CS458/CS558
Introduction to Computer Security

Course Information

- Instructor: Ping Yang
  - Office: P11, Engineering Building
  - Email: pyang@binghamton.edu
  - Office hours (via zoom):
    Mon. & Fri.: 11:30am - 12:30pm
    (Starting Sept. 1 2023)
    Office hours zoom link:
    https://binghamton.zoom.us/j/91807068587

Course Information

- Teaching Assistants

Meng Wang
Email: mwang150@binghamton.edu
Office hours (hybrid): Tue. & Wed. 11am-12:30pm
(starting Aug. 29, 2023)
Office: Room TBA, Engineering Building (in-person)
https://binghamton.zoom.us/j/91807068587 (zoom)

Priti Wakodikar
Email: pwakodi1@binghamton.edu
Office hours (zoom before Sept. 8 and hybrid after Sept.8):
Thur. & Fri. 12:30pm-2pm (starting Aug. 31, 2023)
Office: Room TBA, Engineering Building (in-person)
https://binghamton.zoom.us/j/91807068587 (zoom)

Course Materials

- Textbook (recommended, not required)
  - William Stallings, Cryptography and Network Security Principles and Practice, Fourth/Fifth Edition, ISBN-10: 0-13-187316-2, ISBN-13: 978-0-13-187316-2

- Course website
  http://www.cs.binghamton.edu/~pyang/cs558F23.html
  contains links to some online resources.

- Course materials are available on brightspace system.
  http://brightspace.binghamton.edu
  - Submitting assignments
  - Checking grades

Course Info (Cont.)

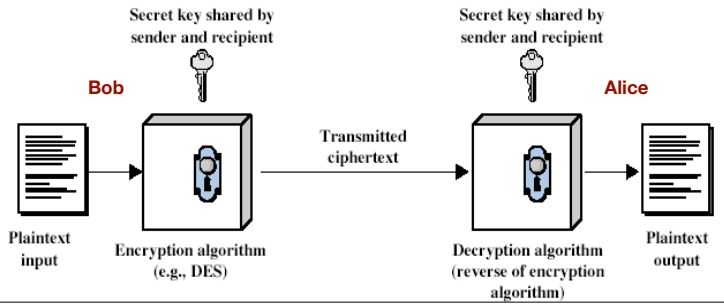
- The CS administrator will create an account for each student on remote.cs.binghamton.edu
- Macbook/Linux: type "ssh remote.cs.binghamton.edu" on terminals.
- Windows: Download SSH secure shell client to access remote.cs.

Prerequisites

- Proficient with programming in C, C++, Java or Python
- Comfortable working and programming in the Unix environment.

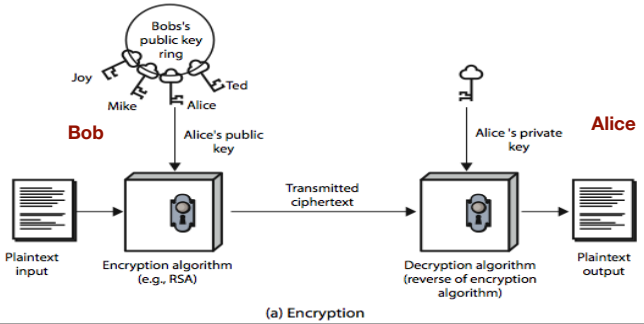
Topics

- A broad introduction to network, computer and information security.
- Topics may include:
  - Cryptography: encryption and decryption techniques
    - Symmetric encryption



Topics

- A broad introduction to network, computer and information security.
- Topics include:
  - Cryptography: encryption and decryption techniques
    - Public-key encryption



Topics

- A broad introduction to network, computer and information security.
- Topics include:
  - Cryptography: encryption and decryption techniques, key management, digital signature, authentication protocols.
  - Web/Email Security: socket programming, SSL, scam websites, phishing emails, social networking security
  - Systems Security: intrusion detection, malicious software.
  - Security Policies and Principles: confidentiality, integrity, availability, access control.
  - Attacks: Buffer overflow attack, SQL injection attack, identity theft, social engineering, deepfake.

Grading

- Three assignments: 44%
  - Assignment 1 (Programming, individual): 15%
  - Assignment 2 (Written, individual/group): 14%
  - Assignment 3 (Programming, individual): 15%
- Course project (presentation/programming/K-12 education/systems, individual/group): 18%
  - Long programming (MS students): must choose programming project
- Exam 1: 18% (in-person, open-book, open-notes)
- Exam 2: 18% (in-person, open-book, open-notes)
- Attendance: 2%

Attendance

- Attendance: 2%
  - Require to attend at least 10 in-person classes (or equivalent) to receive full attendance points.
  - 1 zoom attendance = 0.5 in-person attendance
  - 10 zoom + 5 in-person = 10 in-person
- Attendance extra credits: up to 2%
  - After attending 10 in-person classes (or equivalent)
    - 0.12 point extra credits for each additional in-person attendance
    - 0.06 point extra credits for each additional zoom attendance.

Assignment Policies

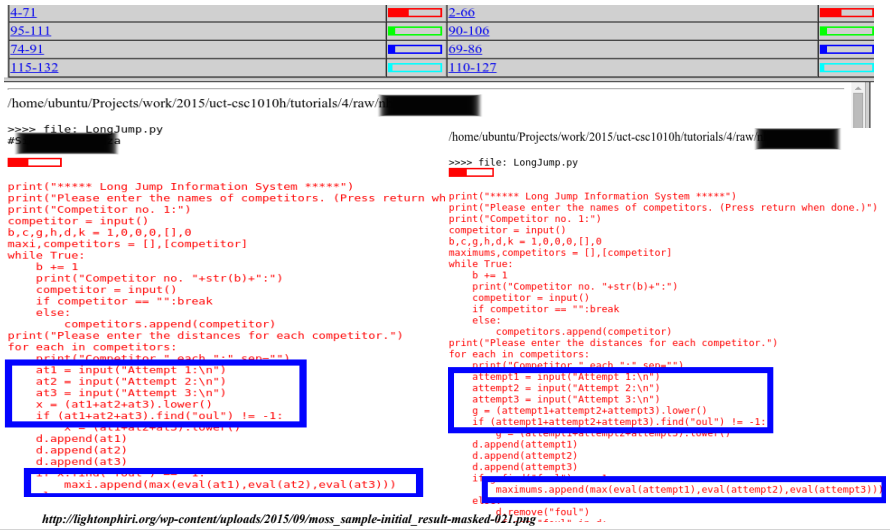
- Assignments
  - Start early, ask questions early, submit on time
  - No assignment will be accepted 24 hours after the deadline.
- Late penalty
  - 1-6hrs: 2.5 points
  - 6-12hrs: 5 points
  - 12-18hrs: 7.5 points
  - 19-24hrs: 10 points

Academic Integrity

- All students should follow Student Academic Honesty Code (http://www2.binghamton.edu/watson/about/honesty-policy.pdf).
- You may discuss the problems with other students, however, you must write your own codes and solutions. Discussing algorithms and solutions to the problem is NOT acceptable.
- Copying an assignment from another student or allowing another student to copy your work.
  - Report to the department and Watson school
  - 0 in the assignment/F in the course

Academic Integrity

- We will use Moss, to detect plagiarism.



http://lightonphiri.org/wp-content/uploads/2015/09/moss_sample-initial_result-masked-021.png

Academic Integrity

- Use chmod 700 <directoryname> command to change the permissions of your working directories before you start working on the assignments.
- The use of generative AI tools for assignments or course project is prohibited (unless permitted by the instructor).
- If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult me before you collaborate.

Sickness & Weather

- If you have covid/flu symptoms, please attend classes through zoom or watch recorded lectures.
- Heavy snow or icy road — the university may cancel classes.

Cybersecurity-focused Program

Cybersecurity Track Within MS Computer Science Program

Required	Elective
1. CS558 Introduction to Computer Security	1. CS527 Mobile Systems Security
2. CS559 Science of Cybersecurity	2. CS553 Software Security
	3. CS580A Hardware and System Security
	4. CS528 Computer Network
	5. CS536 Introduction to Machine Learning
	6. CS580T Topics on Data privacy

Advanced Certificate in Cybersecurity

Foundations	Design	Analysis	Applications
1. Fundamentals of Computer Security (WTSN 551)	1. Cyber-Physical Systems Security (EECE 567)	1. Science of Cybersecurity (CS559)	1. Network Computer Security (EECE 580F)
2. Introduction to Computer Security (CS558)	2. Operating Systems (CS 550)	2. Fundamentals of Steganography (EECE 562)	2. Software Security (CS580c)
		3. Network Security (EECE 657)	3. Mobile Systems Security (CS527X)
		4. Cryptography and Information Security (EECE 560)	4. Operating Systems (CS550)
		5. Hardware-Based Security (EECE 658)	5. Cyber-physical Systems Security (EECE 567)
		6. Contemporary Stats Cybersecurity (EECE 580I)	6. Hardware and Systems Security (CS580A)

https://www.binghamton.edu/watson/graduate/cybersecurity-certificate.html

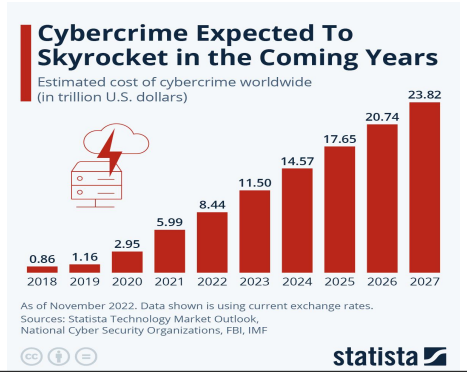
Introduction to Computer Security

What is Computer Security?

- Computer Security (or Cybersecurity) refers to the practice of protecting computer systems, networks, and data from unauthorized access, damage, and disruption.

Why is Cybersecurity Important?

- Protect sensitive information.
- Safeguard critical infrastructure (power grid, transportation, etc.)
- Defend against cybercrimes and prevent financial losses.



Why Is Cybersecurity Important? (Cont.)

Nearly 1 billion emails were exposed in 2021, affecting 1 in 5 internet users.

Data breaches cost businesses an average of $4.35 million in 2022.

53.35 million US citizens were affected by cybercrime in the first half of 2022.

Source (aag-it.com)

Around 236 million ransomware attacks occurred globally in the first half of 2022.

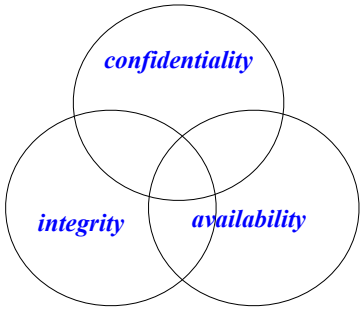
Phishing is the most common cyber threat facing businesses and individuals.

Objectives

- Consists of measures to deter, prevent, detect, and correct security violations that involve the transmission & storage of information

Security Goals: CIA Triad

- Computer security rests on three basic components: confidentiality, integrity, and availability.



Confidentiality, Integrity and Availability

- Confidentiality: only authorized people or system can access the data or resource

Confidentiality, Integrity and Availability

- Confidentiality: only authorized people or system can access the data or resource
- Integrity: assurance that the information is authentic and complete.
  - Data integrity: the assurance that data received is exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay)
  - Origin integrity: the source of data is trustworthy

Confidentiality, Integrity and Availability

- Confidentiality: only authorized people or system can access the data or resource
- Integrity: assurance that the information is authentic and complete.
  - Data integrity: the assurance that data received is exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay)
  - Origin integrity: the source of data is trustworthy
- Availability: people have the ability to use the information or resource desired

Examples: Security Violation

- User *A* transmits a file, which contains sensitive information to user B. User *C*, who is not authorized to read the file, is able to monitor the transmission and capture a copy of the file during its transmission
- User *A* sends a message to B. User *C* modifies the message during its transmission.
- User *C* impersonates *A* to send a message to B.

OSI Security Architecture

OSI Security Architecture

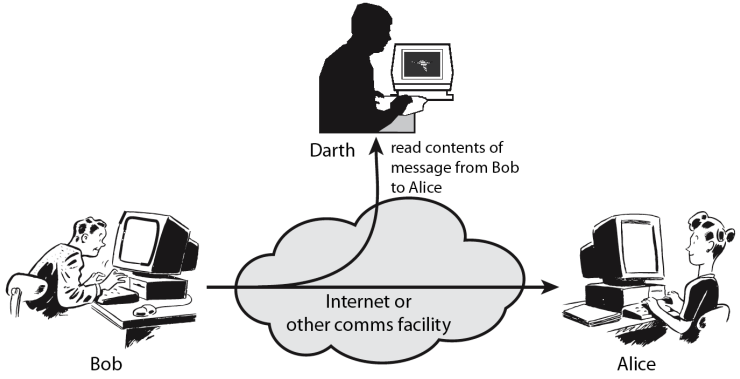
- ITU-T X.800: Security Architecture for OSI
  - ITU-T: International Telecommunication Union, Telecommunication standardization sector
  - OSI: Open Systems Interconnection - an effort to standardize networking
    - Started in 1982 by the International Organization for Standardization (ISO)
  - Systematic way of defining the requirements for security
- 3 aspects of information security:
  - Security attacks
  - Security mechanisms
  - Security services

Security Attacks

- Any action that compromises the security of information owned by an organization
- Information security: how to prevent attacks and to detect attacks on information-based systems
- Can focus of generic types of attacks
  - Passive
  - Active

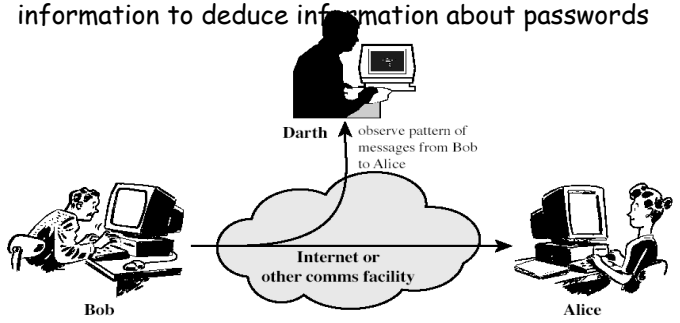
Passive Attacks

- Attempts to learn or make use of the information from the system but does not affect system resources
- The release of mesg. contents: eavesdropping on or monitoring of transmissions.



Passive Attacks

- Traffic analysis: may not be able to extract the information (encryption), but might still be able to observe the pattern of these massages
  - Observe the frequency and length of messages being exchanged.
  - Example: timing attack on the SSH protocol used timing information to deduce information about passwords

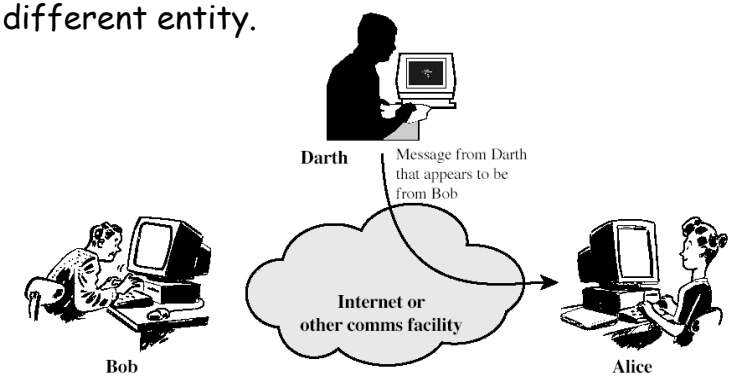


Passive Attacks

- Very difficult to detect because they do not involve any alteration of the data
- It is feasible to prevent the success of these attacks.
- The emphasis in dealing with passive attacks is on prevention rather than detection.

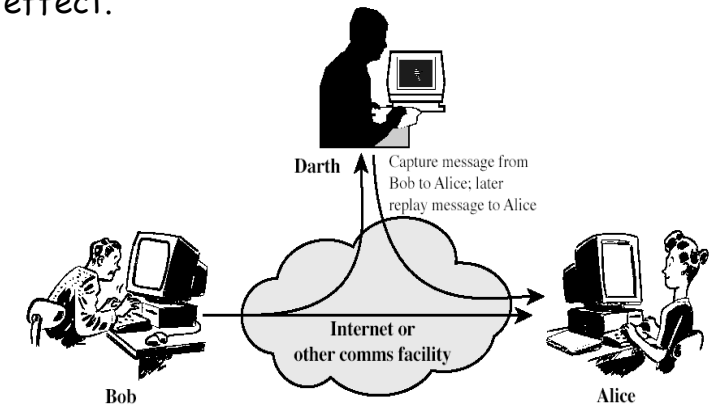
Active Attacks: Masquerade

- Attempts to alter system resources or affect their operation.
- Masquerade: one entity pretends to be a different entity.



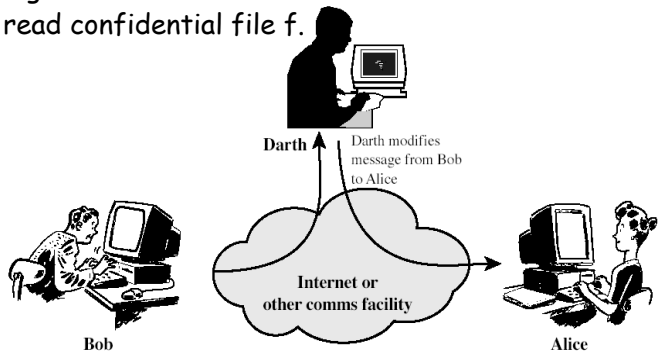
Active Attacks: Replay

- Replay: capture the data unit and transmit to the receiver later to produce an unauthorized effect.



