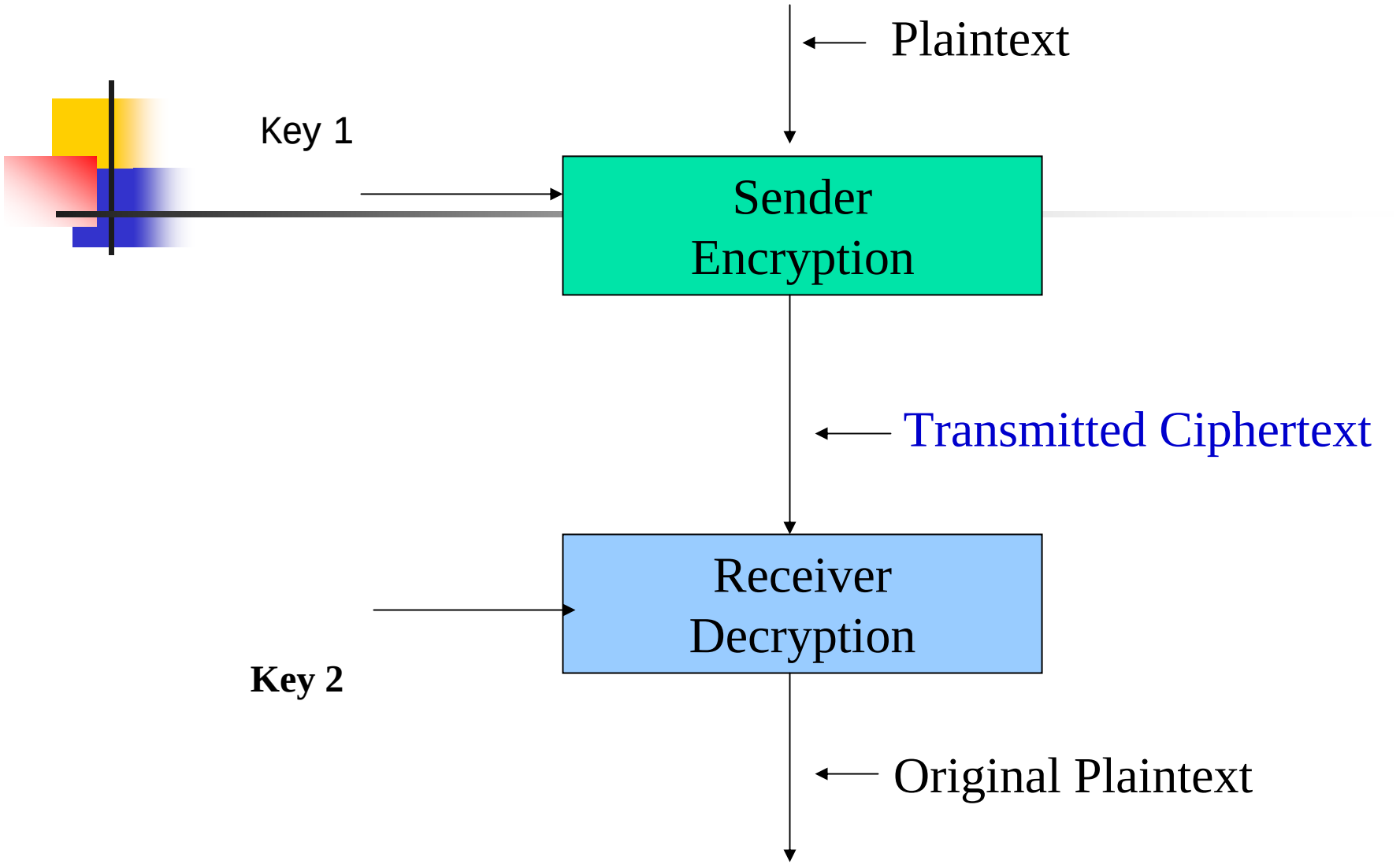


Figure 1 Cryptography


Plaintext
Key 1
Sender
Encryption
Transmitted Ciphertext
Receiver
Decryption
Key 2
Original Plaintext

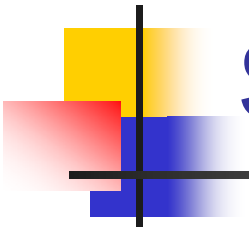
Encryption Algorithm Types

- Algorithms in which the two keys Key 1 and Key 2 are the same are often called
  - Symmetric or
  - Private-key algorithms (since the key needs to be kept private)
- Algorithms in which the two keys are different are often called
  - Asymmetric or
  - Public-key algorithms (since either key 1 or key 2 can be made public depending on the application)

Cryptographic Systems Categorization According to :

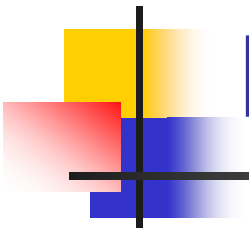
- The operation used in encryption
  - Substitution: each element in the plaintext is mapped into another element
  - Transposition : The elements in the plaintext are rearranged
- The number of keys used:
  - Symmetric (Private-Key) : Both the sender and receiver use the same key
  - Asymmetric (Public Key) : sender and receiver use different keys
- The way the plaintext is processed :
  - Block cipher : inputs are processed one block at a time, producing a corresponding output block
  - Stream cipher : inputs are processed continuously, producing one element at a time.

Advantages of Symmetric Key Systems



- Encryption and authentication performed in a single operation
- Well-known (and trusted) ones perform faster than asymmetric key systems
- Doesn’t require any centralized authority
  - Though key servers help a lot

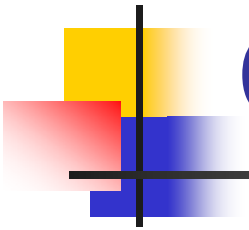
Disadvantages of Symmetric Key Systems



- Encryption and authentication performed in a single operation
  - Makes a signature more difficult
- Non-repudiation hard without servers
- Key distribution can be a problem
- Scaling

Requirements for Secure Symmetric Encryption

- A strong encryption algorithm
- A secret key known to sender/receiver
  - Y = Ek (X) , where X :plaintext , Y : ciphertext
  - X = Dk(Y)
  - Assume encryption algorithm is known
  - Implies a secure channel to distribute key



Commonly used Names

- Alice : initiates a message or protocol
- Bob : second participant
- Carol : third participant
- Trent : trusted middleman
- Eve : eavesdropper
- Mallory : malicious active attacker

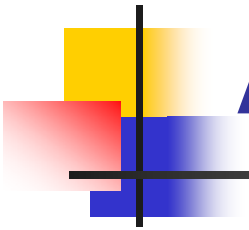
Encryption Security

- Truly Secure : an algorithm is truly secure if encrypted messages cannot be decoded without the key
  - In 1943 Shannon proved that for something to be truly secure, the key must be as long as the message.
  - An example of that is the one-time-pad in which the encoder and decoder both have the same random key which is the same length as the message. Each random key can only be used once

- Security Based on Computational Cost : Here something is secure if it is “infeasible” to break –meaning that no polynomial time algorithm can decode the message

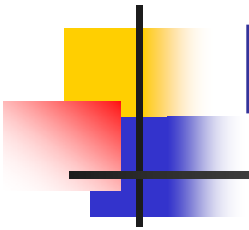
How Cryptography Protects

Protection	Description
Confidentiality	Allow only authorized users to access the information
Authentication	Verify who the sender was and trust the sender is who they claim to be
Integrity	Trust the information has not been altered
Nonrepudiation	Ensure that the sender or receiver cannot deny that a message was sent or received
Access Control	Restrict availability to information



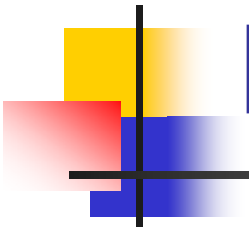
Attacks

- Cryptanalytic Attacks : depends on the nature of the encryption algorithm used
  - Uses information such as plaintext/ciphertext pairs to deduce the key
- Brute Force : try all possible keys – depends on the key length



Forms of Cryptanalysis

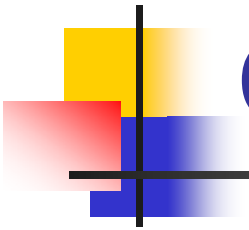
- Analyze an encrypted message and deduce its contents
- Analyze one or more encrypted message to find a common key
- Analyze a cryptosystem to find fundamental flaw



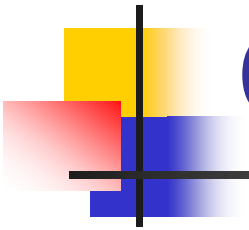
Breaking Cryptosystems

- Most cryptosystems are breakable
- Some just cost more to break than others
- The job of the cryptosystem is to make the cost infeasible
  - Or incommensurate with the benefits extracted

Types of Attacks on Cryptosystems

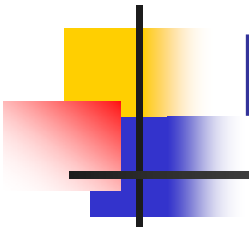


- Ciphertext only
- Known plaintext
- Chosen plaintext
  - Differential cryptanalysis
- Algorithm and ciphertext



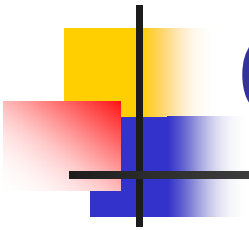
Ciphertext Only

- No prior knowledge of plaintext
- Or details of algorithm
- Must work with probability distributions, patterns of common characters, etc.
- Hardest type of attack



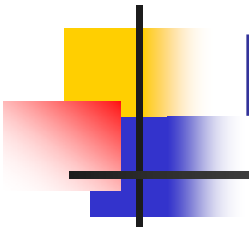
Known Plaintext

- Full or partial
- Cryptanalysis has matching sample of ciphertext and plaintext
- Or may know something about what ciphertext represents
  - E.g. an IP packet with its headers



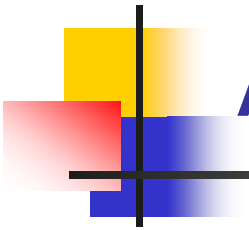
Chosen Plaintext

- Cryptanalyst can submit chosen samples of plaintext to the cryptosystem
- And recover the resulting ciphertext
- Clever choices of plaintext may reveal many details



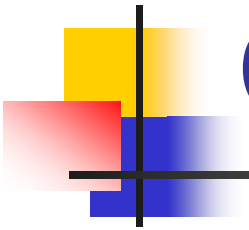
Differential Cryptanalysis

- Iteratively choose plaintexts that differ slightly in carefully chosen ways
- A good crypto algorithm should produce results that don’t help analysis
- But some crypto algorithms are vulnerable to this attack



Algorithm and Ciphertext

- Cryptanalyst knows the algorithm and has a sample of ciphertext
- But not the key, and may not get any more similar ciphertext
- Can use “exhaustive” runs of algorithm against guesses at plaintext
- Password guessers often work this way



Overview

- Classical Cryptography : Symmetric (Private-Key) Algorithms
  - Caesar cipher
  - Vigénere cipher
  - DES (Block cipher)
  - AES
  - IDEA
  - Blowfish , ...etc
- Public Key Cryptography
  - Diffie-Hellman
  - RSA
  - ElGamal
  - Elliptic curve

SYMMETRIC KEY SYSTEMS

Symmetric Encryption Algorithms

- Most common type of cryptographic algorithm (also called private key cryptography)
- Use a single key to encrypt and decrypt a message
- With symmetric encryption, algorithms are designed to decrypt the ciphertext
  - It is essential that the key be kept confidential: if an attacker secured the key, she could decrypt any messages

Symmetric Encryption Algorithms (cont.)

- Can be classified into two distinct categories based on amount of data processed at a time:
  - Stream cipher usually encrypt/decrypt one bit at a time
  - Block cipher operate on blocks of plaintext ; for each input block they produce an output block
- Substitution ciphers substitute one letter or character for another
  - Also known as a monoalphabetic substitution cipher
  - Can be easy to break

Protecting with Symmetric Encryption Algorithms (cont.)

