



Network Security

CSI ZG513 / ES ZG513 / SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Lecture Session – I

Introduction

- **Instructor**

Hemant Rathore

D - 155

Department of Computer Science & Information Systems

BITS Pilani, K K Birla Goa Campus

hemantr@goa.bits-pilani.ac.in

Course Page:

<http://taxila-aws.bits-pilani.ac.in/course/view.php?id=7862>

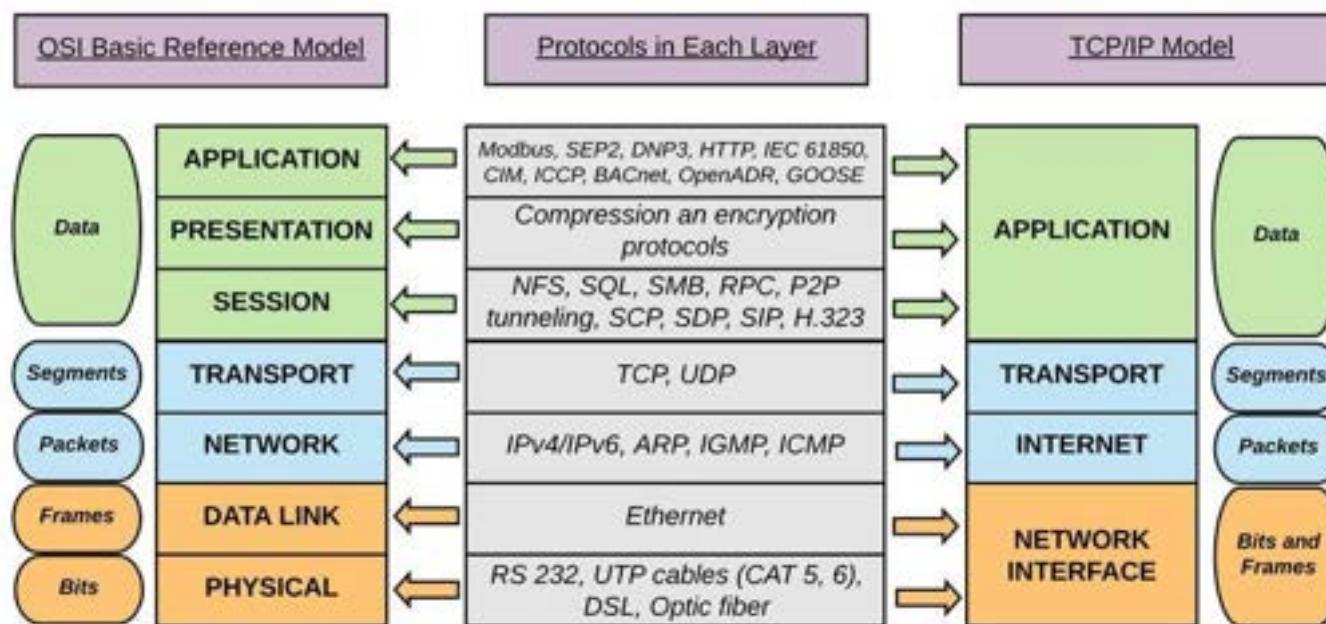
Course Objectives

- Information security is an important area of information technology and this course on Network Security help audience to understand the three important security goals in the networks –
 - Confidentiality
 - Integrity
 - Availability
 - Also Authenticity and Non-repudiation and cryptographic techniques to implement these security goals.



Course Objectives

- The course provides a top down approach to explore the security implementations in different layers - application, transport and network.



Course Objectives

- The course provides a necessary review of mathematical concepts to implement different cryptographic techniques to achieve the network security goals and then provides a deeper dive to the field of cryptography - symmetric and asymmetric key cryptography and methods to implement them.
- The course consolidates and sums up the learning taking few case studies and examples from latest trends and industry deployments.

Text & Reference Books

- Stallings William: Cryptography and Network Security - Principles and Practice, Pearson India, 6th Edition, 2014.
- Forouzan B A, Mukhopadhyay Debdeep : Cryptography and Network Security, McGraw Hill, 2nd Edition, 2010.
- Schneier Bruice: Applied Cryptography : Protocols, Algorithms And Source Code In C, Wiley India, 2nd Edition, Reprint – 2013
- Kurose James F and Keith W. Ross: Computer Networking: A Top-Down Approach, Pearson India, 5th Edition, 2012.

Text & Reference Books

- Christof Paar and Jan Pelzl: Understanding Cryptography - A Textbook for Students and Practitioners, Springer, 1st Edition, 2010.
- Being a WILP level course, no single book is sufficient.
- Materials from different sources.

Evaluation Scheme

- **EC1:** Three Quizzes, each of 5% weightage.
 - February 14 to 24, 2022
 - March 14 to 24, 2022
 - April 14 to 24, 2022
 - **EC2:** Mid-Semester Test 35% (Open Book)
 - 11/03/2022 (FN) 10 AM – 12 Noon
 - **EC3:** Comprehensive Exam 50% (Open Book)
 - 20/05/2022 (FN) 10 AM – 12 Noon
-

Contact Sessions

- 22 Lectures each of 50 minutes.
- CS-1:
 - Network Security and OSI Security Architecture
 - Review of Attacks, Mechanisms and Services, Network Security Model
- CS-2:
 - Network Security Model
 - Techniques to Implement Network Security

Contact Sessions

- CS-3:
 - Cryptography, Classical Encryption
 - Breaking the Cryptosystem
- CS-4:
 - Modular Arithmetic, Groups and Rings
 - One example each in classical substitutive and transposition cipher
- CS-5:
 - Random numbers, its types and usage.
 - TRNG, PRNG, CSPRNG
 - Review of BBS

Contact Sessions

- CS-6:
 - Stream Ciphering
 - RC4 algorithm
 - CS-7:
 - Basic Number Theory
 - Extended Euclidean Algorithm.
 - CS-8:
 - Galois Field
 - Polynomial Arithmetic
 - CS-9:
 - Block Ciphering
 - Confusion and Diffusion Theory
-

Contact Sessions

- CS-10:
 - AES and its importance in security
 - Efficient implementation of AES.
- CS-11:
 - Recapitulation of all the sessions / problem solving before mid-semester exams
- CS-12:
 - Modes of Operation and its applications
 - Multiple Encryption and Meet-in-the Middle Attack
- CS-13:
 - SHA-1 and SHA-3
 - HMAC and CBC-MAC and its Security

Contact Sessions

- CS-14:
 - Model of Asymmetric Key Cryptography
 - Factorization and other methods for Public Key Cryptography
- CS-15:
 - RSA and OAEP
 - Diffe-Hellman Key Exchange and its Security Aspects
- CS-16:
 - Distribution of Symmetric and Asymmetric Key
 - Digital Signature: DSA
 - X.509 Certificate
 - Man-in-the Middle Attack

Contact Sessions

- CS-17:
 - User/Entity Authentication
 - Kerberos
- CS-18:
 - Review of PGP - Authentication and Confidentiality
 - Review of MIME and S/MIME
- CS-19:
 - Review of Web Security
 - Review of SSL and TLS

Contact Sessions

- CS-20:
 - IPSec: Authentication Header and Encapsulated Security Protocol
 - SAD and SPD with inbound/outbound packet processing
- CS-21:
 - Malicious Software and its Detection Techniques
 - Review of Intrusion and Intrusion Detection
- CS-22:
 - Recapitulation of all the sessions / problem solving before comprehensive exams.

Quote

“The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable”

Sun Tzu

Background

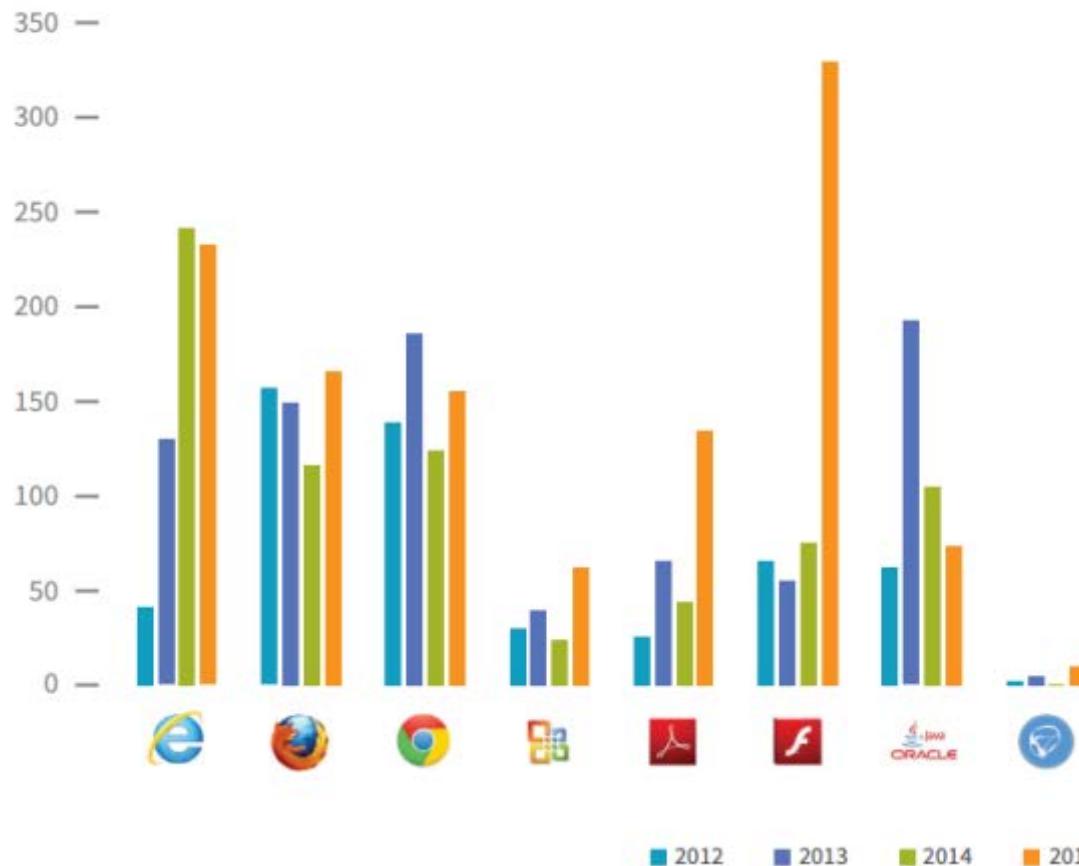
- Information Security requirements have changed in recent times.
 - Traditionally provided by physical and administrative mechanisms.
 - Computer use requires automated tools to protect files and other stored information.
 - Use of networks and communications links requires measures to protect data during transmission.
-

Definitions

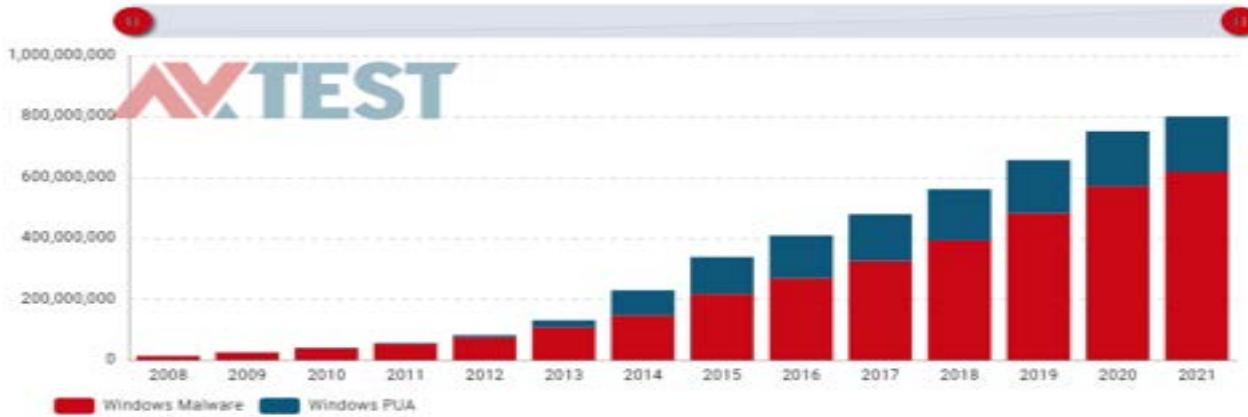
- **Computer Security:** generic name for the collection of tools designed to protect data and to thwart hackers.
 - **Network Security:** measures to protect data during their transmission
 - **Internet Security:** measures to protect data during their transmission over a collection of interconnected networks
-

Security Trends

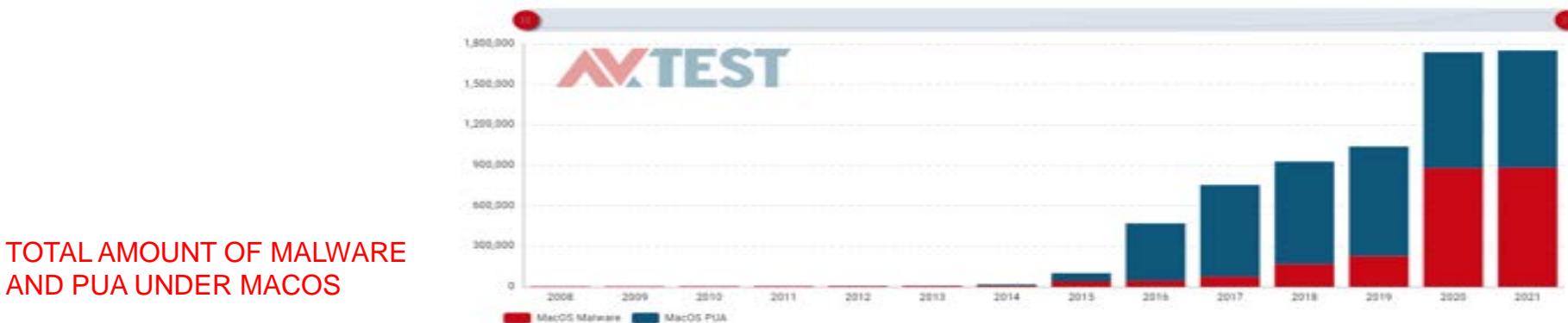
FIGURE 1: VULNERABILITY DYNAMICS 2012-2015



Malware Growth (1)



TOTAL AMOUNT OF MALWARE AND PUA UNDER WINDOWS

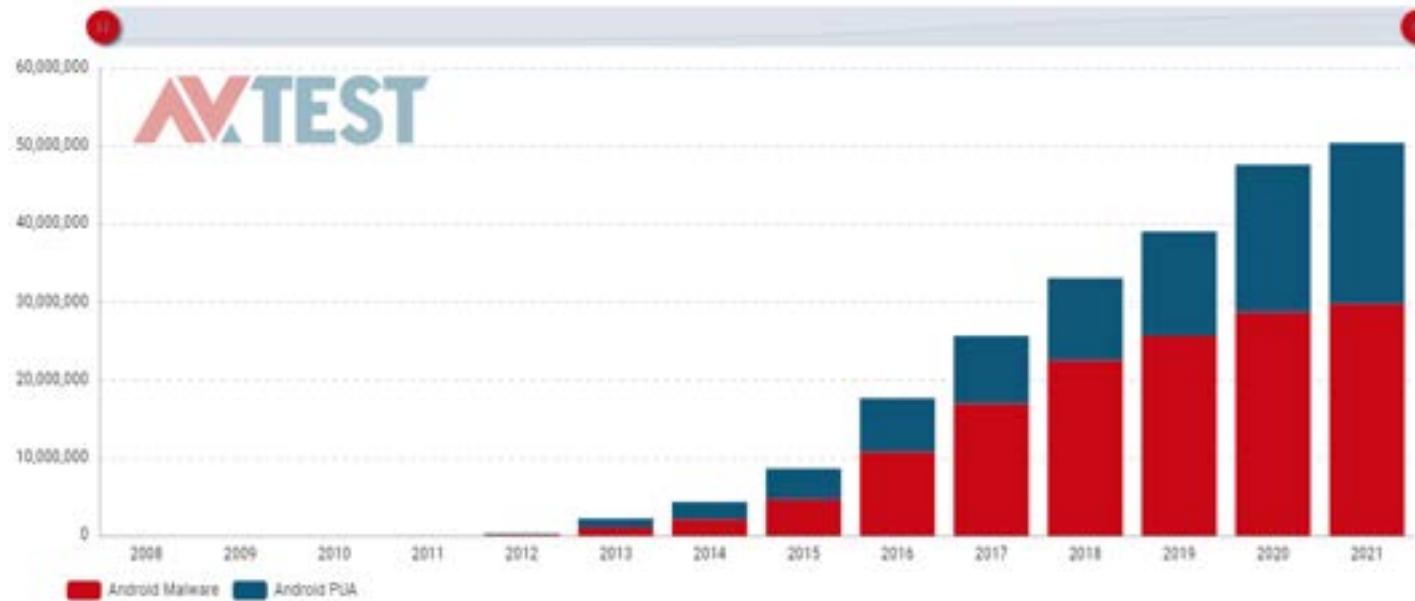


TOTAL AMOUNT OF MALWARE AND PUA UNDER MACOS

Source L: <https://www.av-test.org/en/statistics/malware/>

Source R: <https://www.av-test.org/en/statistics/malware/>

Malware Growth (2)



30 Jan 2018:

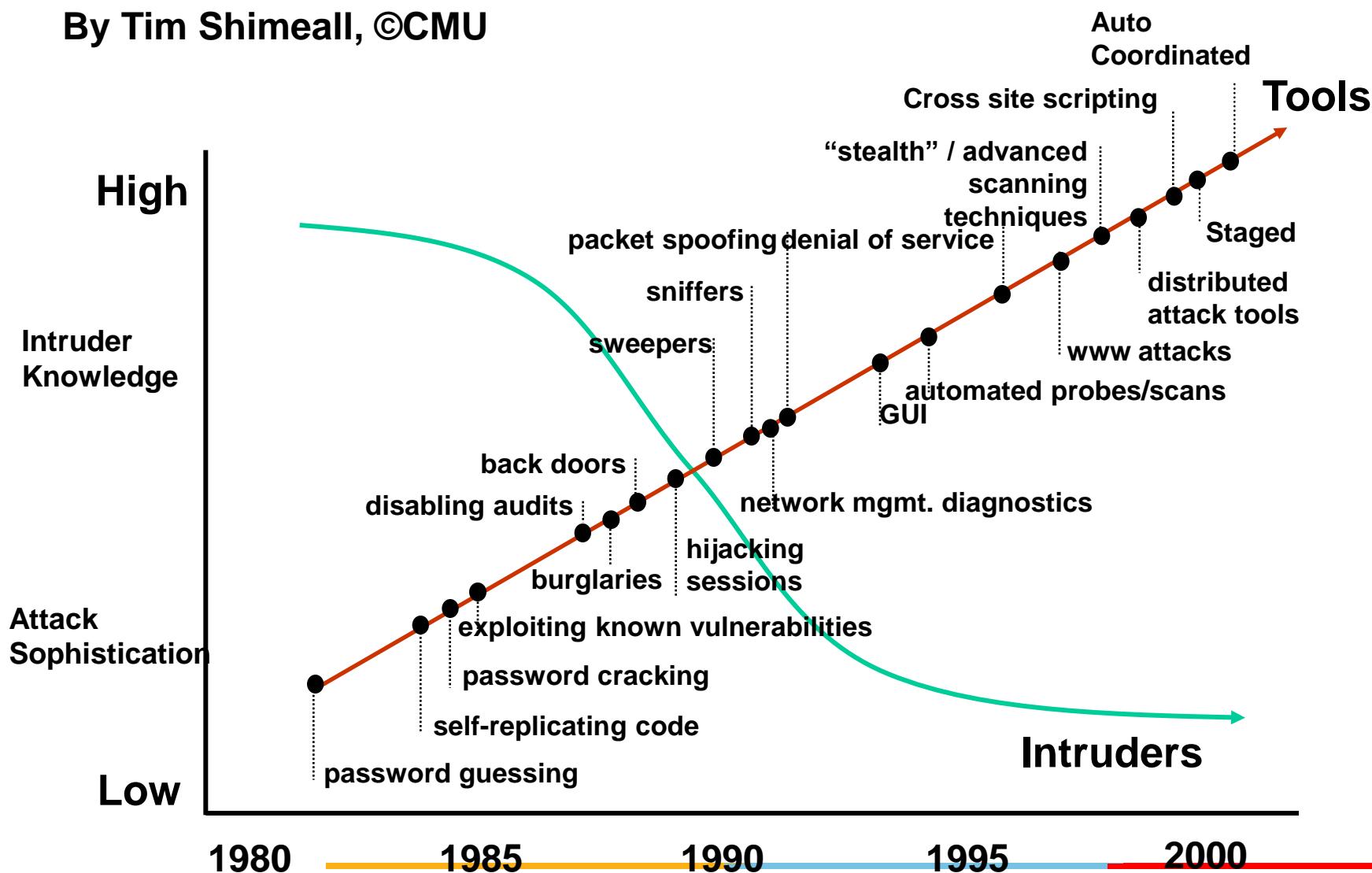
~ 7×10^5 malicious / bad apps are removed from Google Play Store in 2017, which is ~70% more than the apps taken down in 2016

Source T: <https://portal.av-atlas.org/malware/statistics>

Source B: <https://android-developers.googleblog.com/2018/01/how-we-fought-bad-apps-and-malicious.html>

Security Trends

By Tim Shimeall, ©CMU



CSI/FBI 2005 Computer Crime and Security Survey



Thanks!!!
Queries?



Network Security

CSI ZG513 / ES ZG513 / SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Lecture Session – 2

Aspects of Security

- Security level
 - Security attack
 - Security truism
 - Security mechanism
 - Security service
 - Security response teams
 - Security policies.
-

Security Levels

- **Level D1:**

- Lowest level of security.
- Entire system is untrusted.
- No protection for hardware, compromised OS.
- No authentication regarding users i.e. no defined method of determining who is typing at the keyboard and their right to access information stored in the computer.

Security Levels

- Level C1 (Sublevel of level C):

- Discretionary Security Protection System (UNIX).
 - Some protection for hardware exists, not easily compromised, although it is still possible.
 - User must be identified (user ID & pwd) i.e. access rights to programs and information each user has.
 - System administrator can do any activity, hence security can easily compromise of system without any one's knowledge.
 - Not uncommon to find 23 person's know root pwd, in an organization, hence no way exists to distinguish changes made by X to the system yesterday.
-

Security Levels

- Level C2 (Sublevel of level C):

- Addition to C1, create a controlled access environment i.e. users has restricted command to use, files, not only permissions but upon authorization levels.
- Requires that system be audited i.e. writing a audit record for each event that occurs in the system.
- Auditing is used to keep record such as those activities performed by the system administrator, also with the use additional authorization it is possible to perform su job without having root password.
- Additional authorizations shall not be confused with SGID & SUID permissions.

Security Levels

- Level B1 (Sublevel of level B):
 - Labeled Security Protection.
 - The first level that supports multilevel security, such as secret and top secret.
 - An object under mandatory access control cannot have its permission changed by the owner of the file.

Security Levels

- Level B2 (Sublevel of level B):
 - Known as Structured Protection, requires that every object be labeled such as disks, tapes or terminals might have single or multilevel of security assigned to them.
 - This is the first level that starts to address the problem of an object at a higher level of security communicating with another object at a lower level of security.

Security Levels

- Level B3 (Sublevel of level B):
 - Security Domain level, enforces the domain with the installation of hardware, for e.g.
 - Memory management hardware is used to protect the security domain from unauthorized access or modification from objects in different security domains.
 - This level also requires that the user's terminal be connected to the system through a trusted path.

Security Levels

- Level A:
 - Verified Design level, currently the highest level of security validated through orange book, includes a stringent design, control and verification process.
 - To achieve level A, all the component of the lower levels must be included.
 - Design must be mathematically verified and an analysis of covert channels and trusted distributions must be performed.