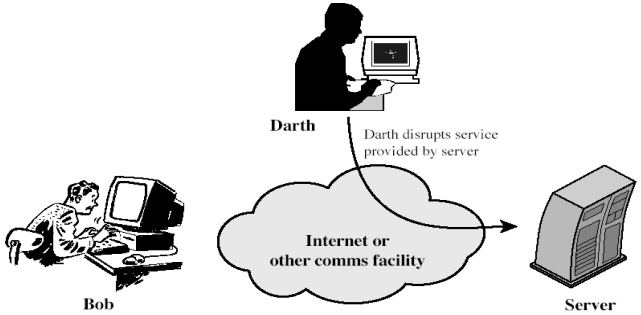
Active Attacks: Modification of Mesg.

- Modification of messages: some portion of a legitimate message is altered, or messages are delayed or reordered
  - E.g. Allow Tom to read confidential file f → allow Darth to read confidential file f.



Active Attacks: DOS

- Denial of service: prevents or inhibits the normal use or management of communications facilities
  - E.g. An entity may suppress all messages directed to a particular destination
  - E.g. disruption of an entire network by overloading it with messages so as to degrade performance



Examples of Recent Cyber Attacks

EQUIFAX

TARGET

solarwinds

RANSOMWARE

Target Data Breach

- During the holiday season of 2013, cybercriminals stole about 40 million credit and debit card information.
- Hackers gained access to Target's network through a third-party HVAC contractor and installed malware on the point-of-sale systems.
- Substantial financial losses (estimated $148 million) and reputation damage to target.
- CEO resigned.

Equifax Data Breach

- Occurred in 2017 at the American credit bureau Equifax.
- Leaked information: Name, DoB, Address, SSN, etc.
- Impacted as many as 148 million U.S. consumers.
- Settlements
  - $300 million for victim compensation.
  - $175 million to 48 states, DC, and Puerto Rico.
  - $100 million fines to the Consumer Financial Protection Bureau.

New Jersey Hospital Ransomware Attack

- In 2020, University Hospital New Jersey in Newark paid a $670K ransom to prevent publishing stolen patient info.
- A ransomware known as SunCrypt, infiltrated a hospital network, stole unencrypted files, and encrypted the data.
- The network was compromised after an employee fell for a phishing scam and provided their network credentials.

Hospital Ransomware Attack (Aug. 2023)

A cyberattack has disrupted hospitals and health care in several states

A cyberattack has affected computers at hospitals in multiple states, forcing some emergency rooms to close and ambulances to be diverted

By PAT EATON-ROBB Associated Press
August 4, 2023, 9:42 AM

Hospital Ransomware Attack (Aug. 2023)

https://www.youtube.com/watch?v=5QN14HGffd0

SolarWind Supply Chain Attack (2020)

- An attack on SolarWinds' Orion software, which provides tools for network and infrastructure monitoring.
  - Malicious code was injected into Orion software updates.
  - Numerous government agencies and private companies unknowingly installed the compromised updates, granting the hackers unauthorized access to their systems.
  - Victims include Department of Treasury, Department of Commerce, Department of Homeland Security, etc.

solarwinds

Security Services

Security Services

- Provided by a system to give a specific kind of protection to system resources.
- Intended to counter security attacks
- Using one or more security mechanisms
- ITU-T X.800 divides these services into 5 categories and 14 specific services.

Security Services (X.800)

- Authentication: assurance that the communicating entity is the one claimed
- Access control: prevention of the unauthorized use of a resource
  - Controls who can have access to a resource.

Security Services (X.800)

- Data confidentiality: protection of data from unauthorized disclosure
  - Protection of transmitted data from passive attacks.
  - Broader service: protects all user data transmitted between two users over a period of time.
  - Narrower service: protection of a single message or specific fields within a message

Security Services (X.800)

- Data integrity: assurance that data received is as sent by an authorized entity
  - Integrity can apply to a stream of messages, a single message, or selected fields within a message.
  - Most useful: total stream protection
    - Connection-oriented integrity service: assures that messages are received as sent with no duplication, insertion, modification and denial of service

Security Services (X.800)

- Nonrepudiation: protection against denial by one of the parties in a communication
  - Proof that the message was sent by the specified party
  - Proof that the message was received by the specified party

Security Mechanism

- Feature designed to detect, prevent, or recover from a security attack
- No single mechanism that will support all services required
- However one particular element underlies many of the security mechanisms in use:
  - cryptographic techniques

Security Mechanisms (X.800)

- Specific security mechanisms:
  - Encipherment: the use of mathematical algorithms to transform data into a form that is not readily intelligible
  - Digital signatures: data appended to a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery
  - Access control: enforce access rights to resources
  - Data integrity: assure the integrity of a data unit or stream of data units.

Security Mechanisms (X.800)

- Specific security mechanisms:
  - Authentication exchange: ensure the identity of an entity by means of information exchange.
  - Traffic padding: the insertion of bits into gaps in a data stream to frustrate traffic analysis
    - Make it difficult for an attacker to distinguish between true data flow and noise
    - Make it difficult to deduce the amount of traffic.

Relationship Between Security Services and Mechanisms

	Mechanism							
Service	Enciph-erment	Digital signature	Access control	Data integrity	Authenti-cation exchange	Traffic padding	Routing control	Notari-zation
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y						Y	
Traffic flow confidentiality	Y					Y	Y	
Data integrity	Y	Y		Y				
Non-repudiation		Y		Y				Y
Availability				Y	Y			

CS458/CS558: Introduction to Computer Security

Basic Terminology

Crypto

- Crypto: from Greek "krypto" = hide
- Cryptology: science of hiding
  - Cryptography: making "secret codes".
  - Cryptanalysis: breaking "secret codes".

Crypto Terminology

- Encryption: converting plaintext to ciphertext
- Decryption: restoring plaintext from ciphertext
- Plaintext: original message
- Ciphertext: coded message

- Cipher: an algorithm for performing encryption
- Secret key: the input of encryption algorithm. The key is independent of the plaintext and the alg..

Cryptography

- Characterize cryptographic system by:
  - Type of encryption operations used
    - Substitution: each element (a bit or a letter) in the plaintext is mapped into another element
    - Transposition: elements in the plaintext are rearranged.
    - Product: multiple stages of substitutions and transpositions
  - Number of keys used
    - Symmetric, Single-key encryption
    - Asymmetric, Two-key or Public-key encryption
  - Way in which plaintext is processed
    - Block: process one block of elements at a time, producing an output block for each input block
    - Stream: process the input elements continuously, producing one element at a time.

Cryptanalysis

- Objective of attacking an encryption system: recover key rather than simply to recover the plaintext of a single ciphertext.
- General approaches:
  - Cryptanalytic attack
    - Rely on the nature of the algorithm + some knowledge of the plaintext (e.g. English or French) or some sample plaintext-ciphertext pairs.
  - Brute-force attack
    - Try every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.

Chapter 2

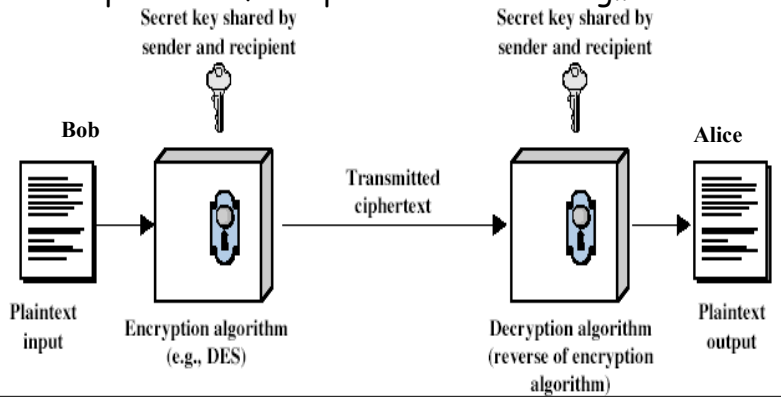
Symmetric Encryption

Symmetric Encryption

- A form of cryptosystem in which encryption and decryption are performed using the same key – single-key encryption
- Was only type prior to invention of public-key in 1970's, and by far most widely used.

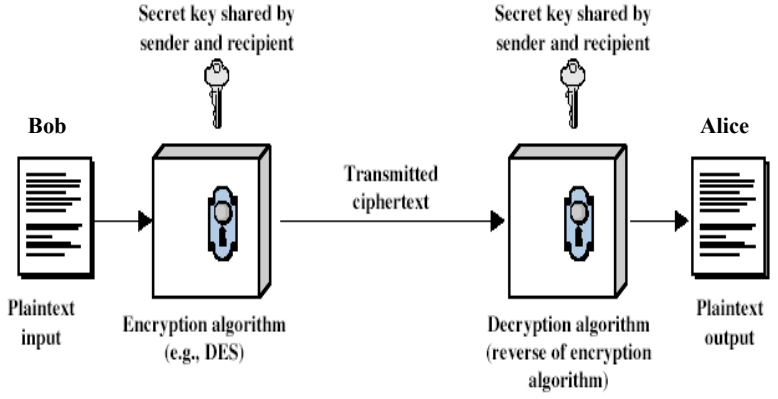
Symmetric Cipher Model

- Encryption algorithm: performs various substitutions a and transformation on the plaintext.
- Secret key: the input of encryption algorithm. The key is independent of the plaintext and the alg..



Symmetric Cipher Model

- Decryption algorithm: the ciphertext and the secret key and produces the original plaintext.



Requirements

- Two requirements for secure use of symmetric encryption:
  - Assume encryption algorithm is known
  - Encryption algorithm needs to be strong so that an attacker should be unable to discover the key or decrypt a ciphertext without knowing the key.
  - Mathematically have:

X: plaintext, K: encryption key, Y: ciphertext

$$Y = E(K,X)$$

$$X = D(K,Y)$$

An opponent can observe Y, but do not have access to K or X.

Requirements

- Two requirements for secure use of symmetric encryption:
  - A secret key known only to sender/receiver
    - Sender and receiver must have obtained copies of secret key in a secure fashion and must keep the key secure.
    - A third party could generate the key and securely deliver it to both source and destination.
    - If someone can discover the key and knows the algorithm, all communication using this key is readable.

Unconditional Security

- Unconditional security
  - No matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to determine the corresponding plaintext.

Computational Security

- Computational security
  - Given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken
    - The time required to break the cipher exceeds the useful lifetime of the information

Brute Force Search

- Simply try every 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 - proportional to key size
- DES: 56-bit, triple DES: 168-bit, AES: > 128 bits
- Time required for various key spaces:

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/μs	Time required at 10^6 decryptions/μs
32	$2^{32} = 4.3 \cdot 10^9$	2^{31} μs= 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \cdot 10^{16}$	2^{55} μs= 1142 years	10.01 hours
128	$2^{128} = 3.4 \cdot 10^{38}$	2^{127} μs= $5.4 \cdot 10^{24}$ years	$5.4 \cdot 10^{18}$ years
168	$2^{168} = 3.7 \cdot 10^{50}$	2^{167} μs= $5.9 \cdot 10^{36}$ years	$5.9 \cdot 10^{30}$ years

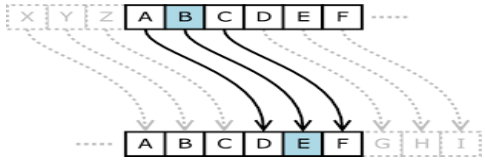
Substitution Cipher

Classical Substitution Ciphers

- Letters of plaintext are replaced by other letters or by numbers or symbols

Caesar Cipher

- The earliest known substitution cipher (by Julius Caesar)
- First attested use in military affairs
- Replaces each letter with the letter standing K places further down the alphabet.
- K = 3:

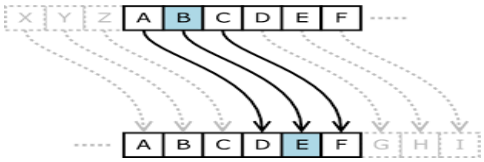


- Example:

```
Plaintext:  meet me after the toga party
```

Caesar Cipher

- The earliest known substitution cipher (by Julius Caesar)
- First attested use in military affairs
- Replaces each letter with the letter standing K places further down the alphabet.
- K = 3:



- Example:

```
Plaintext:  meet me after the toga party
```

```
Ciphertext: PHHW PH DIWHU WKH WRJD SDUWB
```

Cryptanalysis of Caesar Cipher

Cryptanalysis of Caesar Cipher

- A brute force search can be easily performed: simply try all the 25 possible keys - far from security.
  - The language of the plaintext is known and easily recognizable.
- The input may be compressed, make recognition difficult, e.g., E.g. .zip file.

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Caesar Cipher: Attack

- Decrypt the following ciphertext encrypted using the caesar ciper. (The plaintext is an English word)

ROVVY

Caesar Cipher: Attack

- Decrypt the following ciphertext encrypted using the caesar ciper (the plaintext is an English word).

ROVVY

1 q n u u x
2 p m t t w
3 o l s s v
4 n k r r u
5 m j q q t
6 l i p p s
7 k h o o r
8 j g n n q
9 i f m m p
10 h e l l o

Monoalphabetic Cipher

- Allow an arbitrary substitution rather than just shifting the alphabet
- Each plaintext letter maps to a random ciphertext letter

```
Plain:  abcdefghIjklmnopqrstuvwxyz
Cipher:DKVQFIBJWPESCXHTMYAUOLRGZN
```

E.g.

If we wish to replace letters

Monoalphabetic Cipher

- Allow an arbitrary substitution rather than just shifting the alphabet
- Each plaintext letter maps to a random ciphertext letter

```
Plain: abcdefghIjklmnopqrstuvwxyz
Cipher:DKVQFIBJWPESCXHTMYAUOLRGZN
```

E.g.

```
If we wish to replace letters
WI RF RWAJ UH YFTSDVF SFUUFYA
```

Monoalphabetic Cipher Security

- Number of mappings

Monoalphabetic Cipher Security

- The number of mappings
  - 26! = $4*10^{26}$ mappings
- With so many mappings, might think is secure

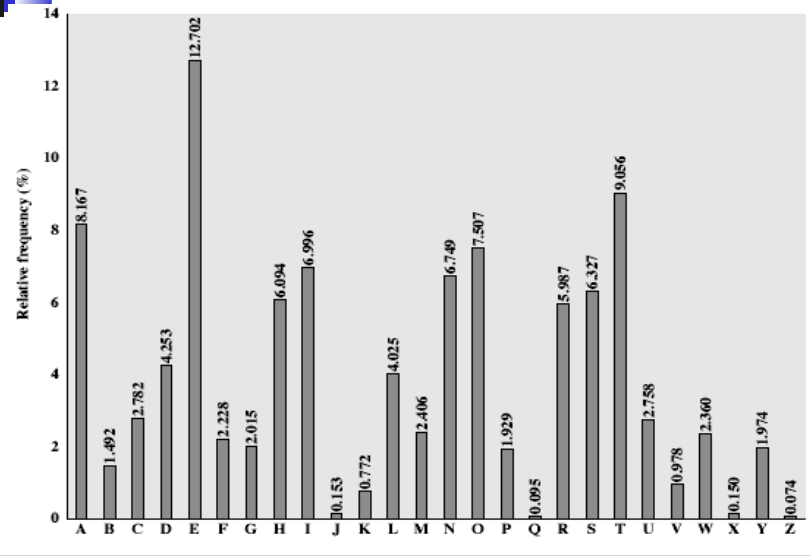
Monoalphabetic Cipher Security

- The cipher line can be any permutation of the 26 alphabetic characters
  - 26! = $4*10^{26}$ mappings
- With so many mappings, might think is secure
- But would be WRONG – if the cryptanalyst knows the nature of the plaintext (e.g. noncompressed English text), then the analyst can exploit the regularities (frequency of letters) of the language

Language Redundancy and Cryptanalysis

- Human languages are redundant. Letters are not equally commonly used.
- In English E is by far the most common letter, followed by T,A,O,I,N,S,R, other letters like Z,J,K,Q,X are fairly rare.
  - If the mesg. is long enough, this technique alone may be sufficient.

English Letter Frequencies



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

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33 (16/120)	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

- P → e

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

- P → e, Z → t

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

- P → e, Z → t, {S,U,O,M,H} → {a,h,i,n,o,r,s}

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

- P → e, Z → t, {S,U,O,M,H} → {a,h,i,n,o,r,s}
- Most common pair: ZW → and hence ZWP → , ZWSZ →

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

- P →e, Z → t, {S,U,O,M,H} → {a,h,i,n,o,r,s}
- Most common pair: ZW → th and hence ZWP → , ZWSZ →

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

- P →e, Z → t, {S,U,O,M,H} → {a,h,i,n,o,r,s}
- Most common pair: ZW → th and hence ZWP →the, ZWSZ →

Example

- UZQSOVUOHXMOPVGPOZPEVAGZWSZOPFPESXUDBMETSXAIZVUEPHZ HMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSUFPOMBZW PFUPZHMDJUDTMOGMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	Tt 2.50	J 0.83	N 0.00
O 7.50	X 4.17	A 1.67	I 0.83	R 0.00
M 6.67				

- P →e, Z → t, {S,U,O,M,H} → {a,h,i,n,o,r,s}
- Most common pair: ZW → th and hence ZWP →the, ZWSZ → that
- Finally: it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow

Playfair Cipher

- Not even the large number of mappings ($4*10^{26}$ mappings) in a monoalphabetic cipher provides security
- One approach to improving security is to encrypt multiple letters – e.g. Playfair Cipher.

Playfair Key Matrix

- A 5X5 matrix of letters based on a keyword
- Fill in letters of keyword (minus duplicates) from left to right and from top to bottom
- Fill rest of matrix with other letters
- Eg. using the keyword MONARCHY
- I and j count as one letter.