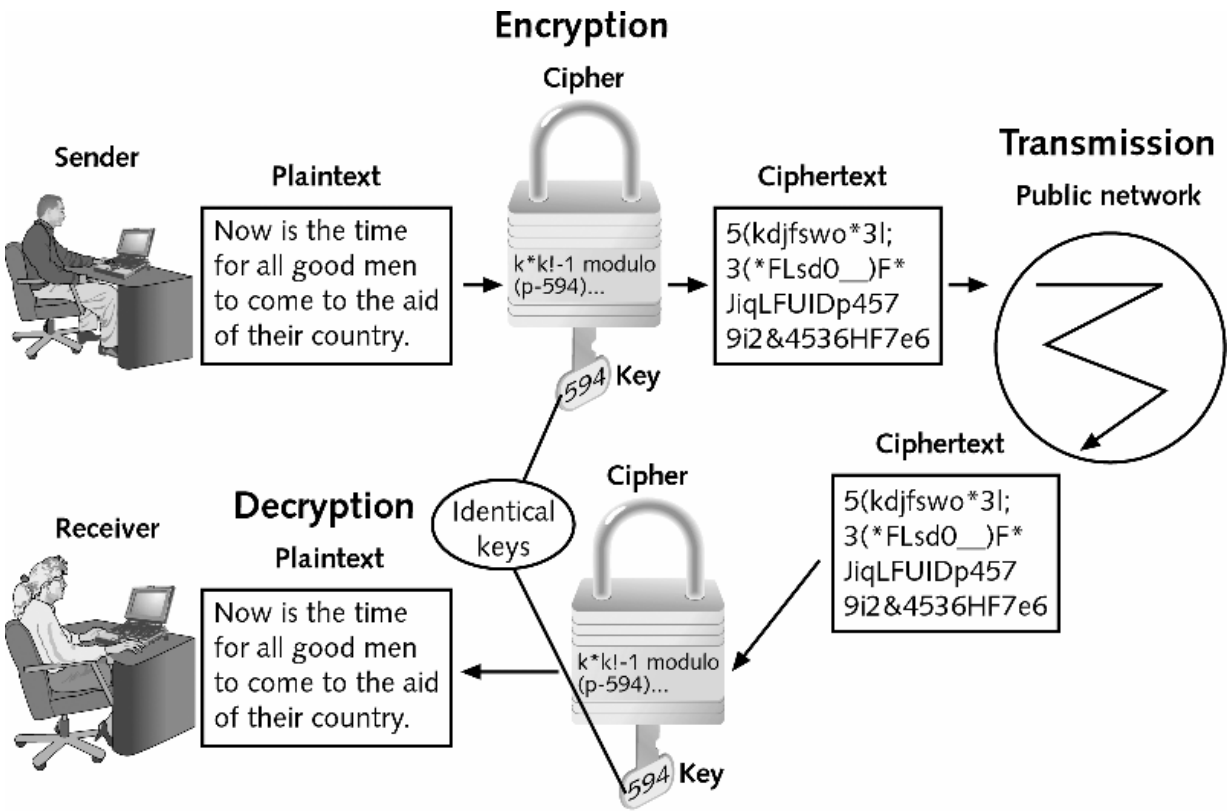
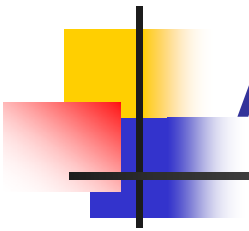


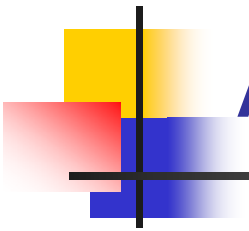
Figure 2 Symmetric encryption

Symmetric Encryption Algorithms (continued)

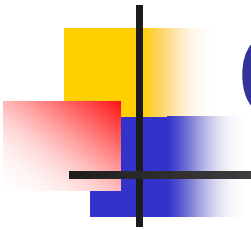


- A polyalphabetic substitution cipher maps a single plaintext character to multiple ciphertext characters
- A transposition cipher rearranges letters without changing them

Symmetric Encryption Algorithms (continued)



- A block cipher manipulates an entire block of plaintext at one time
- The plaintext message is divided into separate blocks of 8 to 16 bytes and then each block is encrypted independently
- The blocks can be randomized for additional security

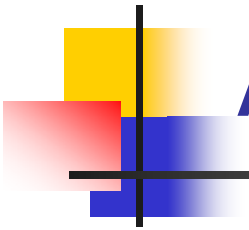


Classical Cryptography

- Sender, receiver share common key
  - Keys may be the same, or trivial to derive from one another
  - Sometimes called symmetric cryptography
- Two basic types
  - Transposition ciphers
  - Substitution ciphers
    - Monoalphabetic Substitution
    - Polyalphabetic Substitution

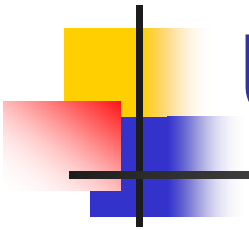
Cryptographic Methods

- Substitution Ciphers : is a type of ciphering system that changes one character or symbol into another. This is easy to break using character frequency analysis. There are two types :
  - Monoalphabetic cipher
  - PolyAlphabetic cipher



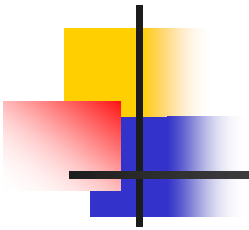
Attacks : Frequency Analysis

- Letter frequency analysis (Same plaintext maps to same ciphertext) :
- Letters are not equally commonly used
- In English E is by far the most common letter, then T , R , N , I , O , A, S
- Other letters are fairly rare : Z , J,K , Q , X



Use in Cryptanalysis

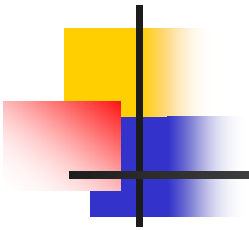
- Key concept – monoalphabetic substitution ciphers do not change relative letter frequencies
- Calculate letter frequencies for ciphertext
- Compare counts/plots against known values
- For monoalphabetic must identify each letter
  - Tables of common double/triple letters help



Caesar Cipher

- A simple substitution cipher
  - Supposedly invented by Julius Caesar
  - Translate each letter a fixed number of positions in the alphabet
  - Reverse by translating in opposite direction

Substitution Cipher
Caesar Cipher

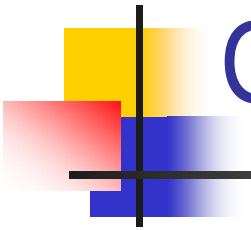


- Change characters in plaintext to produce ciphertext
- Example
  - Plaintext is HELLO WORLD
  - Change each letter to the third letter following it (X goes to A , Y to B , Z to C)
  - Ciphertext is KHOOR ZRUOG

Attacking the Cipher

- Exhaustive search (or guessing)
  - If the key is small enough, try all possible keys until you find the right one
  - Caesar cipher has 26 possible keys
- Statistical analysis (letter frequencies)
  - Count frequency of each encrypted symbol
  - Match to observed frequencies of other symbols in other kinds of data
  - Provides probable mapping of cipher.

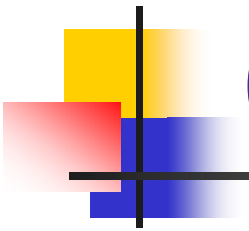
Polyalphabetic Substitution Ciphers



- Another approach to improving security is to use multiple cipher alphabets
- Ciphers that don't always translate a given plaintext character into the same ciphertext character
- Makes cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- For example, use different substitutions for odd and even positions.
  - Example : "transfer account"

    "sszorgds zdbptos"

Transposition Ciphers (Permutation Cipher)



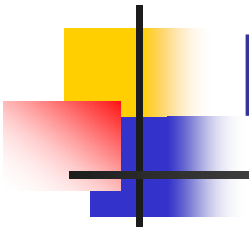
- The characters retain their plaintext form but change their positions to create the ciphertext
- The letters may be written into an array in one order and read out in a different order.
- This kind of cipher is also easily broken if enough ciphertext is available since the same frequency distribution is as the original text.

Transposition Cipher

- Rail-Fence Cipher
  - Write message letters over a number of rows
  - Then read off cipher row by row
  - E.g. write message out as
    - M e m a t r h t g p r y
    - e t e f e t e o a a t
  - Giving ciphertext

    mematrhtgpryetefeteoaat

Row Transposition Ciphers



- A more complex scheme
- Write letters of message out in rows over a specified number of columns
- Then reorder the columns according to some key before reading off the rows

Example

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

Vigenére Cipher

- Works as a poly-alphabetic substitution cipher that depends on a password

GRID :

ABCDEFGHIJKLMNOPQRSTUVWXYZ

BCDEFGHIJKLMNOPQRSTUVWXYZA

CDEFGHIJKLMNOPQRSTUVWXYZAB

.....

ZABCDEFGHIJKLMNOPQRSTUVWXY

Then the password is matched up to the text it is meant to encipher.

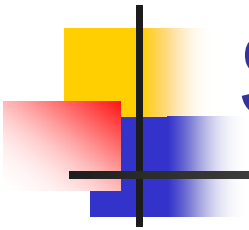
Vigenére Cipher Example

Key : **deceptivedeceptivedeceptive**

wearediscoveredsaveyourself

Ciphertext : zic**vtw**qngrzg**vtw**avzhcqyglmgj

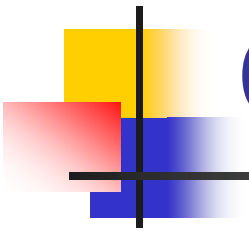
- The password is repeated until one character of the password is matched up with each character of the plaintext.
- Use each key letter as a caesar cipher key
- In this example, the key in the encryption system is the password. The algorithm is simple, but the key should be complex, with the best keys being very long and very random data.



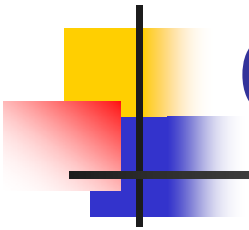
Security of Vigenére Cipher

- Have multiple ciphertext letters for each plaintext letter
- Hence letter frequencies are obscured but not totally lost.
- A good polyalphabetic substitution cipher may smooth out the frequencies; each character may occur almost the same number of times
- However, attacking the code is not difficult, since the character relationships are still preserved
- A good trial and error attack can break the code.

Transposition vs. Substitution Ciphers



- Both are not so good ciphers
- Strong modern ciphers tend to use both
- Transposition scrambles text patterns
- Substitution hides underlying text /bits
- Combining them can achieve both effects



One-Time Pad

- A type of cipher with a random key (a one-time pad) as long as the message.
- How it works:
  - A message is encoded as a binary string
  - A key is a random binary string that is at least as long as the message
  - Encryption is by bit-XOR
- Properties
  - Provides perfect secrecy
  - The only theoretically proven unbreakable encryption system
  - Requires key length at least as long as the message to be sent
  - OTPs can only be used once

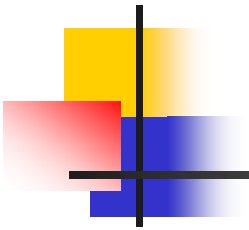
ONE –TIME PAD

- Invented in 1917 by Major Joseph Mauborgne and Gilbert Vernam.
- The method is also known as the **SECRET LOOKUP TABLE** among clinical laboratory specialists,
- and is the only method of encryption sanctioned by the U. S. **HEALTH INFORMATION PORTABILITY AND ACCOUNTABILITY ACT (HIPAA)** (2001, as amended).
- In this method, the sender and receiver agree upon a common secret text, which forms the key.

One Time Pad (cont.)

- Message 010
- Random key 011
- Apply the XOR algorithm : 001

- Any key was equally likely
- Any plaintext could have produced this message with one of those keys
- Key distribution is painful
- Synchronization of keys is vital
- A good random number generator is hard to find



ONE -TIME PAD

- Each key letter is used exactly once, and then discarded forever.
- Ideally, the key should be completely random.