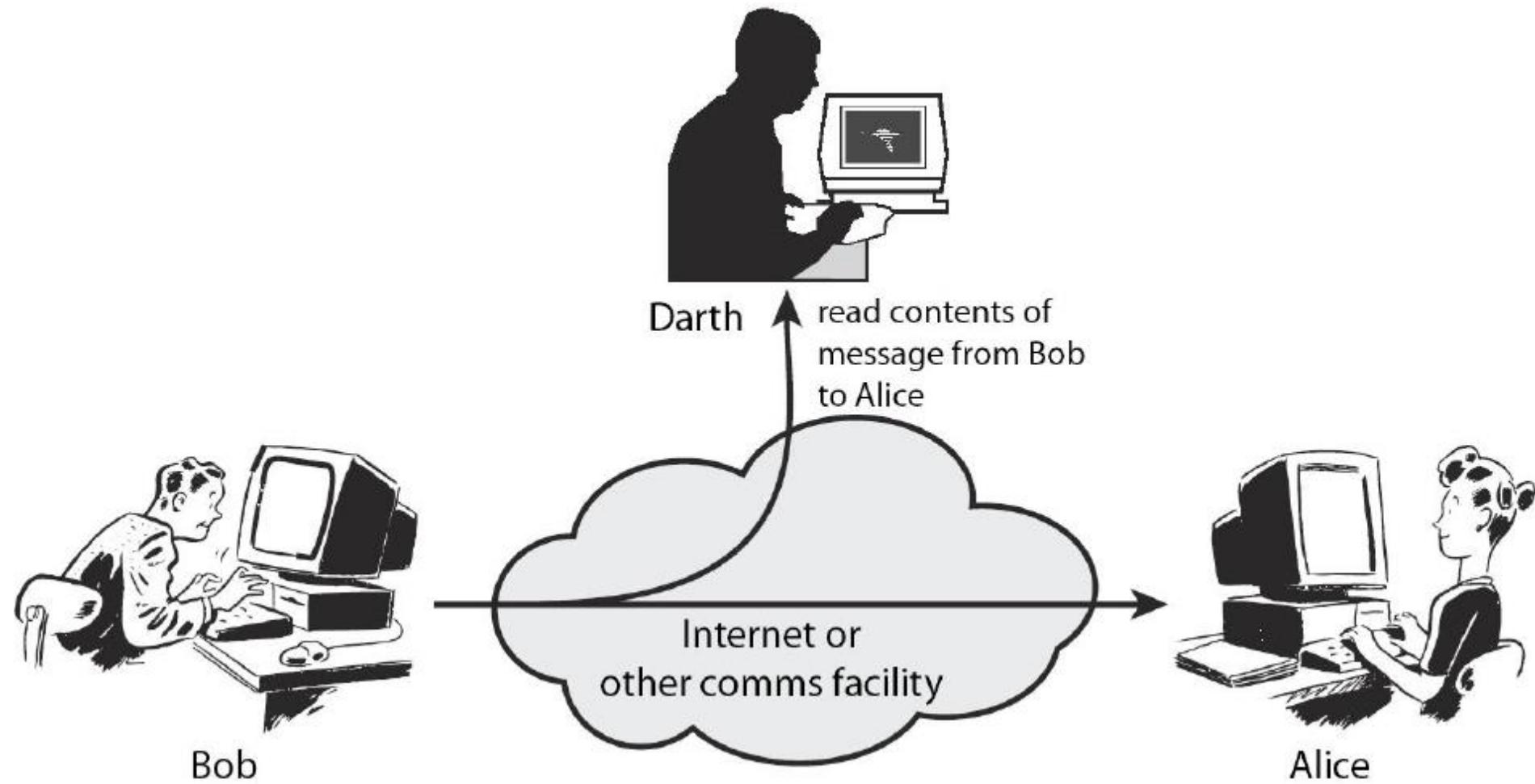
Aspects of Security

- Security level
 - Security attack
 - Security truism
 - Security mechanism
 - Security service
 - Security response teams
 - Security policies.
-

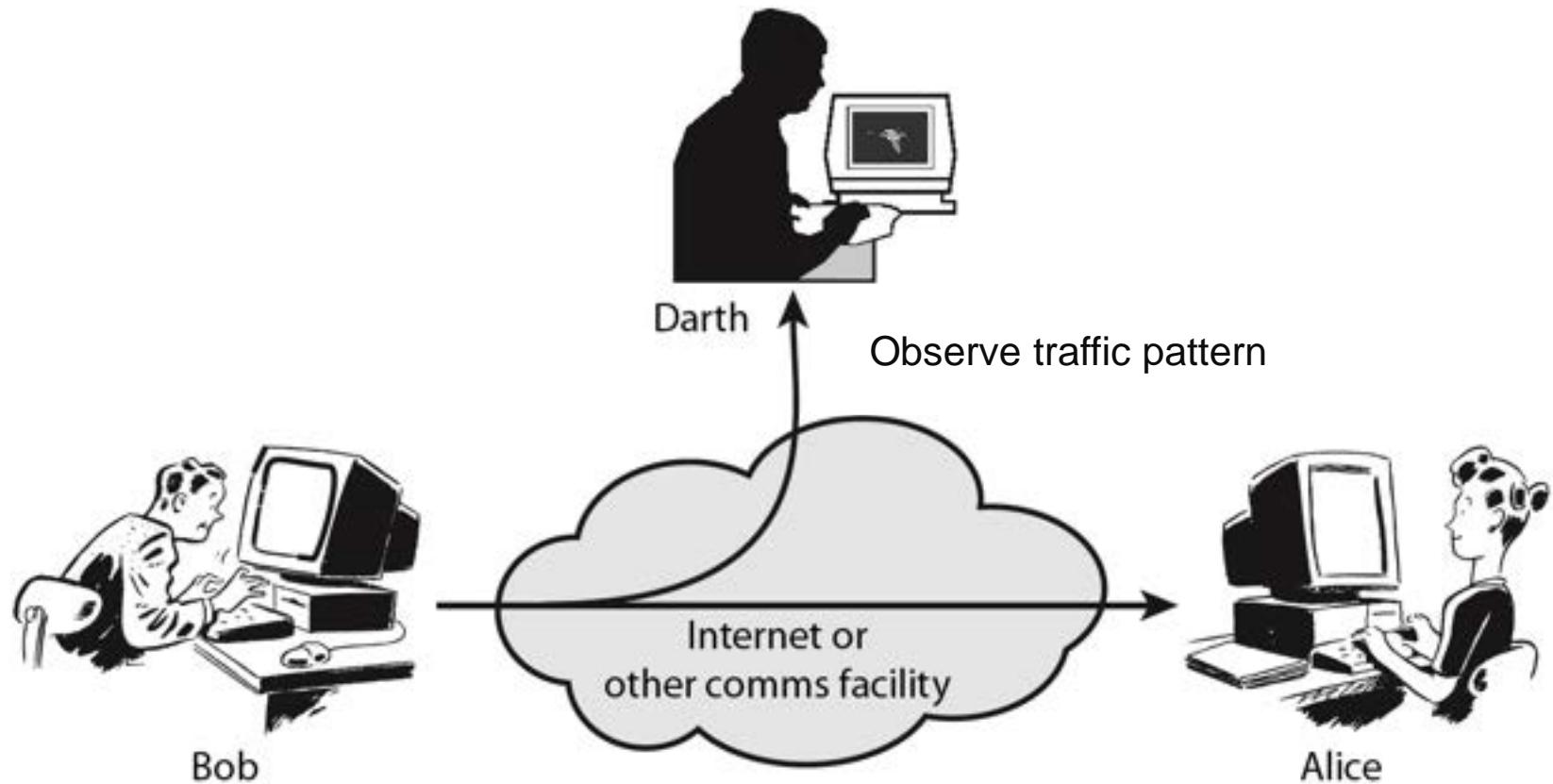
Security Attacks

- Any action that compromises the security of information owned by an organization
- Information security is about how to prevent attacks, or failing that, to detect attacks on information based systems.
- Have a wide range of attacks.
- Focus on generic types of attacks
 - Passive
 - Active

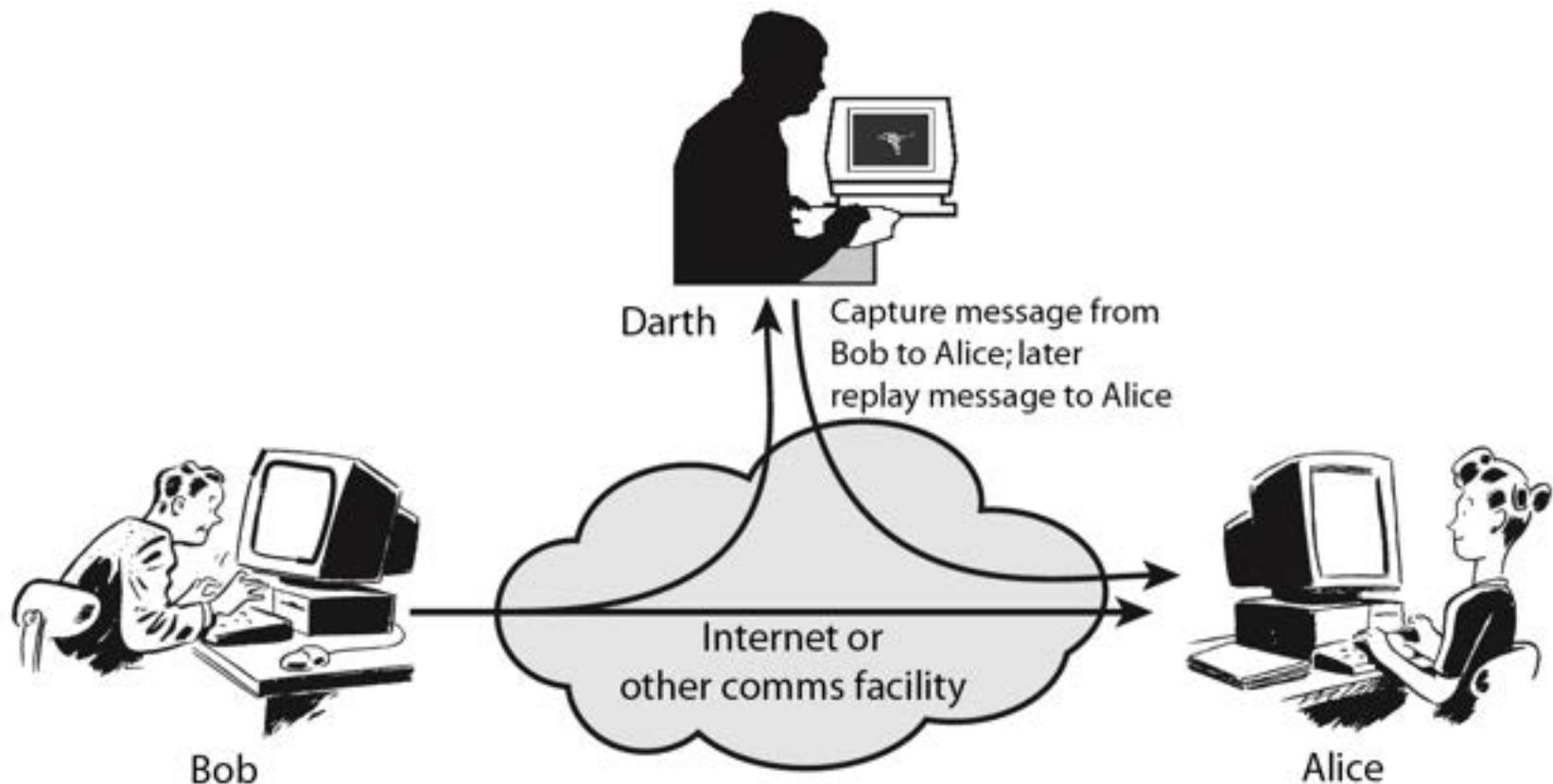
Passive Attacks - Interception



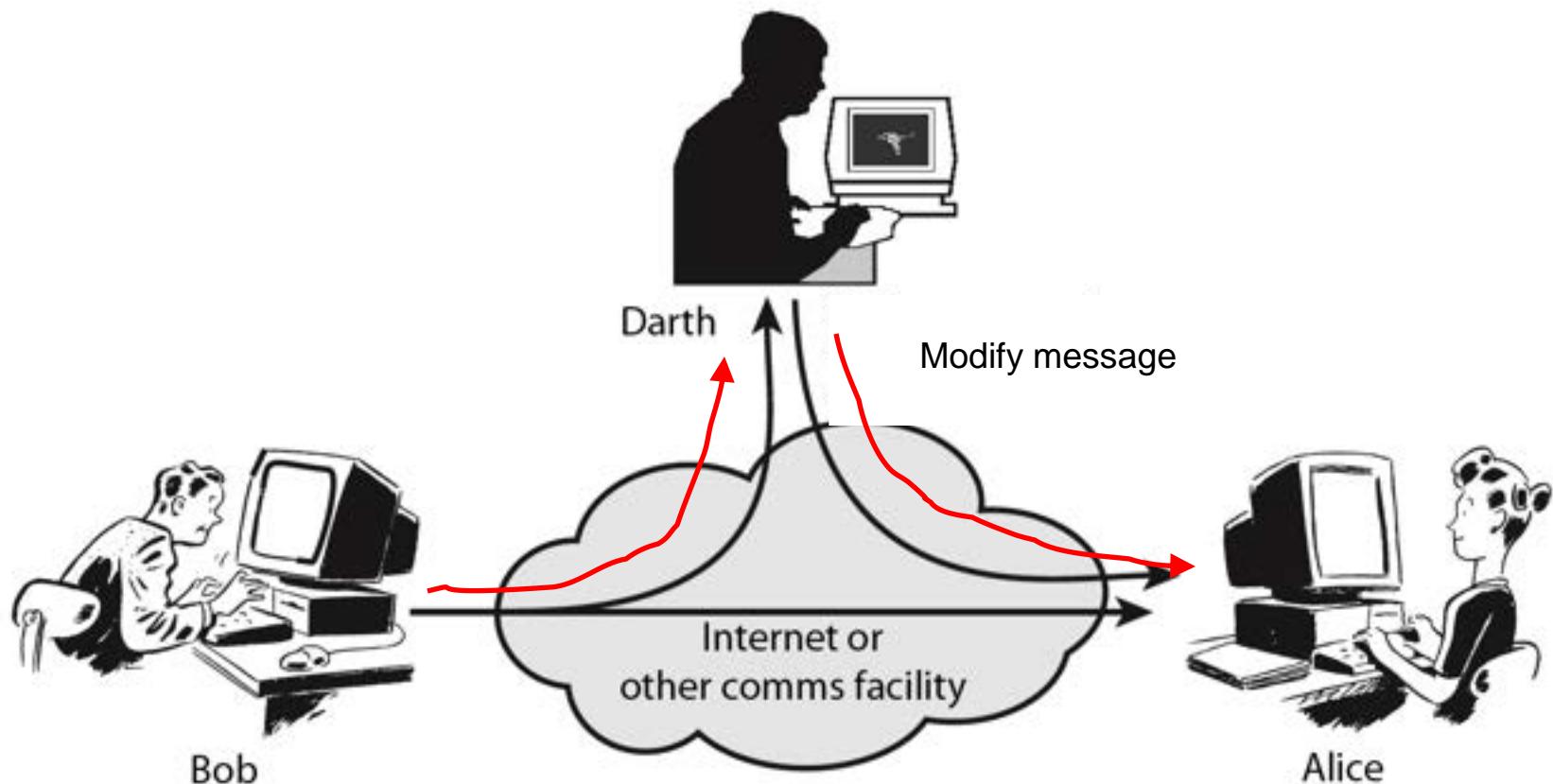
Passive Attacks - Traffic Analysis



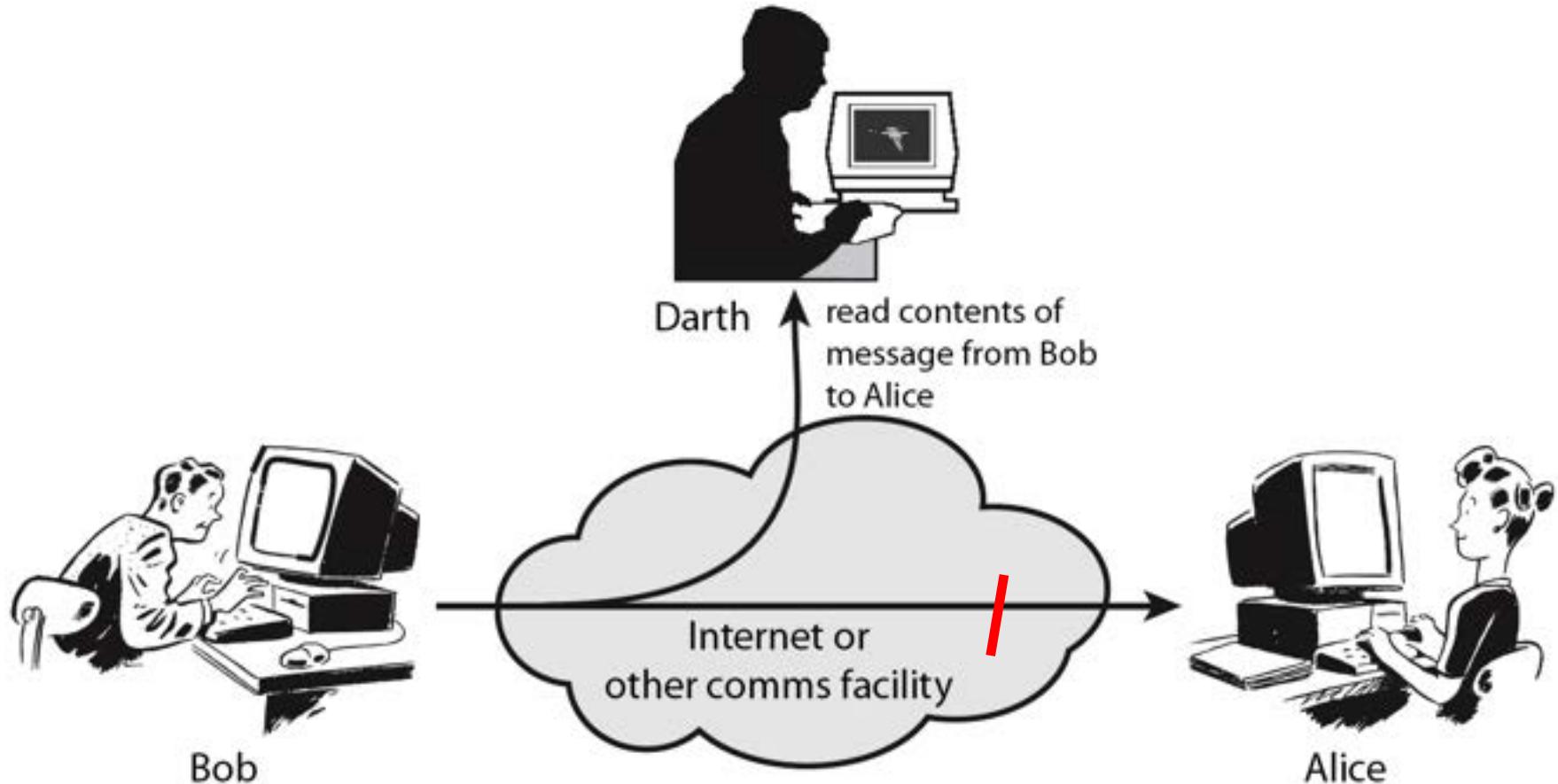
Active Attacks - Reply



Active Attacks - Modification



Active Attacks - Interruption



Handling Attacks

- Passive attacks – focus on Prevention
 - Easy to stop
 - Hard to detect
- Active attacks – focus on Detection and Recovery
 - Hard to stop
 - Easy to detect

Security Policies

- Which is not expressly prohibited is permitted.
 - Which is not expressly permitted is prohibited.
 - What resource are you trying to protect.
 - Who is interested in attacking you.
 - How many security can you afford.
 - What action will be taken when security is compromised in the organization.
 - What action will be tolerated and what will not.
-

Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Last Class

- Course Objectives
- Books and Evaluation Scheme
- Course Content
- Introduction to Network Security
- Security Levels
- Active and Passive Attacks

Aspects of Security

- Security level
 - Security attack
 - Security truism
 - Security mechanism
 - Security service
 - Security response teams
 - Security policies.
-

Security Truism

- There is no such thing as absolute security.
 - Security is always a question of economics.
 - Keep the level of all your defenses at about the same height.
 - Attacker doesn't go through security, but around it.
 - Put your defenses in layers.
 - Its bad idea to rely on “Security through obscurity.”
-

Security Truism

- If you don not run a program, it does not matter if it has security holes.
- A program/protocol is insecure until proven secure.
- A chain is only as strong as its weakest link.
- Security is a tradeoff with convenience.
- Don't underestimate the value of your assets.



Applications of Crypto/NS



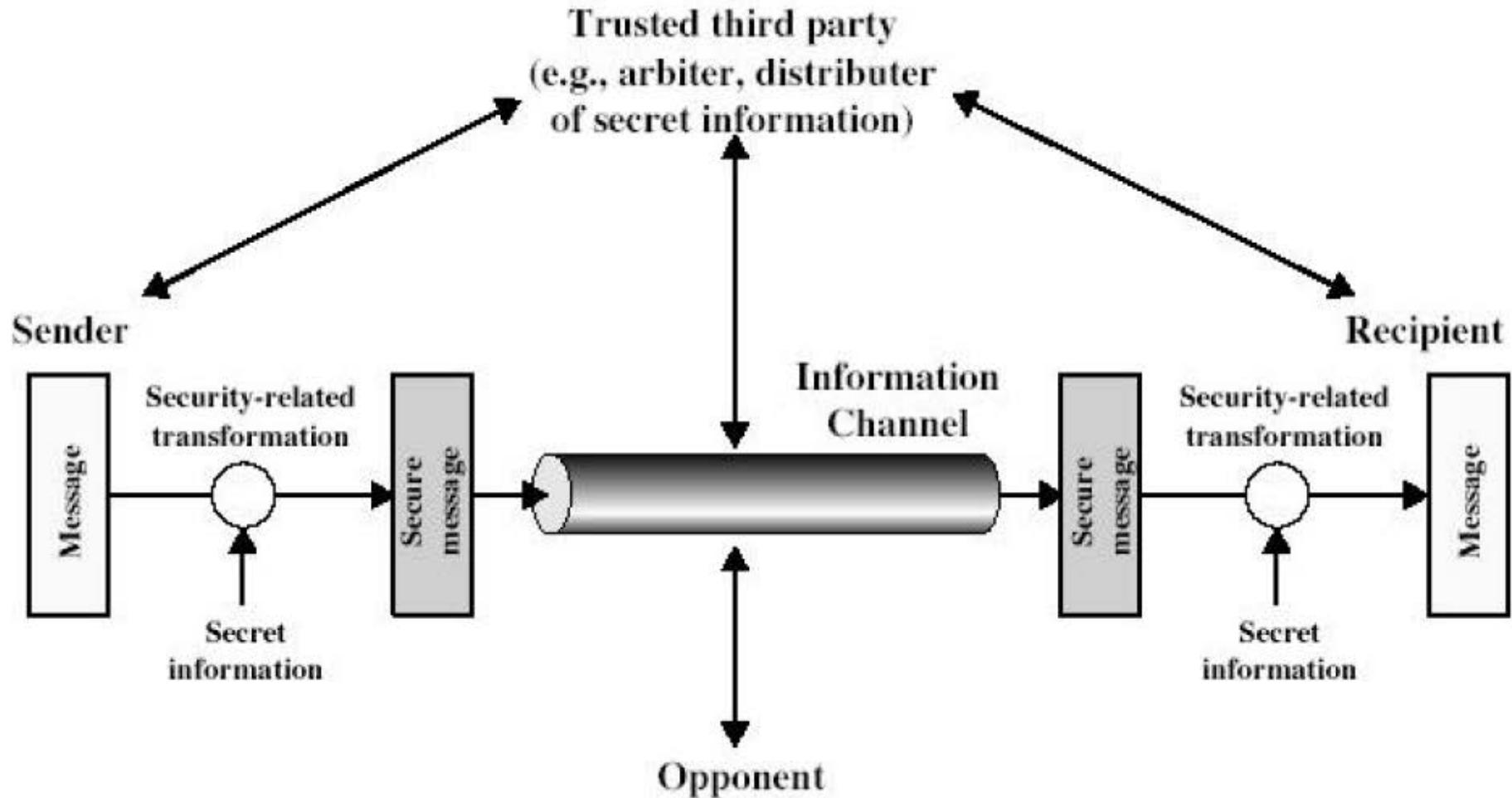
Cryptology

Model for Network Security

- A likes to send a message to B securely.
- Insecure Channel.
- Secure Channel.



Model for Network Security







Cryptography: Terminology

- **Cryptography:** science of secret writing with the goal of hiding the meaning of a message.
 - **Cryptanalysis:** art and science to break the cryptosystem.
 - **Encryption:** method of transforming data (x) into an unreadable format.
 - **Plaintext:** message/data before encryption.
 - **Ciphertext:** message/data after encryption.
 - **Decryption:** method to get back the ' x ' from ' y '.
-

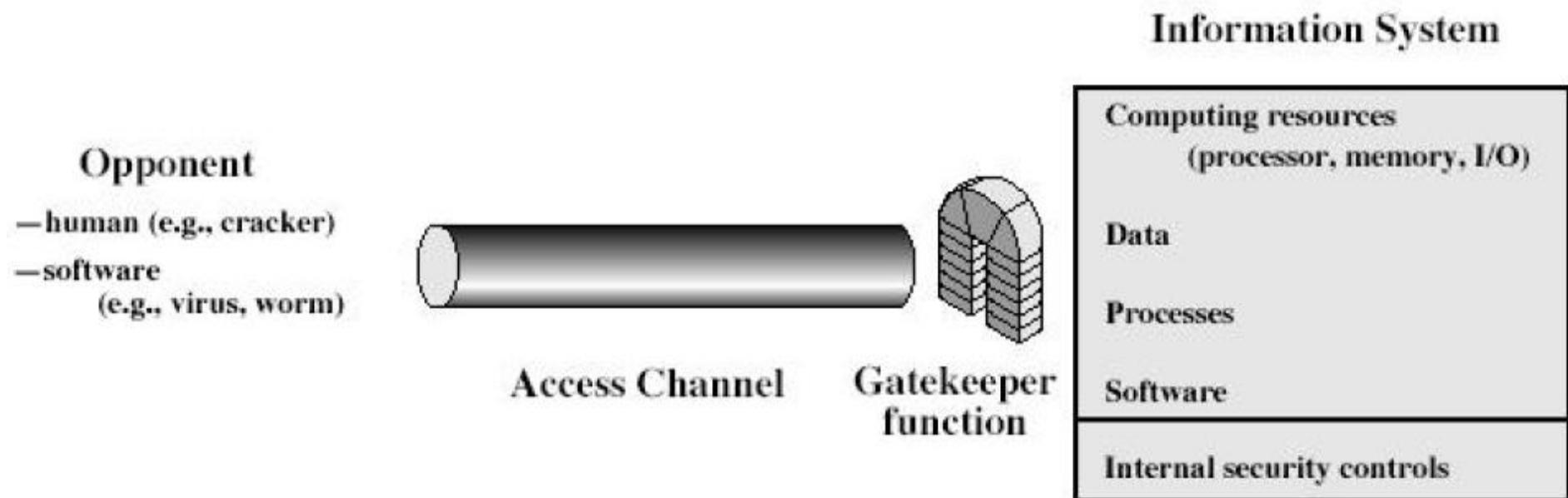
Cryptography: Terminology

- **Cipher/EA:** set of rules/procedures that dictates how to encrypt/decrypt data.
- **Key:** values used in encryption/decryption.
- **Key space:** range of possible values used to construct keys.
- **Key clustering:** when two different keys generate the same 'y' from the same 'x'.
- **Work factor:** estimated time and resources to break a cryptosystem. **No system is unbreakable.**

Model for Network Security

- Using this model requires us to:
 - Design a suitable algorithm for the security transformation.
 - Generate the secret information (keys) used by the algorithm.
 - Develop methods to distribute and share the secret information.
 - Specify a protocol enabling the principals to use the transformation and secret information for a security service

Model for Network Security



Model for Network Security

- Using this model requires us to:
 - Select appropriate gatekeeper functions to identify users
 - Implement security controls to ensure only authorised users access designated information or resources.
 - Trusted computer systems may be useful to help implement this model.
-

Thanks!!!
Queries?



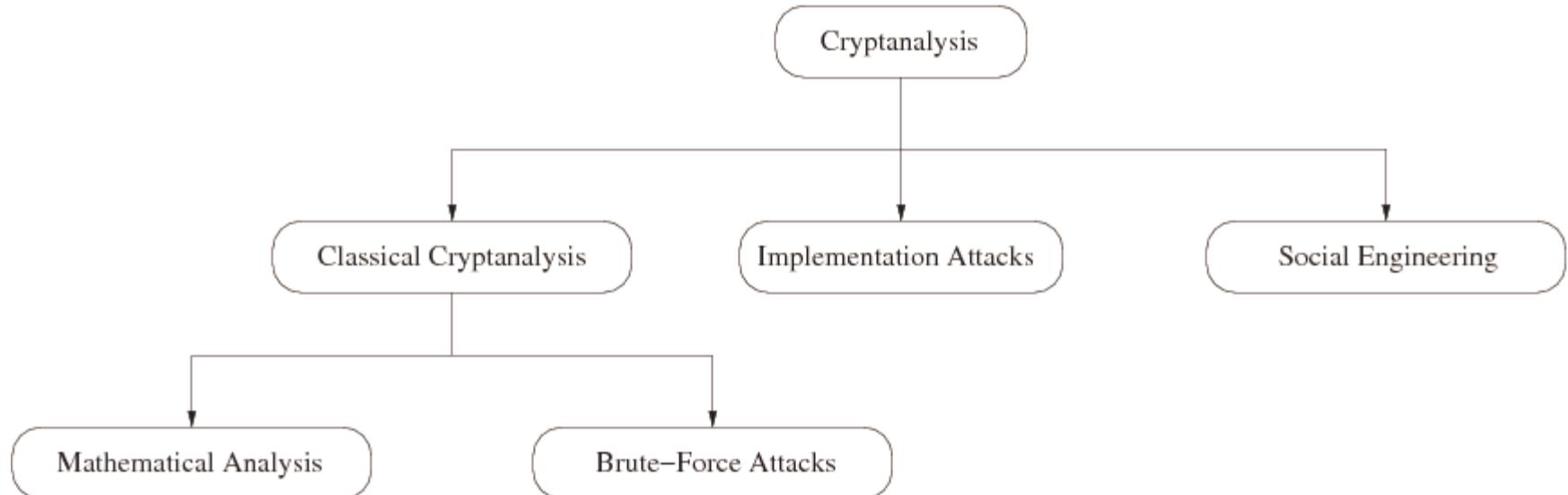
Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Cryptanalysis





Cryptanalysis

Substitution Cipher

Plaintext		Ciphertext
A	->	q
B	->	w
C	->	e

Eg: ABC would be encrypted as qwe

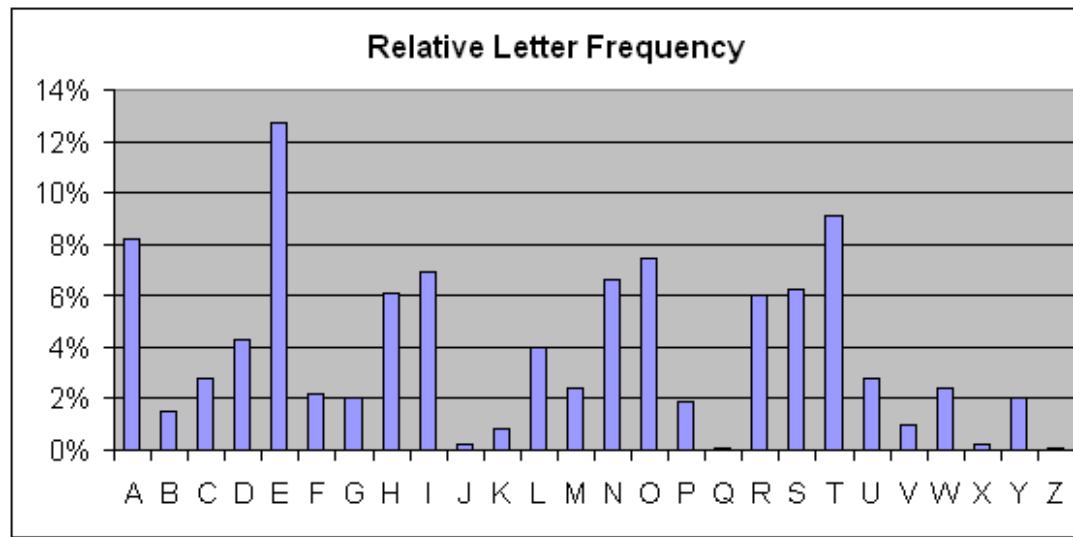
- Ciphertext:
- qwertyuiop
- How secure is the Substitution Cipher ?