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair Cipher: Encryption

- Plaintext is encrypted two letters at a time
  - If a pair is a repeated letter, insert filler x, e.g. balloon → ba lx lo on
  - If both letters fall in the same row, replace each with letter to right with the first element of the row circularly following the last, e.g. ar → rm
  - If both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom), e.g. mu → cm

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair Cipher: Encryption

- Plaintext is encrypted two letters at a time
  - Otherwise, each letter is replaced by the letter in the same row and in the column of the other letter of the pair, e.g. hs → bp, ea → im (or jm)

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair Cipher: Decryption

- To decrypt, use the inverse of the encryption rules and drop any extra x that does not make sense in the final message.
  - Decrypts two letters at a time
  - If both letters fall in the same row, replace each with letter to left, e.g. rm → ar
  - If both letters fall in the same column, replace each with the letter above it, e.g. cm → mu

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair Cipher: Decryption

- Plaintext is encrypted two letters at a time
  - Otherwise, each letter is replaced by the letter in the same row and in the column of the other letter of the pair
  - E.g. bp → hs, im → ea

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Example

- Use the playfair cipher and key word commom to encrypt the message: gdddcgs

=

Example

- Use the playfair cipher and key word commom to encrypt the message: gdddcgs

gd dx dc gs

C	O	M	A	B
D	E	F	G	H
I/J	K	L	N	P
Q	R	S	T	U
V	W	X	Y	Z

gd => he

dx => fv

dc=> id

gs => ft

Security of Playfair Cipher

- Security much improved over monoalphabetic
- Would need a 676 (26*26) entry frequency table to analyse (verses 26 for a monoalphabetic)
- Was widely used for many years
  - eg. by US & British military in WW1
- It is relatively easy to break because it still leaves much of the structure of the plaintext language intact.

Transposition Ciphers

- Consider classical transposition or permutation ciphers
- Hide the message by rearranging the letter order without altering the actual letters used

Rail Fence Cipher

- Write message letters out diagonally over a number of rows
- Then read the letters row by row
- Encrypt the message "meet me after the toga party" with a rail fence of depth 2

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

Ciphertext: MEMATRHTGPRYETEFETEOAAT

- Think: how to encrypt the above message using rail fence cipher of depth 3?

Rail Fence cipher

- Write message letters out diagonally over a number of rows
- Then read the letters row by row
- Encrypt the message "meet me after the toga party" with a rail fence of depth 3

```
m   t   a   e   h   o   p   t
  e   m   f   r   e   g   a   y
    e   e   t   t   t   a   r
```

Ciphertext: MTAEHOPTEMFREGAYEETTTAR

- How to decrypt a ciphertext with 3 rows?

ciphertext: CPEERYOURCIMTSUT

Rail Fence cipher: Decryption

- Example:

ciphertext: **CPEERYOURCIMTSUT**

|row| = 3

- **|cipher| = 16**
- **16/3= 5, 16 mod 3 = 1 ➔**

1st row: 5+1 = 6 letters

2nd row: 5 letters, 3rd row: 5 letters

```
C P E E R Y
O U R C I
M T S U T
```

➔ Plaintext: **computersecurity**

Rail Fence cipher: Decryption

- How to decrypt a ciphertext
  - Let |row| be the number of rows
  - Compute the length of the ciphertext |cipher|
  - Compute the number of letters of each row
  - Write down the ciphertext row by row
  - Read the ciphertext diagonally

Rail Fence cipher

- Write message letters out diagonally over a number of rows
- Then read the letters row by row
- Encrypt the message "meet me after the toga party" with a rail fence of depth 3

```
m   t   a   e   h   o   p   t
  e   m   f   r   e   g   a   y
    e   e   t   t   t   a   r
```

Ciphertext: MTAEHOPTEMFREGAYEETTTAR

- How to decrypt a ciphertext with 3 rows?

ciphertext: CPEERYOURCIMTSUTA

Rail Fence cipher: Decryption

- Example:

ciphertext: **CPEERYOURCIMTSUTA**

|row| = 3

- **|cipher| = 17**
- **17/3= 5, 17 mod 3 = 2 ➔**

1st row: 5+1 = 6 letters

2nd row: 5+1 = 6 letters, 3rd row: 5 letters

```
C P E E R Y
O U R C I M
T S U T A
```

➔ Plaintext: **cotpuseruectriaym**

Row Transposition Ciphers

- A more complex transposition
- 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

Read the 3rd column

```
Key:        3 4 2 1 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

Row Transposition Ciphers

- A more complex transposition
- 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

```
Key:          3 4 2 1 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
```

Read the 3rd column

- how to decrypt a ciphertext using the above key?
  ciphertext: ATHNIERIPTSISORPNSOCZ

Row Transposition Ciphers: Decryption

- How to decrypt a ciphertext?
  ciphertext: ATHNIERIPTSISORPNSOCZ
  - |cipher| = 21, |key| = 7 → |row| = 3