Example

- If the receiver and sender employ the text for George Orwell's 1984 as their one-time pad, then the encryption-key becomes:

  ITWASABRIGHTCOLDDAYINAPRILANDTHECLOCKSWERESTRIKI
  NGTHIRTEENWINSTON....

  Analogously, the decryption-key becomes:

- 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- (I=8 , 26-8 = 18 → S) (T =19 , 26 -19= 7 → H)

  Hence:

  SHEAIAZJSUTHYMPXXACSNALJSPANXHTWYPMYQIEWJW

Example (cont.)

- Encryption of the plaintext, THEQUICKBROWNFOX, yields the ciphertext, BAAQMIDCJXFPPTZA, as follows:

```
      THEQUICKBROWNFOX
(+)   ITWASABRIGHTCOLD
      ____________________

      BAAQMIDCJXFPPTZA
```

Example (cont.)

- that is:

19 07 04 16 20 08 02 10 01 17 14 22 13 05 14 23 (plaintext)
(+) 08 19 22 00 18 00 01 17 08 06 07 19 02 14 11 03 (key)
_____
01 00 00 16 12 08 03 01 09 23 21 15 15 19 25 00 (ciphertext)
B A A Q M I D B J X V P P T Z A

- Decryption of the ciphertext, BAAQMIDBJXVPPTZA, returns the plaintext, THEQUICKBROWNFOX, as follows:

BAAQMIDBJXVPPTZA
(+) SHEAIAZJSUTHYMPX
_____
THEQUICKBROWNFOX

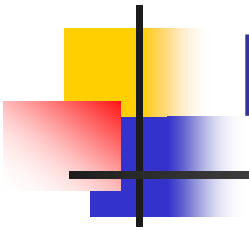
Example (cont.)

- that is:

01 00 00 16 12 08 03 01 09 23 21 15 15 19 25 00
(ciphertext)
(+) 18 07 04 00 08 00 25 09 18 20 19 07 24 12 15 23
(decryption-key)

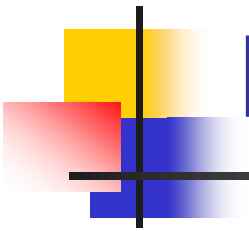
19 07 04 16 20 08 02 10 01 17 14 22 13 05 14 23
(plaintext)

The one-time-pad has the advantage of simplicity and relative security, but it requires the maintenance of a large key, which sender and receiver must maintain in synchrony.



Product Ciphers

- Ciphers using substitutions or transpositions are not secure because of language characteristics
- Hence consider using several ciphers in succession to make it harder :
  - E.g. a substitution followed by a transposition
- This is a bridge from classical ciphers to modern ciphers

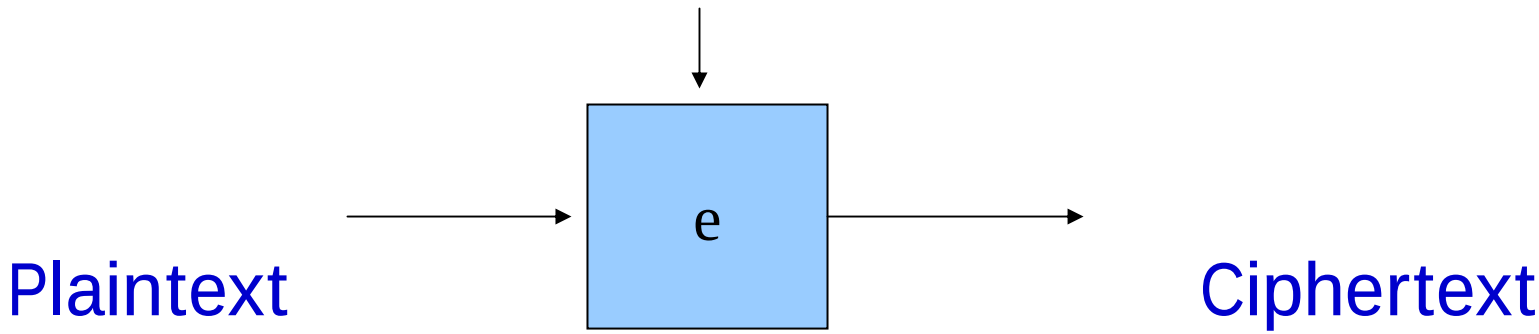


Block vs. Stream Ciphers

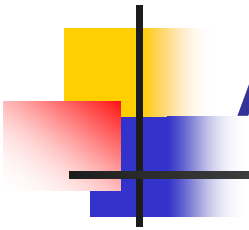
- Block ciphers treat messages as blocks to be then encrypted/decrypted separately
- Stream ciphers process messages a bit or a byte at a time when encrypting/decrypting –e.g. Vigenere
- Many current ciphers are block ciphers –most network-based cryptographic applications

Stream and Block Ciphers

- Stream ciphers convert one symbol of plaintext immediately into one symbol of ciphertext



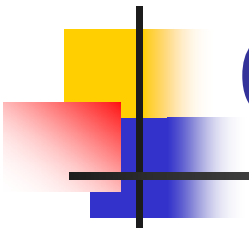
- Block ciphers work on a given sized chunk of data at a time



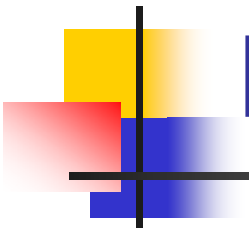
Advantages of Stream Ciphers

- Speed of encryption and decryption
  - Each symbol encrypted as soon as it's possible
- Low error propagation
  - Errors affect only the symbol where the error occurred
- Widely used :
  - SSL (RC4) , Cell Phones , DVD (LFSR)
  - WEP (RC4)

Disadvantages of Stream Ciphers

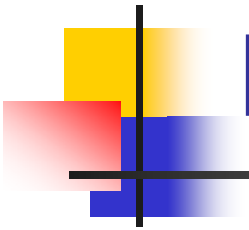


- Low diffusion
  - Each symbol separately encrypted
  - Each ciphertext symbol only contains information about one plaintext symbol
- Susceptible to insertions and modifications
- Not good match for many common uses of cryptography



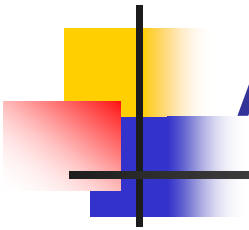
Example Stream Ciphers

- RC4 (widely used)
  - A byte oriented stream cipher
  - Used in Lotus Notes, Oracle Secure SQL, and other products
- Linear Feedback Shift Register:
  - Vulnerable to known plaintext attack
  - A m-stage LFSR can be completely broken given 2m bits output.



Block Ciphers

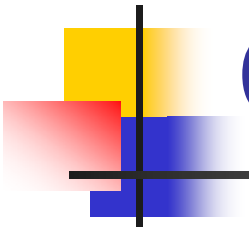
- A Block cipher algorithm operates on a block.
  - DES uses 64-bit blocks, and 56-bit key
  - AES uses 128-bit blocks, key can be 128, 192, 256 bits
- Security of block ciphers
  - When a random key is picked, the cipher should behave like a random permutation



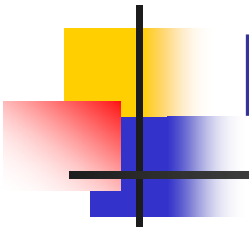
Advantages of Block Ciphers

- Diffusion
  - Easier to make a set of encrypted characters depend on each other
- Immunity to insertions
  - Encrypted text arrives in known lengths

Disadvantages of Block Ciphers



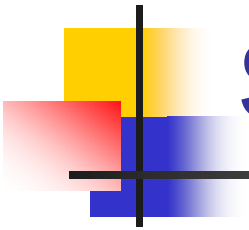
- Slower
  - Need to wait for block of data before encryption/ decryption starts
- Worse error propagation
  - Errors affect entire block



Block Cipher Principles

- Most symmetric block ciphers are based on a Feistel Cipher Structure
- Block ciphers look like an extremely large substitution
- Using idea of a product cipher
- It has complex structure compared to public-key algorithms

Claude Shannon and Substitution-Permutation Ciphers



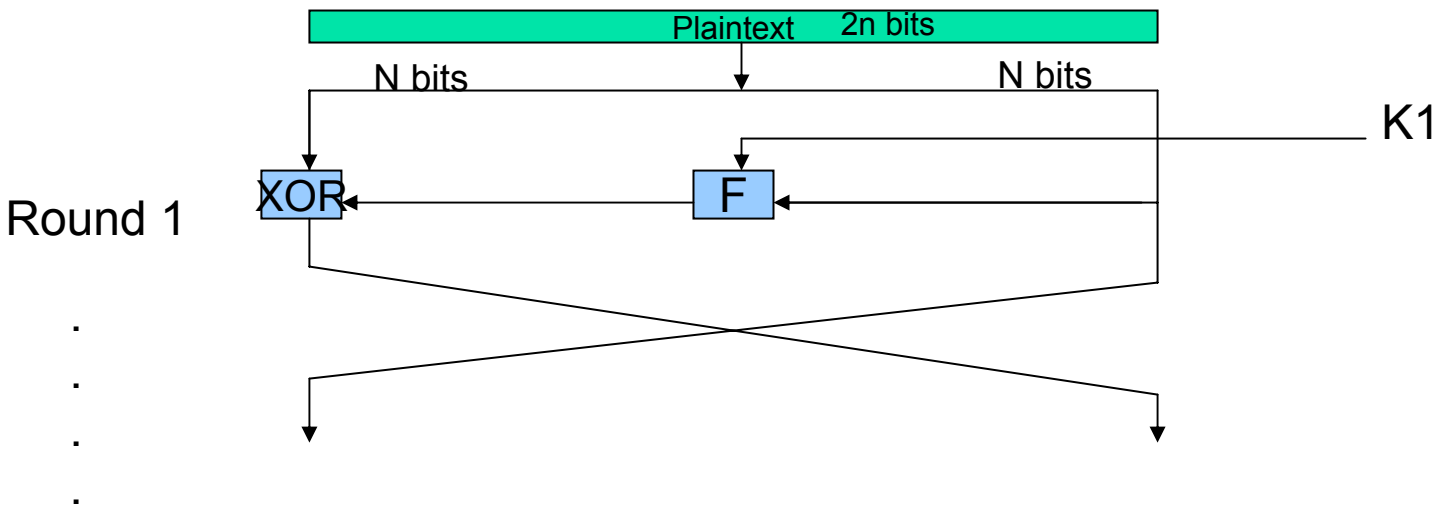
- In 1949 Claude Shannon introduced the idea of substitution-permutation (S-P) networks
  - Modern substitution-transposition product cipher
- These form the basis of modern block ciphers
- S-P networks are based on the primitive cryptographic operations :
  - **Substitution (S-Box)**
  - **Permutation (P-Box)**
- Provide confusion and diffusion of message

Confusion and Diffusion

- Cipher needs to completely obscure statistical properties of original message
- A one-time-pad does this
- More practically Shannon suggested combining elements to obtain:
- Diffusion : dissipates statistical structure of plaintext over bulk of ciphertext (each plaintext bit affects the value of many ciphertext bits)
- Confusion : makes relationship between ciphertext and key as complex as possible – use complex substitution algorithm

Feistel Cipher Structure

- **Partitions input block into two halves**
  - **Process through multiple rounds which**
  - **Perform a substitution on left data half**
  - **Based on round function of right half and subkey**
  - **Then have permutation swapping halves**



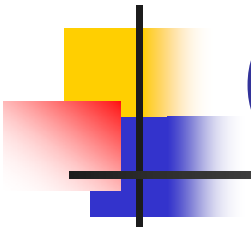
Feistel Cipher Design Principles

- Block size
  - Increasing block provides more security, but reduces the en/decryption speed
- Key size
  - Larger size →greater security, makes exhaustive key searching harder, but may slow cipher (common 64, 128)
- Number of rounds
  - More rounds →more security (Typical 16 rounds)
- Subkey generation
  - Greater complexity makes cryptanalysis harder, but slows cipher
- Round function
  - Greater complexity can make analysis harder, but slows cipher
- Fast software en/decryption and ease of analysis
  - Are more recent concerns for practical use and testing

Feistel Cipher Decryption

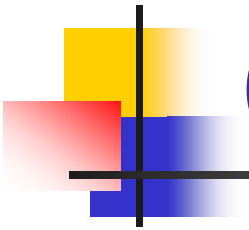
- Use the same encryption algorithm with :
- The ciphertext as the input
- The round keys are applied in reverse order:
  - Use Kn in the first round , and K1 in the 16^{th} round

Data Encryption Standard (DES)



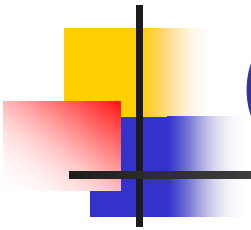
- Most widely used block cipher
- IBM developed Lucifer cipher
  - By team led by Feistel
  - Used 64-bit data blocks with 128-bit key
- In 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES
- Adopted in 1977 by NBS (now NIST)

Data Encryption Standard (DES) (cont.)