Substitution Cipher

- Brute Force Attack:
 - Try every possible key
 - How many keys are there
$$26 * 25 * 24 * 23 \dots \dots 1 = 26! = 2^{88}$$
 - Not possible to solve by todays computer
 - Is Substitution Cipher unbreakable ?

Substitution Cipher

- Letter frequency analysis



- Count the frequency of letters in Cipher text and replace it with help of letter frequency analysis.

Security Services

- Enhance security of data processing systems and information transfers of an organization.
- Intended to counter security attacks.
- Using one or more security mechanisms.
- Often replicates functions normally associated with physical documents, for e.g.
 - Have signatures, dates; need protection from disclosure, tampering, or destruction; be notarized or witnessed; be recorded or licensed.

Security Services

- **X.800:**
“a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers”
 - **RFC 2828:**
“a processing or communication service provided by a system to give a specific kind of protection to system resources”
-

Security Services (X.800)

- **Data Confidentiality** - protection of data from unauthorized disclosure.
 - **Authentication** - assurance that the communicating entity is the one claimed.
 - **Integrity** - assurance that data received is as sent by an authorized entity.
 - **Non-Repudiation** - protection against denial by one of the parties in a communication.
 - **Access Control** - prevention of the unauthorized use of a resource.
-

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 Mechanism (X.800)

- **Specific security mechanisms:**
 - Encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization.

- **Pervasive security mechanisms:**
 - Trusted functionality, security labels, event detection, security audit trails, security recovery .

Security Response Teams

- **Computer Emergency Response Team:**

- The goal of CERT/CC was to address computer security concerns of research users of the internet.
- The promotion of CERT/CC team was to prevent and handle incidents such as internet worms.
- Has ability to immediately confer with experts to diagnose and solve security problems.
- Work with vendors of software systems in order to coordinate the fixes for security problems.
- Help you indirectly in formulating an effective network security policy.
- Operates 24-hour hotline that you can call to report security problems. e.g. someone breaking your system.

Security Response Teams

- **Computer Emergency Response Team:**

- To join the CERTAdvisory mailing list, send a message to
<certrequest@cert.sei.cmu.edu>
- Address:
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 152133890
(412)2687090
E-mail:cert@cert.sei.cmu.edu
URL: <http://www.cert.org>

Security Response Teams

- **NIST Computer Security Resource and Response Clearinghouse:**
 - Besides dealing with standards issues, also has responsibility for computer science & tech. activities.
 - Provide help & information regarding computer security events and incidents.
 - Also interested in raising awareness about computer security vulnerabilities.
 - Publications related to computer security and computer viruses.
 - CSRC, A216 Technology, Gaithersburg, MD 20899,
(301)9755200, <csrc@nist.gov>, <http://csrc.nist.gov>

Security Response Teams

- **DOE Computer Incident Advisory Capability:**
 - Primary responsibility is to assist DOE sites faced with computer security incidents such as intruder attacks, virus, worms attack and so on.
 - Keeps informed of current securityrelated events, and maintain contact with other response teams & agencies.
 - Develops guideline for security incident handlings & develops software for responding to security incident.
 - Analyzes security events and trends, and conducts training and awareness activites to alert and advise sites about vulnerabilites and potential attacks.
 - (415)4228193, <ciac@tiger.llnl.gov>, <http://ciac.llnl.gov>

Security Response Teams

- **NASA Ames Computer Network Security Response Team:**
 - CNSRT (equivalent to CERT) was formed by NASA Ames Research Centre in Aug. 1989.
 - Primary goal is to provide help to Ames users, but also assist other NASA Centers and federal agencies.
 - (415)6940571, <cnsrt@ames.arc.nasa.gov>

Thanks!!!
Queries?



Network Security

SS ZG513

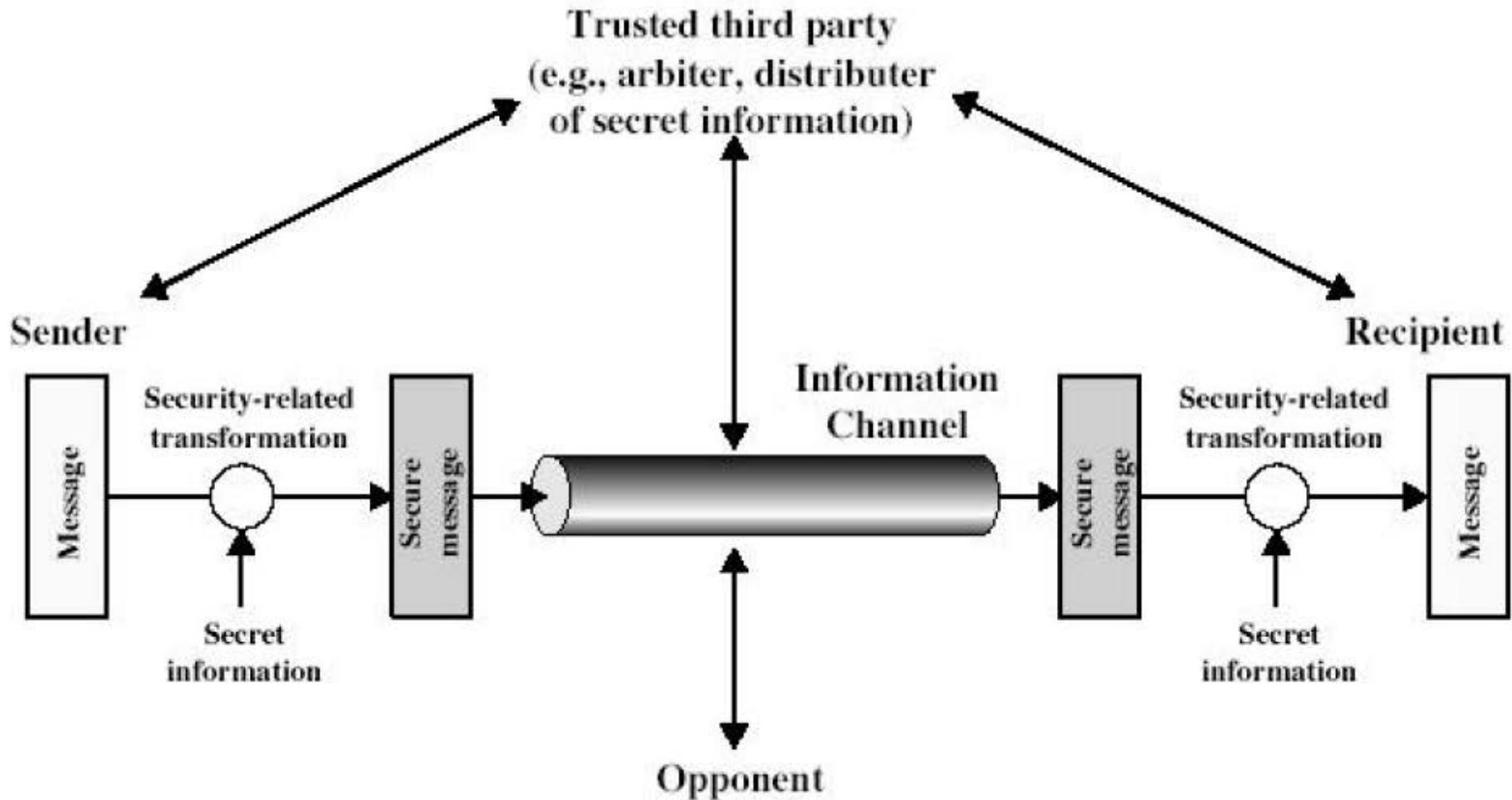
BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Last Class

- Aspects of Security
- Cryptography (Terminology etc.)
- Model for Network Security

Model for Network Security



Security Services

- Enhance security of data processing systems and information transfers of an organization.
- Intended to counter security attacks.
- Using one or more security mechanisms.
- Often replicates functions normally associated with physical documents, for e.g.
 - Have signatures, dates; need protection from disclosure, tampering, or destruction; be notarized or witnessed; be recorded or licensed.

Security Services

- **X.800:**
“a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers”
 - **RFC 2828:**
“a processing or communication service provided by a system to give a specific kind of protection to system resources”
-

Security Services (X.800)

- **Data Confidentiality** - protection of data from unauthorized disclosure.
 - **Authentication** - assurance that the communicating entity is the one claimed.
 - **Integrity** - assurance that data received is as sent by an authorized entity.
 - **Non-Repudiation** - protection against denial by one of the parties in a communication.
 - **Access Control** - prevention of the unauthorized use of a resource.
-

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 Mechanism (X.800)

- **Specific security mechanisms:**
 - Encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization.

- **Pervasive security mechanisms:**
 - Trusted functionality, security labels, event detection, security audit trails, security recovery .

Security Response Teams

- **Computer Emergency Response Team:**

- The goal of CERT/CC was to address computer security concerns of research users of the internet.
- The promotion of CERT/CC team was to prevent and handle incidents such as internet worms.
- Has ability to immediately confer with experts to diagnose and solve security problems.
- Work with vendors of software systems in order to coordinate the fixes for security problems.
- Help you indirectly in formulating an effective network security policy.
- Operates 24-hour hotline that you can call to report security problems. e.g. someone breaking your system.

Security Response Teams

- **Computer Emergency Response Team:**

- To join the CERTAdvisory mailing list, send a message to
<certrequest@cert.sei.cmu.edu>
- Address:
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 152133890
(412)2687090
E-mail:cert@cert.sei.cmu.edu
URL: <http://www.cert.org>

Security Response Teams

- **NIST Computer Security Resource and Response Clearinghouse:**
 - Besides dealing with standards issues, also has responsibility for computer science & tech. activities.
 - Provide help & information regarding computer security events and incidents.
 - Also interested in raising awareness about computer security vulnerabilities.
 - Publications related to computer security and computer viruses.
 - CSRC, A216 Technology, Gaithersburg, MD 20899,
(301)9755200, <csrc@nist.gov>, <http://csrc.nist.gov>

Security Response Teams

- **DOE Computer Incident Advisory Capability:**
 - Primary responsibility is to assist DOE sites faced with computer security incidents such as intruder attacks, virus, worms attack and so on.
 - Keeps informed of current securityrelated events, and maintain contact with other response teams & agencies.
 - Develops guideline for security incident handlings & develops software for responding to security incident.
 - Analyzes security events and trends, and conducts training and awareness activites to alert and advise sites about vulnerabilites and potential attacks.
 - (415)4228193, <ciac@tiger.llnl.gov>, <http://ciac.llnl.gov>

Security Response Teams

- **NASA Ames Computer Network Security Response Team:**
 - CNSRT (equivalent to CERT) was formed by NASA Ames Research Centre in Aug. 1989.
 - Primary goal is to provide help to Ames users, but also assist other NASA Centers and federal agencies.
 - (415)6940571, <cnsrt@ames.arc.nasa.gov>

Modular Arithmetic

Most Cryptosystems are based on finite sets:

- Finite set:
 - Discrete (Sets with integers)
 - Finite (Cardinality of set)

Modular Arithmetic

- Modulus Operation

Let a, r, m be integers where $m > 0$. Then

$$a \equiv r \pmod{m}$$

if $(r-a)$ is divisible by m

“ m ” is called the modulus

“ r ” is called the remainder

Modular Arithmetic

Computation of Remainder

Given $a, m \in \mathbb{Z}$

$$a = q * m + r$$

where q = quotient

r = remainder

m = modulus operation

remainder is not unique

Modular Arithmetic

Equivalence Class:

where all the members of a set has equivalence relation.

$$\begin{aligned}
 A. a &= 10 \ m=5 \\
 5 * 1 + 5 &= 10 \\
 5 * 2 + 0 &= 10 \\
 5 * 3 - 5 &= 10 \\
 5 * -1 + 15 &= 10 \\
 \{.. -5, 0, 5, 10, 15\}
 \end{aligned}$$

Same for $a=15$

$$\begin{aligned}
 5(0) + 15 \\
 5(1) + 10 \\
 5(2) + 5 \\
 5(3) + 0 \\
 5(4) - 5
 \end{aligned}$$

$$\begin{aligned}
 B \ a=11 \ m=5 \\
 5 * 1 + 6 \\
 5 * 0 + 11 \\
 5 * 2 + 1 \\
 5 * 3 - 4 \\
 5 * (-1) + 16 \\
 \{-4, 1, 6, 11, 16\}
 \end{aligned}$$

$$\begin{aligned}
 C \ a=12 \ m=5 \\
 5 * 0 + 12 \\
 5 * 1 + 7 \\
 5 * 2 + 2 \\
 5 * 3 - 3 \\
 5 * (-1) + 17 \\
 \{-3, 2, 7, 12, 17\}
 \end{aligned}$$

$13 * 16 - 8 \text{ mod } 5 = (208 - 8) \text{ mod } 5 = 200 \text{ mod } 5 = 0 \text{ mod } 5 = 0$
 same as using other elements from equivalence class
 13 class D
 16 class B
 8 class D
 so can use smallest number from equi class $(3 * 1) - 3 \text{ mod } 5 = 0$

$$\begin{aligned}
 3^8 \text{ mod } 5 &= 3^4 * 3^4 = 3^2 * 3^2 * 3^2 * 3^2 = 9 * 9 * 9 * 9 = \text{class E} = -1 * -1 * -1 * -1 = 1 \text{ mod } 5 \\
 &= 1
 \end{aligned}$$





Ring in Modular Arithmetic

Definition:

- The Integer Ring, Z_m consist of
 - The set $Z_m = \{ 0, 1, \dots, m-1 \}$
 - Two operations
 - “+” and “*”
for all $a, b, \dots \in Z_m$
 - eg $a + b = c \text{ mod } m$
 - $a * b = d \text{ mod } m$

Integer Ring

- Closure: add and multiply of any two numbers and the result is always in the ring.
- Associative: Addition and Multiplication operations are associative, for all $a, b, \dots \in Z_m$

$$a + (b + c) = (a + b) + c$$

$$a * (b * c) = (a * b) * c$$

- Distributive: for all $a, b, \dots \in Z_m$

$$a * (b + c) = (a * b) + (a * c)$$

Integer Ring

- Neutral Element: 0 with respect to addition, i.e., for all $a \in \mathbb{Z}_m$

$$a + 0 \equiv a \pmod{m}$$

- Neutral Element: 1 with respect to multiplication,
i.e., for all $a \in \mathbb{Z}_m$

$$a * 1 \equiv a \pmod{m}$$

Type text here

Integer Ring

- Additive inverse: For $a \in Z_m$, there is always an additive inverse element $-a$ such that

$$a + (-a) \equiv 0 \pmod{m}$$

- Multiplicative inverse (a^{-1})

$$a * a^{-1} \equiv 1 \pmod{m}$$

exists only for some, **but not for all, elements in Z_m .**

Modular Arithmetic: Applications

Shift/Caesar cipher:

- If $x, y, k \in \mathbb{Z}_{26}$, then

$$y = E_k(x) \equiv (x + k) \bmod 26$$

$$x = D_k(y) \equiv (y - k) \bmod 26$$

- If $k = 10$ and plaintext is CRYPTO = $x_1, x_2, x_3, x_4, x_5, x_6 = 2, 17, 24, 15, 19, 14$ then ciphertext = $y_1, y_2, y_3, y_4, y_5, y_6 = 12, 1, 8, 25, 3, 24$ = MBIZDY
- Only 25 possible keys, hence brute force attack is trivial. Also one can apply letter frequency analysis.
- If arbitrary substitution, then key space is 26!

Modular Arithmetic: Applications

Affine cipher:

- If $x, y, a, b \in \mathbb{Z}_{26}$, then

$$y = E_k(x) \equiv (a \cdot x + b) \bmod 26$$

$$x = D_k(y) \equiv a^{-1} \cdot (y - b) \bmod 26$$

- If $(a, b) = (3, 10)$ and plaintext is CRYPTO = $x_1, x_2, x_3, x_4, x_5, x_6 = 2, 17, 24, 15, 19, 14$ then ciphertext = $y_1, y_2, y_3, y_4, y_5, y_6 = 16, 9, 4, 3, 15, 0$ = QJEDPA
- $12 \times 26 = 312$ possible keys. Larger than caesar cipher but still brute force attack is trivial and letter frequency analysis.
- Correctness.



Historical Ciphers

- **Symmetric ciphers** are also referred as symmetric-key, secret-key and single key. Ancient ciphers was exclusively based on symmetric-key.
 - **Substitution ciphers:**
 - Monalphabetic ciphers
 - Homophonic ciphers
 - Polyalphabetic ciphers
 - Polygram ciphers
 - Running key ciphers
 - **Letter frequency attack**
-

Historical Ciphers

- Transposition ciphers:
 - Simplest: write horizontally and read vertically.
 - key: 2 3 1 7 5 6 4
 - Letters remain same, order changes. While in substitution letter changes, order remain same.
- Combined cipher:
 - Two substitution/transposition cipher in sequence.
 - Substitution and transposition are orthogonal. Hence can be combined to produce a new harder cipher.

Breaking an Algorithm

- Total Break
- Global Deduction
- Instance (local) deduction
- Information Deduction.

Security of Cipher

- Unconditional secure
 - Computationally secure
 - Degree of security: how hard to break.
 - Peer-review.
 - Decoding by reverse engg.
Type text here
 - Data Complexity: Breaking cost \gg Encrypted data cost.
 - Time Complexity: Time require to break \gg Time the data is useful.
 - Storage requirement: Amount of data required to break \gg Amount of available 'x', 'y'.
 - An algorithm is said to have a **security level of n bit** if the best known attack requires 2^n steps.
-

Cryptanalysis Attacks

- Ciphertext only attack
 - Known plaintext attack
 - Chosen plaintext attack
 - Adaptive chosen plaintext attack
 - Chosen ciphertext attack
 - Chosen key attack
 - Rubber hose cryptanalysis
-

Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Previous Lecture

- Modular Arithmetic, Integer Ring, Group.
- Substitution and Transposition Cipher.

Modular Arithmetic: Applications

Shift/Caesar cipher:

- If $x, y, k \in \mathbb{Z}_{26}$, then

$$y = E_k(x) \equiv (x + k) \bmod 26$$

$$x = D_k(y) \equiv (y - k) \bmod 26$$

- If $k = 10$ and plaintext is CRYPTO = $x_1, x_2, x_3, x_4, x_5, x_6 = 2, 17, 24, 15, 19, 14$ then ciphertext = $y_1, y_2, y_3, y_4, y_5, y_6 = 12, 1, 8, 25, 3, 24$ = MBIZDY
- Only 25 possible keys, hence brute force attack is trivial. Also one can apply letter frequency analysis.
- If arbitrary substitution, then key space is 26!

Modular Arithmetic: Applications

Affine cipher:

- If $x, y, a, b \in \mathbb{Z}_{26}$, then

$$y = E_k(x) \equiv (a \cdot x + b) \bmod 26$$

$$x = D_k(y) \equiv a^{-1} \cdot (y - b) \bmod 26$$

- If $(a, b) = (3, 10)$ and plaintext is CRYPTO = $x_1, x_2, x_3, x_4, x_5, x_6 = 2, 17, 24, 15, 19, 14$ then ciphertext = $y_1, y_2, y_3, y_4, y_5, y_6 = 16, 9, 4, 3, 15, 0$ = QJEDPA
- $12 \times 26 = 312$ possible keys. Larger than caesar cipher but still brute force attack is trivial and letter frequency analysis.
- Correctness.

Historical Ciphers

- **Symmetric ciphers** are also referred as symmetric-key, secret-key and single key. Ancient ciphers was exclusively based on symmetric-key.
 - **Substitution ciphers:**
 - Monalphabetic ciphers
 - Homophonic ciphers
 - Polyalphabetic ciphers
 - Polygram ciphers
 - Running key ciphers
 - **Letter frequency attack**
-

Historical Ciphers

- Transposition ciphers:
 - Simplest: write horizontally and read vertically.
 - key: 2 3 1 7 5 6 4
 - Letters remain same, order changes. While in substitution letter changes, order remain same.
- Combined cipher:
 - Two substitution/transposition cipher in sequence.
 - Substitution and transposition are orthogonal. Hence can be combined to produce a new harder cipher.

Breaking an Algorithm

- **Total break:** the attacker deduces the secret key.
 - **Global deduction:** the attacker discovers a functionally equivalent algorithm for encryption and decryption, but without learning the key.
 - **Instance (local) deduction:** the attacker discovers additional plaintexts (or ciphertexts) not previously known.
 - **Information deduction:** the attacker gains some Shannon information about plaintexts (or ciphertexts) not previously known.
-

Security of Cipher

- Unconditional secure
 - Computationally secure
 - Degree of security: how hard to break.
 - Peer-review.
 - Decoding by reverse engg.
 - Data Complexity: Breaking cost \gg Encrypted data cost.
 - Time Complexity: Time require to break \gg Time the data is useful.
 - Storage requirement: Amount of data required to break \gg Amount of available 'x', 'y'.
 - An algorithm is said to have a **security level of n bit** if the best known attack requires 2^n steps.
-

Today's Agenda

- Stream Cipher
- Random Number
- TRNG, PRNG, CSPRNG
- Slides:
 - Cryptography and Network Security by William Stallings.

Stream Cipher vs Block Cipher

Stream Cipher vs Block Cipher

Stream Ciphers

- Encrypt bits individually
- Usually small and fast common in embedded devices (e.g., A5/1 for GSM phones)

Block Ciphers:

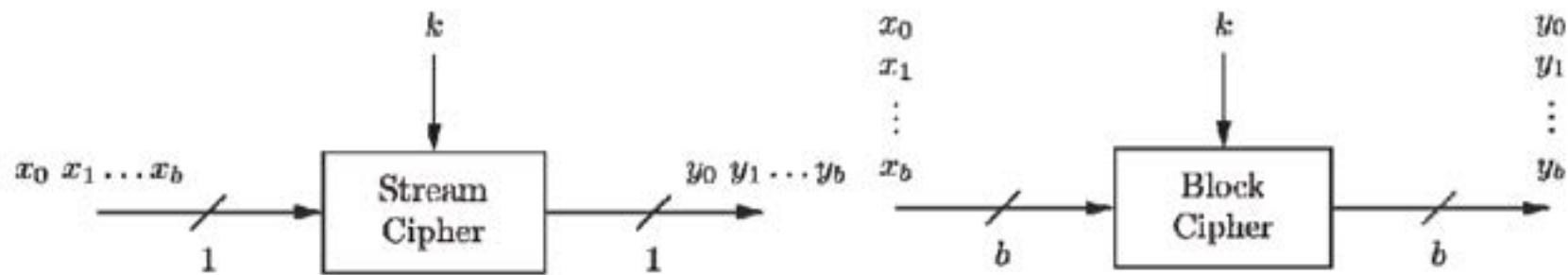
- Always encrypt a full block (several bits)
- Are common for Internet applications

Stream Cipher vs Block Cipher

Stream Cipher vs Block Cipher

Stream Cipher vs Block Cipher

Stream Cipher vs Block Cipher



Random Numbers

- Random Numbers in cryptography
 - nonces in authentication protocols to prevent replay
 - session keys
 - public key generation
 - keystream for a one-time pad
 - Properties of Random Number
 - statistically random
 - uniform distribution
 - independent
 - unpredictability of future values from previous values
-

True Random Number Generator (TRNG)



- Generate “TRUE RANDOM NUMBER”
- True Random Number stem from random physical process
- eg: coin flipping, mouse movement, time b/w clicks
- +ve:
 - True Random Number
- -ve
 - Cannot recreate them

Pseudo Random Number Generators (PRNG)



- Use deterministic algorithmic techniques to create “random numbers”
 - can be recreated
 - although are not truly random
 - can pass many tests of “randomness”
 - known as “pseudorandom numbers”
 - created by “Pseudorandom Number Generators (PRNGs)”
-

Pseudorandom Number Generators



- PNG computation:

$$s_0 = \text{seed}$$

$$s_{i+1} = f(s_i, s_{i-1}, \dots, s_{i-t})$$

- rand() function in ANSI C

$$s_0 = 12345$$

$$s_{i+1} = 1103515245s_i + 12345 \bmod 2^{31}$$

- Most PRNGs have bad cryptographic properties

Cryptographically Secure PRNG

- Given n consecutive bits of output S_n , the following output bits s_{n+1} cannot be predicted



Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

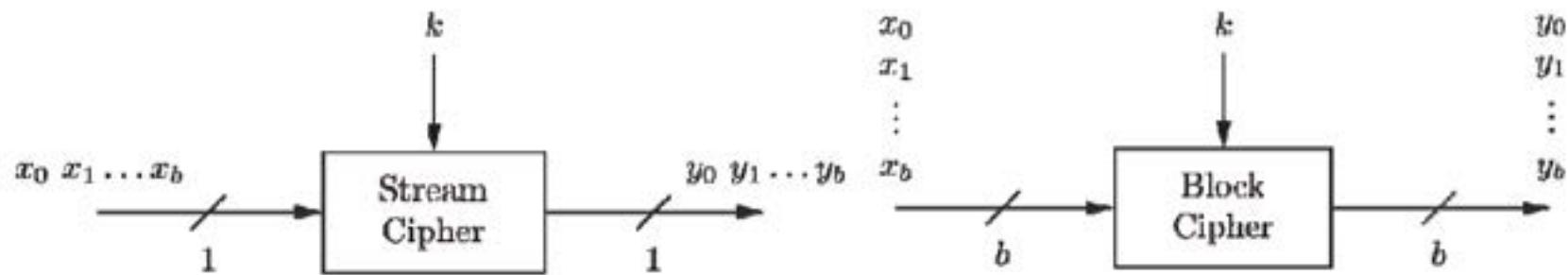
Lecture Session – 7
28 - 01 - 2017

Today's Agenda

Type text here

- LFSR
- RC4
- Extended Euclidean Algorithm

Stream Cipher vs Block Cipher



One Time Pad

- A cryptosystem is unconditionally secure if it cannot be broken even with infinite computational resources.
- Eg: Let plaintext, ciphertext and key

$$x_i, y_i, k_i \in \{0,1\}$$

$$\text{Encryption: } e_k(x) = x \oplus k$$

$$\text{Decryption: } d_k(y) = y \oplus k$$

- OTP is unconditionally secure if key k_i is used once
-

One Time Pad

- Unconditionally secure cryptosystem:

$$y_0 = x_0 \oplus k_0$$

$$y_1 = x_1 \oplus k_1$$

- Every equation is a linear equation with two unknowns
 - For every y_i are $x_i = 0$ and $x_i = 1$ equi-probable
 - Since k_0, k_1, \dots are independent, thus k_i will be Truly Random number
 - It can proved that this systems can not be solved.