```
Key:          3 4 2 1 5 6 7
Ciphertext:   T R A N S P O
              S I T I O N C
              I P H E R S Z
Plaintext:  transpositionciphers
```

Product Ciphers

- Ciphers using substitutions or transpositions are not secure because of language characteristics
- Hence consider using several ciphers in succession to make harder
  - Two substitutions make a more complex substitution
  - Two transpositions make a more complex transposition
  - But a substitution followed by a transposition makes a much harder cipher
    - This is bridge from classical to modern ciphers

Encryption Game (1)

Ciphertext: nGhAmToNbI IvErSiTyUn

Hint: reorder letters

Encryption Game (1)

Ciphertext: nGhAmToNbI IvErSiTyUn

Hint: reorder letters

Plaintext: Binghamton University

Substitution or transposition?

Encryption Game (2)

Ciphertext: 1,20,20,1,3,11, 1,20, 4,1,23,14,!

Hint: numeric to character

Encryption Game (2)

Ciphertext: 1,20,20,1,3,11, 1,20, 4,1,23,14,!

Hint: numeric to character

Plaintext: Attack at Dawn!

Substitution or transposition?

Encryption Game (3)

Ciphertext:, ugewtkva

Hint: caesar cipher

Encryption Game (3)

Ciphertext: ugewtkva

Hint: caesar cipher
K = 2

Plaintext: security

Substitution or transposition?

Chapter 3

Block Ciphers and the Data Encryption Standard (DES)

Block Ciphers

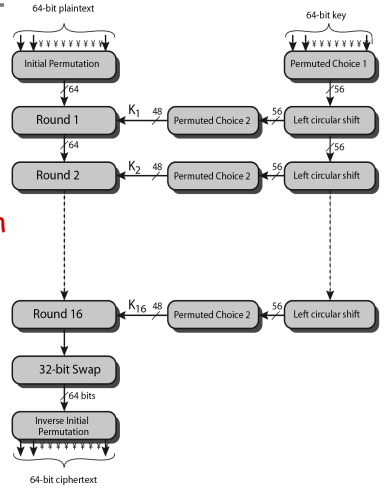
- Block ciphers: a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
  - Typically, a block size of 64 or 128 bits is used
  - Many current ciphers are block ciphers
  - Broader range of applications
  - DES (Data Encryption Standard): one of the most widely used cryptographic algorithms, especially in financial applications.

Data Encryption Standard (DES)

- Most widely used block cipher in world
- Developed in 1974 by IBM and the U.S. government
- 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 reverse the encryption.
- Use of DES has flourished, especially in financial applications

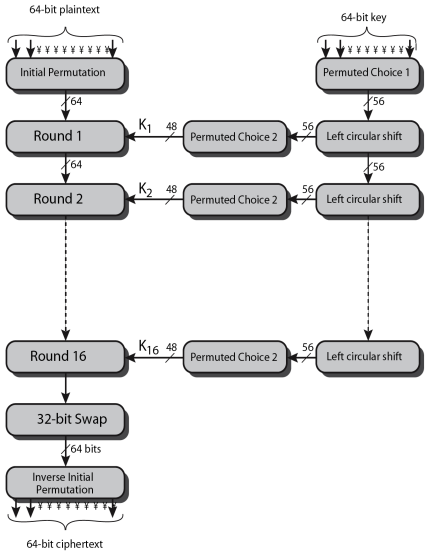
DES Encryption Overview

- The plaintext passes through an initial permutation (IP) that rearranges the bits in plaintext.
- Then passes through 16 rounds of the same function, which involves both permutation and substitution function
- The left and right halves of the output of the last round are swapped, which is then passed through a permutation that is the inverse of IP to produce the 64-bit ciphertext
  - $IP^{-1}(IP(M)) = M$



DES Encryption Overview

- 64-bit key is passed through a permutation function.
- For each 16 round, a subkey K_i is produced by the combination of a left circular shift and a permutation.
- Permutation function is the same for each round, but a different subkey is produce because of the repeated shifts of key bits



Initial Permutation (IP)

Input:

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Initial Permutation (IP)

Output:

M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_2
M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_4
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_6
M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_8
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_9	M_1
M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_3
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_5
M_{63}	M_{55}	M_{47}	M_{39}	M_{31}	M_{23}	M_{15}	M_7

Example

Input: 00100….0

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Example

Input: 00100….0

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

The 48th bit of the output is 1; others are 0

Example

Output: 00100….0

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Example

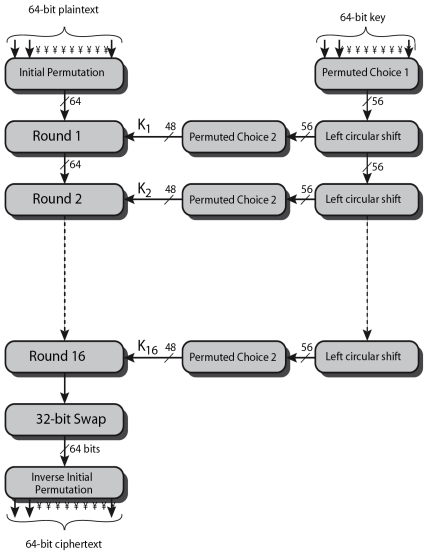
Output: 00100….0

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

The 42nd bit of the input is 1; others are 0

DES Encryption Overview

- The 64-bit key is passed through a Permuted Choice 1 table.



Permuted Choice One (PC-1)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Example: Permuted Choice One

Input: 00100001….0

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Example: Permuted Choice One

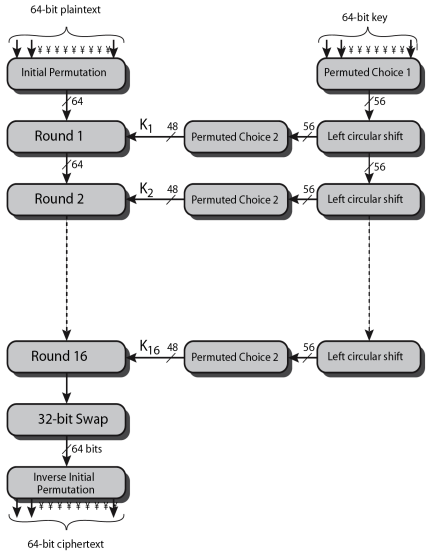
Input: 00100001….0

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Output: the 24th bit is 1; others are 0.

DES Encryption Overview

- For each 16 round, a subkey K_i is produced by the combination of a left circular shift and a permutation.
- The permutation function is the same for each round, but a different subkey is produce because of the repeated shifts of the key bits



Schedule of Left Circular Shifts

- The output of PC-1 is then treated as two 28 bits quantities, labeled C_1 and D_1.
- At each round, C_i and D_i are separately subjected to a circular left shift, of 1 or 2 bits.

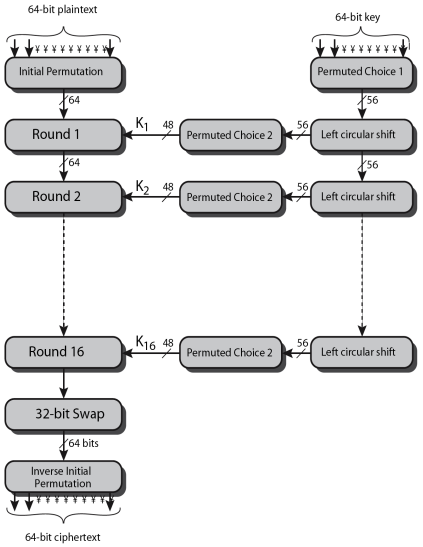
Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

DES Encryption Overview

- The left and right halves of the output of the last round are swapped, which is then passed through a permutation that is the inverse of IP to produce the 64-bit ciphertext
  - $IP^{-1}(IP(M)) = M$



Inverse Initial Permutation (IP^{-1})

Input:

M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_2
M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_4
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_6
M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_8
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_9	M_1
M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_3
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_5

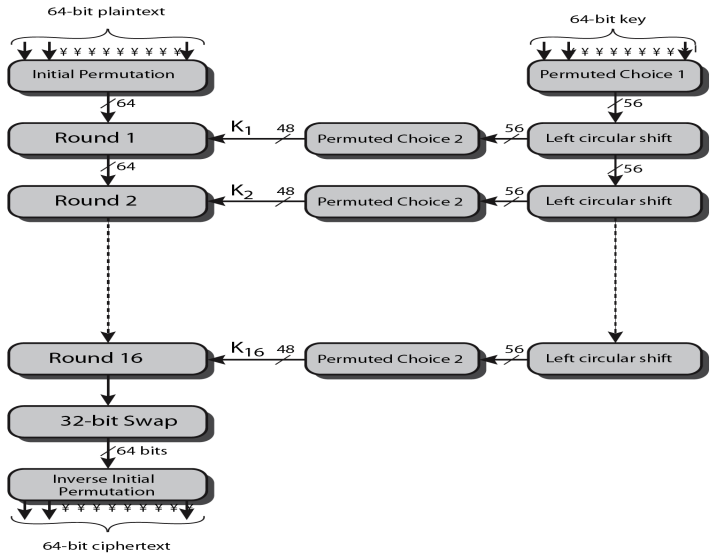
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Inverse Initial Permutation (IP^{-1})

Output:

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

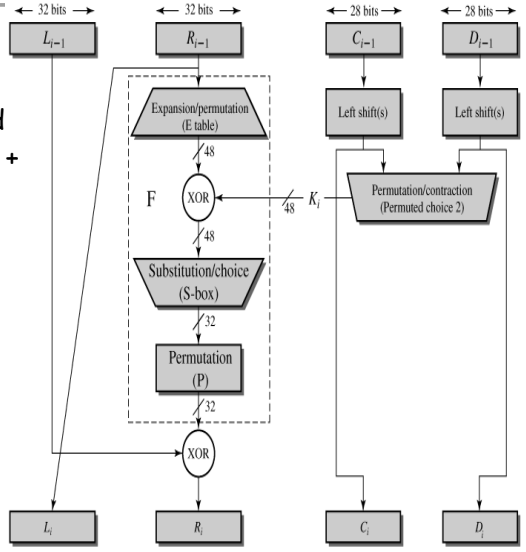
DES Encryption Overview



Single Round of DES Algorithm

- Uses two 32-bit L & R halves. Ki: 48 bits

1. The R input is expanded to 48 bits (permutation + duplication of 16 of R bits); the resulting 48 bits are XORed with Ki.



Expanded Permutation (E)

Input: 0011000….0

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Expanded Permutation (E)

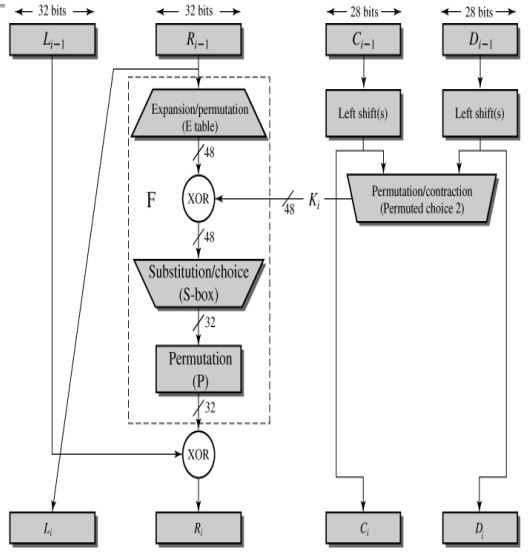
Input: 0011000….0

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

The 4th, 5th, and 7th bit in the output are 1

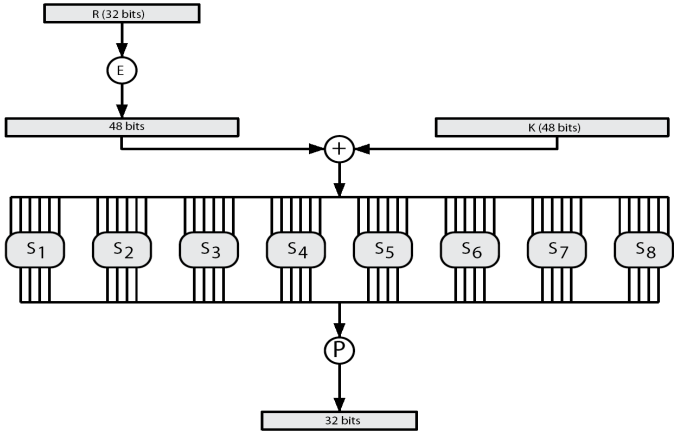
Single Round of DES Algorithm

2. The 48-bits then pass through a substitution function that produces 32-bit output.



S-Boxes

- The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.



S-Boxes

- The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.
  - The first and last bits of the input to S_i form a 2-bit binary number to select one of 4 substitutions defined by the four rows (0, 1, 2, 3) in the table for S_i.
  - The middle 4 bits select one of 16 columns (0-15).
  - E.g. in S_1, input 011001

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-Boxes

- The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.
  - The first and last bits of the input to S_i form a 2-bit binary number to select one of 4 substitutions defined by the four rows (0, 1, 2, 3) in the table for S_i.
  - The middle 4 bits select one of 16 columns (0-15).
  - in S_1, input 011001
    - The row is 01 (row 1)
    - The column is 1100 (column 12)
    - The value is 9 – output is 1001.

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-Boxes

- The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.
  - The first and last bits of the input to S_i form a 2-bit binary number to select one of 4 substitutions defined by the four rows (0, 1, 2, 3) in the table for S_i.
  - The middle 4 bits select one of 16 columns (0-15).
  - Assume the output is 4, what are possible input?

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-Boxes

- The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.
  - The first and last bits of the input to S_i form a 2-bit binary number to select one of 4 substitutions defined by the four rows (0, 1, 2, 3) in the table for S_i.
  - The middle 4 bits select one of 16 columns (0-15).
  - Assume the output is 4, what are possible inputs?

Row 0 column 1: 000010, row 1 column 3: 000111

Row 2 column 0: 100000, row 3 column 4: 101001

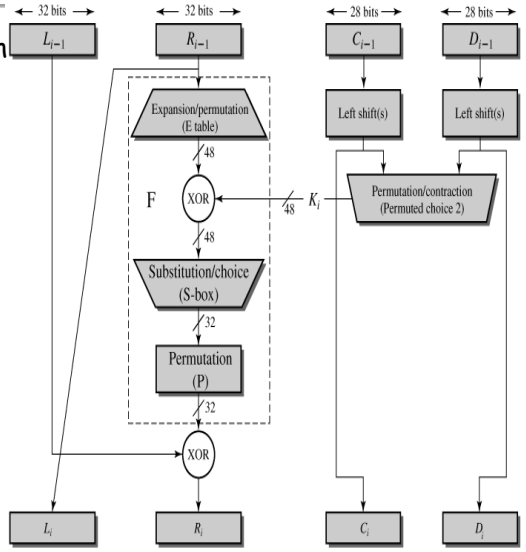
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-Box: Design Criteria

- The design of the round function focuses on the design of s-boxes and on the permutation P.
- The design was primarily aimed at thwarting differential cryptanalysis.
  - Any change to the input to an S-box should result in random-looking changes to the output
  - No output bit of any S-box should be too close a linear function of the input bits

Single Round of DES Algorithm

3. The 32-bit pass through a permutation function.



Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

- The four output bits from each S-box affect six different S-boxes on the next round.

Expanded Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

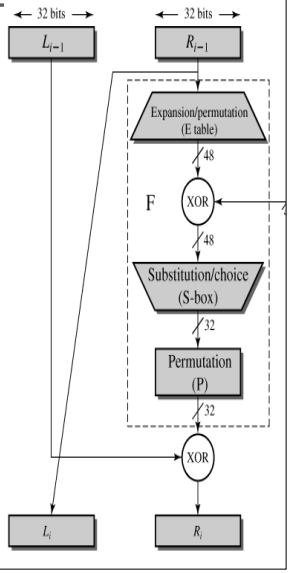
Property: The four output bits from each S-box affect six different S-boxes on the next round.

- Output bits of S1: 1 2 3 4
- After the P table:
  1 => 9, 2 => 17, 3 => 23, 4 => 31

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

9=>12: S2
9=> 14: S3
17=>24: S4
17=> 26: S5
23 => 34: S6
31 => 46: S8

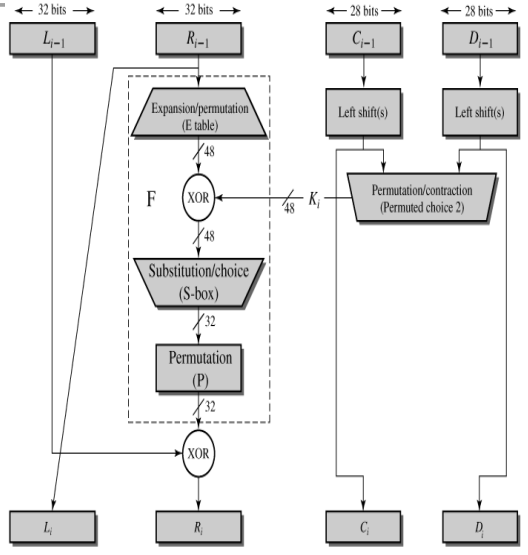
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



Single Round of DES Algorithm

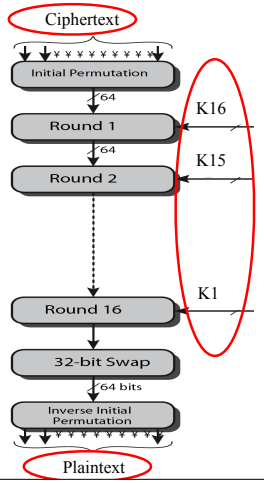
$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i)$



DES Decryption

- Decryption uses the same algorithm as encryption, except that subkeys are used in the reversed order.



DES – Key Size

- 56-bit keys have 2^{56} = 7.2 x 10^{16} values
- Brute force search looks hard
  - Performing one DES encryption per microsecond would take more than 1000 years to break the cipher
- Recent advances have shown is possible
  - In 1977, Diffie and Hellman: technology existed to build a parallel machine with 1 million encryption devices; each performs one encryption per microsecond ➔ 10 hours
  - In 1998, Electronic Frontier Foundation (EFF) had broken a DES encryption using a computer built for less than $250K. The attack took less than 3 days.

CS458/CS558 Introduction to Computer Security

Double-DES?

- Could use 2 DES encrypts on each block

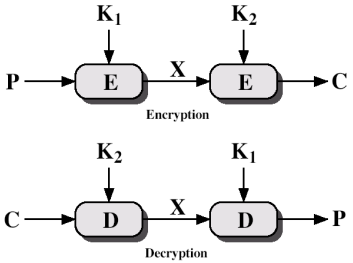
  $C = E(K2, E(K1, P))$

- Issue of reduction to single stage
  - Would it be possible to find a key K3 such that

    $E(K2, E(K1, P)) = E(K3, P)$
  - Answer: NO - proved in 1992

Double-DES?

- Meet-in-the-middle attack
  - Works whenever use a cipher twice
  - Based on the observation:



If C = E(K2, E(K1, P)),
then X = E(K1,P) = D(K2,C)

- Assume that the attacker has a set of plaintext-ciphertext pairs (P1,C1), (P2,C2), …, (Pn,Cn)
  - Encrypt P1 for all 2^{56} keys. Store the result in a table and then sort the table.
  - Decrypt C1 with 2^{56} keys and check the result against the table for a match.
  - If a match occurs, then test the two resulting keys against other plaintext pairs.

Double-DES?

Encrypt P1 using 2^56 Possible keys

K_1	X_1
K_2	X_2
…	…
K_i	X_i
…	…
$K_{2^{56}}$	$X_{2^{56}}$

Match, try (Ki, Kj) on (P2,C2), .., (Pn, Cn)

Decrypt C1 using 2^56 Possible keys

K_1	X'_1
K_2	X'_2
…	…
K_j	X'_j
…	…
$K_{2^{56}}$	$X'_{2^{56}}$

Double-DES?

Encrypt P1 using 2^56 Possible keys

K_1	X_1
K_2	X_2
…	…
K_i	X_i
…	…
$K_{2^{56}}$	$X_{2^{56}}$

Correct? (Ki, Kj) may be the key。

Decrypt C1 using 2^56 Possible keys

K_1	X'_1
K_2	X'_2
…	…
K_j	X'_j
…	…
$K_{2^{56}}$	$X'_{2^{56}}$

Double-DES?

Encrypt P1 using 2^56 Possible keys

K_1	X_1
K_2	X_2
…	…
K_i	X_i
…	…
$K_{2^{56}}$	$X_{2^{56}}$

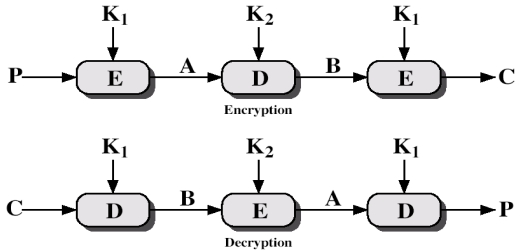
Incorrect? Find another match

Decrypt C1 using 2^56 Possible keys

K_1	X'_1
K_2	X'_2
…	…
K_j	X'_j
…	…
$K_{2^{56}}$	$X'_{2^{56}}$

Triple-DES with Two-Keys

- An obvious counter to the meet-in-the-middle attack is to use three stages of encryption with 3 different keys ➔ requiring key length 168-bits
- Can use 2 keys with E-D-E sequence
  - C = E(K1, D(K2, E(K1,P)))
  - The only advantage of the use of decryption for the second stage is: if K1=K2 then can work with single DES



Triple-DES with Three-Keys