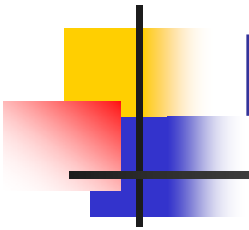


- Probably the best known symmetric key cryptosystem
- Analyzed, altered and approved by the National security Agency
- Adopted as a federal standard

Data Encryption Standard (DES) (cont.)

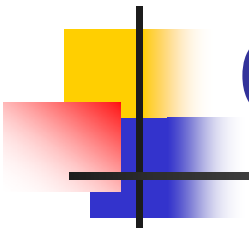


- DES is a block cipher and encrypts data in 64-bit blocks using a 64 bit key. (only 56 bits really used) Outputs 64 bits of ciphertext.
- Uses substitution and permutation
- DES encrypts 64-bit plaintext by executing the algorithm 16 rounds (iterations) each with a round key (48 bits) generated from the user-supplied key (56 bits).
- Each round consists of a substitution followed by a permutation



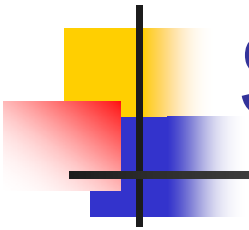
Description of DES Algorithm

- Alternate applications of two different ciphers
  - A product cipher
- Starts by breaking block in half
- The algorithm goes through 16 rounds
- Each round consists of a substitution followed by a permutation



One DES Round

- Select 48 bits from the key
- Expand right half of block to 48 bits
- XOR with key bits
- Look up result in an S-Box
  - Resulting in 32 bits
- Perform a permutation using a P-box
- XOR with left half of block
- Result is new right half
- Old right half becomes new left half

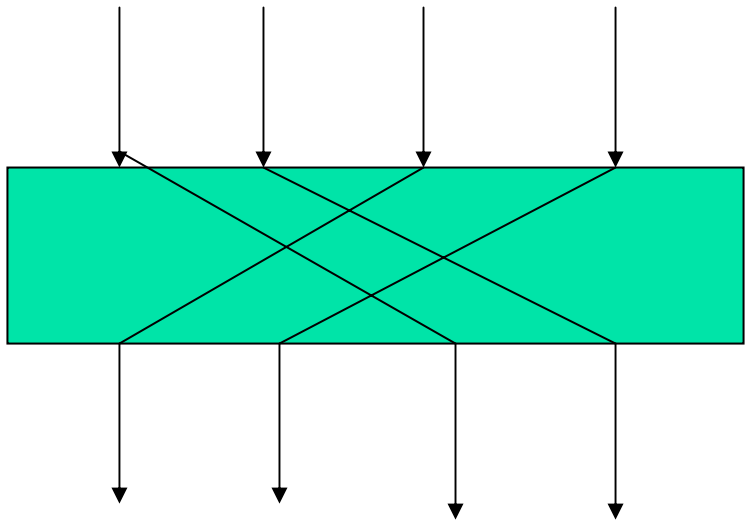


S - Boxes

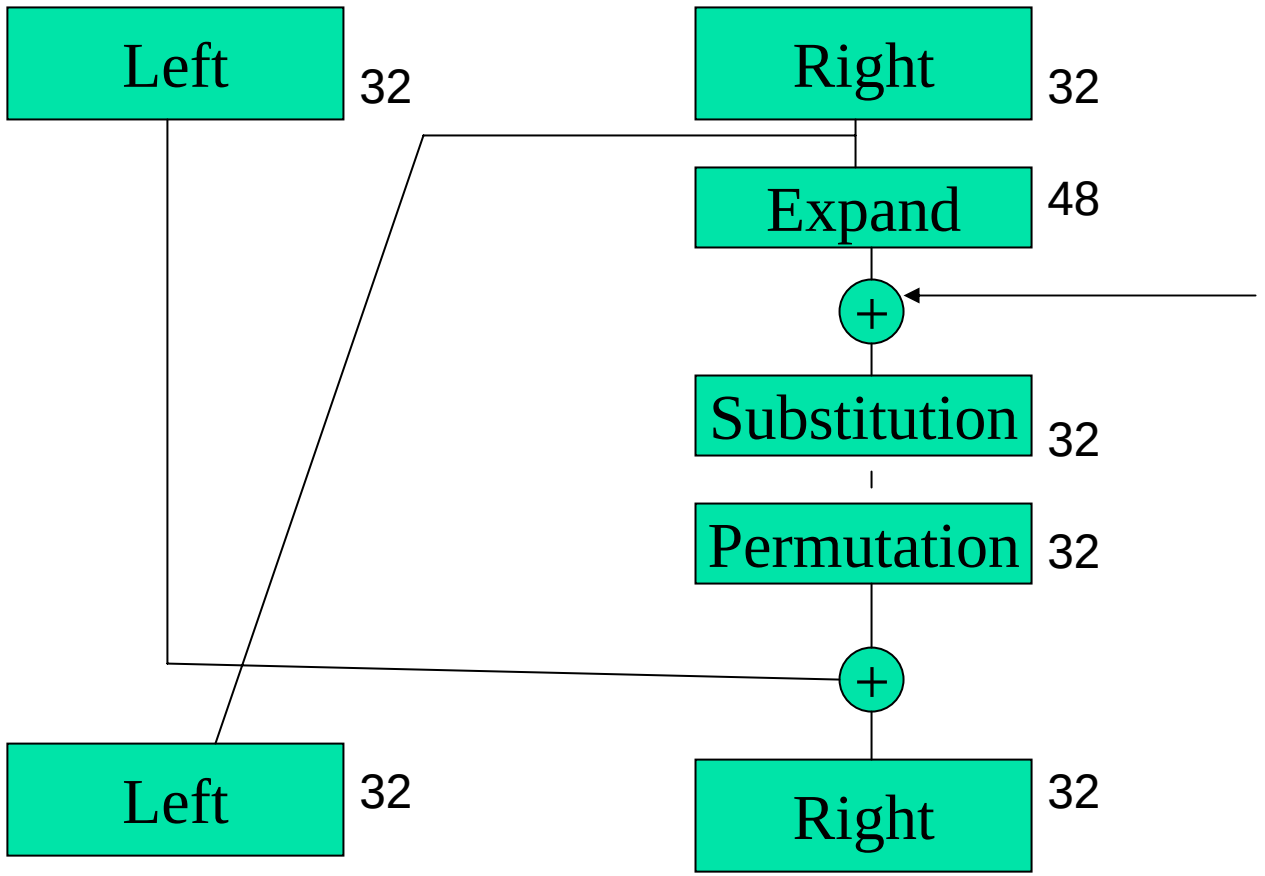
- Substitution Box
- Table lookups to perform substitutions
- Permanently defined for DES
- Eight different S-Boxes
- Choice of contents of S-Boxes believed to strongly impact security of DES

Permutation Box P - Box

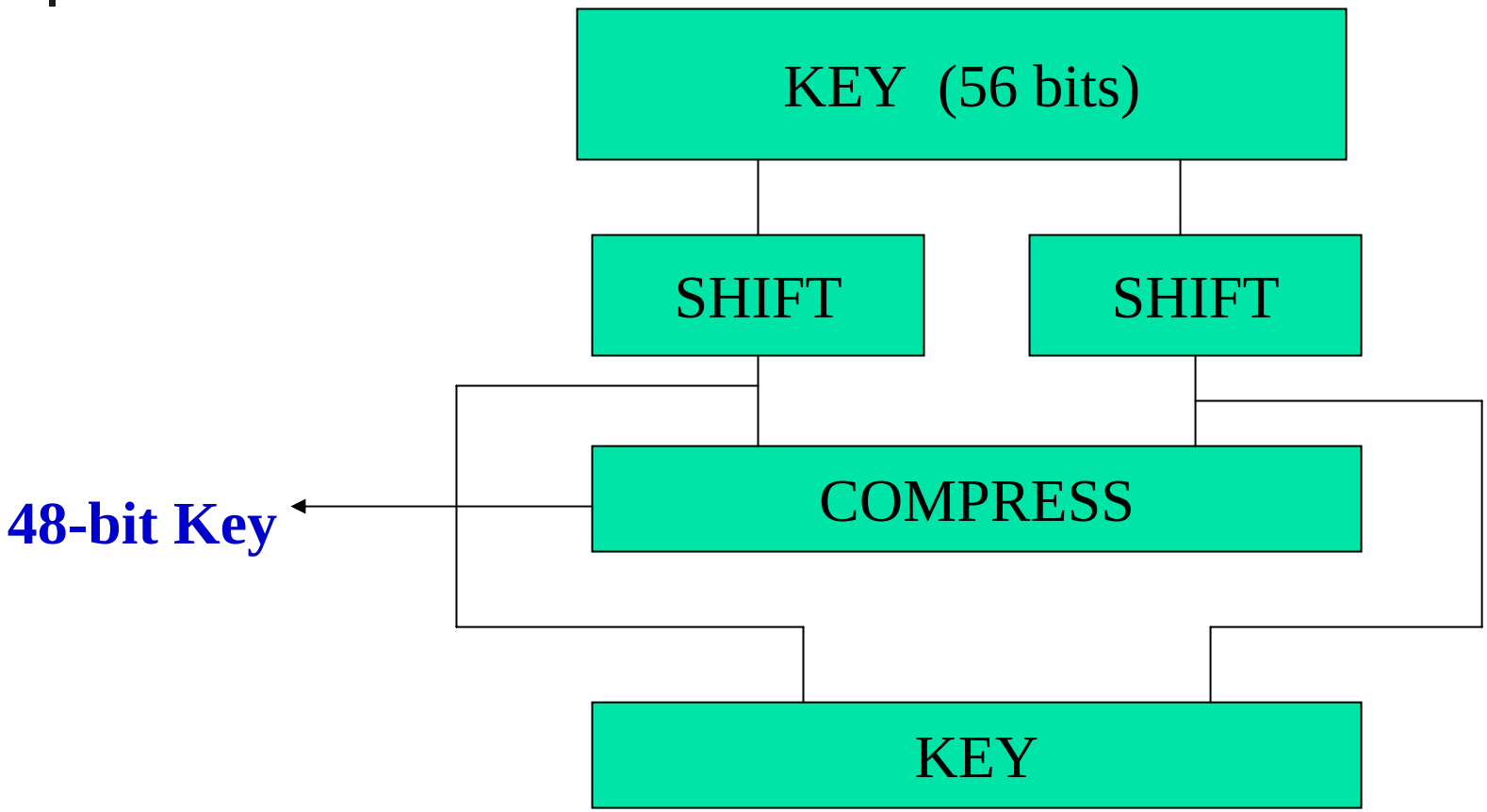
- Maps 32 input bits to 32 output bits
- A single straight permutation

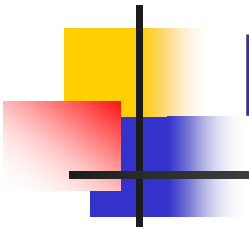


DES Round


Left
32
Right
32
Expand
48
48 bit Key
+
Substitution
32
Permutation
32
+
Left
32
Right
32