Linear Congruential Generator (LCG)

Linear Congruential Generator (LCG)

RC4

- A proprietary cipher owned by RSA.
 - Designed by Ron Rivest.
 - Simple but effective.
 - Variable key size, byte-oriented stream cipher.
 - Widely used (Web SSL/TLS, Wireless WEP).
 - Key formed by the random permutations of all 8-bit values.
 - Permutation is used to scramble input info.
 - One byte is processed at a time.
-

RC4

- Steps:
 - Initialize S to values 0 to 255
 - Initialize T with repeating values of key, K
 - Use T for initial permutation of S
 - Permutated S and generate key stream, k from S
 - Encrypt a byte of plaintext, p by XOR with k

RC4

- Starts with an array S of numbers: 0....255.
- Uses key to well and truly shuffle.
- S forms the internal state of the cipher.
- Encryption continues shuffling array values.
- Sum of shuffled pair selects “stream key” value from permutation

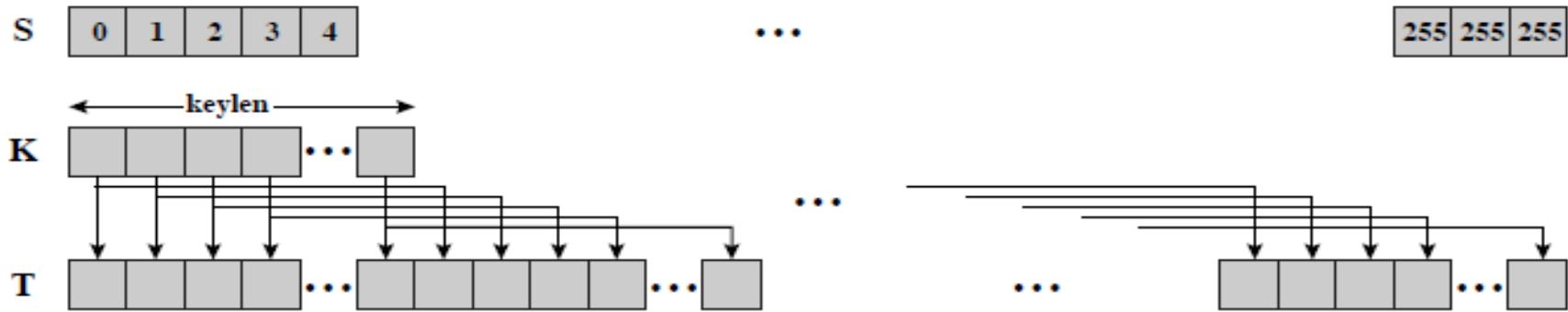
```
/* Initialization */
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
```

RC4

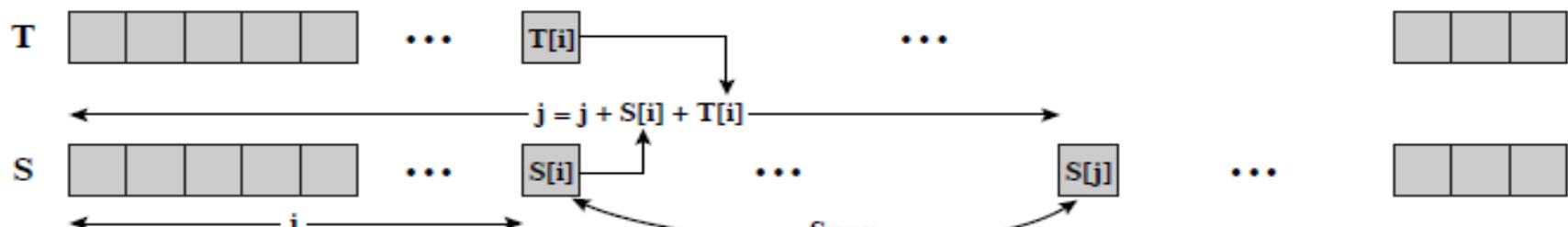
```
/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256;
    Swap (S[i], S[j]);

/* Stream Generation */
i, j = 0;
while (true)
    i = (i + 1) mod 256;
    j = (j + S[i]) mod 256;
    Swap (S[i], S[j]);
    t = (S[i] + S[j]) mod 256;
    k = S[t];
```

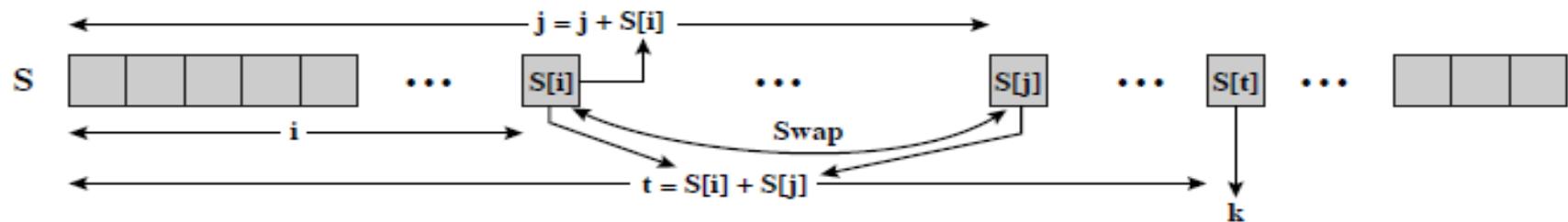
RC4



(a) Initial state of S and T



(b) Initial permutation of S



RC4 Security

- Claimed secure against known attacks.
- Result is very non-linear.
- Key shall not be reused as RC4 is a stream cipher.
- Have a concern with WEP, but due to key handling rather than RC4 itself.

Thanks!!!
Queries?



Network Security

SS ZG513

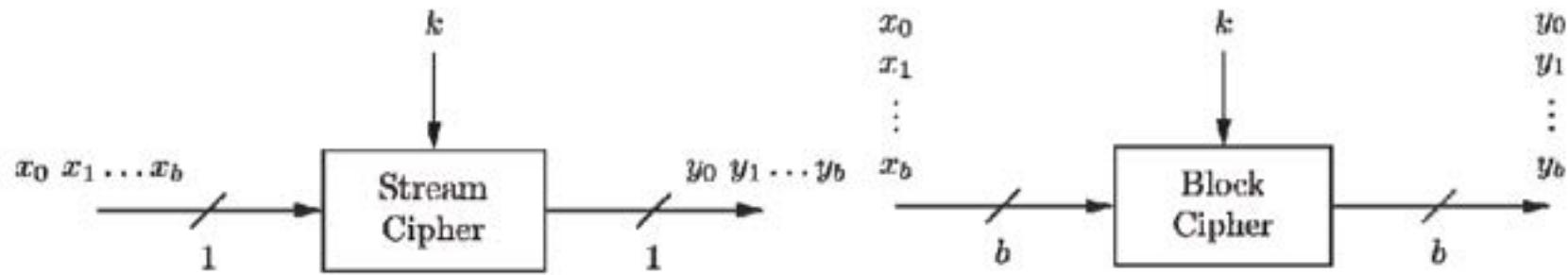
BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Today's Agenda

- RC4
- AES

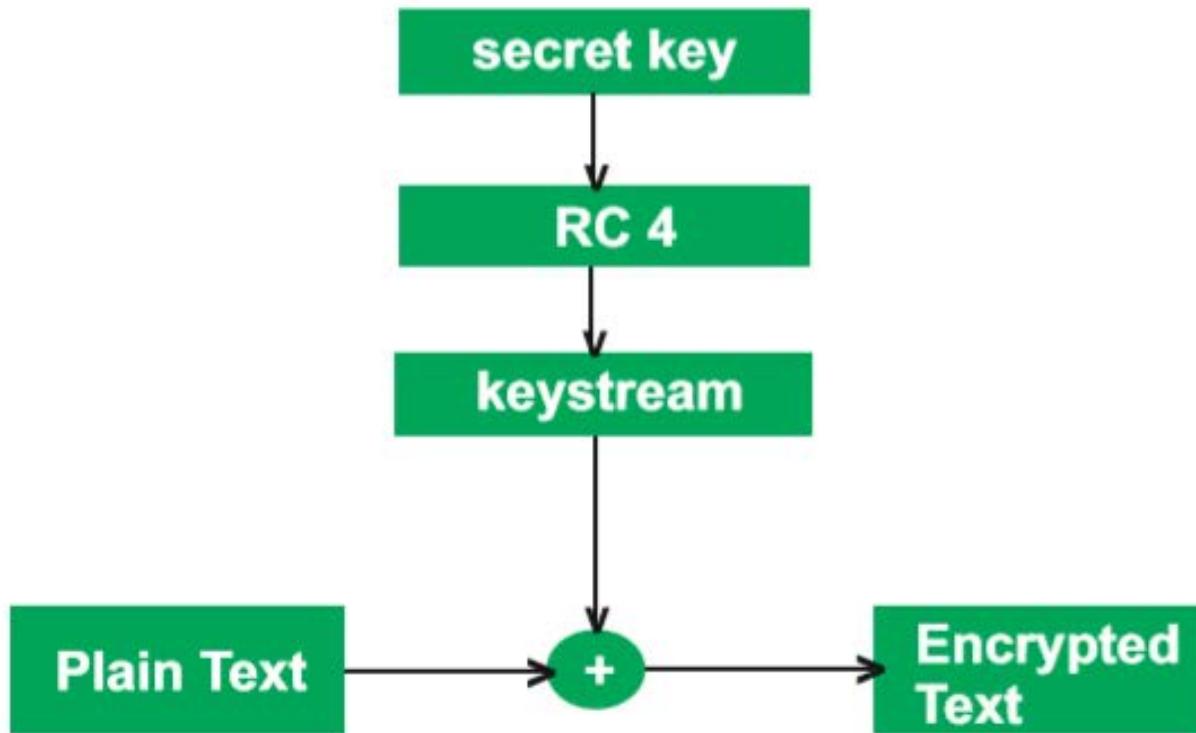
Stream Cipher vs Block Cipher



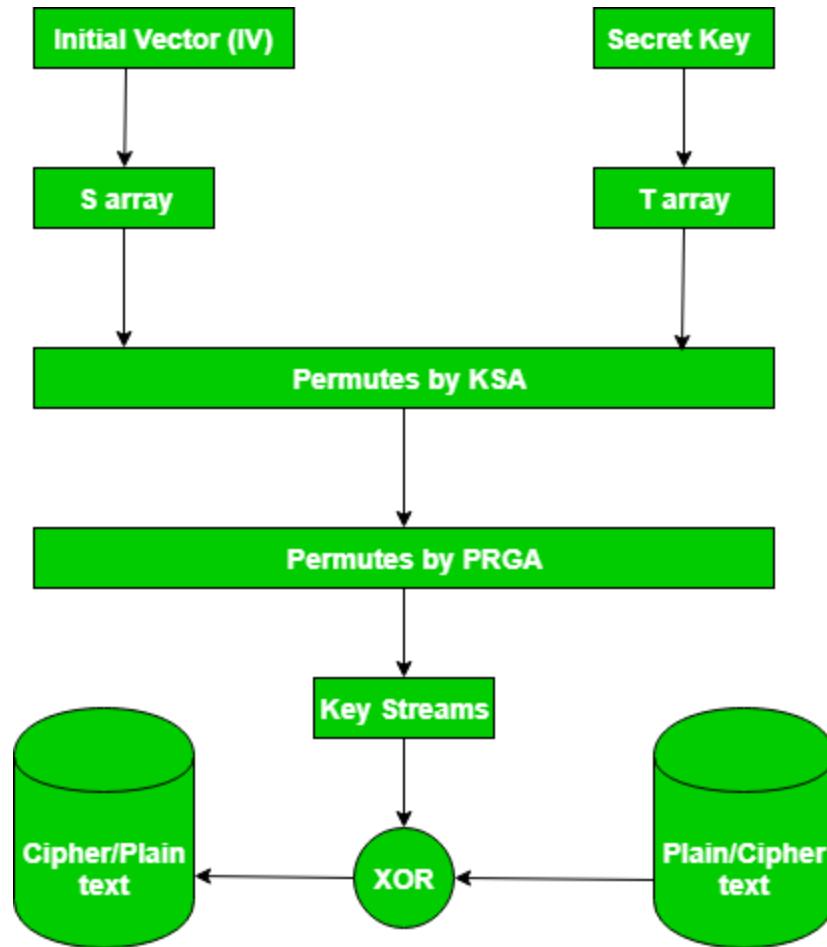
RC4 (Rivest Cipher 4)

- A proprietary cipher owned by RSA.
 - Designed by Ron Rivest.
 - Simple but effective.
 - Variable key size, byte-oriented stream cipher.
 - Widely used (Web SSL/TLS, Wireless WEP).
 - Key formed by the random permutations of all 8-bit values.
 - Permutation is used to scramble input info.
 - One byte is processed at a time.
-

RC 4 BLOCK DIAGRAM



RC4



RC4

- Starts with an array S of numbers: 0....255.
- Uses key to well and truly shuffle.
- S forms the internal state of the cipher.
- Encryption continues shuffling array values.
- Sum of shuffled pair selects “stream key” value from permutation

```
/* Initialization */
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
```

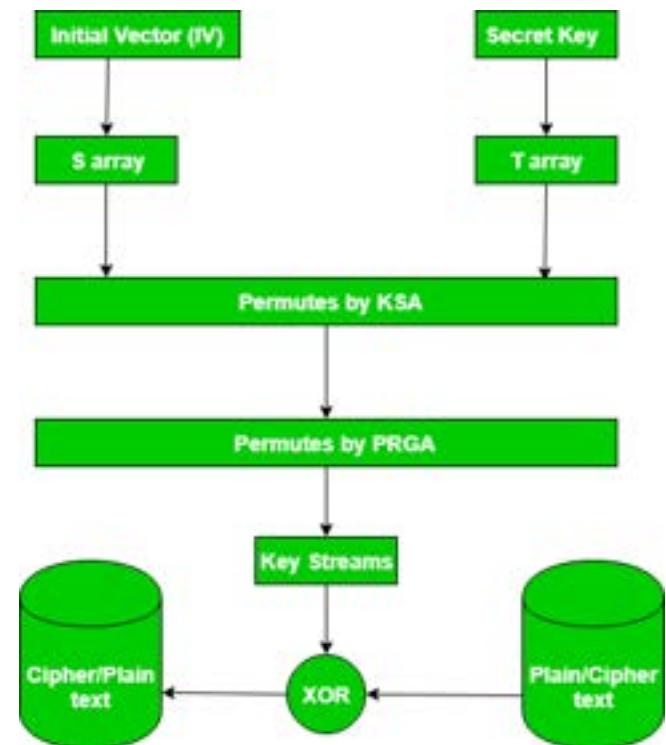
RC4

```

/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256;
    Swap (S[i], S[j]);
}

/* Stream Generation */
i, j = 0;
while (true)
    i = (i + 1) mod 256;
    j = (j + S[i]) mod 256;
    Swap (S[i], S[j]);
    t = (S[i] + S[j]) mod 256;
    k = S[t];

```

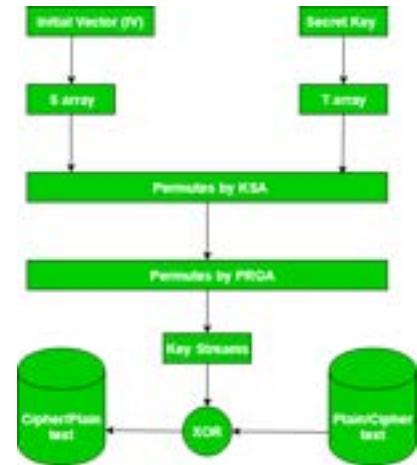


Simplified RC4 (Initialization)

- Suppose the length of S-array is 8

```
/* Initialization */
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
```

- [$S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7]$]
- Initialize S-array = [0, 1, 2, 3, 4, 5, 6, 7]
- Let the Key [3, 1, 4, 1, 5]
- Initialize T-array = [3, 1, 4, 1, 5, 3, 1, 4]

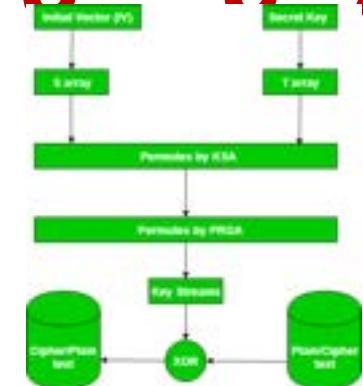


Simplified RC4

(Permute by Key Scheduling Algo)



```
/* Initial Permutation of S */  
j = 0;  
for i = 0 to 255 do  
    j = (j + S[i] + T[i]) mod 256;  
    Swap (S[i], S[j]);
```



		T ₀ =3	T ₁ =1	T ₂ = 4	T ₃ =1	T ₄ =5	T ₅ =3	T ₆ =1	T ₇ =4
i	j	S ₀	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇
	0	0	1	2	3	4	5	6	7
0	3	3	1	2	0	4	5	6	7
1	5	3	5	2	0	4	1	6	7
2	3	3	5	0	2	4	1	6	7
3	6	3	5	0	6	4	1	2	7
4	7	3	5	0	6	7	1	2	4
5	3	3	5	0	1	7	6	2	4
6	6	3	5	0	1	7	6	2	4
7	6	3	5	0	1	7	6	4	2

Simplified RC4

- The final S-array will be

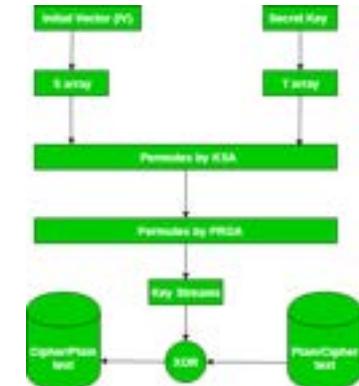
[3, 5, 0, 1, 7, 6, 4, 2]

- Let the PT be [6, 1, 5, 4]

Simplified RC4 (Generate Key Stream)



```
/* Stream Generation */
i, j = 0;
while (true)
    i = (i + 1) mod 256;
    j = (j + S[i]) mod 256;
    Swap (S[i], S[j]);
    t = (S[i] + S[j]) mod 256;
    k = S[t];
```

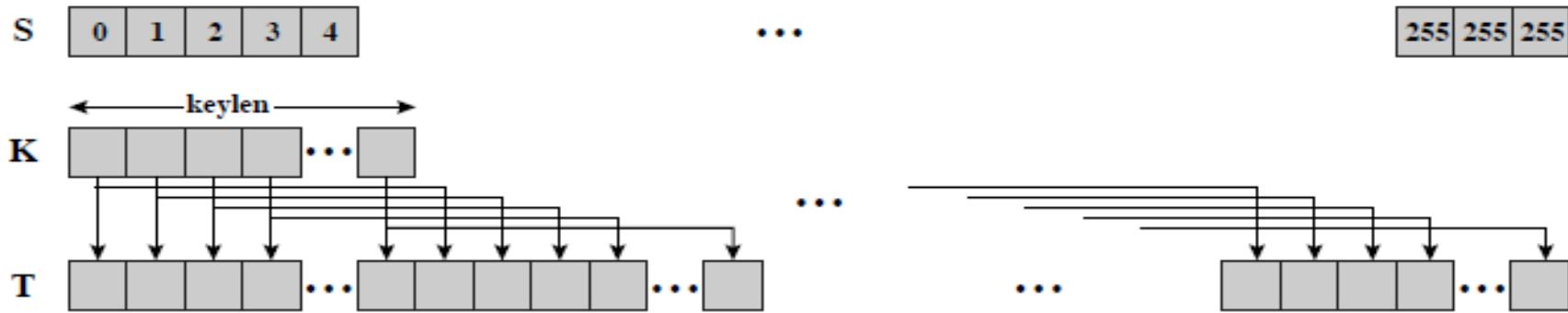


i	j	t	k	S ₀	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇
0	0			3	5	0	1	7	6	4	2
1	5	3	1	3	6	0	1	7	5	4	2
2	5	5	0	3	6	5	1	7	0	4	2
3	6	5	0	3	6	5	4	7	0	1	2
4	5	7	2	3	6	5	4	0	7	1	2

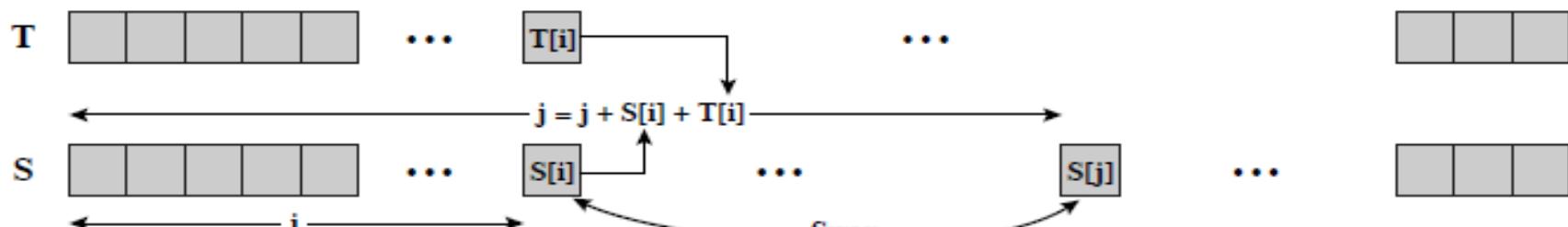
Simplified RC4 (Encrypt)

- PT = [6, 1, 5, 4]
- Key = [1, 0, 0, 2]
- CT = PT XOR Key
- PT = 0110 0001 0101 0100
- Key = 0001 0000 0000 0010
- CT = 0111 0001 0101 0110
- CT = 7 1 5 6

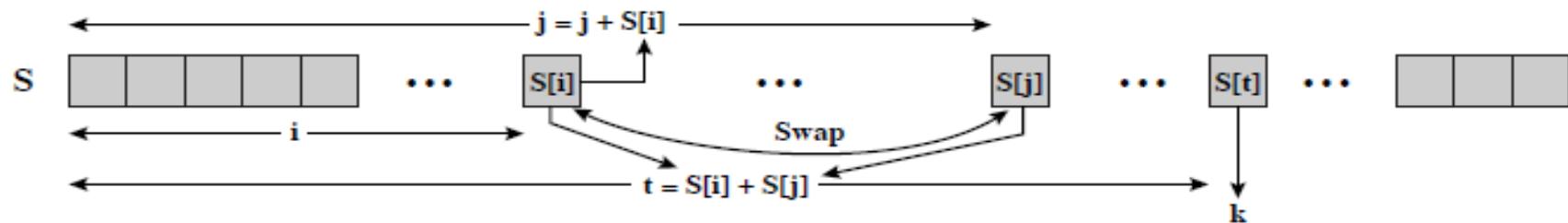
RC4



(a) Initial state of S and T



(b) Initial permutation of S



RC4 Security

- Claimed secure against known attacks.
- Result is very non-linear.
- Key shall not be reused as RC4 is a stream cipher.
- Have a concern with WEP, but due to key handling rather than RC4 itself.

Thanks!!!
Queries?



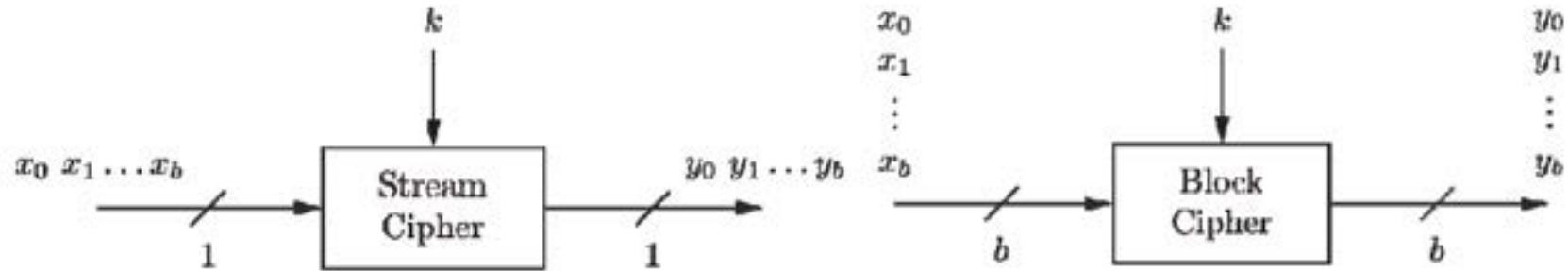
Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

AES History



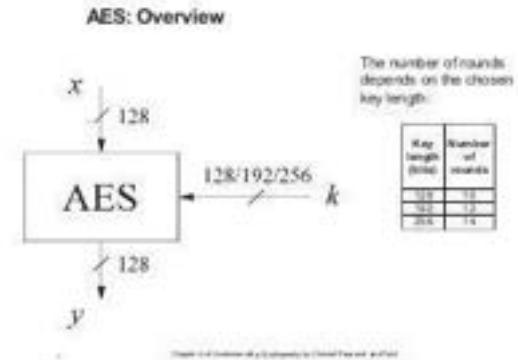
- Call for encryption algorithm by NBS in 1973
- DES (Data Encryption Standard) – 1976 published as standards
 - Block Size : 64 bits
 - Key Size : 54 bits
- Replacement for DES was needed by 1995
 - theoretical attacks
 - exhaustive key search attacks
- Can use Triple-DES – but slow, has small blocks

Introduction to AES

- US NIST issued call for ciphers in 1997
 - 22 submissions
 - 7 did not satisfy all requirements
 - 15 submissions (August 1998)
 - 5 finalists (August 1999)
 - Mars - IBM Corporation
 - RC6 - RSA Laboratories
 - Rijndael - J. Daemen & V. Rijmen
 - Serpent - Eli Biham et al.
 - Twofish - B. Schneier et al..
 - Winner (October 2000): Rijndael.
 - Published FIPS PUB 197 standard by NIST in 2001.
-

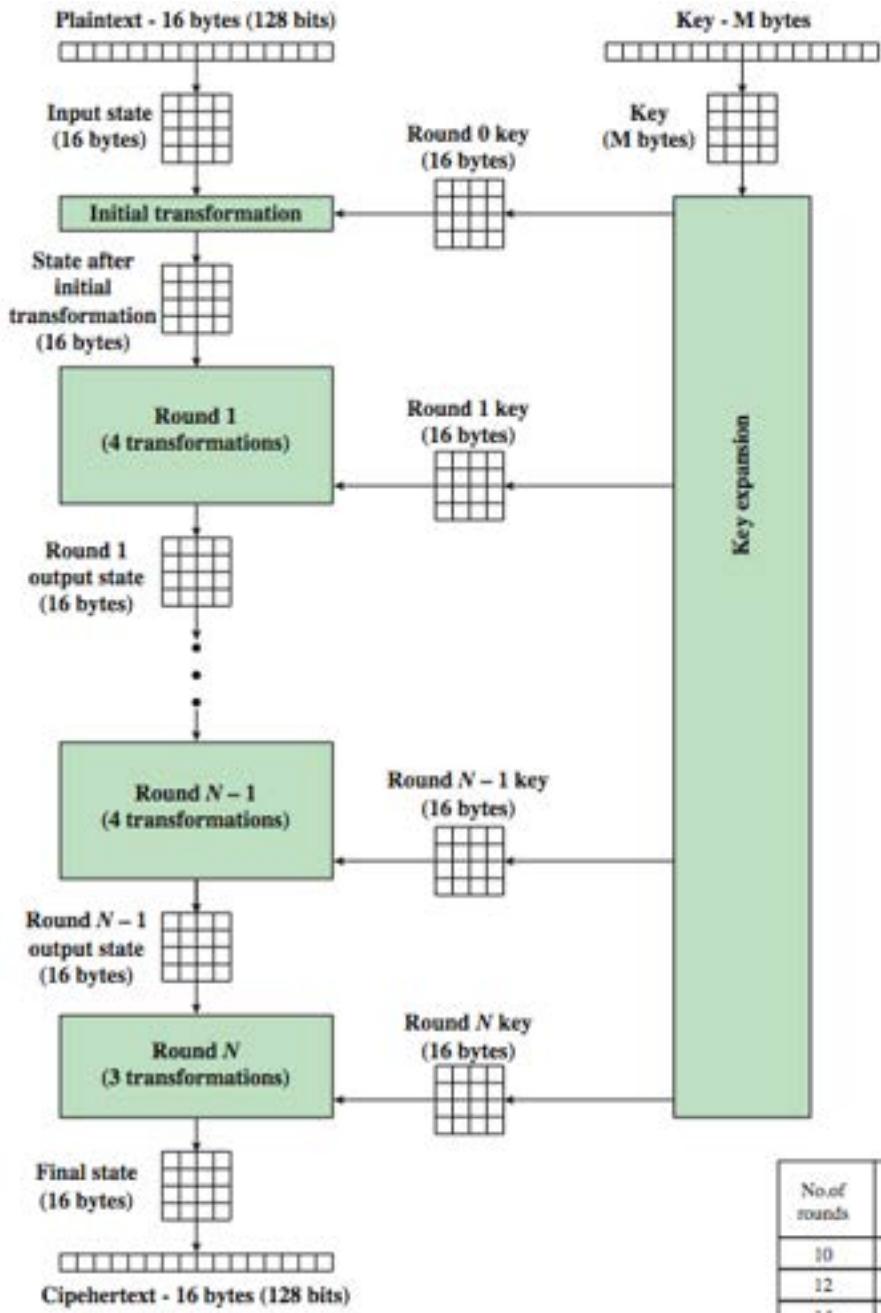
Introduction to AES

- AES is based on a competition, won by Rijmen and Daemen (Rijndael) from Belgium.
- AES
 - Block Size: 128 bits
 - Key Size: 128, 192, 256 bits
(AES-128, AES-192, AES-256)
 - Larger Key size
More secure (More Rounds)
Slower
- An iterative rather than Feistel cipher.
 - Operates on entire data block in every round.