- No current known practical attacks
  - Brute-force: 2^{112}
  - Differential cryptanalysis: > 10^{52} plaintext-ciphertex pairs
- Can use Triple-DES with 3 Keys to avoid these
  - C = E(K3, D(K2, E(K1, P)))
- Has been adopted by some Internet applications, eg PGP, S/MIME

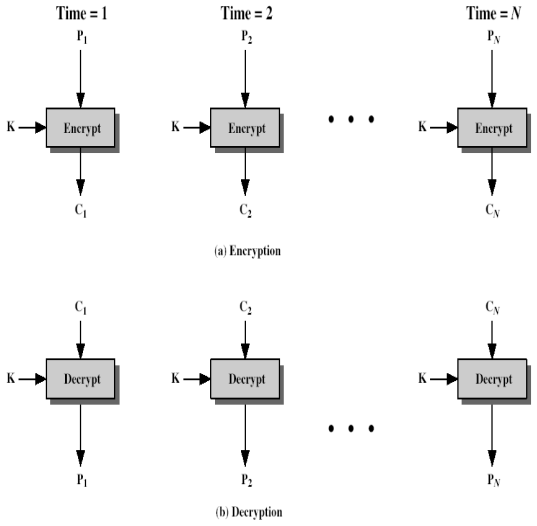
Advanced Encryption Standard (AES)

- The most widely used symmetric cipher
- AES is a subset of the Rijndael block cipher developed by two Belgian researchers.
- AES became effective as a federal government standard on May 26, 2002.
- Extremely efficient in 128-bit form, AES also uses keys of 192 and 256 bits for heavy duty encryption purposes.

Section 6 Block Cipher Modes of Operation

Electronic Codebook Mode (ECB)

- The message is divided into blocks.
- Plaintext is handled one block at a time and each block is encrypted using the same key.

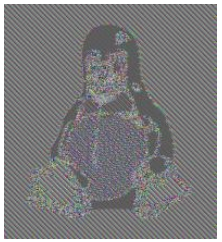


Advantages and Limitations of ECB

- Each block is encrypted independently of the other blocks
- Ideal for a short amount of data, e.g. transmit a DES key securely.
- For lengthy mesg, the ECB mode may not be secure.
  - The same n-bit block of plaintext, if it appears more than once in the mesg., always produces the same ciphertext - does not hide data patterns well.

Example: Disadvantage of ECB

- A pixel-map version of the image on the left was encrypted with ECB mode and with other modes.

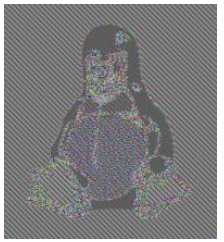


Encrypted using ECB mode

Example: Disadvantage of ECB

- A pixel-map version of the image on the left was encrypted with ECB mode and with other modes.

Original



Encrypted using ECB mode

Encrypted using other modes

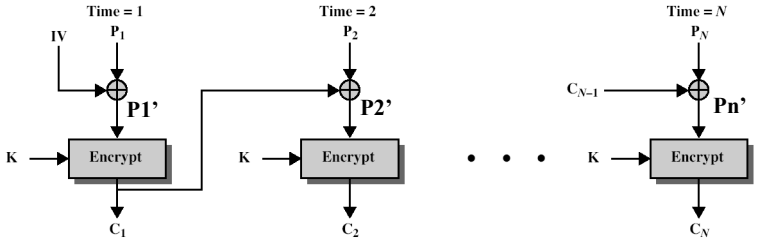
Advantages and Limitations of ECB

- Weakness is due to the encrypted message blocks being independent
  - Would like a technique in which the same plaintext block, if repeated, produces different ciphertext block.

Cipher Block Chaining (CBC)

- The input to the encryption algorithm is the XOR (⊕) of the current plaintext block and the preceding ciphertext block. ($A \oplus A = 0, 0 \oplus A = A$)
- Each ciphertext block is dependent on all plaintext blocks processed up to that point.

$C_j = E(K,[C_{j-1} \oplus P_j]), C_0 = IV$

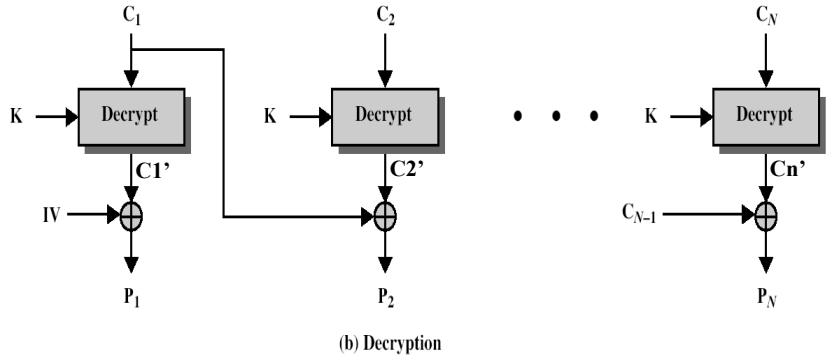


(a) Encryption

Cipher Block Chaining (CBC)

- For decryption, each cipher block is passed through the decryption alg.. The result is XORed with the preceding ciphertext block to produce the plaintext.

  $P_j = C_{j-1} \oplus D(K,C_j)$, $C_0 = IV$



(b) Decryption

Cipher Block Chaining (CBC)

- How to prove that the decryption process is correct?

  Encryption: $C_j = E(K,[C_{j-1} \oplus P_j])$, $C_0 = IV$

  Decryption: $P_j = C_{j-1} \oplus D(K,C_j)$, $C_0 = IV$

  To prove the correctness, we assume $C_j = E(K,[C_{j-1} \oplus P_j])$, $C_0 = IV$, and prove that $C_{j-1} \oplus D(K,C_j)$ is equal to P_j

Cipher Block Chaining (CBC)

- How to prove that the decryption process is correct?

  Encryption: $C_j = E(K,[C_{j-1} \oplus P_j])$, $C_0 = IV$

  Decryption: $P_j = C_{j-1} \oplus D(K,C_j)$, $C_0 = IV$

  Proof:

  $A \oplus A = 0$

  $0 \oplus A = A$

  $$\begin{aligned} & C_{j-1} \oplus D(K,C_j) \\ &= C_{j-1} \oplus D(K, E(K,[C_{j-1} \oplus P_j])) \\ &= C_{j-1} \oplus C_{j-1} \oplus P_j \\ &= 0 \oplus P_j \\ &= P_j \end{aligned}$$

Message Padding

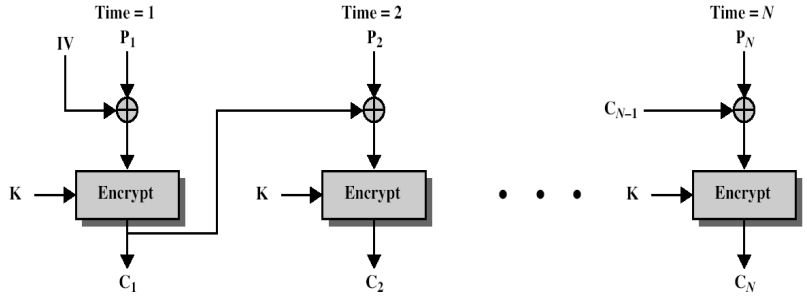
- At end of message must handle a possible last block, which is not as large as block size of cipher

Message Padding

- At end of message must handle a possible last block, which is not as large as block size of cipher
  - Pad either with known non-data value (eg nulls)
  - Or pad last block along with count of pad size
    - eg. [b1 b2 b3 0 0 0 0 5] - 3 data bytes, 5 bytes pad+count

Advantages and Limitations of CBC

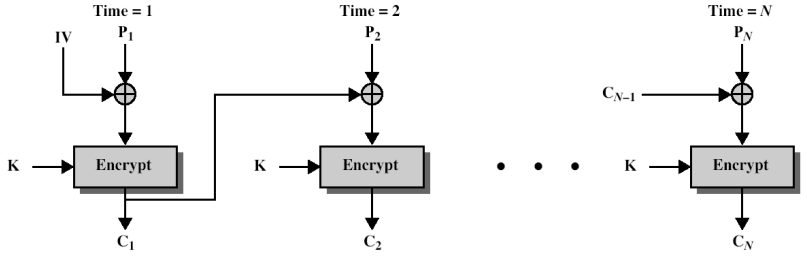
- Advantage:
  - A ciphertext block depends on all blocks before it - any change to one block affects all the following ciphertext blocks



(a) Encryption

Advantages and Limitations of CBC

- Disadvantage:
  - Encryption is sequential (i.e., cannot be parallelized)
  - Need Initialization Vector (IV)
    - Which must be known to sender & receiver
    - IV must either be a fixed value or be sent encrypted in ECB mode before rest of message



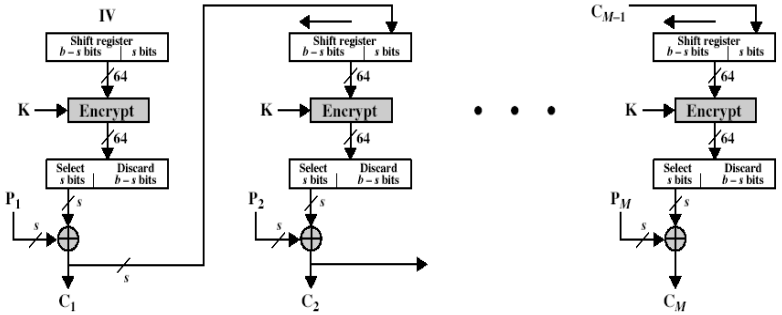
(a) Encryption

Cipher FeedBack Mode (CFB)

- When the data unit is smaller than the block size (e.g. data is only available a bit/byte at a time).
- Convert DES into a stream cipher and can be used to encrypt any number of bits
  - Property of stream cipher: the ciphertext is of the same length as the plaintext.
  - Eliminates the need to pad a mesg.

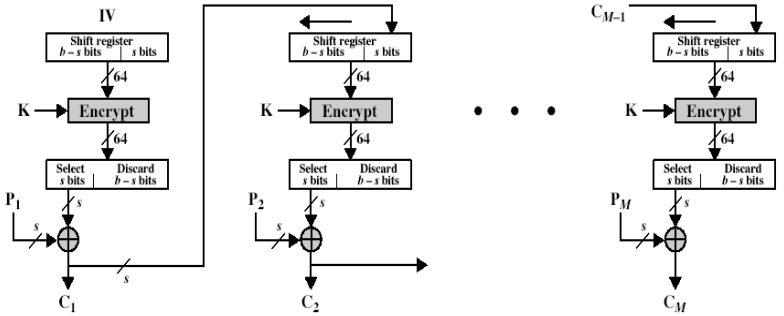
Cipher FeedBack Mode (CFB): Encryption

- E.g. The unit of transmission is s bits.
  - 64-bit shift register is initially set to some initialization vector (IV).
  - The leftmost s bits of the output of the encryption function is XORed with the first unit of plaintext P1 to produce C1.



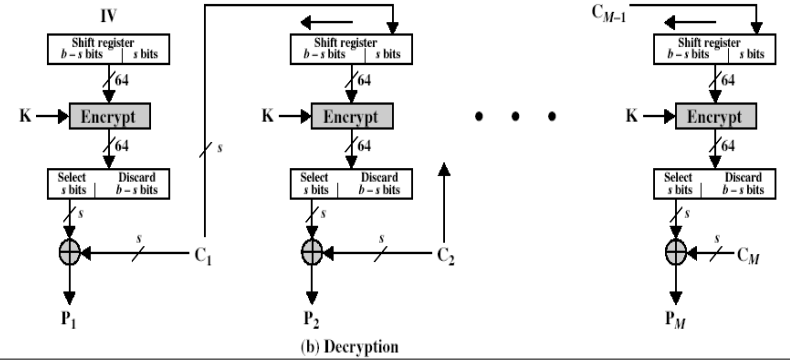
Cipher FeedBack Mode (CFB): Encryption

- E.g. the unit of transmission is s bits.
  - The contents of the shift register are shifted left by s bits and C1 is placed in the rightmost s bits of the shift register.
  - Continue this process until all plaintext units have been encrypted.



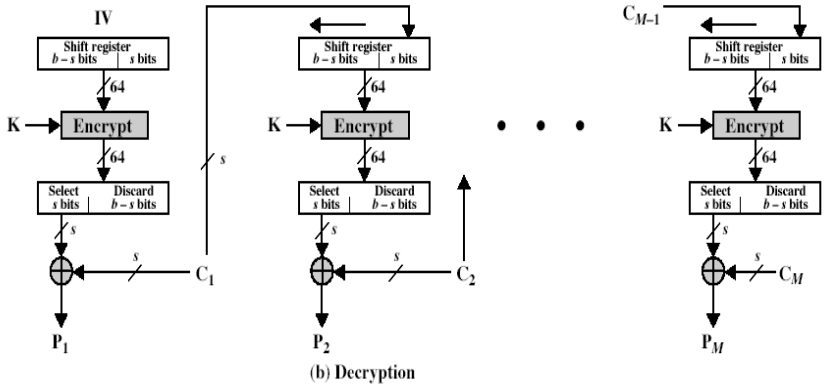
Cipher FeedBack Mode (CFB): Decryption

- E.g. the unit of transmission is s bits.
  - Same scheme as encryption, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit.



Cipher FeedBack Mode (CFB): Decryption

- Question: how to prove that the decryption is correct?



Let Ei be the leftmost s bits of the encrypted shift register at round i

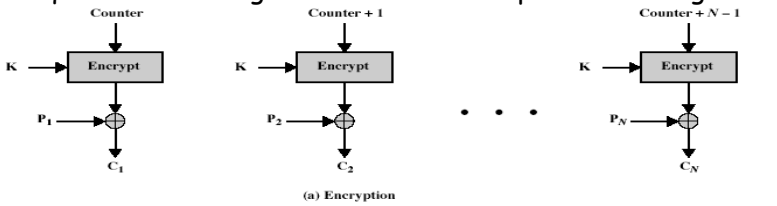
Encryption: $C_i = P_i \oplus E_i$

Dncryption: $P_i = C_i \oplus E_i$

Proof: $C_i \oplus E_i = P_i \oplus E_i \oplus E_i = P_i \oplus 0 = P_i$

Counter (CTR)

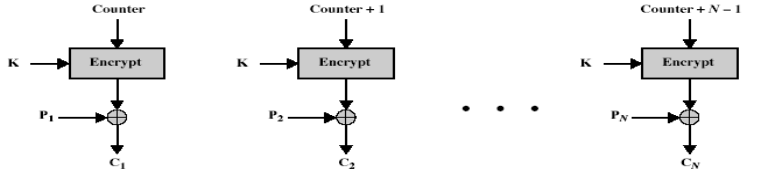
- The counter is initialized to some value and then incremented by 1 for each subsequent block.
- The counter can be any simple function which produces a sequence which is guaranteed not to repeat for a long time



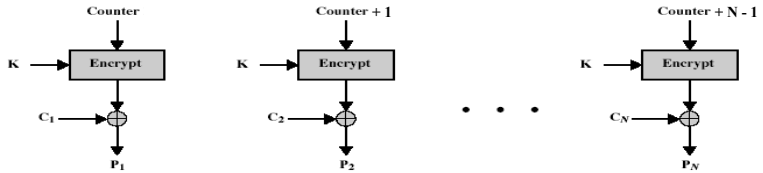
(a) Encryption

Counter (CTR)

- The counter is initialized to some value and then incremented by 1 for each subsequent block.



(a) Encryption



(b) Decryption

Encryption: $C_i = E(\text{Counter} + i - 1) \oplus P_i$

Decryption: $P_i = E(\text{Counter} + i - 1) \oplus C_i$

Advantages of CTR

- Efficiency
  - Encryption/decryption can be done in parallel on multiple blocks of plaintext or ciphertext
  - The execution of the encryption algorithm does not depend on the plaintext and ciphertext - can preprocess in advance.
  - Uses: high-speed network encryptions
- Random access to encrypted data blocks
- Provable security (good as other modes)

Key Distribution

Key Distribution

- For symmetric encryption to work, the two parties must share a secrete key.
- Frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key.
- Key distribution: refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key
- Often secure systems failure due to a break in the key distribution scheme

Key Distribution

- For two parties A and B, key distribution can be achieved in a number of ways:

Key Distribution

- For two parties A and B, key distribution can be achieved in a number of ways:

1. A can select key and physically deliver to B
2. A third party can select & physically deliver key to A & B
3. If A and B have communicated previously, A can transmit the new key to B, encrypted using the old key.
   - If an attacker succeeds in getting one key, then all subsequent keys will be revealed

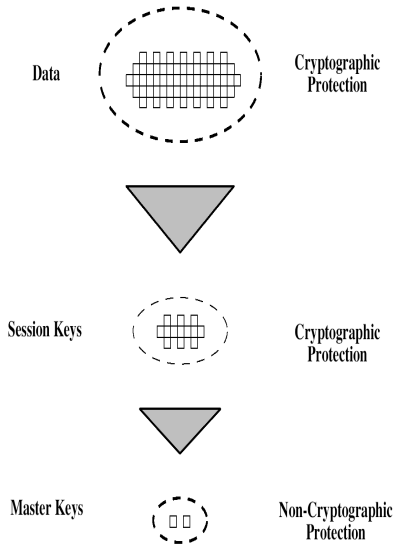
Key Distribution

- For two parties A and B, key distribution can be achieved in a number of ways:

4. If A & B have secure communications with a third party C, C can deliver a key on the encrypted links to A and B
   - A key distribution center is responsible for distributing keys to pairs of users.
   - Each user must share a unique key with the key distribution center for purpose of key distribution.

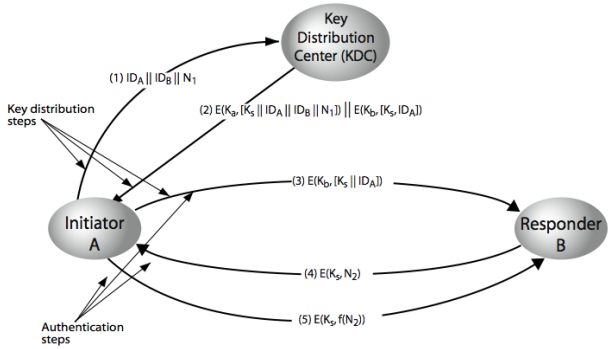
Key Hierarchy

- The use of a key distribution center is based on the use of a hierarchy of keys.
- Master key
  - Used to encrypt session keys
  - Shared by user & key distribution center
- Session key
  - Temporary key
  - Used for encryption of data between users



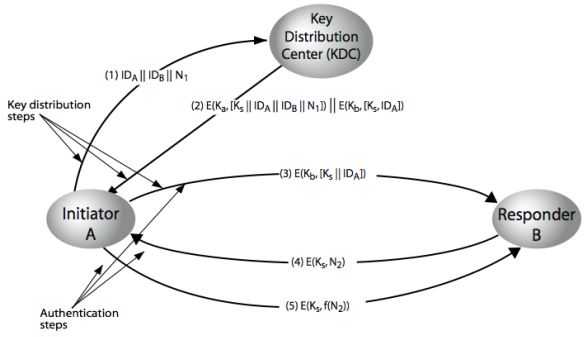
Key Distribution Scenario

- A wishes to establish a logical connection with B and requires a one-time session key to protect the data transmitted over the connection
- A shares the master key K_a with the KDC
- B shares the master key K_b with the KDC



Key Distribution Scenario

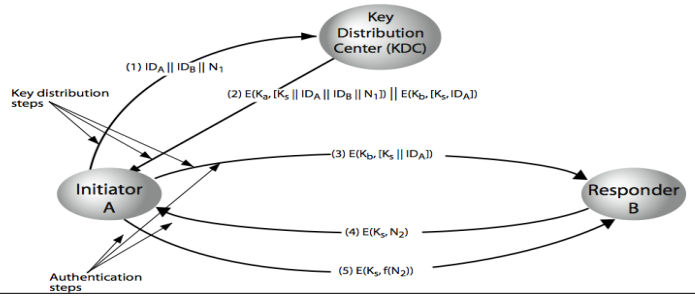
- Msg1: A issues a request to the KDC for a session key to protect a connection to B. The message includes the identity of A and B, and a unique identifier, N1 (nonce).
- Nonce: a random number that is used to demonstrate the freshness of a session – prevent replay attack



Key Distribution Scenario

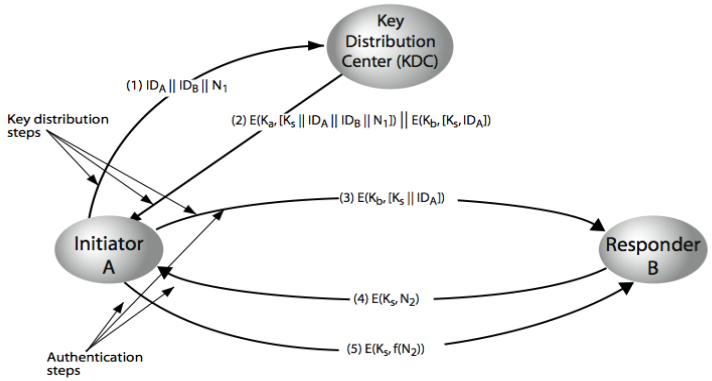
- Msg2: The KDC responds with a message encrypted using K_a

1. The one-time session key k_s
2. The original request message and the nonce
3. Two items for B, encrypted using K_b: the one-time session key k_s and an identity of A, ID_A.



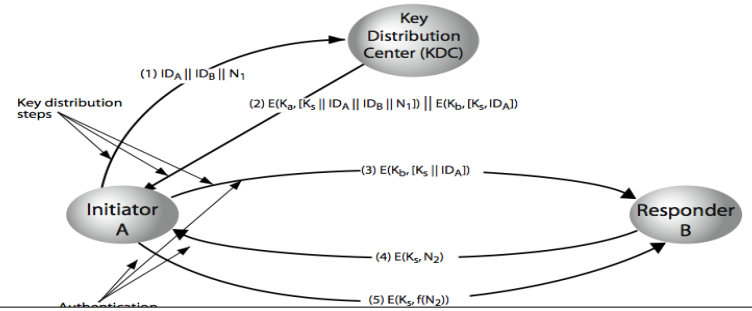
Key Distribution Scenario

- Msg3: A stores the session key for use in the upcoming session and forward to B the information that originated at the KDC for B, E(Kb,[Ks,IDA]).



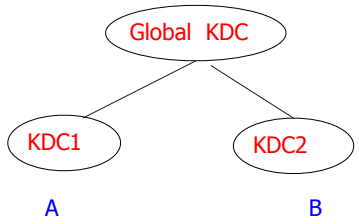
Key Distribution Scenario

- Now, a session key has been securely delivered to A and B.
- Msg4: B sends a nonce N_2 to A using the newly minted session key.
- Msg5: Also using K_s, A responds with $f(N_2)$, where f is a function that perform some transformation on N_2



Hierarchical Key Control

- It is not necessary to limit the key distribution function to a single KDC – for large networks, a hierarchy of KDCs can be established
  - E.g. local KDCs, each responsible for a small domain
  - If two entities are in different domains, then local KDCs can communicate through a global KDC.



CS458/CS558: Introduction to Computer Security

Email Security

Is it possible to send emails using someone else's email address without knowing their account password?

Is it possible to send emails using a non-existing email address?

Email

- Email is one of the most widely used network-based application.
- Every user is uniquely identified by an email address: id@domain
  - id: identifies the user of a domain
  - Domain: identifies the organization or a host machine
- Using a mailbox principle
  - A sender does not require the receiver to be online.

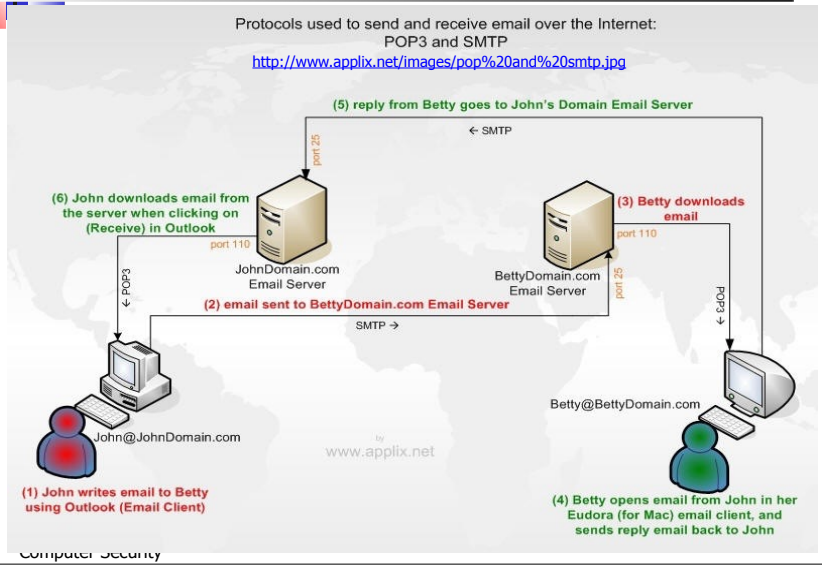
Simple Mail Transfer Protocol (SMTP)

- SMTP: deliver email from the sender's email client to the recipient's email server.
  - Mails that cannot be delivered keep waiting in the spooling area
    - ➢ Client process will repeat its delivery attempts periodically
    - ➢ After several repetitions that mail will be removed from the spooling area.

POP3

- POP3 (Post Office Protocol version 3): handle email between Email Server and the recipient's local Email Client.
  - The email will stay on the recipient's email server until it is explicitly requested to be downloaded by the recipient's Email client (e.g. Outlook or Eudora) over, e.g. POP3 protocol.

Example: SMTP and POP3



SMTP Provides No Security

- Emails can be altered en route
- There is no way to validate the identity of the email source.
  - Email headers (except the first received header) can be easily forged.
  - Received header: the IP address of the last computer through which the message has passed before being delivered

SMTP Commands: Client → Server

HELO: Initiates a conversation with the mail server.

Mail FROM: Indicates who is sending the mail. E.g.

MAIL FROM: <user1@google.com>

RCPT TO: Indicates who is receiving the mail. E.g.

RCPT TO: user2@yahoo.com