DES Key Round



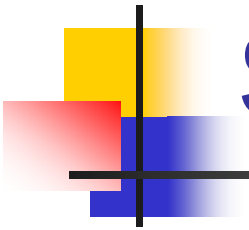


More on DES

- Uses substitutions to provide *confusion*
  - To hide the set of characters sent
- Uses transposition to provide *diffusion*
  - To spread the effects of one plaintext bit into other bits
- Uses only standard arithmetic and logic functions and table lookup

Avalanche Effect

- A small change in the plaintext or the key should result in significant change in the ciphertext. It is a desirable property of encryption algorithm
- Where a change of one input or key bit results in changing approximately half output bits
- Making attempts to “home-in” by guessing keys impossible
- DES exhibits strong avalanche effect



Strength of DES

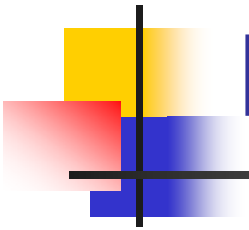
- 56-bit keys have 2^56 = 7.2x10^16 values
- Brute force search looks hard
- Recent advances have shown is possible
  - In 1997 on Internet in a few months
  - In 1998 on dedicated h/w in a few days
  - In 1999 in 22 hrs
  - In 2000 in 3.5hrs by very fast computers
- The main concern with regard to DES, was about the S-boxes.
- DES reasonably resistant to differential cryptanalysis and to timing attacks

Data Encryption Standard

- DES is a mature encryption standard
- It can be implemented in hardware and software
- Its 56-bit key is no longer considered secure enough to be used.
- It is no longer recommended for secure transmission

Data Encryption Standard Modes

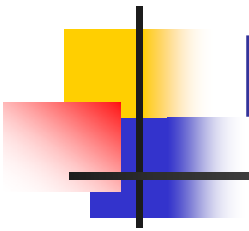
DES Mode	Cipher Algorithm	Operation	Strength
ECB	Block Cipher	Uses a 56-bit key to encrypt 64-bit blocks	Vulnerable to attackers
CBC	Block Cipher	Each message block is linked together	More secure than ECB
OFB	Block cipher that functions like a stream cipher	Results of cipher are added to a message for the next round	Less secure than CFB
CFB	Block Cipher that functions like a stream cipher	Ciphertext created in one round used to encrypt the next round	Very secure but slower than ECB



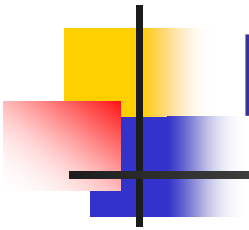
Encryption Modes

- How to encrypt a message
  - A message is divide into blocks
  - Four standard modes were defined for DES
  - Extended to five later, and they can be used with other block ciphers : 3DES and AES
  - Different encryption modes may be used
    - Electronic Code Book (ECB)
    - Cipher Block Chaining (CBC)
    - Cipher Feedback (CFB)
    - Output Feedback (OFB)
    - Counter Mode (CTR).... new mode similar to OFB but encrypts counter rather than any feedback value, used in high speed network encryptions

Block Cipher Encryption Modes : ECB



- Electronic Code Book (ECB) :message is broken into independent blocks which are encrypted.
- Each block is a value which is substituted, like a codebook, hence the name.
- Not secure : it is deterministic; the same data gets encrypted the same way; vulnerable if data repeats, reordering ciphertext determines reordered plaintext.
- Errors in one block do not propagate.
- Identical input produces identical output in this mode

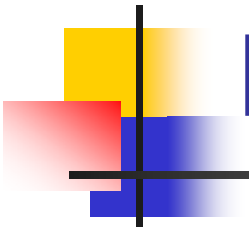


ECB

- The simplest of the encryption modes is the electronic codebook (ECB) mode.
- The message is divided into blocks and each block is encrypted separately.
- The disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks; thus, it does not hide data patterns well.
- In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all.

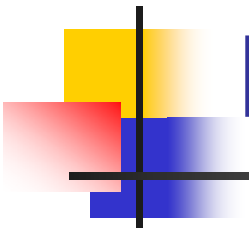
Block Cipher Encryption Modes : CBC

- Cipher Block Chaining (CBC) : next input depends on previous output
- Uses Initial Vector IV to start process
- In this mode, each block is encrypted as in ECB, but a third factor , derived from the previous input, is added.
- In this case, identical input (plaintext) does not produce identical output
- Uses : bulk data encryption, authentication



Properties of CBC

- CBC has been the **most commonly** used mode of operation.
- Its main drawbacks are that encryption is sequential (i.e., it cannot be parallelized), and that the message must be padded to a multiple of the cipher block size.
- Note that a one-bit change in a plaintext affects all following ciphertext blocks, and a plaintext can be recovered from just two adjacent blocks of ciphertext.
- As a consequence, decryption *can* be parallelized, and a one-bit change to the ciphertext causes complete corruption of the corresponding block of plaintext, and inverts the corresponding bit in the following block of plaintext.

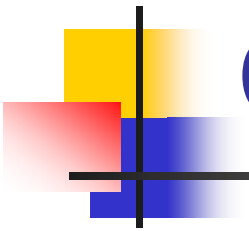


Properties of CBC (cont.)

- Errors in one block propagate.
- Randomized encryption: repeated text gets mapped to different encrypted data.
- When IV is randomized, can be proven to be semantically secure, assuming that the block cipher is Pseudo Random Permutation (PRP)
- Sequential encryption, cannot use parallel hardware
- Need IV known to sender and receiver

Cipher Feedback (CFB)

- Message is treated as a stream of bits
- Added to the output of the block cipher
- Result is feedback for next stage, hence name
- Standard allows any number of bits (1 , 8, or 64 or whatever) to be feedback ...denoted CFB-1 , CFB-8, CFB-64
- Uses : stream data encryption, authentication
- Advantages : appropriate when data arrives in bits/bytes
- Most common stream mode
- Limitation : need to stall while doing block encryption after every n-bits
- Errors propagate for several blocks after the error



Output Feedback (OFB)

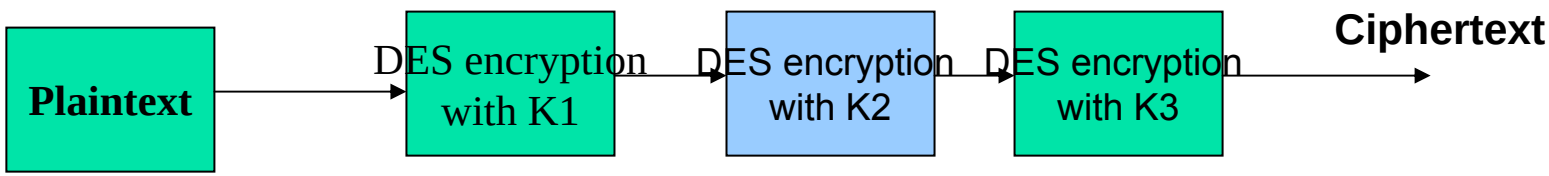
- Message is treated as a stream of bits
- Output of cipher is added to message
- Output is then feedback ...hence name
- Feedback is independent of message
- Used when error feedback a problem
- Research has shown that only OFB-64 should ever be used

Triple Data Encryption Standard (3DES)

- Simple way of increasing security of DES
- Apply DES three times iteratively to each block
  - Thus 1/3 as fast as DES
- Use different key for each encryption
- Effectively doubles the key length of DES
- Approved by NIST
  - Which recommends using it in preference to DES
- Has been adopted by some Internet applications, e.g. S/MIME , PGP

Triple Data Encryption Standard (3DES) (cont.)

- Uses three rounds of encryption instead of just one
- The ciphertext of one round becomes the entire input for the second iteration
- Employs a total of 48 iterations in its encryption (3 iterations times 16 rounds)
- The most secure versions of 3DES use different keys for each round
- Like DES it can suffer from weak keys
- If the same key is used for all 3 rounds, it will be identical to DES



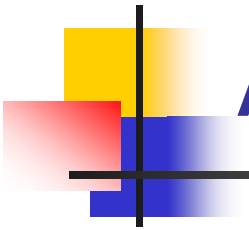
For decryption : Use K3 ,then K2 , then K1 to obtain plaintext

Advanced Encryption Standard (AES)

- A relatively new cryptographic algorithm
- Intended to be the replacement for DES
- Chosen by NIST in 2000
  - Through an open competition(Rijndael-Daemen in Belgium was selected as the AES)
  - Process began with the NIST publishing requirements for a new symmetric algorithm and requesting proposals
- Increased popularity of AES : appears to be gradually replacing DES as was intended
- Various RFCs describe using AES in IPSEC
- Commercial VPNs that use AES are available

Advanced Encryption Standard (AES) (continued)

- Performs three steps on every block (128 bits) of plaintext
- Within step 2, multiple rounds are performed depending upon the key size:
  - 128-bit key performs 9 rounds AES-128
  - 192-bit key performs 11 rounds AES-192
  - 256-bit key uses 13 rounds AES-256
- An iterative rather than feistel cipher ...treats data in 4 groups of 4 bytes
- Operates an entire block in every round
- Criteria :
  - General security
  - S/W and H/W implementation ease



AES (cont.)

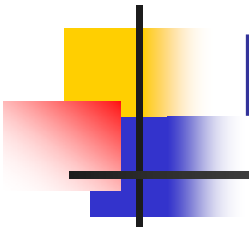
- Has 10/12/14 rounds depending on the key length.
- In each round the following operations are performed
  - Byte substitution (1 S-box used on every byte)
  - Shift rows (permute bytes between groups/columns)
  - Mix columns (subs using matrix multiply of groups)
  - Add round key (XOR state with key material)
- All operations can be combined into XOR and table lookups – hence very fast and efficient

Rivest Cipher (RC)

- Family of cipher algorithms designed by Ron Rivest
- He developed six ciphers, ranging from RC1 to RC6, but did not release RC1 and RC3
- RC2 is a block cipher that processes blocks of 64 bits
- RC4 is a stream cipher that accepts keys up to 128 bits in length (used in Wired Equivalent Privacy WEP encryption standard in 802.11a,b, and g)

International Data Encryption Algorithm (IDEA)

- IDEA algorithm dates back to the early 1990s and is used in European nations
- Block cipher that processes 64 bits with a 128-bit key with 8 rounds
- IDEA is part of PGP

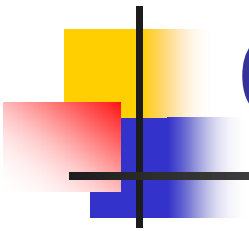


Blowfish

- A symmetric block cipher designed in 1993 by Bruce Schneier , that operates on 64-bit blocks
- Can have a key length from 32 to 448 bits
- Characteristics
  - Fast implementation on 32-bit processors
  - Compact in use of memory
  - Simple structure for implementation
  - Variable security by varying key size
- Has been implemented in various products

Asymmetric Key Systems

History of Public Key Cryptography

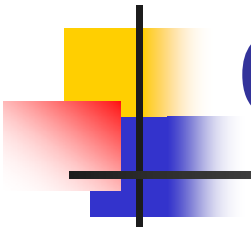


- Invented by Diffie and Hellman in 1976
- Merkle and Hellman developed Knapsack algorithm in 1978
- Rivest-Shamir-Adelman developed RSA in 1978
  - Most popular public key algorithm

Asymmetric Encryption Algorithms

- The primary weakness of symmetric encryption algorithm is keeping the single key secure
- This weakness, known as key management, poses a number of significant challenges
- Asymmetric encryption (or public key cryptography) uses two keys instead of one
  - The private key typically is used to encrypt the message
  - The public key decrypts the message

Practical Use of Public Key Cryptography



- Keys are created in pairs
- One key is kept secret by the owner
- The other is made public to the world
- If you want to send an encrypted message to someone, encrypt with his public key
  - Only he has private key to decrypt

Asymmetric Encryption Algorithms (continued)