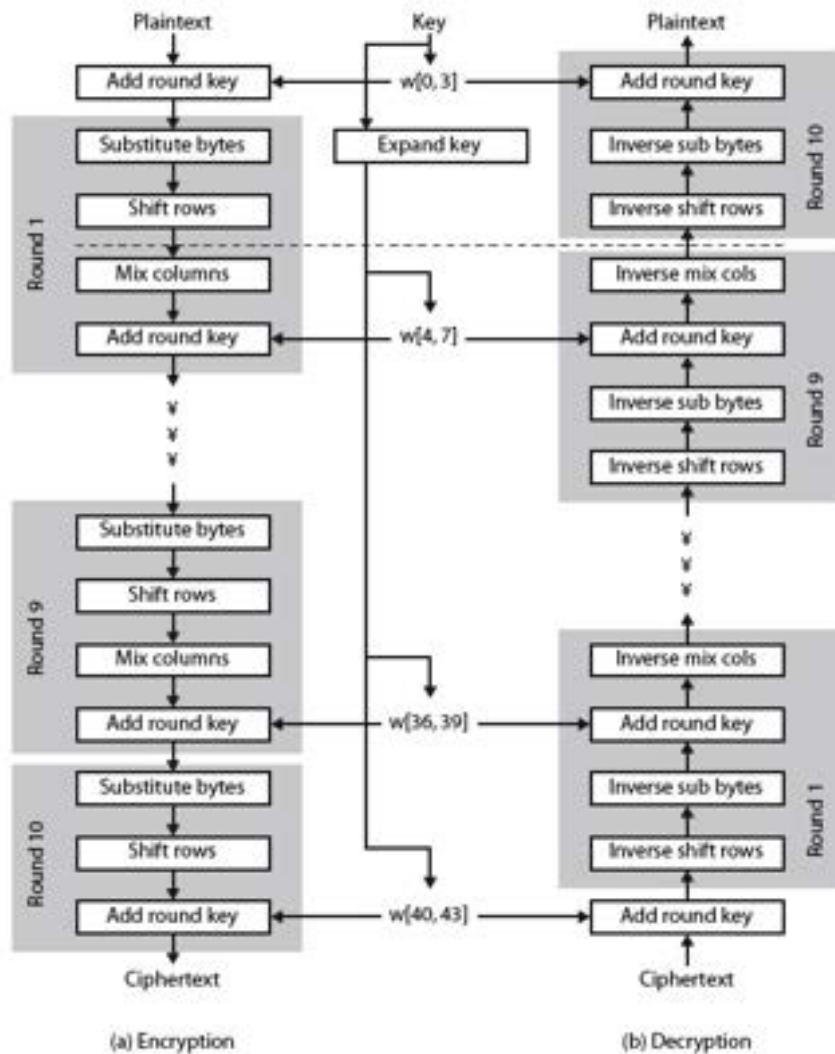


AES



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

AES Structure



(a) Encryption

(b) Decryption

Overview of AES

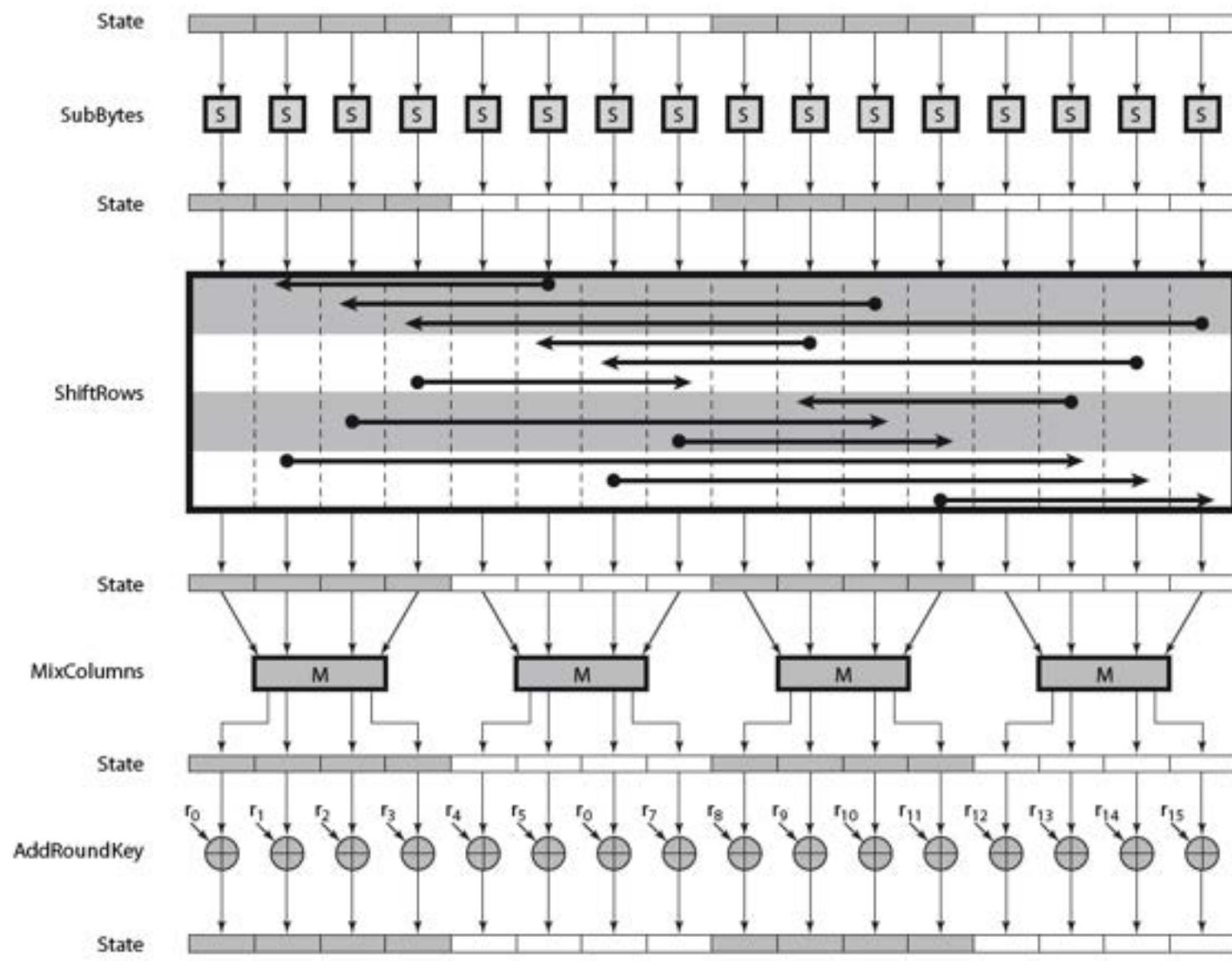
AES consists of:

- Confusion Layer
 - Confusion: each bit of the cipher text should depend on several parts of the key.
 - Byte substitution layer (S-box):
Introduces confusion to the data by transforming the data non-linearly and assures that changes in individual state bits propagate quickly across the data path.
 - Diffusion layer
 - Diffusion: if we change a single bit in the plaintext, then (statistically) half of the bits in the cipher text should change.
 - Provides diffusion over all the state, consists two sublayers ShiftRows and MixColumn layer.
-

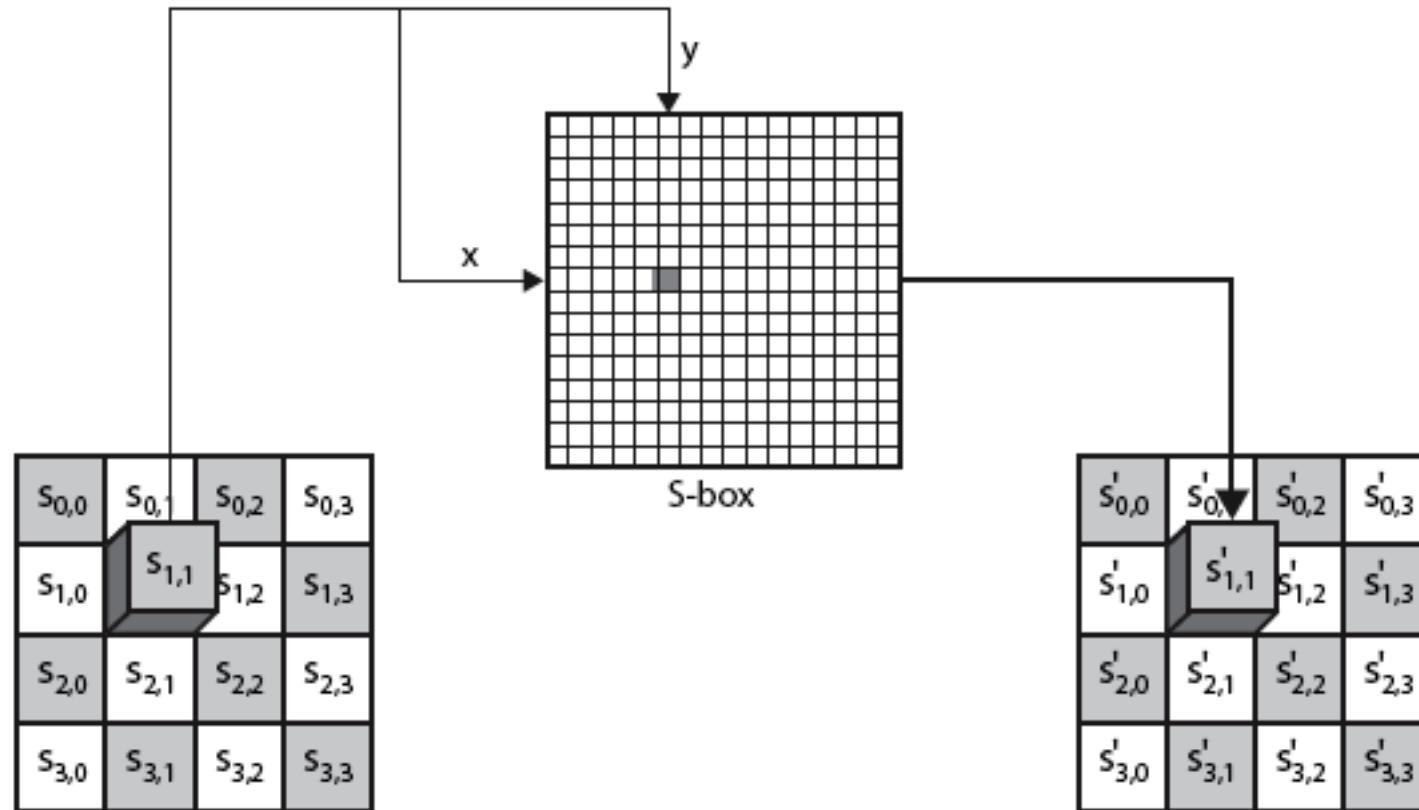
Internal Structure of AES

- The 128-bit data path consists of 16 bytes is arranged in a 4-by-4 byte matrix
- The key bytes are arranged into a matrix with 4 rows and 4 (128-bit key), 6 (192-bit key), 8 (256-bit key) columns.





SubBytes



SubBytes

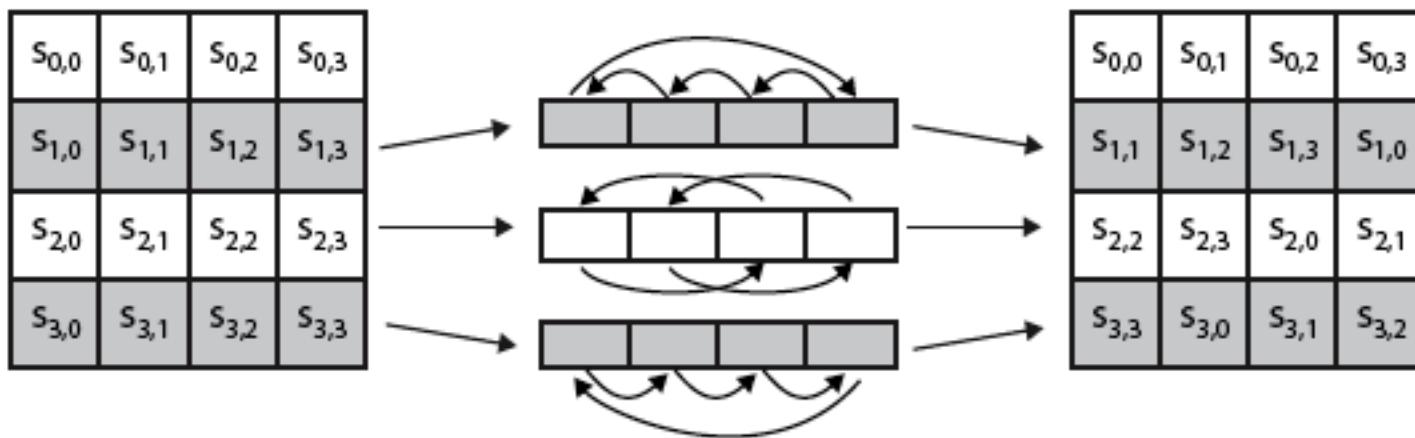
- Simple substitution of each byte
- uses one table of 16x16 bytes containing a permutation of all 256 8-bit values
- Each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits)
- S-box constructed using defined transformation of values in $\text{GF}(2^8)$
- Designed to be resistant to all known attacks

SubBytes

	y																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76	
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0	
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15	
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75	
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84	
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF	
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8	
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2	
X	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	ID	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES S-Box

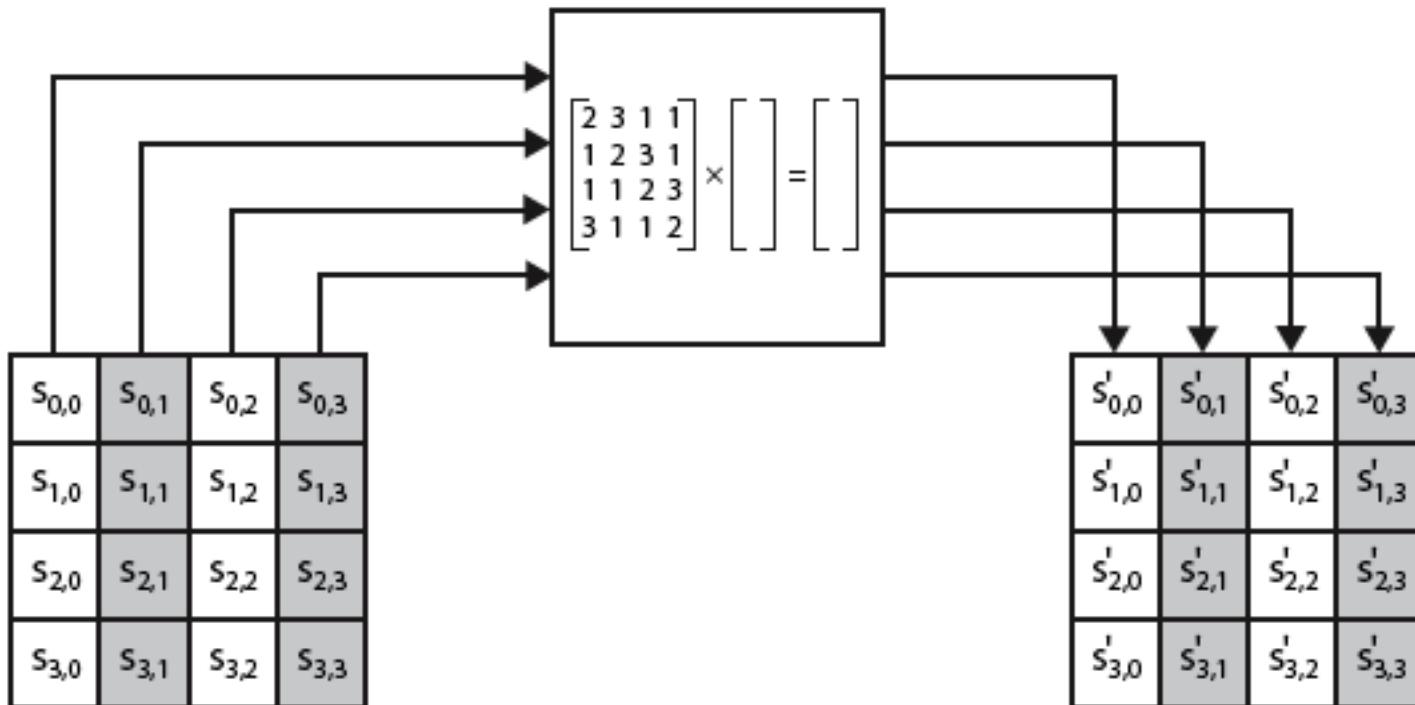
Shift Rows



Shift Rows

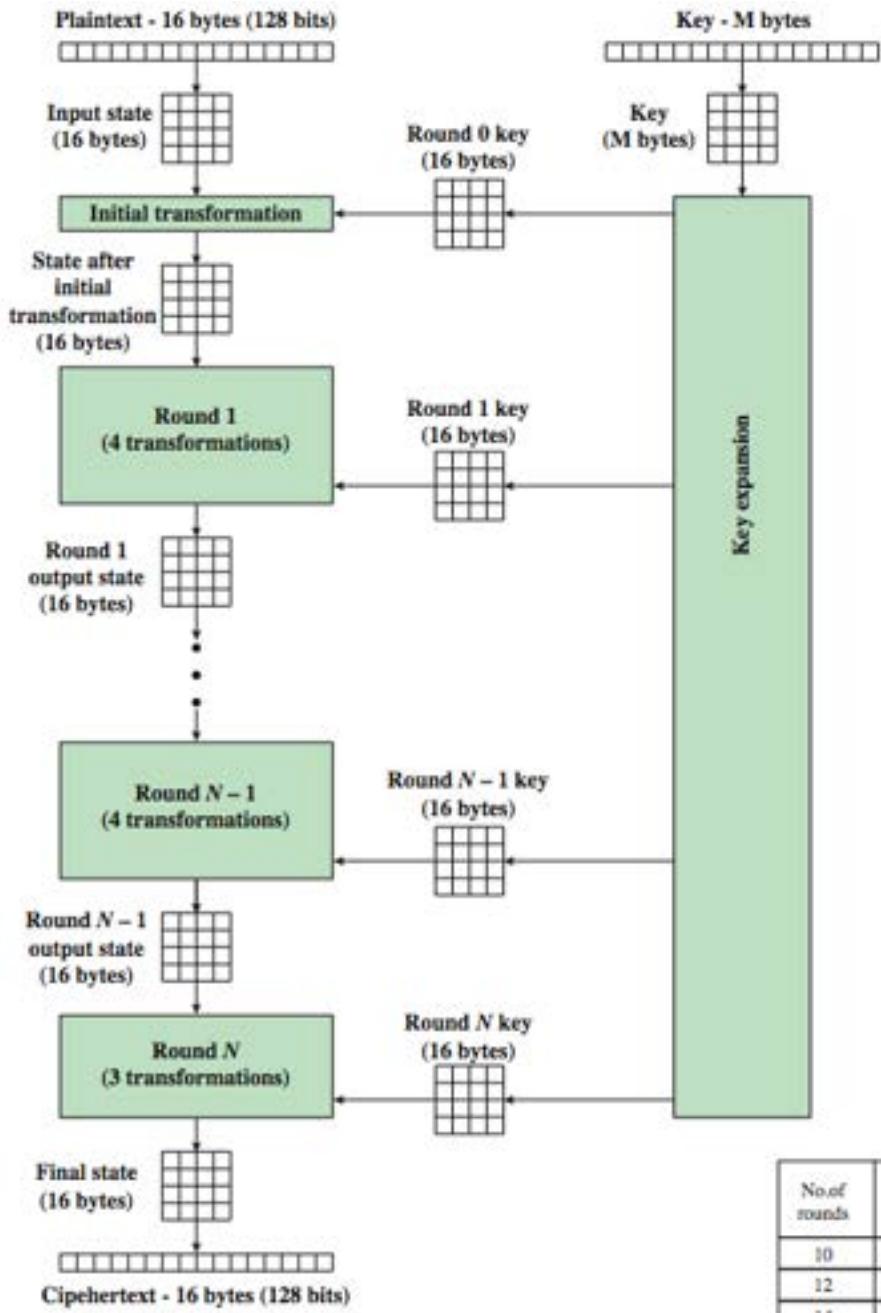
- a circular byte shift in each
 - 1st row is unchanged
 - 2nd row does 1 byte circular shift to left
 - 3rd row does 2 byte circular shift to left
 - 4th row does 3 byte circular shift to left
- decrypt inverts using shifts to right
- since state is processed by columns, this step permutes bytes between the columns

Mix Columns





AES



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Implementation and Security

- Efficient in software and hardware.
 - AES is part of numerous open standard such as Ipsec/TLS.
 - Will dominant encryption algorithm for many years.
 - Not based on Feistel networks. Its basic operations use Galois field arithmetic and provide strong diffusion and confusion.
 - Provides excellent long-term security against brute-force attack (128/ 192/ 256 bits key).
 - Studied intensively in 1990s and no attacks have been found that are better than brute-force.
 - Pre-computations adds a small latency to the decryption operation relative to encryption.
-

Thanks!!!
Queries?



SS ZG513

Network Security

Introduction – Information Security

Objectives

BITS Pilani Revision 1.0
Work Integrated Learning Programmes

Information Security Objectives



Continuing...

- **Example-1:** Alice transmits a file to Bob. The file contains sensitive information (e.g. design secrets) that is to be protected from disclosure. Eve, who is not authorized to read the file, is able to capture a copy of the file during its transmission.
- **Example-2:** Eve works in a bank and maliciously changes the details of a customer in the IT system.
- **Example-3:** One Monday morning, when customers log in to their bank accounts maintained with the XYZ Bank Limited, the login was denied. Downtime was not communicated earlier.

What concerns are we talking about?

The three security objectives for the information systems often referred to as the **CIA triad**.

Confidentiality

Integrity

Availability



Information Security Objectives

Concluded.

Confidentiality: Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.

- Hiding trade secrets from competitors.
- Bank account details of customers.
- Country's military combat plans.

Integrity: Guarding against improper information modification or destruction. A loss of integrity is the unauthorized modification or destruction of information.

- Tampering with the opinion poll data.
- Data associated with the unique ids (e.g. PAN, Aadhar, Driving License) are maliciously changed.

Availability: Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.

- Railway reservation cannot be done from Railway's website during few specific hours.
- Online banking not available for a particular bank. Customers are in lurch.

Points to Ponder !



-
1. Can there be more objectives for the information security beyond Confidentiality, Integrity and Availability?

How about:

- **Authenticity** – Information being able to be verified. Confidence in information source and transmission.
- **Accountability** - Ability to trace a security breach to a responsible party.

CIA triad is more established, but the necessity for the above two is also important and practiced by information security professionals.

3. Information security appears to be a vast area, is there any systematic approach to meet the objectives of it?

Major Standardization Bodies

In Information Security

IETF: The Internet Engineering Task Force ([IETF](#)) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.



ITU-T: One of the three sectors of the International Telecommunication Union ([ITU](#)); it coordinates standards for telecommunications.



NIST: National Institute of Standards and Technology ([NIST](#)) is the US federal technology agency that works with industry to develop and apply technology, measurements, and standards.



ISO: The International Organization for Standardization (ISO) is an international standard-setting body composed of representatives from various national standards organizations. It works in several areas including networking and security.



Information Security



Definition

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).

Reference: NIST Computer Security Handbook [NIST95]

The OSI Security Architecture

ITU-T Recommendation [X.800](#), Security Architecture for Open System Interconnection (OSI), defines a systematic approach.