You can indicate more than one user by issuing multiple RCPT commands.

DATA: Indicates that you are about to send the text (or body) of the message. The message ends with '.'

QUIT: Indicates that the conversation is over.

SMTP Replies: Server → Client

- 220: service ready
- 250: requested mail action OK, completed
- 421: service is not available
- 450: requested action aborted
- 500: syntax error
- ……

Example

Connect to port 25
HELO mail1.com
250 … Pleased to meet you
MAIL FROM: user1@mail1.com
250 OK
RCPT TO: cs5712013@gmail.com
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
test this function
.
250 OK
QUIT
Connection closed by foreign host.

Demonstration

We can use any email address to sent email!

Spam and Phishing Emails

Spam Email

- Spam emails: unsolicited bulk email
  - are sent to a large group of individuals in an effort to force the email onto people who would otherwise choose not to receive this message.

Detecting Spam Email

- Based on the IP address, email address, or domain name from which the spam email is sent. However,
  - The from and reply-to headers can be forged
  - The spammer can hide the IP address using bot-networks or open proxy
- Based on contents or patterns.
- User engagement.

Email Phishing

- Designed to steal your valuable personal data, such as credit card numbers, passwords, account data, or other information.
- Sometimes include official-looking logos and other information taken directly from legitimate Web sites.

https://www.istockphoto.com/vector/phishing-scam-hacker-attack-gm956400244-261133477

How to tell if an email is a phishing email?
What are the telling attributes?

Asking for Personal Information

- Asking for confidential information through email
  - Businesses will not ask you to send passwords, SSNs, or other personal information through e-mail.

Your account has been compromised. Please email me the following information as soon as possible so that we can reset your password.
- Current password
- New password
- SSN
- Birth date
- Phone number

Convey a Sense of Urgency

Convey a Sense of Urgency

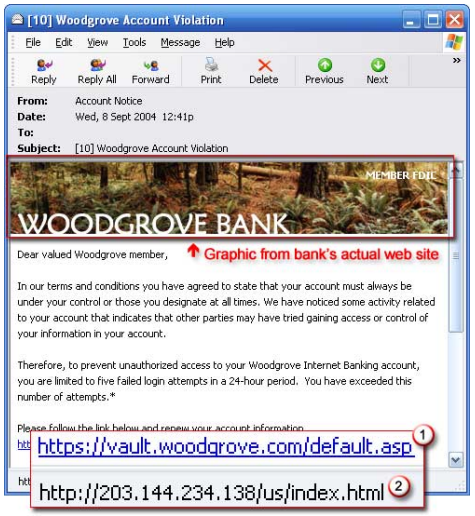
- Convey a sense of urgency so that you will respond immediately without thinking.
- Examples:

> If you don't respond within 48 hours, your account will be closed.

> Your response is required because your account might have been compromised.

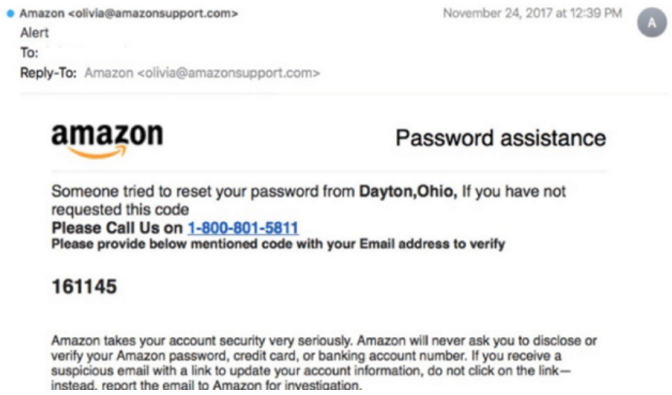
Asking to Click a Link

- Dear Valued Customer
  - Phishing e-mail messages are usually sent out in bulk and often do not contain your first or last name.
- Asking to click the link
  - The link you see does not take you to that address but somewhere different, usually a phony Web site.



Incorrect Phone Number

- Not a legitimate organization's phone number.


Amazon <olivia@amazonsupport.com> November 24, 2017 at 12:39 PM
Alert
To:
Reply-To: Amazon <olivia@amazonsupport.com>

amazon Password assistance

Someone tried to reset your password from **Dayton,Ohio,** If you have not requested this code
Please Call Us on 1-800-801-5811
Please provide below mentioned code with your Email address to verify

161145

Amazon takes your account security very seriously. Amazon will never ask you to disclose or verify your Amazon password, credit card, or banking account number. If you receive a suspicious email with a link to update your account information, do not click on the link—instead, report the email to Amazon for investigation.

Incorrect Email Address

Subject: [External Email] Quick request
Date: Thu, 26 Aug 2021 15:09:19 +0100
From: A B<ab83@gmail.com>

Kindly send me your available cell number.
A B
Professor
Computer Science
Binghamton University

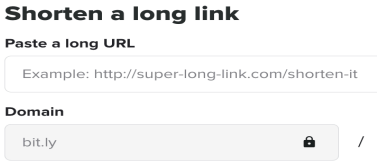
Typos and Grammatical Errors

Typos and Grammatical Errors

- Scammers introduce spelling errors to penetrate through spam filters.
  - Spam filters look out for keywords and phrases commonly found in phishing emails.
- Misspellings and grammatical errors make the email look more authentic and believable.

Shortened URL

- Online services are provided to shorten long URLs to make it easy to type in a web address.
  - E.g., http://bit.ly/fkgylsd
  - The URL is shortened using a redirection which links to the web page that has a long URL.
- Shortened URL makes it difficult to tell where your web browser will take you.
- Therefore, it is often used by scammers to evade detection and direct people to phishing sites.



The Email is Too Good To Be True

- Offering money

We were married for 30 years without a child and he died after a brief illness and since his death I decided not to remarry due to my religious belief and old age. When my late husband was alive he deposited the sum of three million five hundred thousand Euros (3.5 Million Euros) with a Bank here. Presently this money is still in the custody of the Bank. Recently, my Doctor told me that I would not last for the next four months due to my cancer illness.

Having known my condition I decided to donate this money to churches, organizations or good people that will utilize this fund in a way I am going to instruct for a proper idea on how this fund would be used.

I want you to use this money for charity organizations, orphanages, widows and other people that are in need. I took this decision because I don't have any child that will inherit this money. Moreover, my husband's relatives are not close to me since I developed cancer illness and it had been their wish to see me dead so as to enable them to inherit his wealth hence we have no Child. This people are not worthy of this inheritance. This is why I am taking this decision to contact you and donate this fund to you for the charity works.And 30% of this money will be for your time and effort, while 70% goes to charities

Phishing Emails Received by Binghamton Faculty/Students

- https://www.binghamton.edu/its/about/organization/information-security/phish-tank/index.html

Activities: Phishing Emails (1)

From: **Donald G Nieman** <adamsmith16465@gmail.com>
Date: Thu, Dec 16, 2021 at 11:40 AM
Subject: [External Email] Urgent and important
To: < @binghamton.edu>

--
Are you free right now?, I need your personal cellphone number.
Donald G Nieman
Executive Vice President for Academic Affairs and Provost

Activities: Phishing Emails (1)

From: **Donald G Nieman** <adamsmith16465@gmail.com>
Date: Thu, Dec 16, 2021 at 11:40 AM
Subject: [External Email] Urgent and important
To: < @binghamton.edu>

--
Are you free right now?, I need your personal cellphone number.
Donald G Nieman
Executive Vice President for Academic Affairs and Provost

Activity: Phishing Emails (2)

brij@systemart.com 1:01 PM (11 minutes ago)
to me

Hello,

My name is **Brij**. I am a Technical recruiter with an IT staffing firm name **Systemart, LLC**. We are a leading Global Software Development and business process solutions firm, seeking experienced IT professionals for our clients.

Job Title : Senior Art Director (1032821)
Location : Newark, NJ
Client : Audible
Duration : 6 months
Rate : $ 73/ Hourly

Please reply back with your updated resume if you are interested and would like to apply for this position.
Also help with below details.

Can start asap / 1 week / 2 weeks	
Correct contact no	
Last four OR five digits of SSN	
DOB (Date of birth), only month and date	

JOB DESCRIPTION:
Primary Responsibilities:
- Create strategic, conceptual solutions for experiences that are part of a customer journey and rooted in the brand.
- Leverage your strength in UX/UI and design systems to solve complex business problems with designs that scale across web, app, and email surfaces and globally across marketplaces.

Activity: Phishing Emails (2)

brij@systemart.com 1:01 PM (11 minutes ago)
to me

Hello,

My name is **Brij**. I am a Technical recruiter with an IT staffing firm name **Systemart, LLC**. We are a leading Global Software Development and business process solutions firm, seeking experienced IT professionals for our clients.

Job Title : Senior Art Director (1032821)
Location : Newark, NJ
Client : Audible
Duration : 6 months
Rate : $ 73/ Hourly

Please reply back with your updated resume if you are interested and would like to apply for this position. Also help with below details.

Can start asap / 1 week / 2 weeks	
Correct contact no	
Last four OR five digits of SSN	
DOB (Date of birth), only month and date	

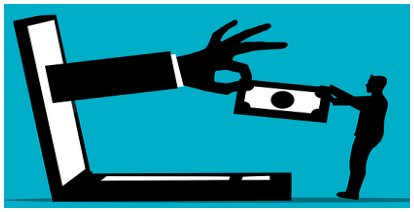
JOB DESCRIPTION:

Primary Responsibilities:

- Create strategic, conceptual solutions for experiences that are part of a customer journey and rooted in the brand.
- Leverage your strength in UX/UI and design systems to solve complex business problems with designs that scale across web, app, and email surfaces and globally across marketplaces.

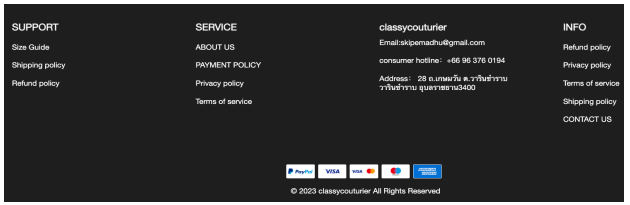
Scam Websites

- Aim to steal your valuable personal data (credit card numbers, passwords, etc.) and money.
- Many scam websites connect to each other.



Scam Website May look Legitimate

- Many scam websites look legitimate.
  - SSL connection (i.e., https).
  - Customer service support.
  - Supporting credit card and Paypal payment.



Scam Website May look Legitimate

- Many scam websites look legitimate.
  - Order confirmation email.
  - Tracking number.
  - Mailing “something” to customers.

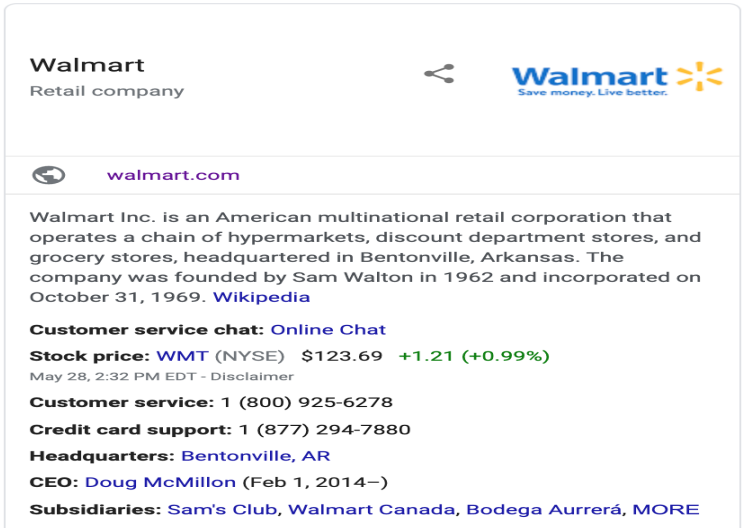
Hi,
Thank you so much for your order.
We have sent it out and the tracking no. is
Pls track your package here.
If the status does not update you can track in 2-3 days or email us first.
Normally you will receive your order within 10-15 days.
If you have any questions,pls feel free to contact us.

What are some ways to identify scam websites?

Very Low Price on Expensive Products

- Scam websites often provide low price for expensive products.

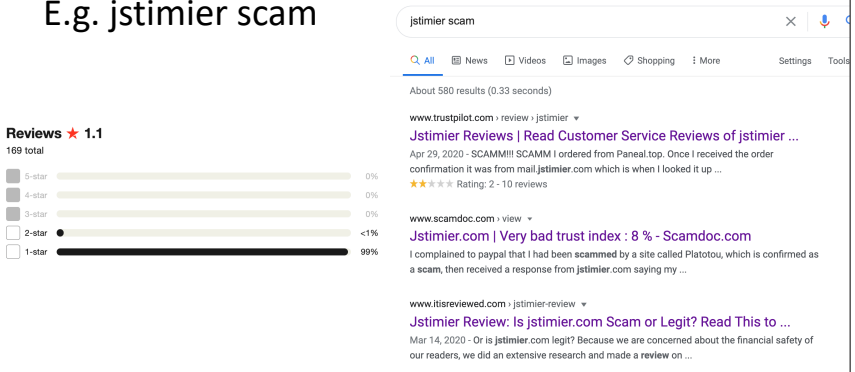
Google the Website Name



Google the Website Name (Cont.)

- Google the website name followed by word "scam"

E.g. jstimier scam



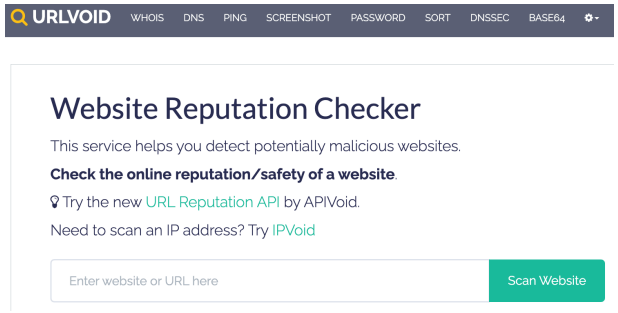
https://www.trustpilot.com/review/jstimier.com

Verify The Phone Number and Address

- Call the phone number in the contact page.
- Google the address in the contact page and check if the address is a residential address.
- Did not provide a phone number or company address? Likely scam.

Use Website Reputation Tracker

- Use website reputation tracker https://www.urlvoid.com
  - Created within one year? Probably scam
  - More than 10 years ago? Google the website name



Check Customer Review Comments

- Do the products on the website have customer review comments?

  No - probably scam

Still Want to Purchase?

- Use Paypal + credit card instead of Paypal balance or credit card to purchase.

- Keep track of the status of the shipment
  - Different location
  - Wrong item
  - Did not receive the product

Reporting Scam/Phishing Websites

- Report to Google safebrowsing, which helps remove website from Google Search results.

Google Reporting Page

Dispute

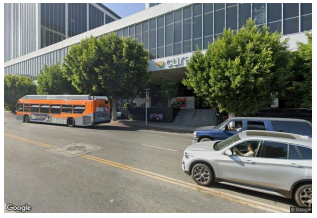
- Take a screenshot of the email containing the tracking number.
- Take a screenshot of the tracking information.
- Take a picture of the envelope you receive.
- Take a picture of the item you receive with the envelop.
- Other evidences (e.g., purchased a very large item, but the tracking details indicate that it was delivered in mailbox).

Which of the following are not trustworthy?

1. https://icuracao.com/
2. https://boomlon.com
3. https://pilosaleltd.com/

1. https://icuracao.com/

- The website was created on 03/08/2007.
- Many products have customer reviews.
- Google search shows that the address is curacao business center



https://www.propertyshark.com/mason/Property/16326459/1605-W-Olympic-Blvd-Los-Angeles-CA-90015/

2. boomlon.com

- The website was created on 06/30/2023.
- Contact information

Phone & Email

Our Presale Phone Line is available in English for all our customers.
Call Time From MON-SUN 10:00-19:00 EST
Due to peak season we are experiencing a high volume of calls at the moment and our phone agents will only offer assistance for presales questions. Please kindly wait patiently for our agents to take your call. During peak times, it may take up to 10-15 minutes to reach us.