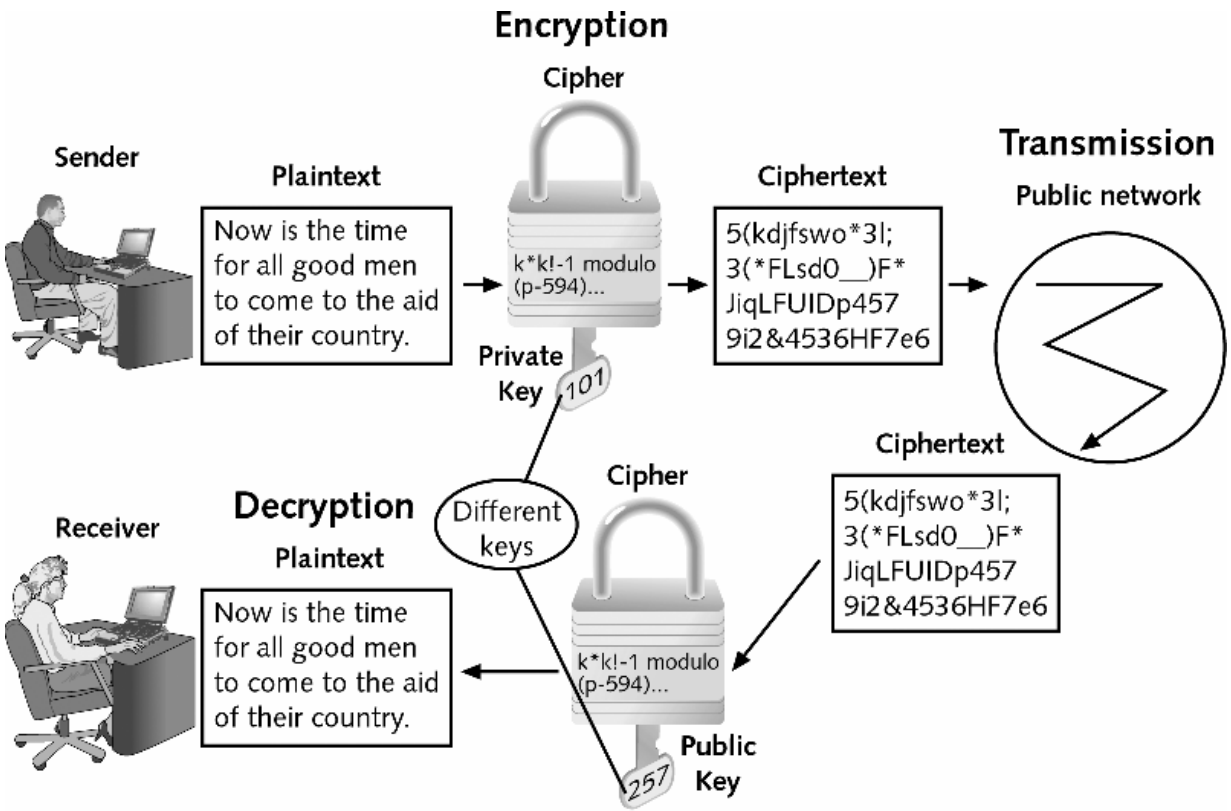
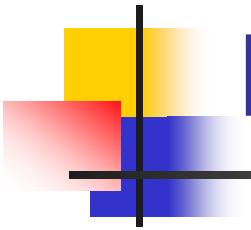
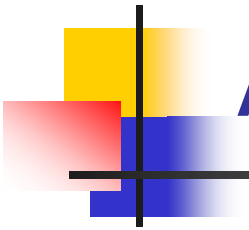


Figure 3 Asymmetric encryption

Authentication With Shared Keys



- If only two people know the key, and I didn't create a properly encrypted message-
  - The other person must have
- But what if he claims he didn't?
- Or what if there are more than two?
  - Requires authentication servers

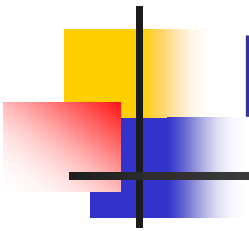


Authentication With Public Key

- If I want to “sign” a message, encrypt it with my private key
- Only I know private key, so no one else could create that message
- Everyone knows my public key, so everyone can check my claim directly

Rivest Shamir Adleman (RSA)

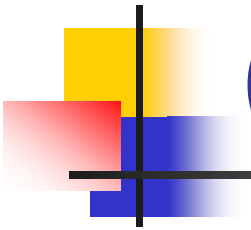
- Asymmetric algorithm published in 1977 and patented by MIT in 1983
- Most common asymmetric encryption and authentication algorithm
- Included as part of the Web browsers from Microsoft and Netscape as well as other commercial products
- RSA algorithm's security is based on the difficulty of factoring large numbers (in particular, products of large primes)
- Compared to symmetric key systems, RSA has nice scaling properties



Background : Number Theory

- Prime Number
  - An integer whose only factors are 1 and itself
  - There are an infinite number of prime numbers
  - All integers can be expressed as a product of (powers of) primes : 48 =2*2*2*2*3
  - Factorization is the process of finding the prime factors of a number
  - This is a hard problem for large numbers

Greatest Common Divisor (GCD)



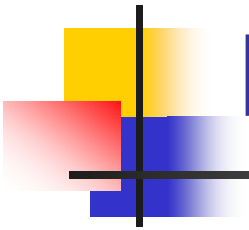
- Also known as greatest common factor
- The largest number that evenly divides two numbers
  - GCD (15,25) = 5

Number Theory (cont.)

- Relatively Prime :
  - Two numbers x and y are relatively prime if their GCD =1
  - No common factors except 1
  - Example : 38 and 55 are relatively prime
    - 38 = 2* 19
    - 55 = 5* 11

Number Theory (cont.)

- Totient function Φ(n)
  - Number of positive integers less than n and relatively prime to n
  - Example : Φ(10) = 4
    - 1,3,7,9 are relatively prime to 10
  - Example : Φ(21) = 12
    - 1,2,4,5,8,10,13,16,17,19,20 are relatively prime to 21



RSA

- Pick two random large primes p and q . For maximum security, choose them to be the same length
- Compute their product n=pq
- Then Φ(n) = (p-1)(q-1)
- Randomly choose the encryption key e, such that :
  e and (p-1)(q-1) are relatively prime i.e. e relatively prime to Φ(n)
- Then, find the decryption key d, such that :
  ed mod(p-1)(q-1) =1 i.e. ed mod Φ(n) =1
  i.e ed = k(p-1)(q-1) +1 , for some integer k
- The public key (e,n)
- The private key (d,n)
- To encrypt a message m : c = m^e (mod n)
- To decrypt a cipher c : m = c^d (mod n)
- The security of RSA is based on the difficulty of factoring large numbers

RSA EXAMPLE

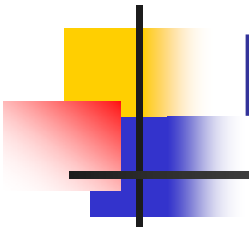
- Take p = 7 , q = 11
- So n = 77 and Φ(n) = 60
- Alice chooses e = 17, making d = 53
- Bob wants to send Alice secret message HELLO (07 04 11 11 14)
  - 07^17 mod 77 = 28
  - 04^17 mod 77 = 16
  - 11^17 mod 77 = 44
  - 14^17 mod 77 = 42
  - Bob sends 28 16 44 44 42

RSA EXAMPLE (cont.)

- Alice receives 28 16 44 44 42
- Alice uses private key, d=53, to decrypt message
  - 28^53 mod 77 =07
  - 16^53 mod 77 =04
  - 44^53 mod 77 = 11
  - 42^53 mod 77 =14
- Alice translates message to letters to read HELLO

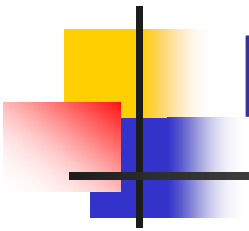
RSA Example : Encryption and Authentication

- Alice wants to send Bob message Hello both encrypted and authenticated (integrity – checked)
  - Alice's keys : public(17,77) ; private 53
  - Bob's keys : public : (37,77) ; private : 13
- Alice encrypts Hello (07 04 11 11 14) :
  - (07^53 mod 77)^37 mod 77 = 07
  - (04^53 mod 77)^37 mod 77 = 37
  - .....
- Alice sends 07 37 44 44 14



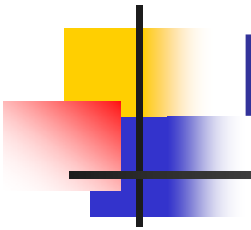
RSA Keys

- Keys are functions of a pair of 100-200 digit prime numbers
- Relationship between public and private key is complex
- Recovering plaintext without private key (even knowing public key) is supposedly equivalent to factoring product of the prime numbers



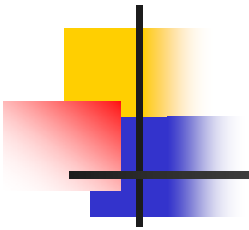
RSA Security Services

- Confidentiality
  - Only the owner of the private key knows it, so text encrypted with public key cannot be read by anyone except the owner of the private key
- Authentication
  - Only the owner of the private key knows it, so text encrypted with the private key must have been generated by the owner



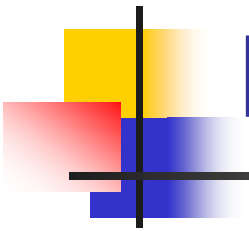
RSA Security Services (cont.)

- Integrity
  - Encrypted letters cannot be changed undetectably without knowing private key
- Non-Repudiation
  - Message encrypted with private key came from someone who knew it



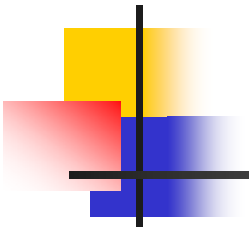
RSA - Warning

- Encrypt message in blocks considerably larger than the previous examples
  - If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
  - Attacker cannot alter letters, but can rearrange them and alter message meaning



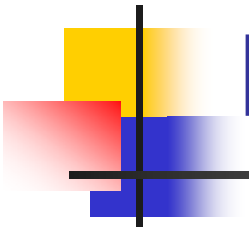
Key Management Issues

- To communicate via shared key cryptography, key must be distributed
  - In trusted fashion
- To communicate via public key cryptography, need to find out each other's public key
  - "Simply publish public keys"



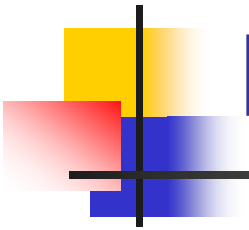
Issues of Key Publication

- Security of public key cryptography depends on using the right public key
- If I am fooled into using the wrong one, that key's owner reads my message
- Need high assurance that a given key belongs to a particular person
- Which requires a *key distribution infrastructure*



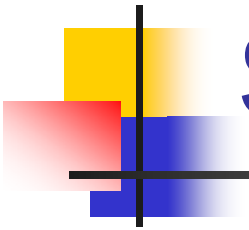
RSA vs. Block Ciphers

- The performance of RSA relative to block ciphers is relatively unimpressive
- IDEA block cipher implemented in software can achieve 800 Kbits /sec in comparison to 16 kbits /sec for RSA !
- Because of its slowness, RSA is typically used to exchange secret keys for block ciphers that are then used to encode the message
- RSA is used in X.509 (ITU standard) , PGP, PEM, Entrust, and many other software packages



DES vs. RSA

- DES is much more complex
- However, DES uses only simple arithmetic, logic, and table lookup
- RSA uses exponentiation to large powers
  - Computationally 1000 times more expensive in hardware, 100 times in software
  - Key selection also more expensive
  - RSA originally patented, but now is public domain



Security of RSA

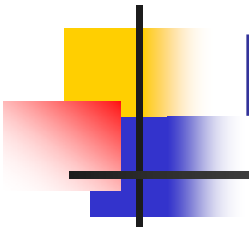
- Conjectured that security depends on factoring large numbers
- In 2003, a 576 bit RSA key was successfully factored
  - Using supercomputers at three German universities and other hardware
- Size of the key will keep on increasing

ELGAMAL Algorithm

- Is based on the difficulty of calculating discrete logarithms
- Pick a prime p
- Pick some g, such that : g is a generator for Zp
- Pick random x and calculate y, such that :