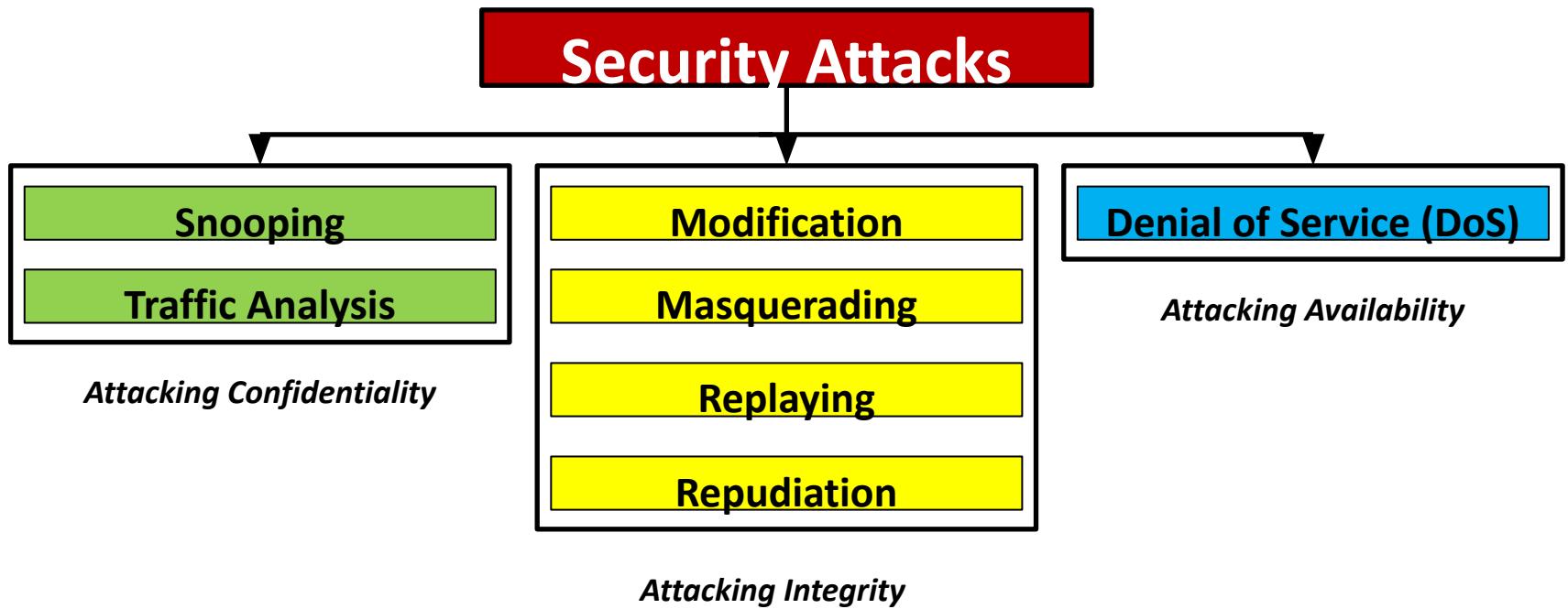
The OSI security architecture focuses on security attacks, mechanisms, and services.

These can be defined as:

- **Security Attack:** Any action that compromises the security of information owned by an organization.
 - **Security Mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.
 - **Security Service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.
-

Security Attacks

Types as described in X.800



Security Attacks

Explanation and Examples

Attacking Confidentiality	Snooping	Data is intercepted by an unauthorized person. E.g. Tapping
	Traffic Analysis	May be the data is masked, so no information can be extracted but some patterns like - sender, receiver, message length, time of the message etc. can be extracted to make intelligent guesses.

Attacking Integrity	Modification	Some portion of a legitimate message is altered or the message is delayed.
	Masquerading	One entity pretends to be a different entity. E.g. Hoax bank sites.
	Replaying	Subsequent retransmission of a captured message to produce an unauthorized effect. E.g. Bill payment fake reminders.
	Repudiation	Sender denies that it sent the message or the receiver denies that it received the message.

Attacking Availability	Denial of Service (DoS)	Slowing down or totally interrupt the service of the system. E.g. multiple requests to bring an exam result server down.
-------------------------------	-------------------------	--

Security Attacks



Another View - Active versus Passive

Passive Attacks – The attacker's goal is to just obtain the information. The attack does not harm the system.

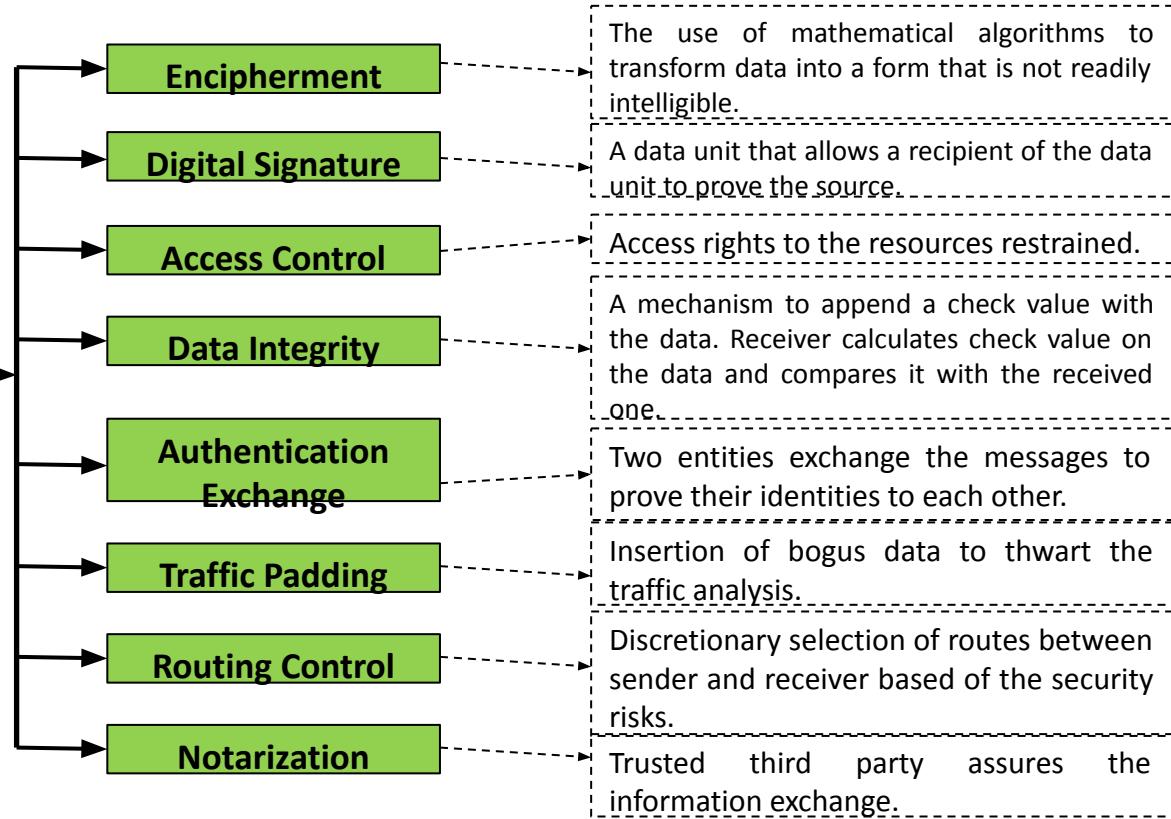
Active Attacks – The attacker changes the data or harms the system.

Attacks	Attacking	Type
Snooping, Traffic Analysis	Confidentiality	PASSIVE
Modification, Masquerading, Replaying, Repudiation	Integrity	ACTIVE
Denial of Service (DoS)	Availability	ACTIVE

Security Mechanisms

As described in X.800

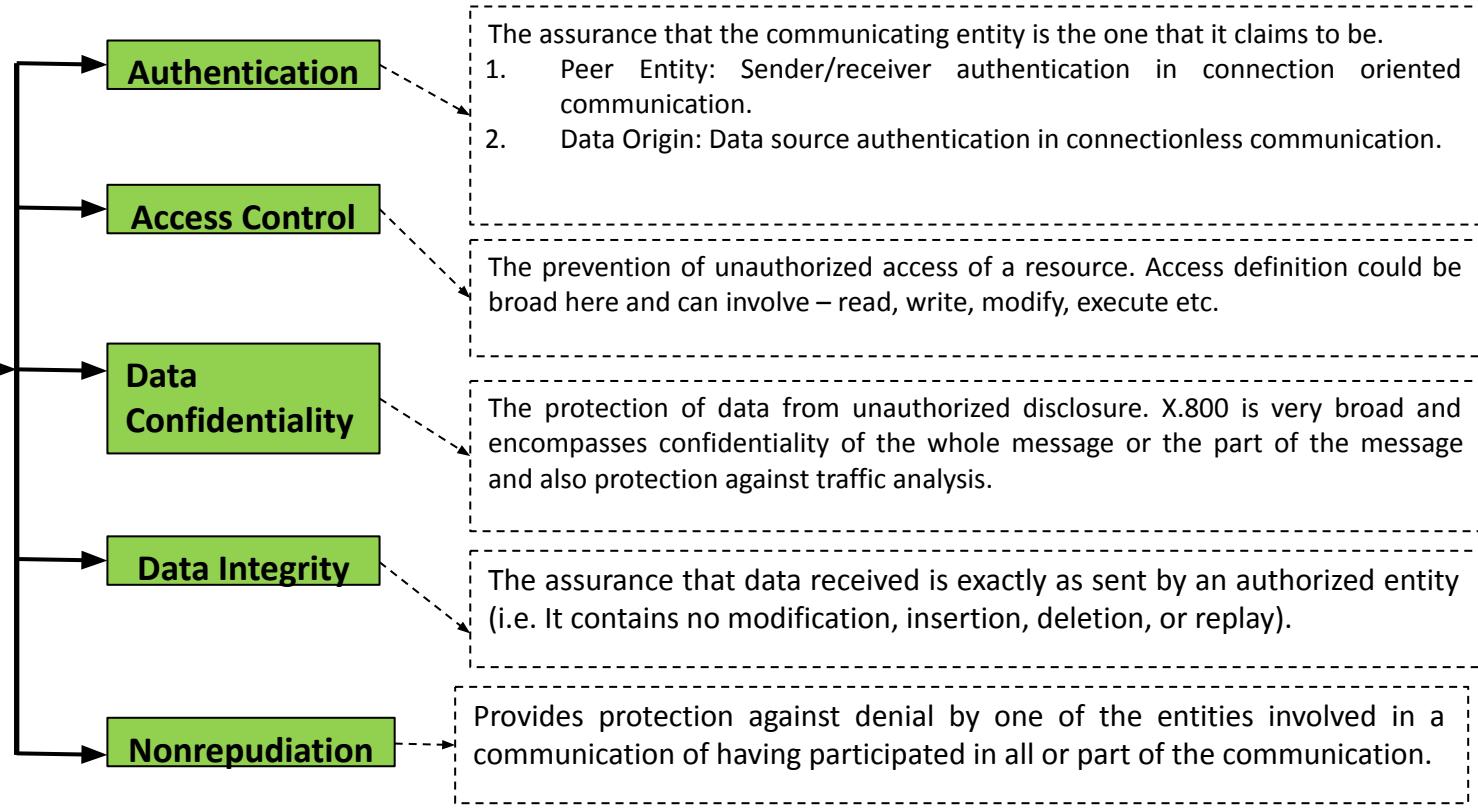
Security Mechanisms



Security Services

As described in X.800

Security Services



Availability & Availability Service

Availability: Both X.800 and RFC-4949 define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorized entity. A variety of attacks can result in the loss of or reduction in availability.

The availability service addresses the security concerns raised by Denial of Service attacks. It can be treated as sixth type of security service.

Security Mechanisms and Services



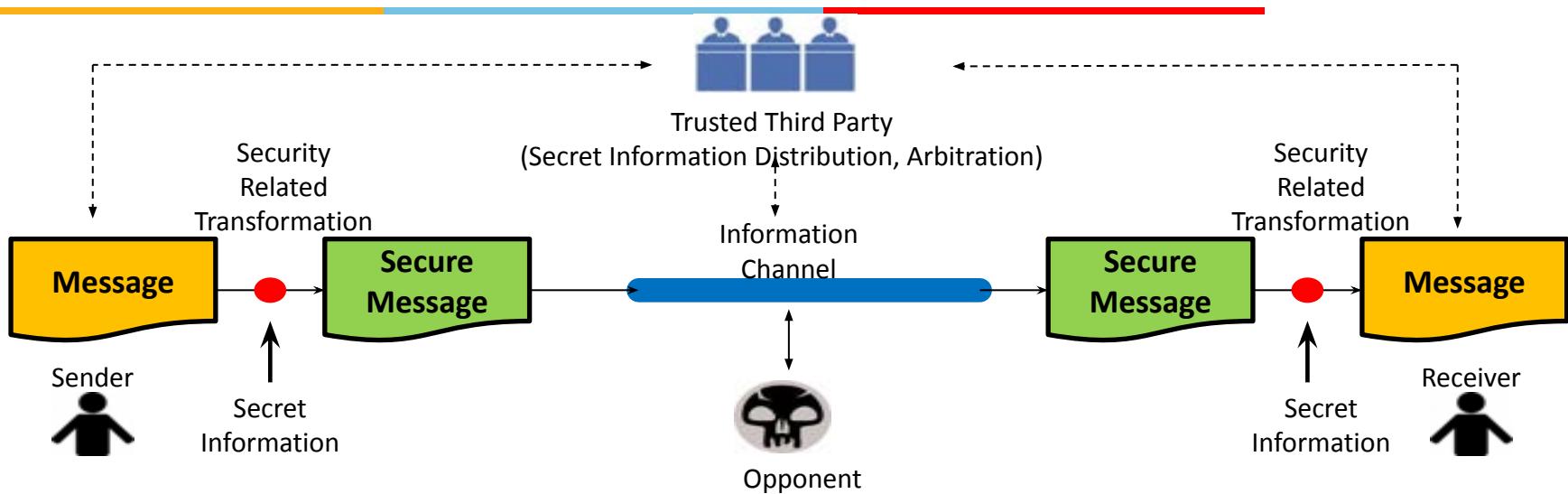
What is the difference?

Security Service is a processing or communication service that is provided by a system to give a specific kind of protection to the system resources. Security services are implemented by security mechanisms. [RFC-4949]. A mechanism or combination of mechanisms are used to implement one or more services.

SERVICE	MECHANISM							
	Enchipherment	Digital signature	Access control	Data integrity	Authentication exchange	Traffic padding	Routing control	Notarization
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				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

Sample mapping:
mechanisms to services

A Model for Network Security



- A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the sender and receiver.
- An opponent may present a threat to the confidentiality of the message that is being transmitted.
- Using a secret information, sender secures the original message (encrypted or ciphered) and using the same or different secret information receiver recovers the original message (decrypted or deciphered).
- A trusted third party distributes the secret information to both the sender and receiver.

Techniques to Implement Security Mechanisms



Cryptography: in Greek it means “*secret writing*”. In the network security it means the science of transforming the messages to make them secure and immune to attacks.

- i. Symmetric-Key Encipherment
- ii. Asymmetric (or Public) Key Encipherment
- iii. Data Integrity
- iv. Mutual Trust

Steganography: in Greek it means “*covered writing*”. In contrast with cryptography, it means concealing the message itself by covering it with something else. Example:

A letter is written on the paper using onion juice or ammonia salts which would not be visible unless exposed to heat.

This course primarily deals with Cryptographic Techniques in Network Security.

Symmetric-Key Encipherment

Sender encrypts the message using an encryption algorithm and the receiver decrypts the message using a decryption algorithm. Symmetric-Key Encipherment uses a single secret key for both encryption and decryption.

It is analogous to that the sender puts the message in a box and locks the box with a shared key. Receiver opens the box with the same shared key and gets the message.



Asymmetric-Key Encipherment

To send a secure message, the sender first encrypts the message using receiver's public key. To decrypt the message, the receiver uses its own private key.



Data Integrity and Mutual Trust

Data Integrity: Different cryptographic techniques to ensure the data integrity. E.g. Hashing and Message Authentication.

Mutual Trust: Different methods for key generation and distribution. Entity authentication and notarization methods.

Terminology

- **Plaintext** – An original message in its ‘as-it-is’ form.
 - **Ciphertext** – Coded message. Cannot be understood just by reading it.
 - **Encryption (Enciphering)** – The process of converting plaintext to ciphertext.
 - **Decryption (Deciphering)** – The process of restoring plaintext from ciphertext.
 - **Cryptographic System (Cipher)** - A scheme used for encryption.
 - **Cryptography** – The area of study for many schemes used for encryption.
 - **Cryptanalysis** – The area in which techniques are used for deciphering a message without any knowledge of the enciphering details. Colloquially called ‘breaking the code’.
 - **Cryptology** - The areas of cryptography and cryptanalysis.
-

Kerckhoffs' Principle

Let us keep always in mind

Kerckhoffs' Principle: Auguste Kerckhoffs, a 19th century professor of languages and an cryptographer in Paris formulated that one should always assume that an adversary, knows the encryption and decryption algorithm. The resistance of the cipher to attack must be based only on the secrecy of the key.

Shannon's Maxim: Kerckhoffs's principle was kind of reformulated by Claude Shannon as "the enemy knows the system", i.e., "one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them".

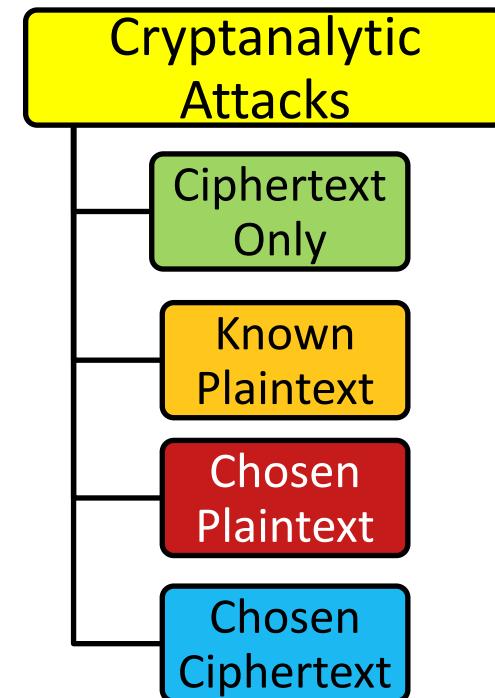
Foundational Assumptions
while Designing Cryptosystems

Cryptography: Dimensions

1. **Operations:** The type of operations used for transforming plaintext to ciphertext. The fundamental requirement is that no information be lost. Two techniques:
 - a) ***Substitution:*** each element in the plaintext is mapped into another element.
Example - **APPLE** can be decrypted as **BQQMF** using some rule. This rule is the key.
 - a) ***Transposition:*** elements in the plaintext are rearranged.
Example - **APPLE** can be decrypted as **ELPPA** using some rule. This rule is the key.
- Most systems, deploy 'product systems', that involve multiple stages of substitutions and transpositions.
3. **The number of keys:** If both sender and receiver use the same key, the system is referred to as ***symmetric, single-key, secret-key, classical,*** or ***conventional*** encryption. If the sender and receiver use different keys, the system is referred to as ***asymmetric*** or ***public-key*** encryption.
4. **Processing:** There are two types of processing:
 - a) ***Block cipher*** processes the input one block of elements at a time, producing an output block for each input block.
 - b) ***Stream cipher*** processes the input elements continuously, producing output one element at a time, as it goes along.

Cryptanalytic Attacks

- Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs.
 - Tries to exploit the characteristics of the algorithm to find out a specific plaintext or to find out the key being used.
1. **Ciphertext Only** – Analyst has least amount of information (algorithm and ciphertext).
 2. **Known Plaintext** – Analyst has some more information like plaintext and ciphertext pairs or some pattern, header, trailer, or message banners.
 3. **Chosen Plaintext** – Analyst successfully make sender to insert a message in the plaintext that is chosen by the analyst. Helps him to pick patterns to reveal the overall structure.
 4. **Chosen Ciphertext** – The analyst tries to analyse - how to influence the plaintext to produce the chosen ciphertext with an overall objective to find out the key.



Brute-Force Attacks

- A brute-force attack involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained.

- To make the possible deciphering more difficult the following approaches are practiced:
 - Usage of symbols also in addition to language characters.
 - Compressed ciphertext.
 - Padding.

Security Criteria



- An encryption scheme is ***unconditionally secure*** if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available and no matter how much time an attacker or analyst has.
- There is no encryption algorithm that is unconditionally secure. Therefore, the aim is to meet one of these criteria:
 1. The ***cost*** of breaking the cipher exceeds the ***value*** of the encrypted information.
 2. The ***time*** required to break the cipher exceeds the ***useful lifetime*** of the information.
- An encryption scheme is said to be ***computationally secure*** if either of the above two criteria are met.

Substitution Techniques



Caesar Cipher

- Caesar cipher is the earliest known and the simplest example substitution cipher. It is said to be used by Julius Caesar.
- The Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.
- As an example, usage of Caesar Cipher in English using the following key (e.g. 'a' is substituted by 'D' and 'x' by 'A')

plain: a b c d e f g h i j k l m n o p q r s t u v w x y z (plaintext - lower case)
cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C (ciphertext - upper case)

- If a sentence **meet me after party** is to be encrypted, its Caesar cipher will be: **PHHW PH DIWHU SDUWB**
- Mathematically, if all plaintext alphabets are represented as numerals in ascending order(a=0, b=1.....z=25) then a plaintext letter p can be encrypted to cipher letter C as: $C = E(3, p)$, where E is a substitution encryption function. Or,

$$C = E(3, p) = (p+3) \text{ mod } 26$$

- In stead of 3, if it needs to be generalized with any number k:

$$\begin{aligned}C &= E(k, p) = (p+k) \text{ mod } 26 \\p &= D(k, C) = (C-k) \text{ mod } 26\end{aligned}$$

*for encryption
for decryption*

Examples

(1) Using $K = 15$ and Caesar Cipher, encrypt the message “hello”.

Plain Letter	Numeral Order (p)	Cipher Letter Numeral Order $C = (p+K) \text{ mod } 26$	Corresponding Cipher Letter
h	7	$(7+15) \text{ mod } 26 = 22$	W
e	4	$(4+15) \text{ mod } 26 = 19$	T
l	11	$(11+15) \text{ mod } 26 = 0$	A
l	11	$(11+15) \text{ mod } 26 = 0$	A
o	14	$(14+15) \text{ mod } 26 = 3$	D

(2) Using $K = 15$ and Caesar Cipher, decrypt the message “WTAAD”.

Cipher Letter	Numeral Order (C)	Cipher Letter Numeral Order $p = (C-K) \text{ mod } 26$	Corresponding Cipher Letter
W	22	$(22-15) \text{ mod } 26 = 7$	h
T	19	$(19-15) \text{ mod } 26 = 4$	e
A	0	$(0-15) \text{ mod } 26 = 11$	l
A	0	$(0-15) \text{ mod } 26 = 11$	l
D	3	$(3-15) \text{ mod } 26 = 14$	o

$r = a \text{ mod } n$, if a is negative, then the final $r = r+n$. E.g.

$r = -13 \text{ mod } 12 = -1$ and final $r = -1+12 = 11$

$r = -7 \text{ mod } 10 = -7$ and final $r = -7+10 = 3$

Cryptanalysis on Caesar Cipher

Weaknesses in the Technique:

- Encryption and decryption algorithms are known.
- There are only 25 keys. If the attacker knows it is Caesar Cipher on English alphabets, all the values of k from 1 to 25 can be tried out as part of brute-force cryptanalysis.
- The language is easily recognizable.

Lesson Learnt:

- Let us assume opponent can find out the algorithms.
- Select a technique that permits very large key space – a lot of time required to try all.
- Language abbreviation or compression can be done. If a plaintext file is first zipped and then encrypted it will be impractical to decrypt such file. In this case more characters will be there to decipher.

XcZ¥Ä-Ulä:¥f»zt□Ö~u...zEöÄÖVy¤Iä%Çlô]¥KRIBö{M' :UI□æù:PÍz]b...\$†×
E3öfÄ"ÆÍl=li:Ø]rcç)J2£ö@□lêqcÄIB[Ó|¶äidi‡ I«ý ïæ+ë\$ú%í-IVý mNYe%L
Iýíí'È-¶+⁵]ylüküsÖ□?IÖ^%ùÑYd· 8'¶qcèlx· s|[Kft"eA" _ýia-£I'nll‡@
mÄÜ;JÅæ[¶l-ØR±äiZ]ÖKý [uO·]é*ri\$~8%a{k¥7çý ll+]ýëëÖ>I -iöZ4ò9iGlo
Pþl+IB-uwlý èUÜ>*+II·Sa'&-ÑAY ãNÖR<!U»=o«ó",Gt_?801iu□ë-iFHëÖô

Sample of Compressed Text

Playfair Cipher

Introduction

- Playfair cipher was invented by British scientist **Sir Charles Wheatstone** in 1854, but it bears the name of his friend **Baron Playfair**. It was used during World War – I and II by British and US armies. Its steps follow as below.
- It uses a 5x5 matrix of letters using a key word. For example a matrix is constructed below using a keyword **NETWORK**. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter. If I or J is in the keyword, two other consecutive letters can be selected and counted as one.

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

Keyword

Playfair Cipher

Procedure

- If the two letters of the digram fall in the same row, their next letters to the right is (wraparound fashion) taken as their replacement. E.g. ***ac = BR, rk = KA***. While decrypting, left letter is taken.
- If the two letters of the digram fall in the same column, the letters beneath them (wraparound fashion) are taken as their replacement. E.g. ***nu = RN, gp = PX***. While decrypting upper letter is taken.
- Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. E.g. ***fq = HM, gs = IP (or JP)***. Same while decrypting.
- Repeating plaintext letters that are in the same digram are separated with a filler letter, such as x, so that ***balloon*** would be treated as ***ba lx lo on*** and its corresponding ciphertext will be ***CB PU SN NE***. Filler is removed after decrypting.
- Filler can also be used to complete the incomplete digram (the last one). Filler is removed after decrypting.

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

Playfair Cipher

Example

Let us now construct a matrix using key word LGDBAQ:

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

Let us now encrypt the word "GERMAN" using the above matrix. It will be first converted into digrams like GE, RM and AN.

Equivalent of GE would be DC.

Equivalent of RM would be SK.

Equivalent of AN would be BO.

So, GERMAN would be encrypted as DCSKBO.

Polyalphabetic Cipher

- In Caesar Cipher and Playfair Cipher, each plaintext character is encrypted with a known ciphertext character. The ciphertext character for a plaintext character can be computed in advance and it is fixed if the key is known. These types of ciphers are called ***monoalphabetic ciphers***.
- One of the ways to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message.
- The general name for this approach is ***polyalphabetic substitution cipher***.
- All these techniques have the following features in common:
 - I. A set of related monoalphabetic substitution rules is used.
 - II. A key determines which particular rule is chosen for a given transformation.



Vigenère Cipher

- One of the simplest polyalphabetic cipher. Named after ***Blaise de Vigenère***, a 19th century cryptographer in France.
- There is a sequence of n plaintext letters:
 $P = p_0, p_1, p_2, p_3, \dots, p_{n-1}$
- There is a key consisting of m letters (assuming m < n)
 $K = k_0, k_1, k_2, k_3, \dots, k_{m-1}$
- The sequence of n ciphertext letters are calculated as:
 $C_i = (p_i + k_{i \bmod m}) \bmod 26$; where i = 0 to (n-1)
- Similarly, the plaintext is restored from ciphertext as:
 $p_i = (C_i - k_{i \bmod m}) \bmod 26$; where i = 0 to (n-1)
- Since we assumed that m < n, so (i mod m) will wraparound (reuse) the values of key letters.

Vigenère Cipher

Example

Keyword = deceptive

Plaintext message = we are discovered save yourself

Ciphertext will be calculated as follows:

key: deceptive deceptivedeceptive

plaintext: we are discovered save yourself

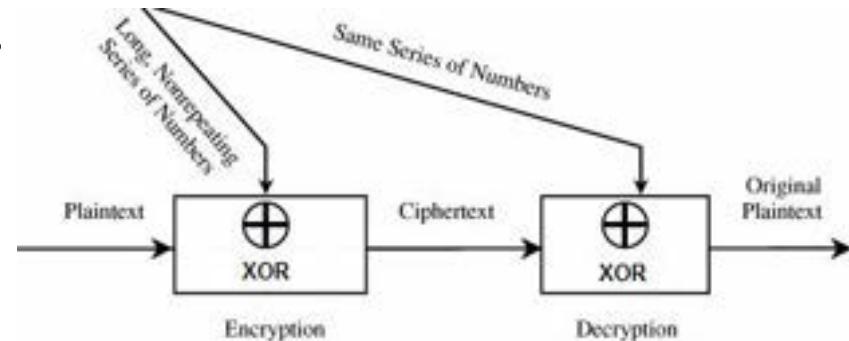
a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Vernam Cipher



- If the key is as long as the plaintext, it could provide an ultimate defence against attack.
- AT&T Engineer **Gilbert Vernam** devised such a system in 1918.
- His system works on binary data (bits) rather than letters. The system can be expressed as follows $c_i = p_i \oplus k_i$, where \oplus is the binary Exclusive-OR (XOR) operation.