If you have any questions, please Email us at customer@boomlon.com for assistance.

3. https://pilosaleltd.com/

- The website was created in 2021.
- The address is a residential address.
- A google search on “pilosaleltd scam” returns many results showing that it is likely a scam website.

E.g., https://www.scamwatcher.com/scam/view/606204
https://www.trustpilot.com/review/www.piloltd.com

CS458/558: Introduction to Computer Security

Chapter 9
Public-Key Cryptography

Public-Key Cryptography

- Symmetric key cryptography uses one key, shared by both sender and receiver
- If this key is disclosed, communications are compromised
- Can we use symmetric key encryption to protect sender from receiver forging a message and claiming is sent by sender?

Public-Key Cryptography

- Symmetric key cryptography uses one key, shared by both sender and receiver
- If this key is disclosed, communications are compromised
- Can we use symmetric key encryption to protect sender from receiver forging a message and claiming is sent by sender?
  - John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.
  - Mary may forge a different message and claim that it came from John

Public-Key Cryptography

- Public invention due to Whitfield Diffie & Martin Hellman at Stanford University in 1976.
- Public-key/two-key/asymmetric cryptography involves the use of two keys:
  - A public-key, which may be known by anybody, and can be used to encrypt messages, and verify signatures
  - A private-key, known only to the recipient, used to decrypt messages, and sign (create) signatures
- Is asymmetric because
  - Those who encrypt messages or verify signatures may not decrypt messages or create signatures

Public-key cryptography: Misconceptions

- Misconception 1: Public-key encryption is more secure from cryptanalysis than symmetric encryption
  - The security depends on the length of the key and the computational work involved in breaking a cipher.

Public-key cryptography: Misconceptions

- Misconception 1: Public-key encryption is more secure from cryptanalysis than symmetric encryption
  - The security depends on the length of the key and the computational work involved in breaking a cipher.

- Misconception 2: Public-key encryption is a general-purpose technique that has made symmetric encryption obsolete.
  - Computation overhead of public-key encryption

Why Public-Key Cryptography?

- Developed to address two key issues:
  - Key distribution – how to have secure communications in general without having to trust a KDC
  - Digital signatures – how to verify a message comes intact from the claimed sender

Requirements for Public-Key Cryptography

- It is computationally easy for a party B to generate a pair: public key PU_b, private key PR_b
- The two keys can be applied in either order.

$$M = D(PU_b, E(PR_b, M)) = D(PR_b, E(PU_b, M))$$

- It is computationally easy for sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext.

$$C = E(PU_b, M)$$

- It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message.

Requirements for Public-Key Cryptography (Cont.)

- It is computationally infeasible for an adversary, knowing the public key PU_b, to determine the private key PR_b.
- It is computationally infeasible for an adversary, knowing the public key PU_b and the ciphertext C encrypted using PU_b, to recover the original message M.
- These are formidable requirements – only a few algorithms (e.g. RSA) have received widespread acceptance.

Public-Key Cryptosystems: Secrecy

- A produces plaintext X = [X1,X2,…,Xn]
- The message is intended for destination B.
- A has two keys: a public key PU_a, and a private key PR_a.
- B has two keys: a public key PU_b, and a private key PR_b.

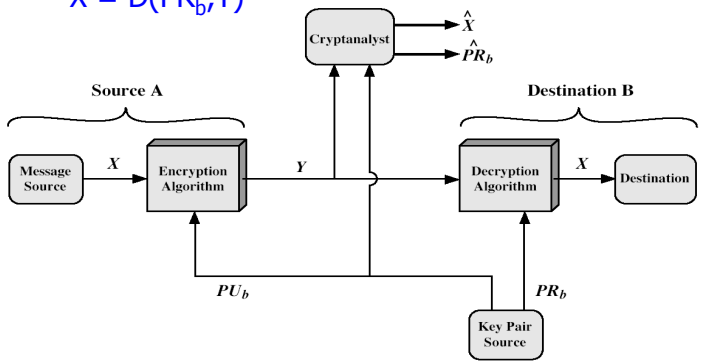
Public-Key Cryptosystems: Secrecy

- A forms the ciphertext Y = [Y1,Y2,…,Yn]:

$$Y = E(PU_b, X)$$

- The receiver is able to invert the transformation

$$X = D(PR_b, Y)$$



Public-Key Cryptosystems: Digital Signature

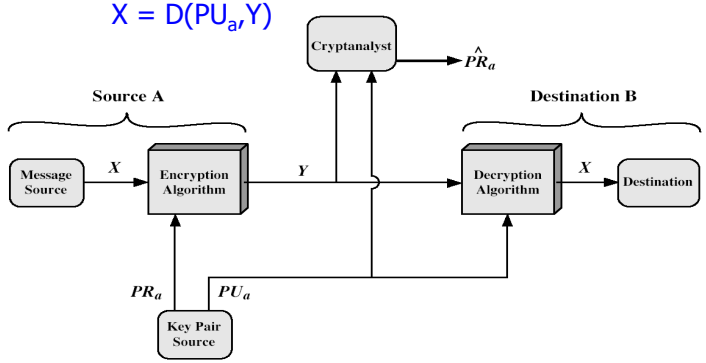
Public-Key Cryptosystems: Digital Signature

- A prepares a message to B and encrypts it using A's private key before transmitting it.

  $Y = E(PR_a, X)$

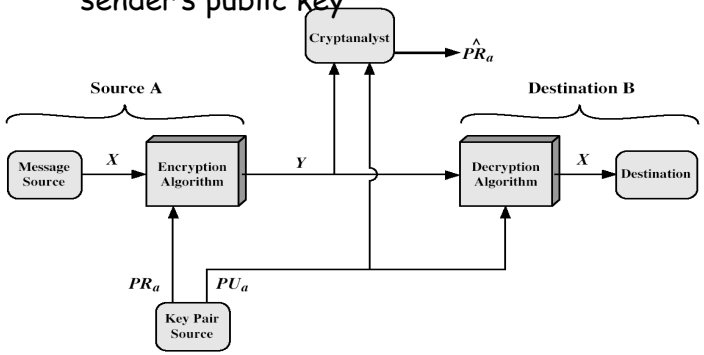
- B decrypts the message using A's public key

  $X = D(PU_a, Y)$



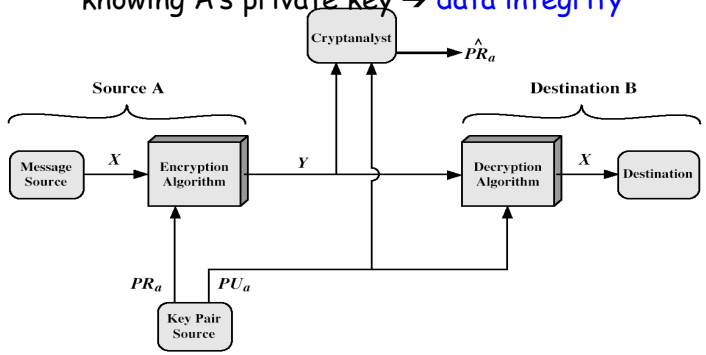
Public-Key Cryptosystems: Digital Signature

- Does not provide confidentiality.
  - The message being sent is safe from alteration but not from eavesdropping.
  - Any observer can decrypt the message using the sender's public key



Public-Key Cryptosystems: Digital Signature

- Because the message was encrypted using A's private key, only A could have prepared the message
  - Serves as digital signature.
  - It is impossible to alter the message without knowing A's private key → data integrity



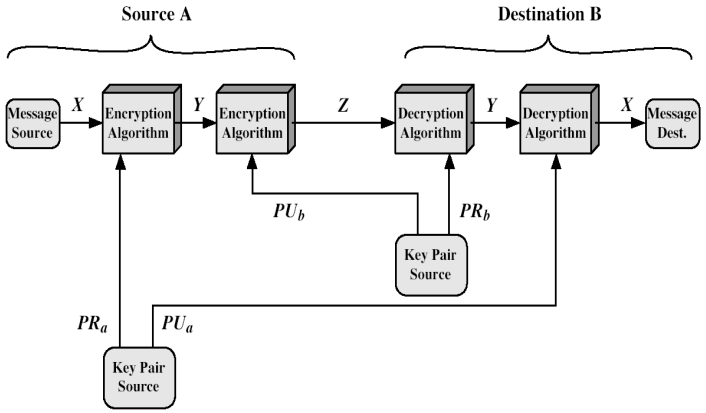
Public-Key Cryptosystems: Digital Signature and Secrecy

Public-Key Cryptosystems: Digital Signature and Secrecy

- Double use of the public-key scheme:

$Z = E(PU_b, E(PR_a, X))$

$X = D(PU_a, D(PR_b, Z))$



Conventional vs. Public-Key Encryption

Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Chapter 8

Introduction to Number Theory

Prime Numbers

- Prime numbers play a critical role both in number theory and in cryptography

Prime Numbers

- Prime numbers play a critical role both in number theory and in cryptography
- An integer $p > 1$ is a prime number if and only if its only divisors are ± 1 and $\pm p$
  - Eg. 2,3,5,7 are prime, 4,6,8,9,10 are not
- Any integer $a > 1$ can be factored in a unique way as $a = p_1^{a1} p_2^{a2} \ldots p_t^{at}$

  $p_1 < p_2 < \ldots < p_t$ are prime numbers and a_i are positive integers.

  eg. 91= ; 3600=

Prime Numbers

- Prime numbers play a critical role both in number theory and in cryptography
- An integer $p > 1$ is a prime number if and only if its only divisors are ± 1 and $\pm p$
  - Eg. 2,3,5,7 are prime, 4,6,8,9,10 are not
- Any integer $a > 1$ can be factored in a unique way as $a = p_1^{a1} p_2^{a2} \ldots p_t^{at}$

  $p_1 < p_2 < \ldots < p_t$ are prime numbers and a_i are positive integers.

  eg. 91=7 * 13 ; 3600=

Prime Numbers

- Prime numbers play a critical role both in number theory and in cryptography
- An integer $p > 1$ is a prime number if and only if its only divisors are ± 1 and $\pm p$
  - Eg. 2,3,5,7 are prime, 4,6,8,9,10 are not
- Any integer $a > 1$ can be factored in a unique way as $a = p_1^{a1} p_2^{a2} \ldots p_t^{at}$

  $p_1 < p_2 < \ldots < p_t$ are prime numbers and a_i are positive integers.

  eg. 91=7 * 13 ; 3600=$2^4 * 3^2 * 5^2$

Greatest Common Divisor (gcd)

- The greatest common divisor of integers a and b, expressed gcd(a,b):

Greatest Common Divisor (gcd)

- The greatest common divisor of integers a and b, expressed gcd(a,b):
  - The largest positive integer that divides both numbers without remainder

Greatest Common Divisor (gcd)

- The greatest common divisor of integers a and b, expressed gcd(a,b):
  - The largest positive integer that divides both numbers without remainder
- Can determine the greatest common divisor by comparing their prime factorizations and using least powers

  eg. $300=2^1x3^1x5^2$, $18=2^1x3^2$ hence

  gcd(18,300)=

Greatest Common Divisor (gcd)

- The greatest common divisor of integers a and b, expressed gcd(a,b):
  - The largest positive integer that divides both numbers without remainder
- Can determine the greatest common divisor by comparing their prime factorizations and using least powers

  eg. $300=2^1x3^1x5^2$, $18=2^1x3^2$ hence

  gcd(18,300)= $2^1x3^1x5^0=6$

Greatest Common Divisor (gcd)

- gcd(x,y) = x if y ==0
  = gcd(y, (x mod y)) if x>=y and y>0

e.g.
gcd(300, 18) =

Greatest Common Divisor (gcd)

- gcd(x,y) = x if y ==0
  = gcd(y, (x mod y)) if x>=y and y>0

e.g.
gcd(300, 18) = gcd(18, (300 mod 18))
= gcd(18, 12)

Greatest Common Divisor (gcd)

- gcd(x,y) = x if y ==0
  = gcd(y, (x mod y)) if x>=y and y>0

e.g.
gcd(300, 18) = gcd(18, (300 mod 18))
= gcd(18, 12)
= gcd(12, (18 mod 12))
= gcd(12, 6)

Greatest Common Divisor (gcd)

- gcd(x,y) = x if y ==0
  = gcd(y, (x mod y)) if x>=y and y>0

e.g.
gcd(300, 18) = gcd(18, (300 mod 18))
= gcd(18, 12)
= gcd(12, (18 mod 12))
= gcd(12, 6)
= gcd(6, 0) = 6

Fermat's Theorem

- Fermat's Theorem: If p is a prime number and a < p is a positive integer not divisible by p, then
- $a^{p-1} \bmod p = 1$.
  - E.g. p =3, a = 2 → $a^{p-1} \bmod p = 4 \bmod 3 = 1$.
- Also $a^p \bmod p = a$
- Useful in public key and primality testing

Fermat's Theorem

- Fermat's Theorem: If p is a prime number and a < p is a positive integer not divisible by p, then
- $a^{p-1} \bmod p = 1$.
- Use Fermat Theorem and the modular property

$[(a1 \bmod n) * .....* (am \bmod n)] \bmod n = (a1*...*am) \bmod n$

to compute $5^{303} \bmod 11$

Fermat's Theorem

- Fermat's Theorem: If p is a prime number and a < p is a positive integer not divisible by p, then

$a^{p-1} \bmod p = 1$.

- Use Fermat Theorem and the modular property

$[(a1 \bmod n) * .....* (am \bmod n)] \bmod n = (a1*...*am) \bmod n$

to compute $5^{303} \bmod 11$

$5^{303} \bmod 11 = ((5^{10})^{30} * 5^3) \bmod 11$
$= (5^{10} * 5^{10} * \ldots * 5^{10} * 5^3) \bmod 11$
$= [(5^{10} \bmod 11) * \ldots * (5^{10} \bmod 11) * (5^3 \bmod 11)] \bmod 11$
$= 5^3 \bmod 11 = 4$

Euler Totient Function ø(n)

- Euler Totient Function ø(n): the number of positive integers less than n and relatively prime to n.
  - m is a relatively prime to n if gcd(m,n)=1
  - ø(37)

Euler Totient Function ø(n)

- Euler Totient Function ø(n): the number of positive integers less than n and relatively prime to n.
  - m is a relatively prime to n if gcd(m,n)=1
  - ø(37) = 36: all integers from 1 through 36 are relatively prime to 37.
  - For a prime number p, ø(p)

Euler Totient Function ø(n)

- Euler Totient Function ø(n): the number of positive integers less than n and relatively prime to n.
  - m is a relatively prime to n if gcd(m,n)=1
  - ø(37) = 36: all integers from 1 through 36 are relatively prime to 37.
  - For a prime number p, ø(p) = p-1
  - ø(35) =

Euler Totient Function ø(n)

- Euler Totient Function ø(n): the number of positive integers less than n and relatively prime to n.
  - m is a relatively prime to n if gcd(m,n)=1
  - ø(37) = 36: all integers from 1 through 36 are relatively prime to 37.
  - For a prime number p, ø(p) = p-1
  - ø(35) = 24:
    - 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34.

Euler Totient Function ø(n)

- Two prime numbers p and q with $p \neq q$, then

  $ø(pq) = ø(p)*ø(q) = (p-1)*(q-1)$

  - The set of integers less than pq is {1, 2, …, pq-1}

Euler Totient Function ø(n)

- Two prime numbers p and q with $p \neq q$, then

  $ø(pq) = ø(p)*ø(q) = (p-1)*(q-1)$

  - The set of integers less than pq is {1, 2, …, pq-1}
  - The integers in this set that are not relatively prime to n: {p, 2p, …., (q-1)p} and {q, 2q, …, (p-1)q}

  $ø(pq) = (pq - 1) – [(q-1) + (p-1)]$
  $= pq – p – q +1$
  $= (p-1) * (q-1)$
  $= ø(p)*ø(q)$

  - E.g. $ø(21) = (3–1)*(7–1) = 2*6 = 12$

Euler's Theorem

- Euler's Theorem: for every a and n that are relatively prime, $a^{ø(n)} \bmod n = 1$

  a=3;n=10;

  $ø(10)=4;\ 3^4 \bmod 10 = 81 \bmod 10 = 1$

  a=2;n=11;

  $ø(11)=10;\ 2^{10} \bmod 11 = 1024 \bmod 11 = 1$

Primality Testing

- For many cryptographic algorithms, it is necessary to select one or more very large prime numbers at random

Primality Testing

- For cryptographic algorithms, it is necessary to select one or more very large prime numbers at random
- Naïve algorithm: divide by all numbers in turn less than the square root of the number
  - Only works for small numbers

Miller Rabin Algorithm

- Background
  - $n-1 = 2^k q$ with $n > 3$, n odd, $k > 0$, q odd
    - Divide $(n-1)$ by 2 until the result is an odd number.
- Property
  - Let $n > 2$ be a prime number, a be an integer $1 < a < n-1$, and $n-1 = 2^k q$. Then one of the following two conditions is true: 1) $a^q \bmod n = 1$ or 2) there exists $1 \le j \le k$ such that $a^{(2^{j-1}q)} \bmod n = n - 1$.

Miller Rabin Algorithm

- Background
  - $n-1 = 2^k q$ with $n > 3$, n odd, $k > 0$, q odd