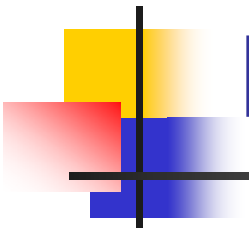
  y = g^x (mod p)
- Distribute p,g, and y as the public key. The private key will be x
- To encrypt, choose a random number k such that :

  k is relatively prime to p-1
- Calculate a =g^k (mod p) and b= y^k m (mod p)
- The cipher is the pair (a,b)
- To decrypt, calculate m = b/a^x (mod p)



Diffie-Hellman Key Exchange

- Unlike RSA, the Diffie-Hellman algorithm does not encrypt and decrypt text
- Allows two users to establish a secret key over an insecure medium without prior secrets.
- Strength of Diffie-Hellman is that it allows two users to share a secret key securely over a public network
- Once the key has been shared, both parties can use it to encrypt and decrypt messages using symmetric cryptography

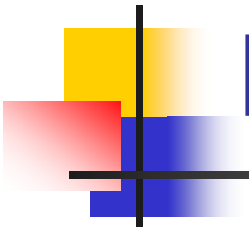


Diffie-Hellman Key Exchange

- Two system parameters p and g
  - Public values that may be used by all the users in a system
  - Parameter p is a prime number and (p-1)/2 is also a prime number
  - Parameter g (usually called a generator) is an integer less than p, such that for every number n with 0‹ n ‹ p , there is a power k of g such that n = g^k mod p

Diffie-Hellman Key Exchange

- Suppose Alice and Bob want to establish a shared secret key
- Alice generates a random private value *a* and Bob generates a random private value *b* where *a* and *b* are integers
- Alice and Bob derive their public values using parameters p and g and their private values
  - Alice's public value R1 = *g^a* mod *p*
  - Bob's public value R2 = g^b mod *p*
- Alice and Bob exchange their public values
- Alice computes *R2^ a = (g^b* mod *p)^a =K1*
- Bob computes *R1^b = (g^a* mod *p)^b =K2*
- Since K1 = K2 =K , Alice and Bob now have a shared secret key *k*

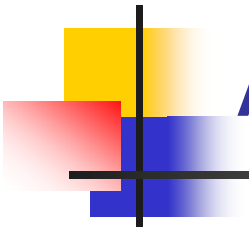


Example

- Assume G = 7 and p =23
- Alice chooses a =3 and calculates R1=7^3 mod 23 = 21
- Bob chooses b =6 and calculates R2=7^6 mod 23 =4
- Bob sends R2 to Alice
- Alice sends R1 to Bob
- Alice calculates the symmetric key

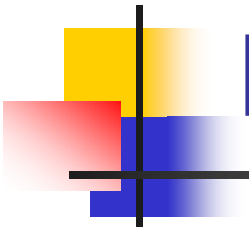
  K=4^3mod 23 = 18
- Bob calculates the symmetric key

  K =21^6mod23 =18



Attacks against DH

- Diffie-Hellman Key exchange is secure against a passive attacker
- How can an attacker disrupt the protocol?
  - man-in-the middle
  - Modify public values as they are exchanged
- How to fortify the protocol against active attackers?
  - Create a certified list of public values
  - Recent research suggests to authenticate strangers that incorporate the DH protocol in the system



Elliptic Curve Cryptography

- First proposed in the mid-1980s
- Instead of using prime numbers, uses elliptic curves
- An elliptic curve is a function drawn on an X-Y axis as a gently curved line
- By adding the values of two points on the curve, you can arrive at a third point on the curve

Hashing Algorithms

- One of the three categories of cryptographic algorithms is known as hashing

Defining Hashing (continued)

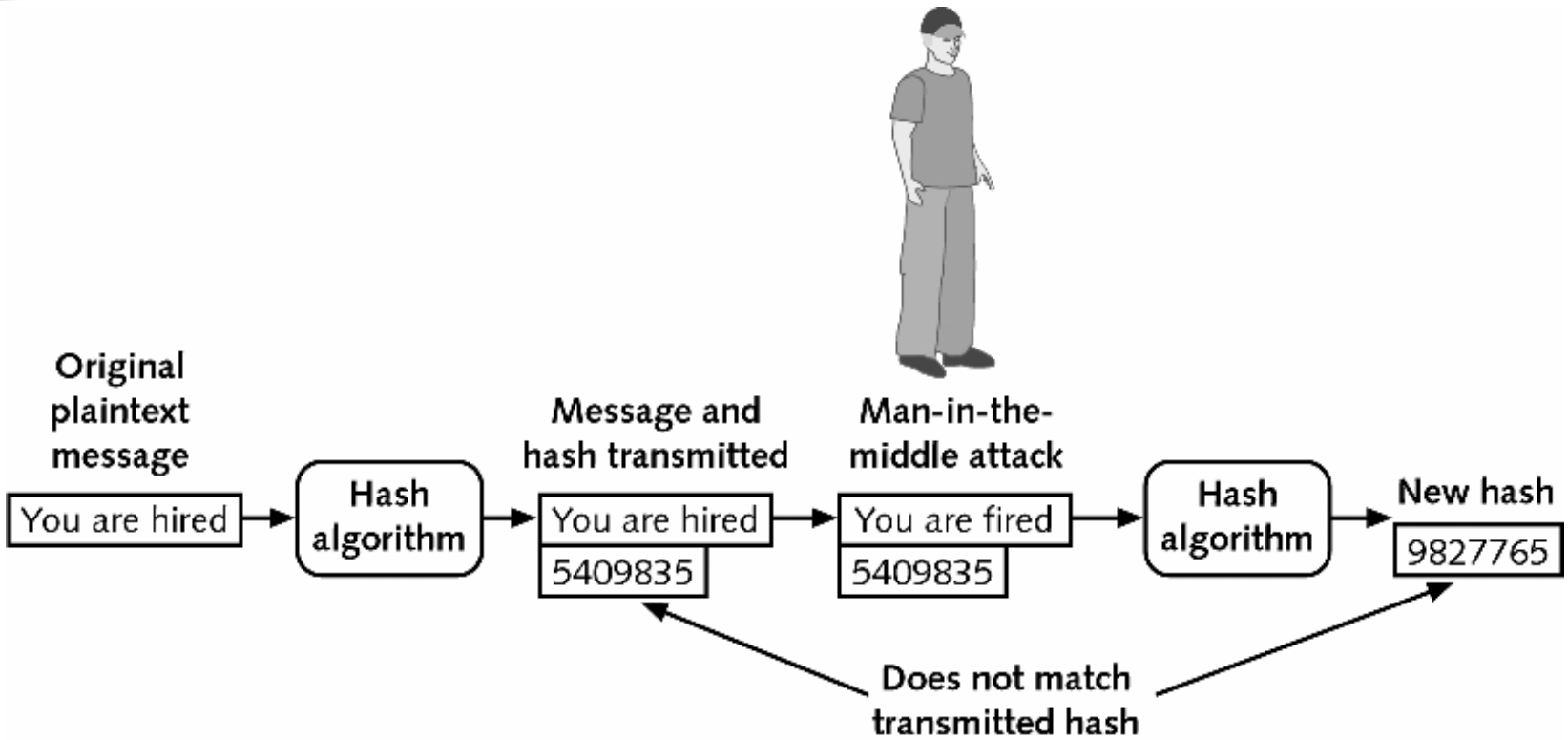
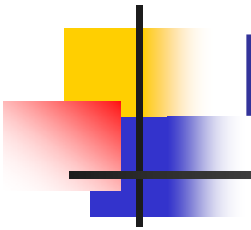
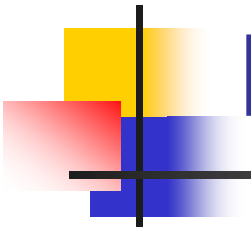


Figure Man-in-the-middle attack thwarted by hashing



Defining Hashing

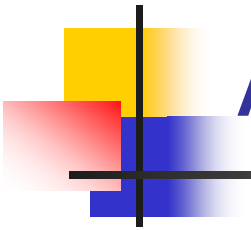
- Hashing, also called a one-way hash, creates a ciphertext from plaintext
- Cryptographic hashing follows this same basic approach
- Hash algorithms verify the accuracy of a value without transmitting the value itself and subjecting it to attacks
- A practical use of a hash algorithm is with automatic teller machine (ATM) cards



Hashing (continued)

- Hashing is typically used in two ways:
  - To determine whether a password a user enters is correct without transmitting the password itself
  - To determine the integrity of a message or contents of a file

Characteristics of Hash Algorithm



Message Digest (MD)

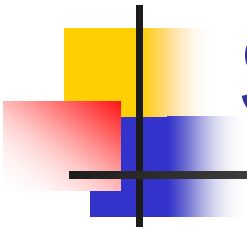
- Message digest 2 (MD2) takes plaintext of any length and creates a hash 128 bits long
  - MD2 divides the message into 128-bit sections
  - If the message is less than 128 bits, data known as padding is added
- Message digest 4 (MD4) was developed in 1990 for computers that processed 32 bits at a time
  - Takes plaintext and creates a hash of 128 bits
  - The plaintext message itself is padded to a length of 512 bits

Message Digest (MD) (continued)

- Message digest 5 (MD5) is a revision of MD4 designed to address its weaknesses
  - The length of a message is padded to 512 bits
  - The hash algorithm then uses four variables of 32 bits each in a round-robin fashion to create a value that is compressed to generate the hash

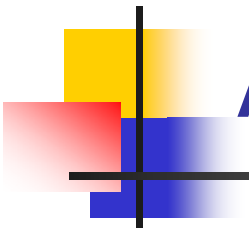
Secure Hash Algorithm (SHA)

- Endorsed by NIST
- Reduces input data of up to 2^64 bits to 160 bit digest
- Patterned after MD4 but creates a hash that is 160 bits in length instead of 128 bits
- The longer hash makes it more resistant to attacks
- SHA pads messages less than 512 bits with zeros and an integer that describes the original length of the message



Security Protocols

- Digital Signatures
- Key Exchange
- Authentication
- Some other protocols

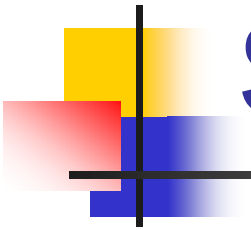


Authentication Protocol

- This is often used in password programs where a host needs to authenticate that a user is who they say they are
- Alice sends p to the host
- The host computes H(p)
- The host already has a table which has y and checks to see if H(p) = y
- H is a one-way function
  - A one-way function y=H(x) is a function where it is easy to compute y from x , but “hard” to compute x from y.

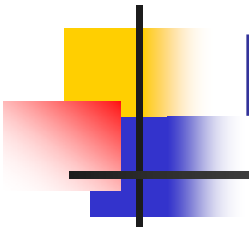
Authentication Protocol (cont.)