- Because of the properties of XOR function, the decryption also involves the same bitwise operation. $p_i = c_i \oplus k_i$.

One Time Pad



- ❑ A US army signal corps officer **Joseph Mauborgne** proposed an improvement over Vernam Cipher in 1914.
- ❑ Mauborgne suggested using a random key that is as long as the message, so that the key need not be repeated. In addition, the key is to be used to encrypt and decrypt a single message, and then it is discarded.

```
ciphertext: ANKYODKYUREPFJBYOJDSPREYIUNOFDOIUFPLUYTS  
key: pxlmvmsydofovrvzwc tnlebnecvgduplicahfzzlunvih  
plaintext: mr mustard with the candlestick in the hall
```

```
ciphertext: ANKYODKYUREPFJBYOJDSPREYIUNOFDOIUFPLUYTS  
key: pftaoomivdaaxaoufhklllmlhsadaoatewbafavovuhwt  
plaintext: miss scarlet with the knife in the library
```

Same ciphertext but different keys yield different plaintexts.

- ❑ One Time Pad is considered as ultimate security in cryptography. But these two issues make it practically unusable:
 - i. Large number of random keys.
 - ii. Secure channel to communicate these keys.

English Letters

Relative Frequency

Letter	Frequency	Letter	Frequency	Letter	Frequency	Letter	Frequency
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

Courtesy: Lewand, R. *Cryptological Mathematics*. Washington, DC: Mathematical Association of America, 2000.

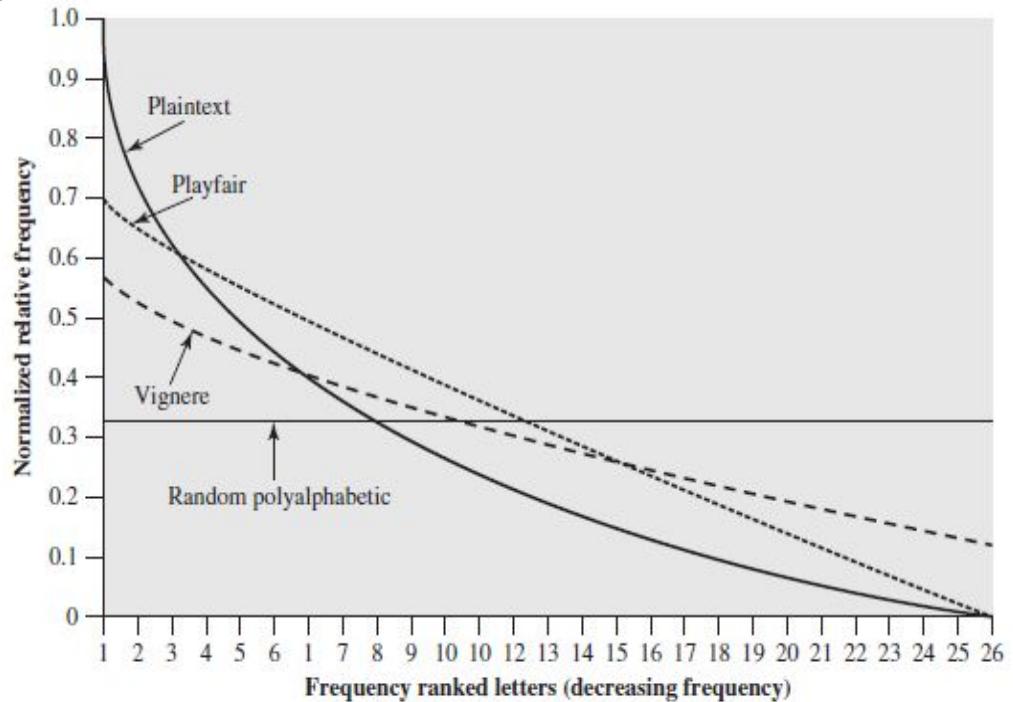
The above table shows the relative frequency of the English letters in decreasing order.
E is used most and Z is used the least.

Relative Performance

The line labelled plaintext plots a typical frequency distribution of the 26 alphabetic characters (no distinction between upper and lower case) in ordinary text.

This is also the frequency distribution of any monoalphabetic substitution cipher, because the frequency values for individual letters are the same, just with different letters substituted for the original letters.

The number of occurrences of each letter in the text is counted and divided by the number of occurrences of the most frequently used letter.



Random Polyalphabetic algorithms yield best results with no correlation of most frequently used letters in the plaintext and their corresponding ciphertexts.

Transposition Technique

- All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol.
- A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a ***transposition cipher***.
- E.g. plaintext is written row wise and encrypted reading it column wise based on the key. Multiple iteration can be done to make the forged reconstruction difficult.

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

First Transposition

Key:	4 3 1 2 5 6 7
Input:	t t n a a p t
	m t s u o a o
	d w c o i x k
	n l y p e t z
Output:	NSCYAUOPTTWLTMDNAOIEPAKTTOKZ

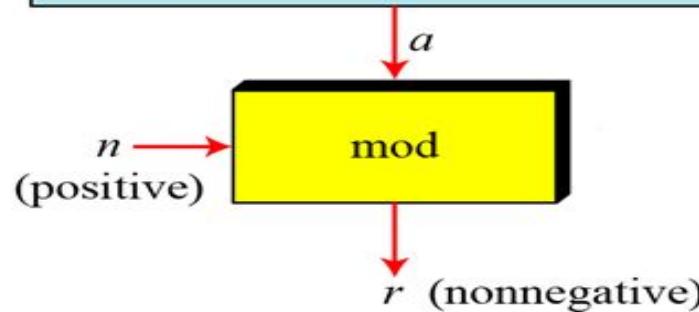
Second Transposition

Numbers

- The ***set of integers***, denoted by **Z**, contains all integral numbers (with no fraction) from negative infinity to positive infinity.
$$Z = \{-\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty\}$$
- A ***rational number*** is any number that can be expressed as p/q of two integers, p and q, with the denominator q not equal to zero. Since q may be equal to 1, every integer is a rational number. E.g. 23, 1.5, 22/7, -7, 1.14141414.....
- An ***irrational number*** is any real number that cannot be expressed as p/q of two integers p and q. Irrational numbers cannot be represented as terminating or repeating decimals. E.g. $\pi = 3.1412857\dots$
- The ***real numbers*** include all rational and irrational numbers.
- ***Prime numbers*** has only two divisors 1 and itself.

Modular Arithmetic

- $a = q \times n + r$
- The input n is called the *modulus*.
- The output r is called the *remainder* or *residue*
- The operation is called *modulo* or just *mod*.
- The *mod* operator is defined in such a way that it gives the non-negative residue r as an output.
- If r is negative then $\mathbf{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ | residue becomes positive.

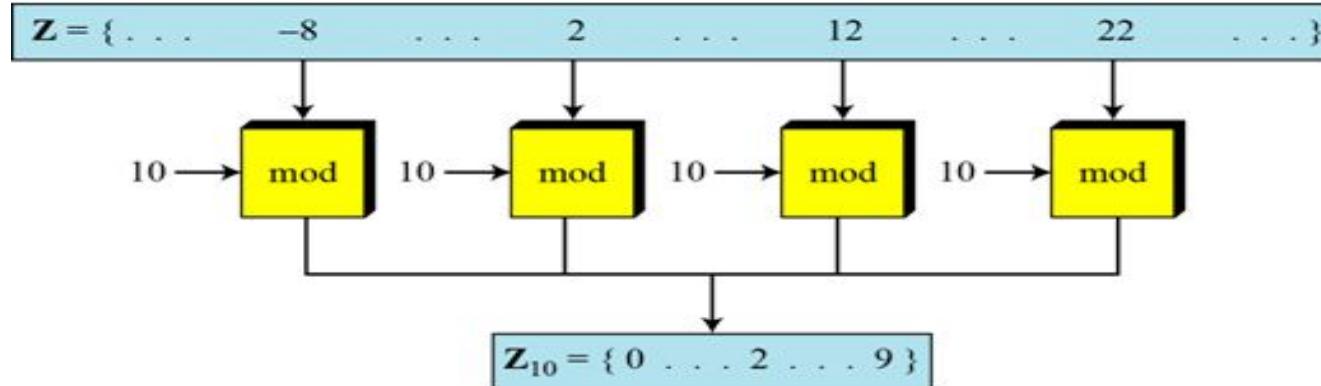


Examples

- $27 \bmod 5 = 2$
- $36 \bmod 12 = 0$
- $-18 \bmod 14 = -4 = -4+14 = 10$
- $-7 \bmod 10 = -7 = -7+10 = 3$

Congruence

- Two integers a and b are said to be congruent if $(a \bmod n) = (b \bmod n)$.
- It is written as $a \equiv b \pmod{n}$.



$$-8 \equiv 2 \equiv 12 \equiv 22 \pmod{10}$$

Congruence Relationship

Examples

- $2 \equiv 12 \pmod{10}$
 - $3 \equiv 8 \pmod{5}$
 - $8 \equiv 13 \pmod{5}$
 - $-8 \equiv 12 \pmod{10}$
-

Greatest Common Divisor (GCD)

- The greatest common divisor of a and b , $\gcd(a, b)$, is the largest integer that divides both a and b . Also it is defined that $\gcd(0, 0) = 0$.

 - The greatest common divisor is to be positive, so
 - $\gcd(a, b) = \gcd(a, -b)$
 - $= \gcd(-a, b)$
 - $= \gcd(-a, -b)$
 - In general, $\gcd(a, b) = \gcd(|a|, |b|)$

 - a and b are **relatively prime** if $\gcd(a, b) = 1$
-

Examples

- GCD of 24 and 27 = 3
- GCD of 25, 34 = 1 and also 25 and 34 are relatively prime.
- GCD of -5, 30 = 5
- GCD of 16, -40 = 8

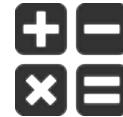
Euclidean Algorithm

To find out the GCD

q	r ₁	r ₂	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

GCD of 2740 and 1760 is 20

Matrix - Multiplication



Example-1

$$\begin{matrix} \text{C} & \text{A} & \text{B} \\ \left[\begin{matrix} 5 & 3 \end{matrix} \right] & = & \left[\begin{matrix} 5 & 2 & 1 \end{matrix} \right] \times \left[\begin{matrix} 7 \\ 8 \\ 2 \end{matrix} \right] \end{matrix}$$

→ ↓

In which: $53 = 5 \times 7 + 2 \times 8 + 1 \times 2$

Example-2

$$\begin{matrix} \text{C} & \text{A} & \text{B} \\ \left[\begin{matrix} 52 \\ 41 \end{matrix} \right] & = & \left[\begin{matrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{matrix} \right] \times \left[\begin{matrix} 7 & 3 & 2 & 1 \\ 8 & 0 & 0 & 2 \\ 1 & 3 & 4 & 0 \end{matrix} \right] \end{matrix}$$

$$52 = 5 \times 7 + 2 \times 8 + 1 \times 1$$



Matrix - Determinant

1. If $m = 1$, $\det(\mathbf{A}) = a_{11}$
2. If $m > 1$, $\det(\mathbf{A}) = \sum_{i=1 \text{ and } j=1..m} (-1)^{i+j} \times a_{ij} \times \det(\mathbf{A}_{ij})$

Where \mathbf{A}_{ij} is a matrix obtained from \mathbf{A} by deleting i^{th} row and j^{th} column.

$$\det \begin{bmatrix} 5 & 2 \\ 3 & 4 \end{bmatrix} = (-1)^{1+1} \times 5 \times \det [4] + (-1)^{1+2} \times 2 \times \det [3] \longrightarrow 5 \times 4 - 2 \times 3 = 14$$

Example

or
$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11} \times a_{22} - a_{12} \times a_{21}$$

Matrix – Inverse and Identity



Multiplicative Inverse (or just inverse) M^{-1} of a square matrix M is defined in such a way that $M \times M^{-1} = M^{-1} \times M = I$, where I is the identity matrix.

In an identity matrix I , all the elements are 0 except main diagonal elements from upper left to lower right which are all 1.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2x2 Identity Matrix

It is not always possible to have an integer multiplicative inverse of an integer matrix.

Residue Matrix

- $11 \times 19 = 209 = 1 \pmod{26}$
So, 11 and 19 are multiplicative inverse in modulo-26 arithmetic.

- Z is the set of integers. When a number is divided by n the remainder is always from 0 to $(n-1)$. Z_n represents this set of 0 to $(n-1)$ elements. E.g. $Z_5 = \{0, 1, 2, 3, 4\}$. **Z_n is the residue set.**
Cryptography uses residue matrices where all elements of a matrix are drawn from a set Z_n . If $n = 26$, it means all elements of a Z_{26} matrix will be drawn from $\{0, 1, 2, \dots, 25\}$.

- A residue matrix Z_n will have a multiplicative inverse matrix, if the determinant of that matrix has a multiplicative inverse in set Z_n .

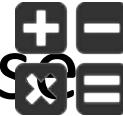
Mathematically, if **GCD (det(A), n) = 1** for a matrix A, it will have a multiplicative inverse. GCD is Greatest Common Divisor.

Multiplicative Inverse in $Z_{n \times \equiv}$

Extended Euclidean Method

q	n_1	n_2	r	t_1	t_2	t
2	26	11	4	0	1	-2
2	11	4	3	1	-2	5
1	4	3	1	-2	5	-7
3	3	1	0	5	-7	26
	1	0		-7	26	

- Let us find out if 11 (n_2) has a multiplicative inverse in Z_{26} ($n_1=26$) using **Extended Euclidean Method**.
Here, q = quotient and r = remainder.
- In the beginning, the temporary numbers, $t_1 = 0$ and $t_2 = 1$ and $t = t_1 - q \cdot t_2$ for all the steps.
- After all the operations, since n_1 and n_2 reduces to 1 and 0, so their GCD is 1 and hence there is a multiplicative inverse of 11 in Z_{26} .
- Multiplicative inverse = (last t_1) mod 26 = -7 mod 26 = 19
- Verification: $(19 \times 11) \bmod 26 = 209 \bmod 26 = 1$.
- **Hence, 11 and 19 are multiplicative inverse in Z_{26} .**



Matrix - Multiplicative Inverse

If a square matrix A has a non-zero determinant then the multiplicative inverse of the matrix is calculated as:

$$[A^{-1}]_{ij} = (\det(A))^{-1} \cdot (-1)^{i+j} \cdot D_{ji}$$

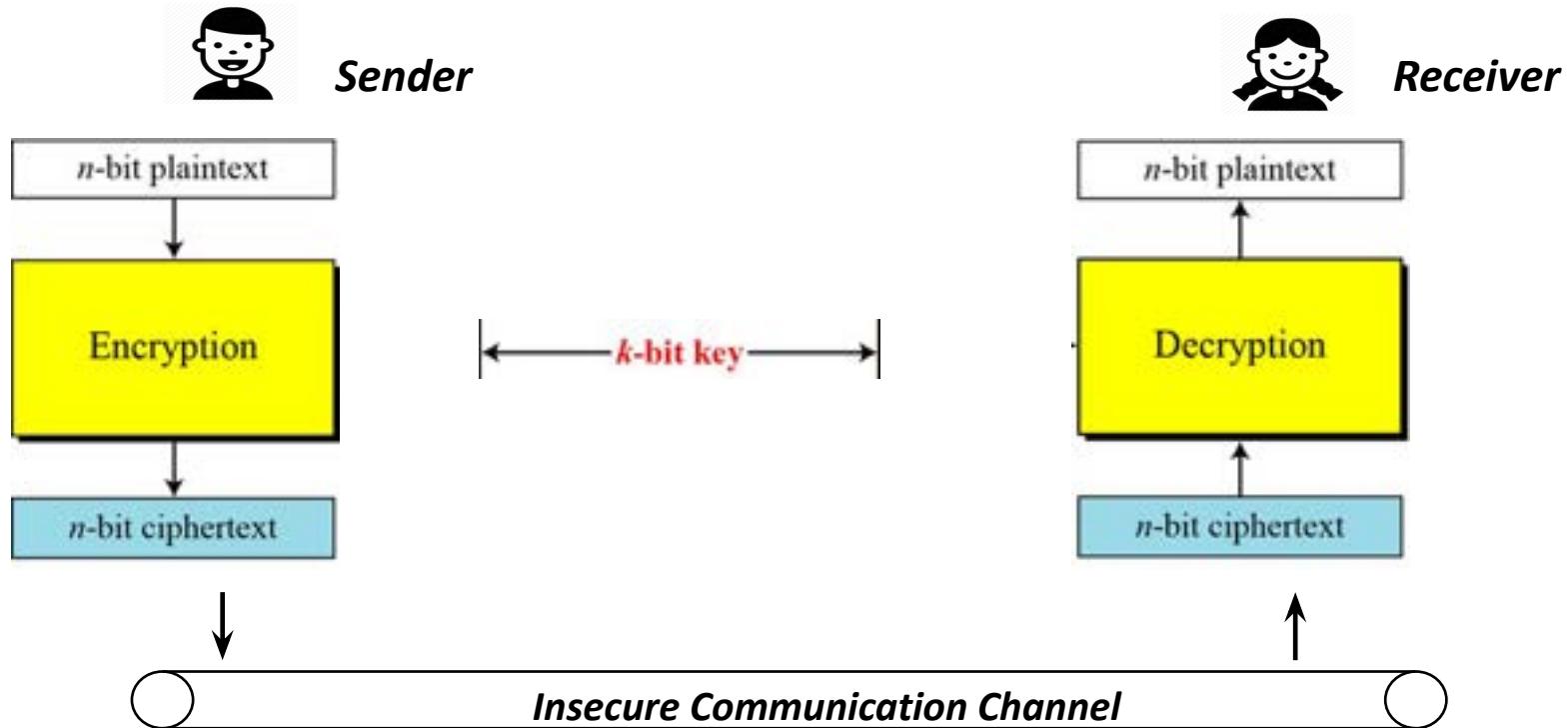
Where:

$(\det(A))^{-1}$ = Multiplicative inverse of $\det(A)$ in Z_{26} . It means using extended Euclidean method the GCD of $\det(A)$ and 26 is to be found out. If it is 1, it means $(\det(A))^{-1}$ exists.

D_{ji} = Determinant of the matrix deleting j^{th} row and i^{th} column.

Refer to the worksheet for detailed example.

Concept of a Block Cipher



Challenges in Block Ciphering

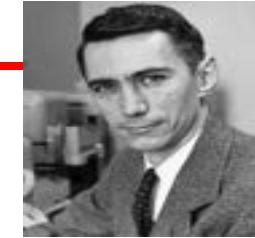
- How the key will be generated and distributed?
 - Selection of block ciphering algorithm to encrypt and decrypt.
 - How do we know these algorithms are strong enough?
-

Diffusion and Confusion

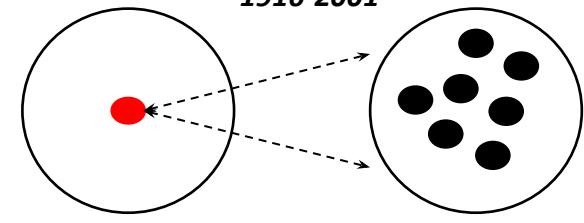


By Claude Shannon

- We have seen in classical ciphering techniques like Caesar Cipher that one letter was substituted by another letter with some known rule.
- It was very easy to de-cipher the ciphertext if that rule is known.
- We need more complexity in encryption so that a ciphertext cannot be easily de-ciphered.
- The terms **diffusion** and **confusion** in cryptography were introduced by Claude Shannon who was an American mathematician and engineer and considered as a father of information theory.
- **Diffusion** means each plaintext letter affect the value of many cipher text letters. For example, a plaintext letter 'B' need not be ciphertext letter 'E' all the time. It could be something else also.
- **Confusion** means the statistical relationship between the rule to encrypt and the ciphertext produced using should be as complex as possible. So it is very difficult to break the rule.
- Using diffusion and confusion theory as foundation principle, many modern ciphering algorithms work.



*Claude Shannon
1916-2001*



Diffusion Theory

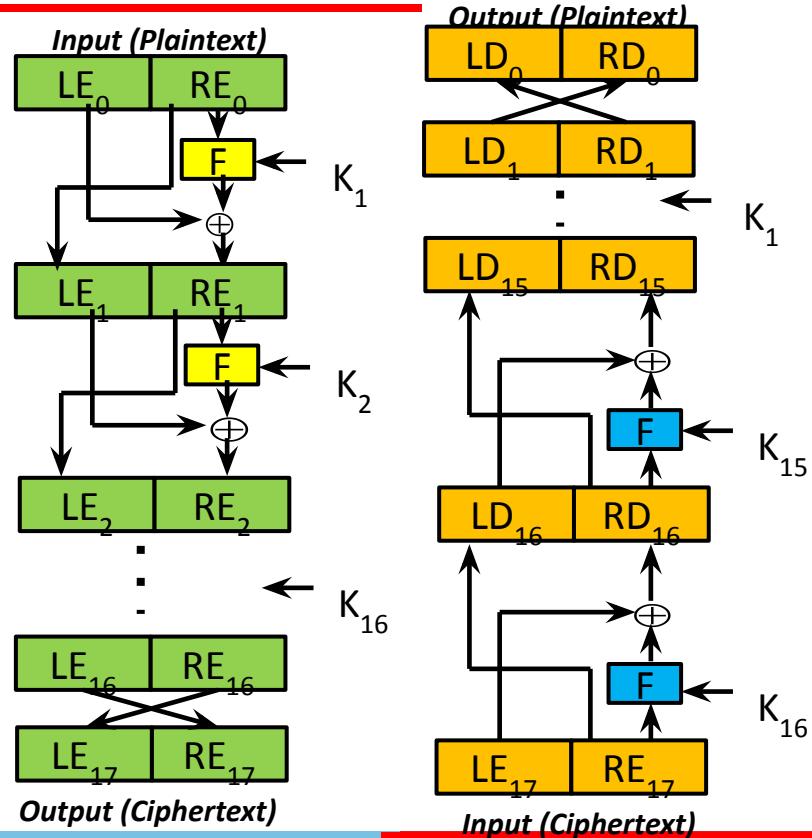


Confusion Theory

Feistel Cipher Structure

Encryption and Decryption

- The plaintext is divided into two equal halves LE_i and RE_i . E stands for Encryption., L stands for left and R stands for right. Initially $i = 0$.
- A subkey K_{i+1} is derived from the key K. So, initially the derived key is K_1 .
- A function F is defined which takes subkey K_{i+1} and right half RE_i of the data as inputs.
- The output of the function F is XORed with the left half of the data (LE_i) and made the new right half (RE_{i+1}).
- Right half (RE_i) of the previous round makes the left half for the next round (LE_{i+1}).
- The iteration is repeated 16 times and the left and right halves are swapped.
- The final output is the ciphertext.
- The same steps are used for decryption. (D stands for Decryption)
- In all steps, **substitution** is performed on the left half XORing it with the output of function F and then **permutation** is performed by swapping the right and left halves.
- Confusion and Diffusion is induced through multiple iterations.



Choice of Parameters and Design features



- Block Size:**
- Key Size:**
- Number of Rounds**
- Subkey Generation**
- Function**

There are two other considerations:

- Fast Encryption and Decryption**
- Ease of Analysis**

Data Encryption Standard (DES)

Introduced by NIST in 1977



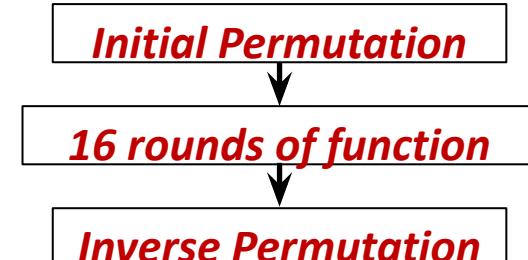
Widely used in communication systems until recently.

Its derivation 3-DES is still used in many systems which is basically “three times DES”

Block size is 64 bits. Key size is also 64 bits but out of 64 bits 56 bits are selected using a rule.

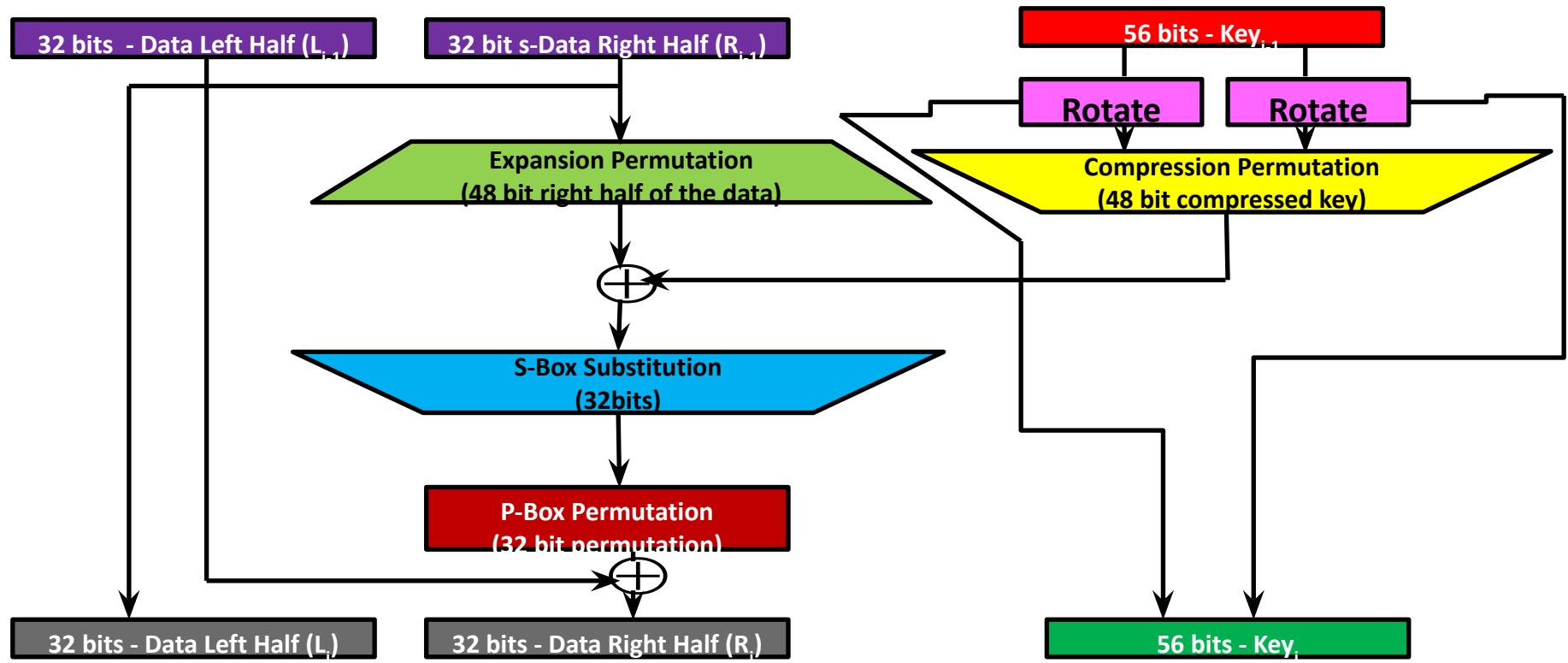
Three phases of operations. The second phase consists of 16 rounds.

After end of three phases, cipher text is obtained.



Reverse sequence to obtain plaintext.