    - Divide $(n-1)$ by 2 until the result is an odd number.
- Property
  - Let $n > 2$ be a prime number, a be an integer $1 < a < n-1$, and $n-1 = 2^k q$. Then one of the following two conditions is true: 1) $a^q \bmod n = 1$ or 2) there exists $1 \le j \le k$ such that $a^{(2^{j-1}q)} \bmod n = n - 1$.

However, if the above condition is met, n may not be a prime.

E.g. $n=2047=23*89$, then $n-1 = 2*1023$.

$2^{1023} \bmod 2047 = 1$, but 2047 is not a prime

Miller Rabin Algorithm

- Algorithm: check if n is a prime

1. Find integers $k > 0$, q odd, so that $(n-1)=2^k q$
2. Select a random integer $1<a<n-1$
3. if $a^q \bmod n = 1$ then return ("maybe prime");
4. for j = 1 to k do
   if $a^{2^{j-1}q} \bmod n = n-1$ then return("maybe prime")

//n is definitely not prime

5. return ("not prime")

Probabilistic Considerations

- It was shown that given an odd number n that is not prime and a randomly chosen integer $1 < a < n-1$, the probability that the algorithm fails to detect that n is not a prime is $< \frac{1}{4}$

Probabilistic Considerations

- It was shown that given an odd number n that is not prime and a randomly chosen integer $1 < a < n-1$, the probability that the algorithm fails to detect that n is not a prime is $< \frac{1}{4}$
- Hence if repeat test with different a, then chance n is prime after t tests is:
  - Pr(n maybe a prime after t tests) = $(1/4)^t$
  - eg. for $t=10$ this probability is $< 10^{-6}$

Section 9.2 The RSA Algorithm

RSA

- By Rivest, Shamir & Adleman of MIT in 1977
- Best known & widely used public-key scheme
- The RSA scheme is a block cipher
  - A typical size is 1024 bits.

Algorithm

- Each block has a value less than some number n
- Encryption and decryption are of the following form for some plaintext block M and ciphertext block C.

$C = M^e \bmod n$

$M = C^d \bmod n$

Property of modular arithmetic

$[(a1 \bmod n) * .....* (am \bmod n)] \bmod n$

$= (a1*...*am) \bmod n$

Thus: $M = C^d \bmod n = (M^e \bmod n)^d \bmod n$

$= (M^e)^d \bmod n = M^{ed} \bmod n$

Determining e and d

- Find values of e, d, n s.t. $M^{ed} \bmod n = M$ for all $M < n$.
- Theorem:

  If $e*d=1+k.\phi(n)$ (or $e*d \bmod \phi(n) = 1$) where $\gcd(e,\phi(n)) = 1$, then $M^{ed} \bmod n = M$.

RSA Algorithm

Theorem:

If $e*d=1+k.\phi(n)$ (or $e*d \bmod \phi(n) = 1$) where $\gcd(e,\phi(n)) = 1$, then $M^{ed} \bmod n = M$.

- Find values of e, d, n such that $M^{ed} \bmod n = M$ for all $M < n$
  - Selecting two large primes p and q
  - Computing $n=p*q$
  - $\phi(n)=(p-1)(q-1)$
  - Selecting at random the encryption key e where $1<e<\phi(n)$, $\gcd(e,\phi(n)) = 1$
  - Solve following equation to find decryption key d

    $e*d \bmod \phi(n) = 1$ and $0 \le d \le n$

RSA Use

- To encrypt a message M the sender:
  - Obtains public key of recipient $PU=\{e,n\}$
  - Computes: $C = M^e \bmod n$, where $0 \le M < n$
- To decrypt the ciphertext C the owner:
  - Uses their private key $PR=\{d,n\}$
  - Computes: $M = C^d \bmod n$
- Can also use the private key to encrypt the message and use the public key to decrypt the message

RSA Example - Key Setup

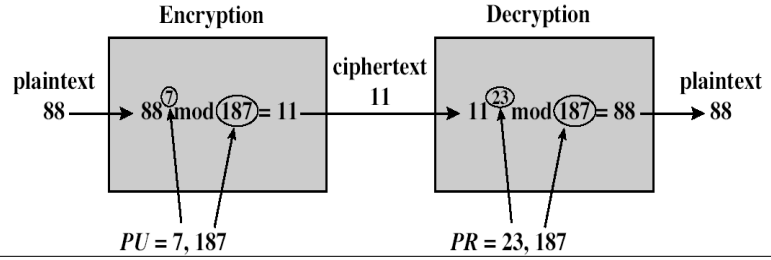
1. Select primes: p=17 & q=11
2. Compute $n = p*q = 17 \times 11 = 187$
3. Compute $\phi(n)=(p-1)(q-1)=16 \times 10=160$
4. Select e: $\gcd(e,160)=1$; choose e=7
5. Determine d: $d*e \bmod 160 = 1$ and $d < 160$. Value is d=23 since $23*7=161= 160+1$
6. Publish public key $PU=\{7,187\}$
7. Keep private key $PR=\{23,187\}$

RSA Example - En/Decryption

- Sample RSA encryption/decryption is:
- Given message M = 88 (nb. 88<187)
- Encryption:

  $C = 88^7 \bmod 187 = 11$
- Decryption:

  $M = 11^{23} \bmod 187 = 88$



RSA Requirements

- Encryption and decryption are of the following form for some plaintext block M and ciphertext block C.

  $C = M^e \bmod n$

  $M = C^d \bmod n = M^{ed} \bmod n$
- The following requirements must be met:
  - Requirement 1: It is possible to find values of e, d, n such that $M^{ed} \bmod n = M$ for all $M < n$
  - Requirement 2: It is relatively easy to calculate $M^e \bmod n$ and $C^d \bmod n$ for all values of $M < n$
  - Requirement 3: It is infeasible to determine d given e and n

RSA Security

- Possible approaches to attacking RSA are:
  - Brute force attacks
  - Mathematical attacks:
    - Factor n into its two prime factors
    - Determine ø(n) directly without determining p and q.
    - Determine d directly.

CS458/CS558 Introduction to Computer Security

Chapter 10
Key Management

Key Management

- Public-key encryption helps address key distribution problems:
  - Use of public-key encryption to distribute secret keys

Distribution of Public Keys

- Several techniques have been proposed for the distribution of public keys:
  - Public announcement
  - Public-key authority
  - Public-key certificates

Public Announcement

- Users distribute public keys to recipients or broadcast to community at large

Public Announcement

- Users distribute public keys to recipients or broadcast to community at large

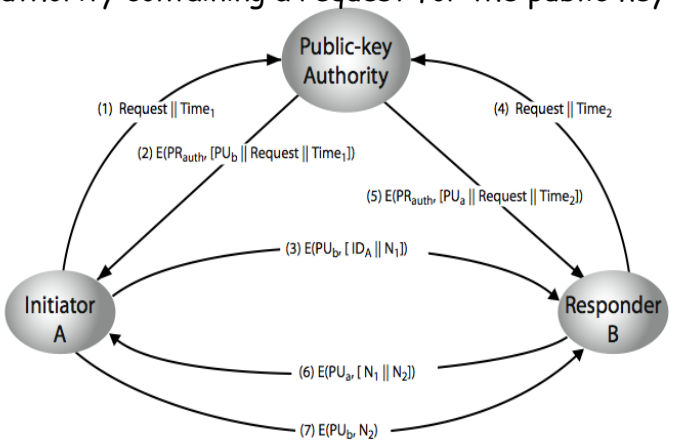
- Major weakness: forgery
  - Anyone can create a key claiming to be someone else and broadcast it
  - Until forgery is discovered can masquerade as

Public-Key Authority

- A central authority maintains a dynamic directory of public keys of all participants {name, public-key}.
- Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure communication.
- Requires users to know public key for the directory. Only the authority knows the corresponding private key.
- Users interact with directory to obtain any desired public key securely.

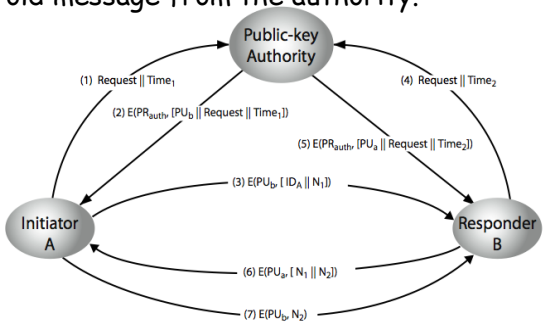
Public-Key Authority

1. A sends a timestamped mesg to the public-key authority containing a request for the public key of B.



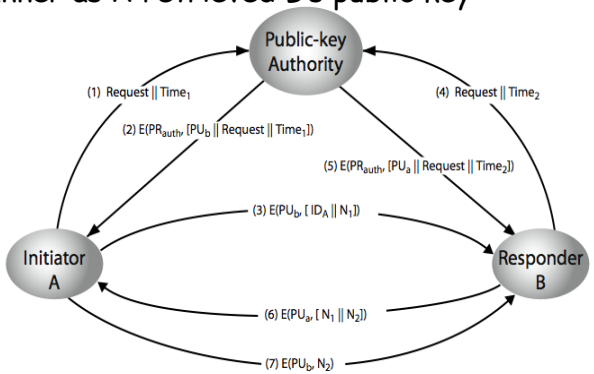
Public-Key Authority

2. The authority responds with a mesg that is encrypted using the authority's private key.
   - B's public key
   - The original request: match with the request
   - The original timestamp: A can determine that this is not an old message from the authority.



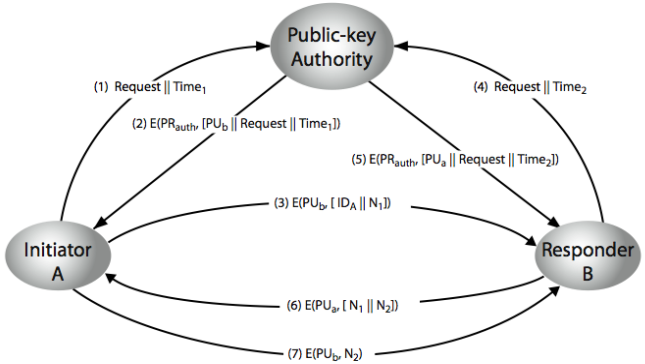
Public-Key Authority

3. A stores B's public key and uses it to encrypt a mesg to B containing an identifier of A and a nonce N1.
4. B retrieves A' public key from the authority in the same manner as A retrieved B's public key



Public-Key Authority

5. B sends a mesg to A encrypted using A's public key that contains A's nonce and a nonce generated by B.
6. A returns N2 encrypted using B's public key, to ensure B that its correspondent is A.



Drawback: Public-Key Authority

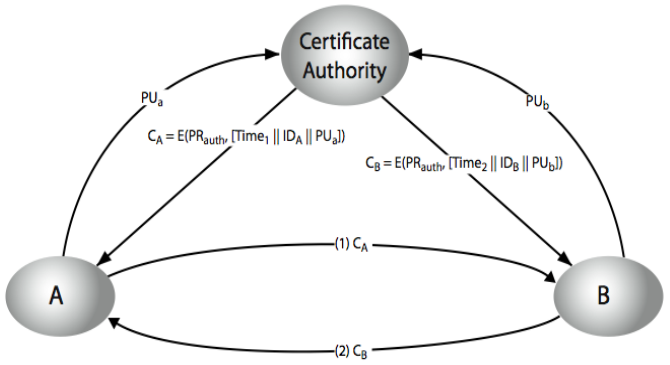
- Drawback of public-key authority: the public-key authority is a bottleneck because a user must appeal to the authority for a public key for every other user that it wishes to contact.

Public-Key Certificates

- Certificates allow key exchange without contacting a public-key authority
- A certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party (certificate authority).
- A user can present his/her public key to the authority in a secure manner and obtain a certificate.
- The user then publishes the certificate.
- Other participant can verify that the certificate was created by the authority.

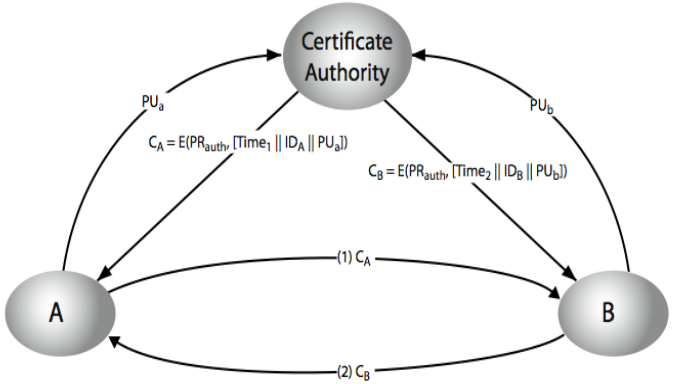
Public-Key Certificates

1. Participant A applies to the certificate authority, supplying a public key and requesting a certificate.
   - Application must be in person or by some form of secure authenticated communication.



Public-Key Certificates

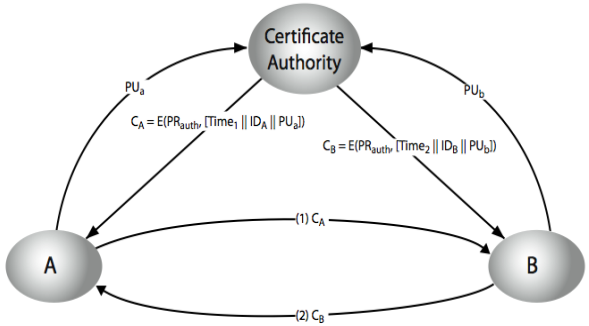
2. The authority provides the certificate of the form CA = E(PRauth, [Time1||IDa||PUa]), where PRauth is authority's private key and Time1 is a timestamp.



Public-Key Certificates

3. A may then pass this certificate on to any other participant, who reads and verifies that the certificate comes from the certificate authority (by decrypting the certificate using authority's public key):

D(PUauth, CA) = D(PUauth, E(PRauth, [T||IDA||PUa]))= (T||IDA||PUa)



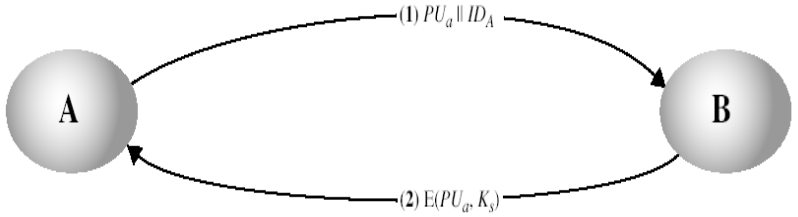
Distribution of Secret Keys Using Public-Key Cryptography

- Use previous methods to obtain public-key
- Can use for secrecy or authentication
- But public-key algorithms are slow
- So usually want to use symmetric-key encryption to protect message contents - need to distribute the session key
- Public-key cryptography can be used to distribute the session keys.

Simple Secret Key Distribution

- Proposed by Merkle in 1979

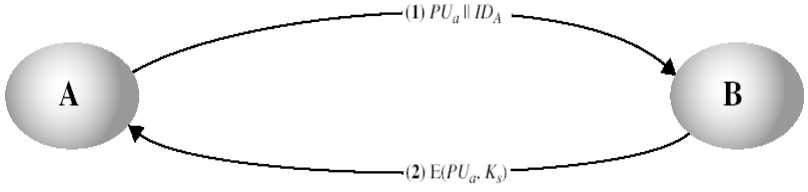
1. A generates a new temporary public key pair
2. A sends B the public key and their identity
3. B generates a session key K, sends it to A encrypted using the supplied public key
4. A decrypts the session key



Simple Secret Key Distribution

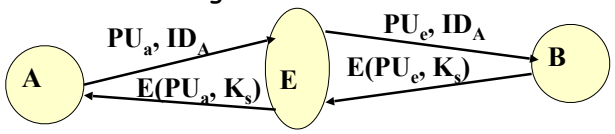
- Proposed by Merkle in 1979

5. A discards the public key pair and B discards A' public key
6. At the completion of the exchange, A and B discard the session key



Simple Secret Key Distribution: Attack

- Problem: man-in-the-middle attack – an adversary can intercept messages and then replay the intercepted message or send another message.



- A transmits a message intended for B consisting of PU_a and A's identifier ID_A
- E intercepts the message, creates its own public/private key pair $\{PU_e, PR_e\}$ and transmits $PU_e||ID_A$ to B
- B generates a secret key K_s and transmits $E(PU_e,K_s)$
- E intercepts the message, and learns Ks by computing $D(PR_e, E(PU_e, K_s))$

Chapter 10.2
Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange

- The first public-key algorithm proposed by Diffie & Hellman in 1976
- The purpose is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages.
- Used in a number of commercial products

Diffie-Hellman Key Exchange

- An integer a is a primitive root of a prime number q if $a \bmod q$, $a^2 \bmod q$, …, $a^{q-1} \bmod q$ are distinct and consist of the integers from 1 through $q-1$ in some permutation.

- Is 2 a primitive root of 5?

Diffie-Hellman Key Exchange

- An integer a is a primitive root of a prime number q if $a \bmod q$, $a^2 \bmod q$, …, $a^{q-1} \bmod q$ are distinct and consist of the integers from 1 through $q-1$ in some permutation.

- Is 2 a primitive root of 5?

  Yes

Diffie-Hellman Key Exchange

- An integer a is a primitive root of a prime number q if $a \bmod q$, $a^2 \bmod q$, …, $a^{q-1} \bmod q$ are distinct and consist of the integers from 1 through $q-1$ in some permutation.
- Agree on two numbers:
  - A prime number q
  - An integer a that is the primitive root of q
- Each user generates his/her key
  - Chooses a private key (number): $x < q$
  - Computes their public key: $y = a^x \bmod q$
- Each user keeps the x value private and makes the y value available publicly

Diffie-Hellman Key Exchange

Property of modular arithmetic

$[(a1 \bmod n) *...* (am \bmod n)] \bmod n = (a1*...*am) \bmod n$

- Shared session key for users A & B is K_{AB}:

$K_{AB} = a^{x_A.x_B} \bmod q$

$= y_A^{x_B} \bmod q$ (which B can compute)

$= y_B^{x_A} \bmod q$ (which A can compute)

- K_{AB} is used as session key in private-key encryption scheme between A and B
- Question: How to prove that $y_A^{x_B} \bmod q = y_B^{x_A} \bmod q$

Diffie-Hellman Key Exchange

Property of modular arithmetic

$[(a1 \bmod n) *...* (am \bmod n)] \bmod n = (a1*...*am) \bmod n$

- prove that $y_A^{x_B} \bmod q = y_B^{x_A} \bmod q$

$y_A^{x_B} \bmod q$

$= (a^{x_A} \bmod q)^{x_B} \bmod q$

$= [(a^{x_A} \bmod q)*... *(a^{x_A} \bmod q)] \bmod q$ (x_B terms)

$= a^{x_A x_B} \bmod q = a^{x_B x_A} \bmod q$

$= [(a^{x_B} \bmod q)*... *(a^{x_B} \bmod q)] \bmod q$ (x_A terms)

$= y_B^{x_A} \bmod q$

Diffie-Hellman Example

- Users A & B who wish to swap keys:
- Agree on prime $q=353$ **and** $a=3$
- Select random private keys:
  - A chooses $x_A=97$, B chooses $x_B=233$
- Compute respective public keys:
  - $y_A=3^{97} \bmod 353 = 40$ **(A)**
  - $y_B=3^{233} \bmod 353 = 248$ **(B)**
- Compute shared session key as:
  - $K_{AB}= y_B^{x_A} \bmod 353 = 248^{97} \bmod 353 = 160$ **(A)**
  - $K_{AB}= y_A^{x_B} \bmod 353 = 40^{233} \bmod 353 = 160$ **(B)**

Summary: Diffie-Hellman Key Exchange