- Sequences of Passwords:
  - To make it less likely that Malory will be able to pretend to be Alice to the host
  - Sequences of passwords are used by having Alice give the host a random number R
  - The host computes x0 = R , x1 = H(R) , x2=H(H(R)),.....xn= H^n(R) and gives all but xn to Alice.
  - The host keeps xn



Sequences of Passwords

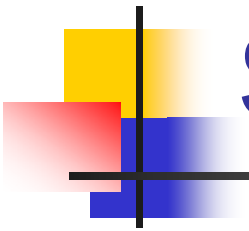
- The first time Alice logs on, she uses xn-1 as the password
- The host computes H(xn-1) and compares it to xn
- If the two are equivalent, Alice has been authenticated
- Now the host replaces xn with xn-1
- The next time Alice logs on, she uses xn-2
- Because each password is only used once by Alice, and H is a one-way function, sequences of passwords are more secure



Digital Signature

- In some cases, secrecy is not required but authentication is
- The data must be guaranteed to be that which was originally sent
- Using DS an encrypted hash of the message is transmitted along with the message
- Helps to prove that the person sending the message with a public key is whom he/she claims to be
- Also proves that the message was not altered and that it was sent in the first place

Desirable Properties of Digital Signatures



- Unforgeable
- Verifiable
- Non-repudiable
- Cheap to compute and verify
- Non-reusable
- No reliance on trusted authority
- Signed document is unchangeable

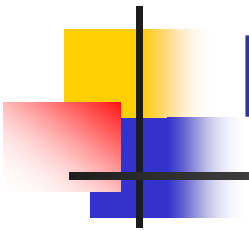
Digital Signature With Shared Key Encryption

- Requires a trusted third party
- Signer encrypts documents with secret key shared with third party
- Receiver checks validity of signature by consulting with trusted third party
- Third party required so receiver can't forge the signature

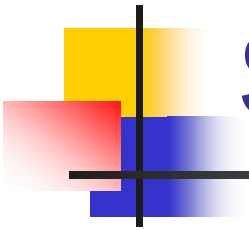
Digital Signature With Public Key Encryption

- Signer encrypts document with his private key
- Receiver checks validity by decrypting with signer's public key
- Only signer has the private key
  - So no trusted third party required
- But receiver must be certain that he has the right public key

Problems With Simple Encryption Approach

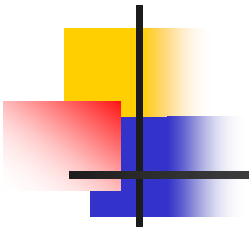


- Computationally expensive
  - Especially with public key approach
- Document is encrypted
  - Must be decrypted for use
  - If in regular use, must store encrypted and decrypted versions



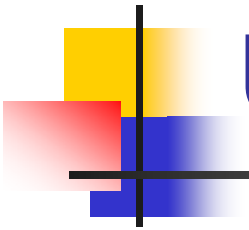
Secure Hash Algorithm

- A method of protecting data from modification
- Doesn’t actually prevent modification
- But gives strong evidence that modification did or did not occur
- Typically used with digital signatures



Idea Behind Secure Hashes

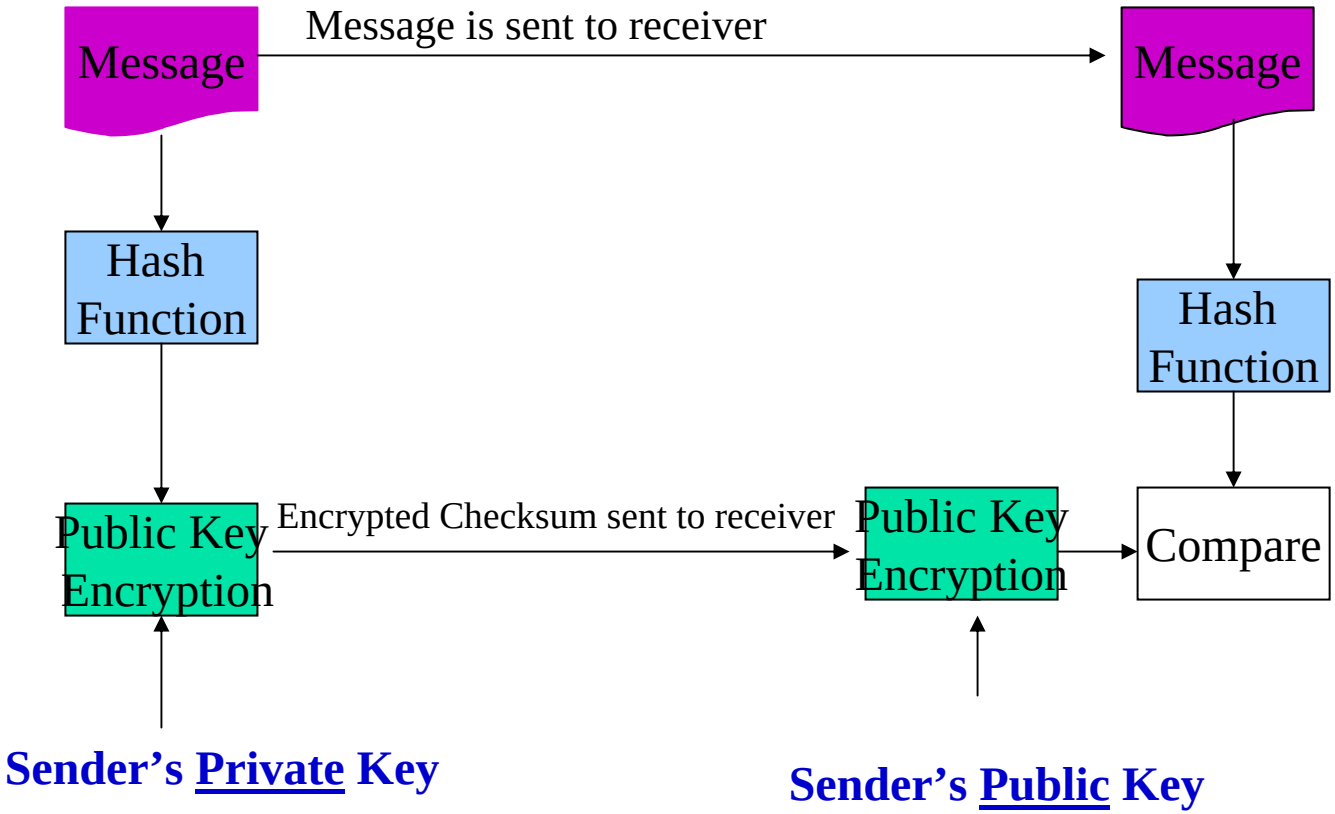
- Apply a one-way cryptographic function (hash) function to data in question
- Producing a much shorter result
- Attach the cryptographic hash to the data before sending
- When necessary, repeat the function on the data and compare to the hash value

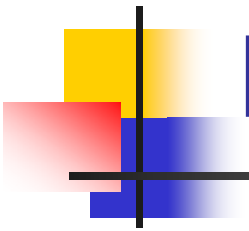


Use of Cryptographic Hashes

- Must assume opponent also has hashing function
- And it doesn't use secret key
- So opponent can substitute a different message with a different hash
- How to prevent this?
- And what (if anything) would secure hashes actually be useful for?

Hashing and Signatures





Benefits of Cryptography

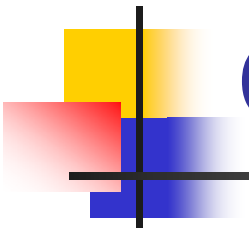
- Five key elements:
  - Confidentiality
  - Authentication
  - Integrity
  - Nonrepudiation
  - Access control

Benefits of Cryptography (continued)

Table How to implement the benefits of cryptography

Protection	Description	How Implemented
Confidentiality	Allow only authorized users to access the information.	Symmetric cryptography and asymmetric cryptography
Authentication	Verify the sender and trust the sender is whom they claim to be.	Asymmetric cryptography and hashing
Integrity	Trust the information has not been altered.	Hashing and digital signatures
Nonrepudiation	Ensure that the sender cannot deny a message was delivered.	Hashing and digital signatures
Access Control	Restrict availability to information.	Symmetric cryptography and asymmetric cryptography

Pretty Good Privacy (PGP) and GNU Privacy Guard (GPG)



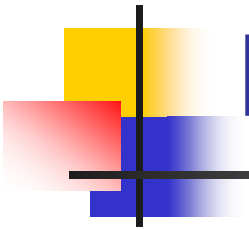
- PGP is perhaps most widely used asymmetric cryptography system for encrypting e-mail messages on Windows systems
  - Commercial product
- GPG is a free product

PGP How it Works

- PGP uses both Asymmetric and Symmetric key cryptography.
- The user creates a pair of keys (public and private)
- User's wishing to send him messages will encrypt the messages with the user's public key.
- The user encrypts the message with a symmetric key, and then encrypts the symmetric key with the public key of the intended recipient.
- Both the encrypted key and the message is then sent.
- The receiver's version of PGP will first decrypt the symmetric key with the private key supplied by the recipient , and then uses the resulting decrypted key to decrypt the rest of the message.

Pretty Good Privacy (PGP) and GNU Privacy Guard (GPG) (cont.)

- GPG versions run on Windows, UNIX, and Linux operating systems
- PGP and GPG use both asymmetric and symmetric cryptography
- PGP can utilize two different public key algorithms - RSA or Diffie Hellman.
- The RSA version uses IDEA algorithm to generate a short symmetric key to be used to encrypt the message and RSA to encrypt the short IDEA key.



PGP

- Functions much like S/MIME by encrypting messages using digital signatures
- A user can sign an e-mail message without encrypting it, verifying the sender but not preventing anyone from seeing the contents
- First compresses the message
  - Reduces patterns and enhances resistance to cryptanalysis
- Creates a session key (a one-time-only secret key)
  - This key is a number generated from random movements of the mouse and keystrokes typed

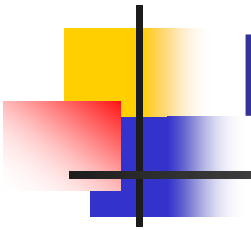
E-Mail Encryption

- Two technologies used to protect e-mail messages as they are being transported:
  - Secure/Multipurpose Internet Mail Extensions
  - Pretty Good Privacy PGP

Secure/Multipurpose Internet Mail Extensions (S/MIME)

- Protocol that adds digital signatures and encryption to Multipurpose Internet Mail Extension (MIME) messages
- Provides these features:
  - Digital signatures
  - Interoperability
  - Message privacy
  - Seamless integration
  - Tamper detection

Microsoft Windows Encrypting File System (EFS)

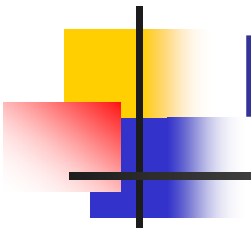


- Encryption scheme for Windows 2000, Windows XP Professional, and Windows 2003 Server operating systems that use the NTFS file system
- Uses asymmetric cryptography and a per-file encryption key to encrypt and decrypt data
- When a user encrypts a file, EFS generates a file encryption key (FEK) to encrypt the data

Microsoft Windows Encrypting File System (EFS) (continued)

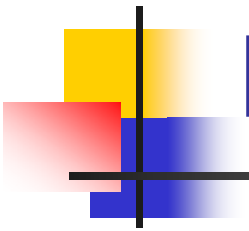
- The FEK is encrypted with the user's public key and the encrypted FEK is then stored with the file
- EFS is enabled by default
- When using Microsoft EFT, the tasks recommended are listed on page 293 of the text

UNIX Pluggable Authentication Modules (PAM)



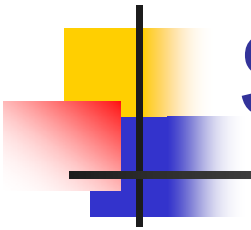
- When UNIX was originally developed, authenticating a user was accomplished by requesting a password from the user and checking whether the entered password corresponded to the encrypted password stored in the user database /etc/passwd
- Each new authentication scheme requires all the necessary programs, such as login and ftp, to be rewritten to support it

UNIX Pluggable Authentication Modules (PAM) (continued)



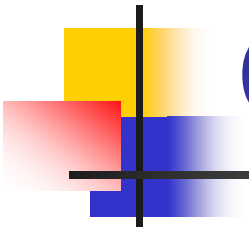
- A solution is to use PAMs
- Provides a way to develop programs that are independent of the authentication scheme

Linux Cryptographic File System (CFS)



- Linux users can add one of several cryptographic systems to encrypt files
- One of the most common is the CFS
- Other Linux cryptographic options are listed on pages 294 and 295 of the text

Characteristics of Good Ciphers



- Well matched to requirements of application
  - Amount of secrecy required should match labor to achieve it
- Freedom from complexity
  - The more complex algorithms or key choices are, the worse

Characteristics of Good Ciphers (cont.)

- Simplicity of implementation
  - Seemingly more important for hand ciphering
  - But relates to probability of errors in computer implementations
- Errors should not propagate

Characteristics of Good Ciphers (cont.)

- Ciphertext size should be same as plaintext size
- Encryption should maximize *confusion*
  - Relation between plaintext and ciphertext should be complex
- Encryption should maximize *diffusion*
  - Plaintext information should be distributed throughout ciphertext