One DES Round



Usage of Random Numbers

- To avoid replay attacks
 - Cryptographic Key Generation
 - In Stream Ciphering
-

Randomness

The following two criteria are used to validate that a sequence of numbers is random:

1. Uniform Distribution

Example: 11111111011111

Probability of occurring 1 is higher so it can not be claimed random

2. Independence

Example: 4, 8, 12, 16,

The next random number can be guessed. Generation is not independent.

Types of Random Numbers

- (1) Pseudo Random Numbers (PRN)
 - (2) True Random Numbers (TRN)
-

Linear Congruential PRNG

- Also referred as ***Lehmer Random Number Generator*** after US mathematician D. H. Lehmer.
- The algorithm is parameterized with four integers:

- m the modulus, $m > 0$
- a the multiplier, $0 < a < m$
- c the increment, $0 \leq c < m$
- X_0 the starting value or the seed, $0 \leq X_0 < m$

- The sequence of random numbers is obtained via:

$$X_{n+1} = (aX_n + c) \bmod m$$

This will produce integers in the range $0 \leq X_n < m$

- The choice of a , c and m is critical:
 - Let us say if $a = 7$, $c = 0$, $m = 32$ and $X_0 = 1$. This would generate $\{7, 17, 23, 1, 7, \dots\}$ which is not satisfactory because out of 32 values only 4 are being used.
 - If a is changed to 5, then the sequence would be $\{5, 25, 29, 17, 21, 9, 13, 1, 5, \dots\}$, which is relatively more satisfactory because 8 out of 32 numbers are being used.
- Normally, the value of m is kept very large – such as the largest unsigned integer for a computer. E.g. $2^{32}-1$.
- If random values are generated from 0 to $(m-1)$ it is called ***full period generator***.

Blum Blum Shub (BBS)

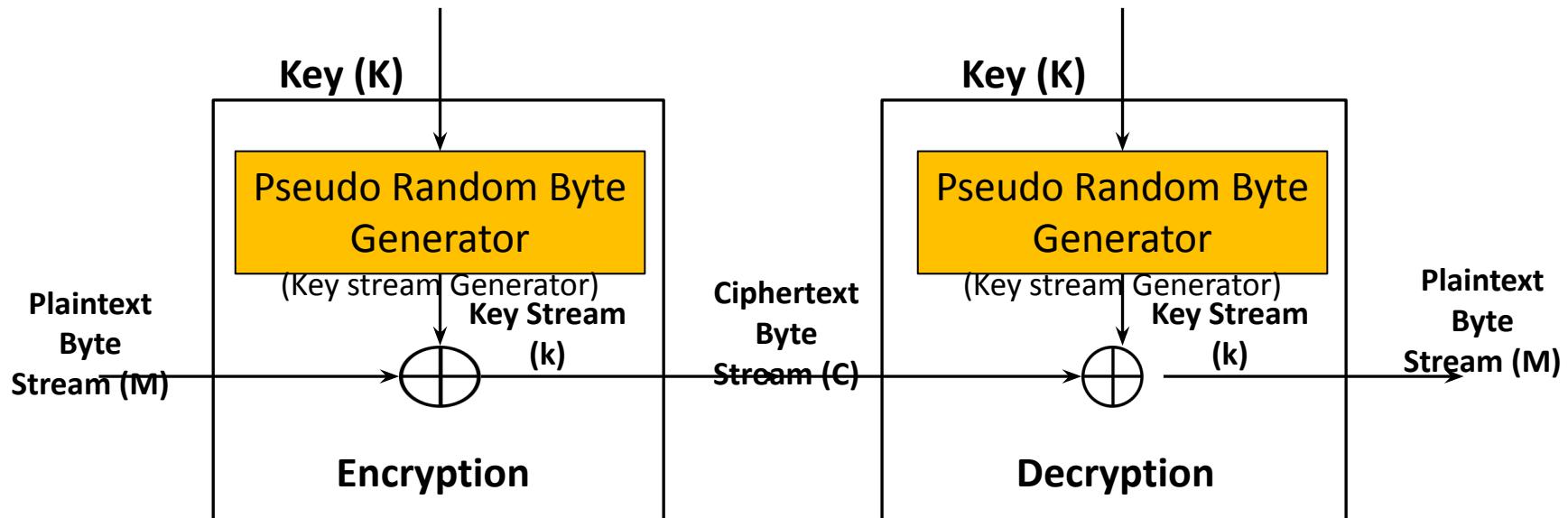


Generator

```
X0 = s2 mod n
for i = 1 to ∞
{
    Xi = (Xi-1)2 mod n
    Bi = Xi mod 2
}
```

- $n = P \times Q$
- Where P and Q are two prime numbers which provide remainder as 3 when divided by 4.
- s is random number, relatively prime to n ; this is equivalent to saying that neither P nor Q is a factor of s.
- X_i is the sequence of random numbers.
- B_i is the least significant bit of X_i which is used as random bit if needed for the application.

Stream Cipher



RC4 Stream Cipher



Ron Rivest

- Designed in 1987 by **Ron Rivest** for RSA Security.
It is a variable key size stream cipher with byte-oriented operations.
- Earlier it was kept secret by its developers. In September 1994, the RC4 algorithm was anonymously posted on the Internet on the Cypherpunks anonymous remailers list.
- RC4 is used in the following security protocols:
 1. Secure Sockets Layer/Transport Layer Security (SSL/TLS).
 2. Wired Equivalent Privacy (WEP) protocol and the newer WiFi Protected Access (WPA) protocol that are part of the IEEE 802.11 wireless LAN standards.
- Easy to explain and implement.

RC4 Operations

Initialization

First Step: Initialization

1. S is a state vector to store 256 bytes, with elements S[0], S[1].....S[255].
2. The entries of S are set equal to the values from 0 through 255 in ascending order; that is, S[0] = 0, S[1] = 1, S[255] = 255.
3. A key (K) of variable length <= 256 bytes is chosen.
4. A temporary vector T of 256 bytes is filled with the values of K. If K = 256, then all of K is filled in T, else K is repeated as required to fill T.

```
/* Initialization*/  
for i = 0 to 255 do  
{  
    S[i] = i;  
    T[i] = K[i mod key_length];  
}
```

RC4 Operations

Initial Permutation

Second Step: Initial Permutation

1. T is used to produce the initial permutation of S.
2. This involves starting with S[0] and going through to S[255], and for each S[i], swapping S[i] with another byte in S according to a scheme dictated by T[i].

```
/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
{
    j = (j + S[i] + T[i]) mod 256;
    swap (S[i], S[j]);
}
```

RC4 Operations

Stream Generation

Third Step: Stream Generation

1. Once the S vector is initially permuted, the input key (K) is no longer used.
2. Stream generation involves cycling through all the elements of S[i], and for each S[i], swapping S[i] with another byte in S according to a scheme dictated by the current configuration of S.
3. As an output a random stream bytes values (k) are generated.

```
/* Stream Generation */  
i, j = 0;  
while (true)  
{  
    i = (i + 1) mod 256;  
    j = (j + S[i]) mod 256;  
    swap (S[i], S[j]);  
    t = (S[i] + S[j]) mod 256;  
    k = S[t];  
}
```

```
/* Initialization */
for i = 0 to 255 do
{
    S[i] = i;
    T[i] = K[i mod key_length];
}

/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
{
    j = (j + S[i] + T[i]) mod 256;
    swap (S[i], S[j]);
}

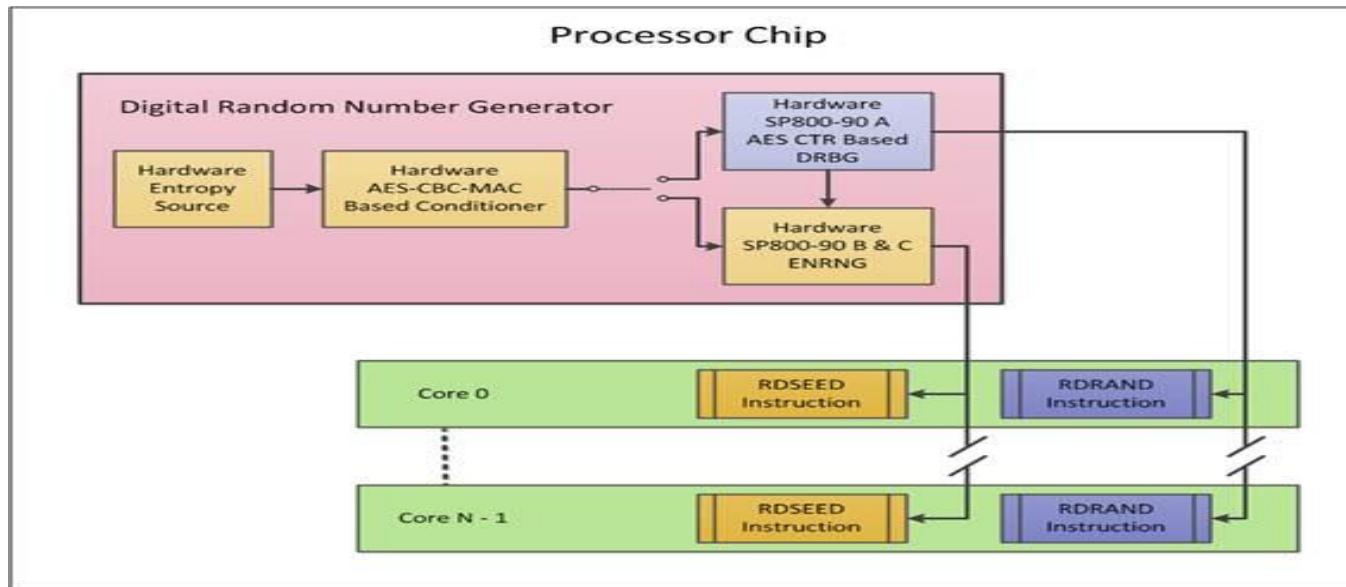
/* Stream Generation */
i, j = 0;
while (true)
{
    i = (i + 1) mod 256;
    j = (j + S[i]) mod 256;
    swap (S[i], S[j]);
    t = (S[i] + S[j]) mod 256;
    k = S[t];
}
```

Intel® Digital Random Number Generator (DRNG)

AES-CBC-MAC: AES Cipher Block Chaining Message Authentication Code

DRBG: Deterministic Random Bit Generator

ENRNG: Enhanced, Nondeterministic Random Number Generator.





Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems



Modes of Operations

- Block cipher is more than just an encryption algorithm: can be used as
 - different way of block encryption
 - stream cipher
 - hash function
 - MACs
 - PRNG
 - key establishment protocols



Modes of Operations

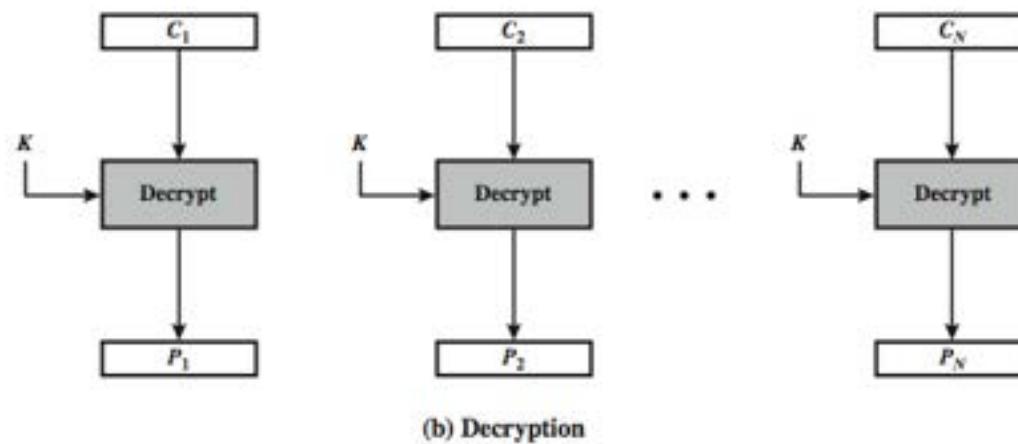
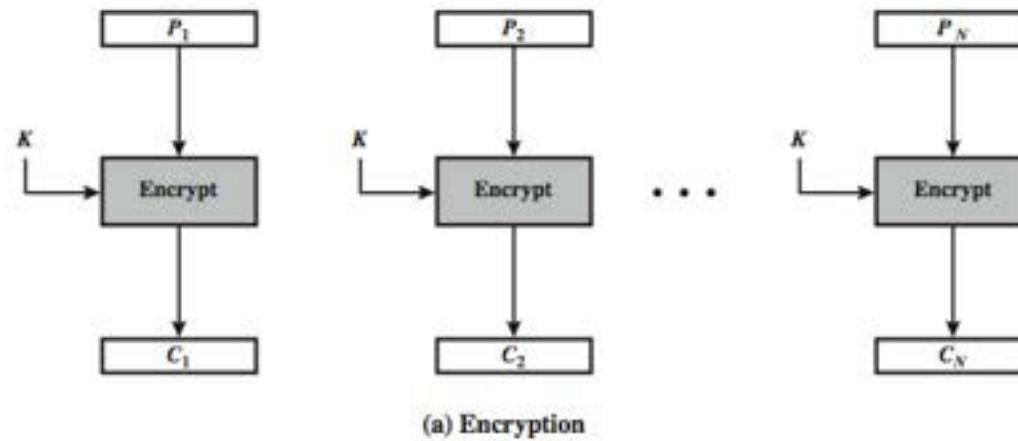
Popular modes of operations:

- Electronic Code Book Mode
- Cipher Block Chaining Mode
- Output Feedback Mode
- Cipher Feedback Mode



Electronic Code Book Mode (ECB)

Electronic Code Book Mode (ECB)



Electronic Code Book Mode

- If message > block size, then partitioned into block size
- Padding, if plaintext is not an exact multiple of cipher.
- Each block is encrypted separately
- +ve
 - Synchronization not required.
 - Bit error only to the corresponding blocks.
 - Efficient encryption and decryption (parallelization).
- -ve
 - Highly deterministic, Traffic analysis, Reordering.
 - Susceptible to substitution attack.





Attack on ECB

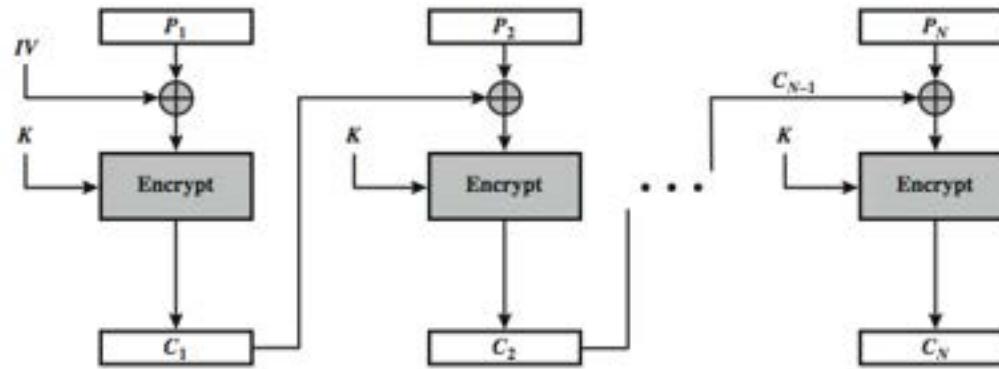


Attack on ECB

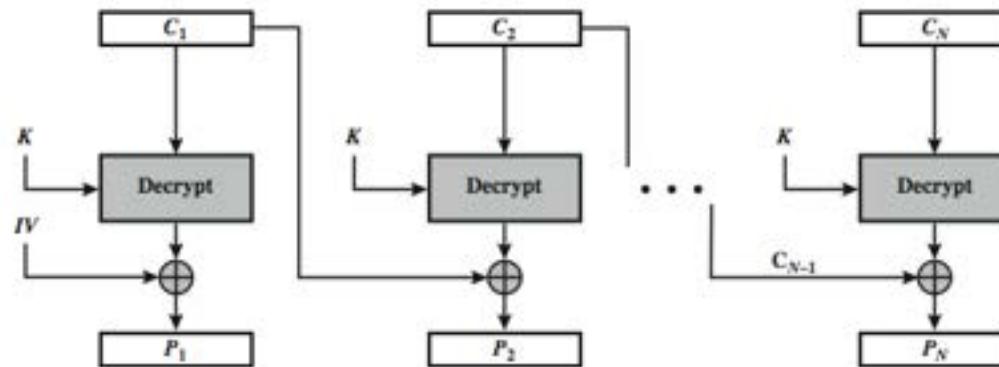
Cipher Block Chaining Mode

Cipher Block Chaining Mode

Cipher Block Chaining Mode



(a) Encryption



(b) Decryption

Cipher Block Chaining Mode

- CBC
 - Encryption should be chained together
 - Randomized using IV
- +ve
 - probabilistic due to IV
 - cipher text block depends on all blocks before it
 - Substitution attack similar to ECB not possible

Output Feedback Mode (OFB)

Output Feedback Mode(OFB)

Output Feedback Mode

- Synchronous cipher, similar to RC4
- Non-deterministic
- Main computation are independent of plaintext
- For efficiency, precompute S_i

Cipher Feedback Mode

Cipher Feedback Mode

- Asynchronous cipher, similar to RC4.
 - Non-deterministic.
 - Encryption and Decryption are same operation (stream cipher).
 - Can be used to encrypt 1 byte, i.e. short plaintext.
-

Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Multiple Encryption

Double Encryption:

Multiple Encryption

Double Encryption:

- Very little increase in resistance (Brute-force) over a single encryption.
- Naive Brute-force attack: $2^k \cdot 2^k = 2^{2k}$.
- Meet-in-Middle attack: $2^k + 2^k = 2^{k+1}$
- Space required for the attack: $2^k \cdot (n+k)$.
- For $k > n$ verify addition key with different pairs.





Multiple Encryption

Triple Encryption:

- MIM reduces the effective key length of triple encryption from 3k to 2k .
- Effective key length of 3DES is 112 bits opposed to 168 bits, which are actually used.

$$y = e_{k_3}(e_{k_2}(e_{k_1}(x))) \quad y = e_{k_1}(e_{k_2}^{-1}(e_{k_3}(x))) \quad y = e_{k_3}(x)$$

- Given 'l' subsequent encryption with a block cipher with key 'k' bits and block size of 'n' bits, as well as 't' pairs of plaintext and ciphertext, then the expected number of false keys which encrypts all plaintext to the corresponding ciphertext is given by $2^{k - tn}$.





Exhaustive Key Search

- If $k > n$, then found k_i may not be the used key.
 $AES_{k_i}(x) \stackrel{?}{=} y_i \quad i = 1, 2, 3, \dots, 2^{192}$
- Multiple pairs (x_i, y_i) are needed to find the correct key.
- Length of the (x, y) required to break the cipher with brute-force attack to bring the number of spurious key to zero is referred as unicity distance.
- Effectiveness of brute-force attack.
- What is the likelihood that key is in the both sets.
- Given a block cipher with a key length ' k ' bits and block size ' n ' bits as well as ' t ' plaintext-ciphertext pairs, then the expected number of false keys which encrypts all plaintext to the corresponding ciphertext is $2^{k - tn}$.

Multiple Encryption

Key Whitening:

- Extremely simple technique, additional computational load is negligible makes the block cipher more resistant to brute-force attack.
$$y = e_{k,k_1,k_2}(x) = e_k(x \oplus k_1) \oplus k_2.$$
$$x = e_{k,k_1,k_2}^{-1}(y) = e_k^{-1}(y \oplus k_2) \oplus k_1$$
 - Effective key length of 3DES is 112 bits opposed to 168 bits, which are actually used.
 - Useful for cipher which are relatively strong against analytical attack, but posses too short key-space.
 - Naive Brute-force attack: 2^{k+2n} .
 - MIM $\approx 2^{k+n}$.
 - If 2^m pairs, advanced attack complexity reduces to 2^{k+n-m} operations.
-

Thanks!!!
Queries?



BITS Pilani
K K Birla Goa Campus

Network Security

SS ZG513

Hemant Rathore
Department of Computer Science and Information Systems



Hash Function



Hash Function

- A widely used protocol to compute a unique representation of message (message digest / finger print / hash value).
- No key.
- Digital signatures, passwords, key derivation, data traffic, checksum, speed up the lookup tables, duplicates.



Hash Function

- Principle: Takes a string of arbitrary length and maps it to a fixed-length output, referred to as hash value.
- Practical Requirements: Arbitrary message size, Fixed output length, Finger print should be highly sensitive, Efficiency.
- Security Requirements: Pre-image resistance, Second pre-image resistance, collision resistance, Non-correlation, Near collision resistance, partial-image resistance.



Types of Hash Function

- Block cipher based hash functions:
 - Use block cipher such as AES to construct hash function.
- Dedicated Hash functions:
 - Specifically designed to serve as a hash function.
 - Fact: Any compression function 'f' which is collision resistant can be extended to a collision resistant hash function.
 - Merkle-Damgard Construction: Blocks are processed sequentially by the hash function, which has a compression function at its heart.



Block Cipher based Hash Functions



Block Cipher based Hash Functions (2)



Dedicated Hash functions

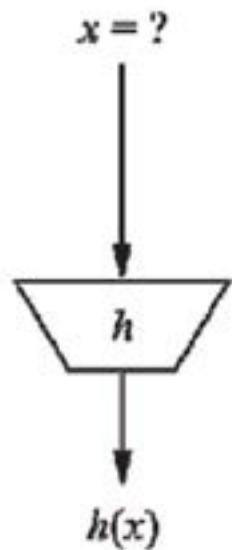


Dedicated Hash functions (2)

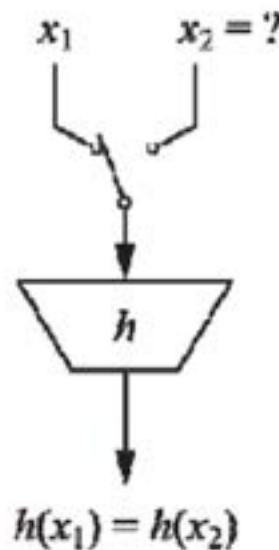


Requirements of Hash Function

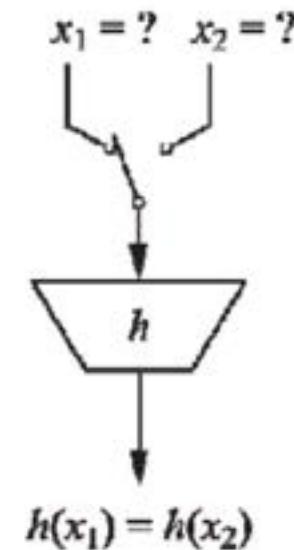
Requirements of Hash Function



preimage resistance



second preimage
resistance



collision resistance



Requirements of Hash Function

- Preimage resistance / One way function:

Given x it should be easy to calculate $h(x)$ -> one way
but given $h(x)$ it should be computationally infeasible to
compute x

- Second preimage resistance / Weak Collision Resistance

Given x_1 , and thus $h(x_1)$, it is computationally infeasible
to find any x_2 such that $h(x_1) = h(x_2)$.

- Collision resistance:

It is computationally infeasible to find any pairs $x_1 \neq x_2$
such that $h(x_1) = h(x_2)$.



2nd pre-image attack



Why Collision Attack



2nd Preimage Attack



Collision Attack



Birthday Paradox



Hash Function (Collision Attack)

$$t \approx 2^{(n+1)/2} \sqrt{\ln\left(\frac{1}{1-\lambda}\right)}.$$

λ	Hash output length				
	128 bit	160 bit	256 bit	384 bit	512 bit
0.5	2^{65}	2^{81}	2^{129}	2^{193}	2^{257}
0.9	2^{67}	2^{82}	2^{130}	2^{194}	2^{258}



Hash Function from Block Cipher

Davies Meyer: $H_i = H_{i-1} \oplus E_{x_i}(H_{i-1})$

Matyas-Meyer-Oseas: $H_i = E_{g(H_i - I)}(x_i) \oplus x_i$



Hash Function from Block Cipher

Miyaguchi-Preneel: $H_i = E_{g(H_{i-1})}(x_i) \oplus H_{i-1} \oplus x_i$



Hash Function from Block Cipher

Hirose Construction: The final output is $H_i \parallel G_i$

$$H_i = E_{H_{i-1}} \cdot x_i (G_{i-1} \oplus c) \oplus (G_{i-1} \oplus c)$$

$$G_i = E_{H_{i-1}} \cdot x_i (G_{i-1}) \oplus G_{i-1}$$



Hash Construction



Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Hash Construction

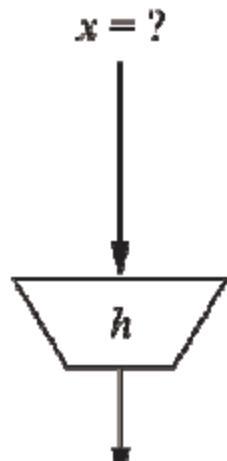
Types of Hash Function

- Block cipher based hash functions:
 - Use block cipher such as AES to construct hash function.
- Dedicated Hash functions:
 - Specifically designed to serve as a hash function.
 - Fact: Any compression function 'f' which is collision resistant can be extended to a collision resistant hash function.
 - Merkle-Damgard Construction: Blocks are processed sequentially by the hash function, which has a compression function at its heart.

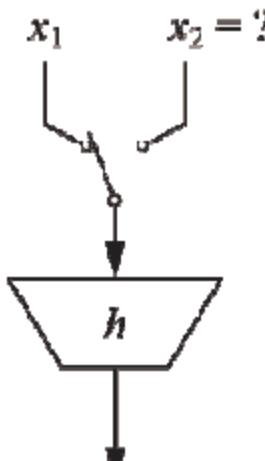
Hash Function

- Principle: Takes a string of arbitrary length and maps it to a fixed-length output, referred to as hash value.
- Practical Requirements: Arbitrary message size, Fixed output length, Finger print should be highly sensitive, Efficiency.
- Security Requirements: Pre-image resistance, Second pre-image resistance, collision resistance, Non-correlation, Near collision resistance, partial-image resistance.

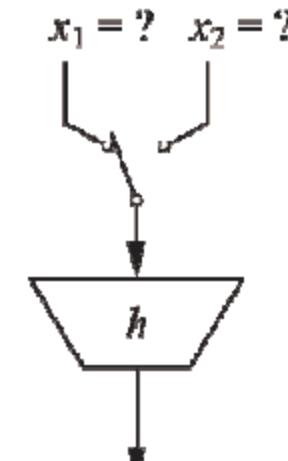
Requirements of Hash Function



preimage resistance



second preimage
resistance



collision resistance

Requirements of Hash Function

- Preimage resistance / One way function:

Given x it should be easy to calculate $h(x)$ -> one way
but given $h(x)$ it should be computationally infeasible to
compute x

- Second preimage resistance / Weak Collision Resistance

Given x_1 , and thus $h(x_1)$, it is computationally infeasible
to find any x_2 such that $h(x_1) = h(x_2)$.

- Collision resistance:

It is computationally infeasible to find any pairs $x_1 \neq x_2$
such that $h(x_1) = h(x_2)$.

Dedicated Hash Function

- **MD4 family of hash functions:**

Algorithm	Output [bit]	Input [bit]	No. of rounds	Collisions found
MD5	128	512	64	yes
SHA-1	160	512	80	not yet
SHA-2	SHA-224	224	512	64
	SHA-256	256	512	64
	SHA-384	384	1024	80
	SHA-512	512	1024	80
				no

λ	Hash output length				
	128 bit	160 bit	256 bit	384 bit	512 bit
0.5	2^{65}	2^{81}	2^{129}	2^{193}	2^{257}
0.9	2^{67}	2^{82}	2^{130}	2^{194}	2^{258}



SHA-1 High Level View

Pre-processing

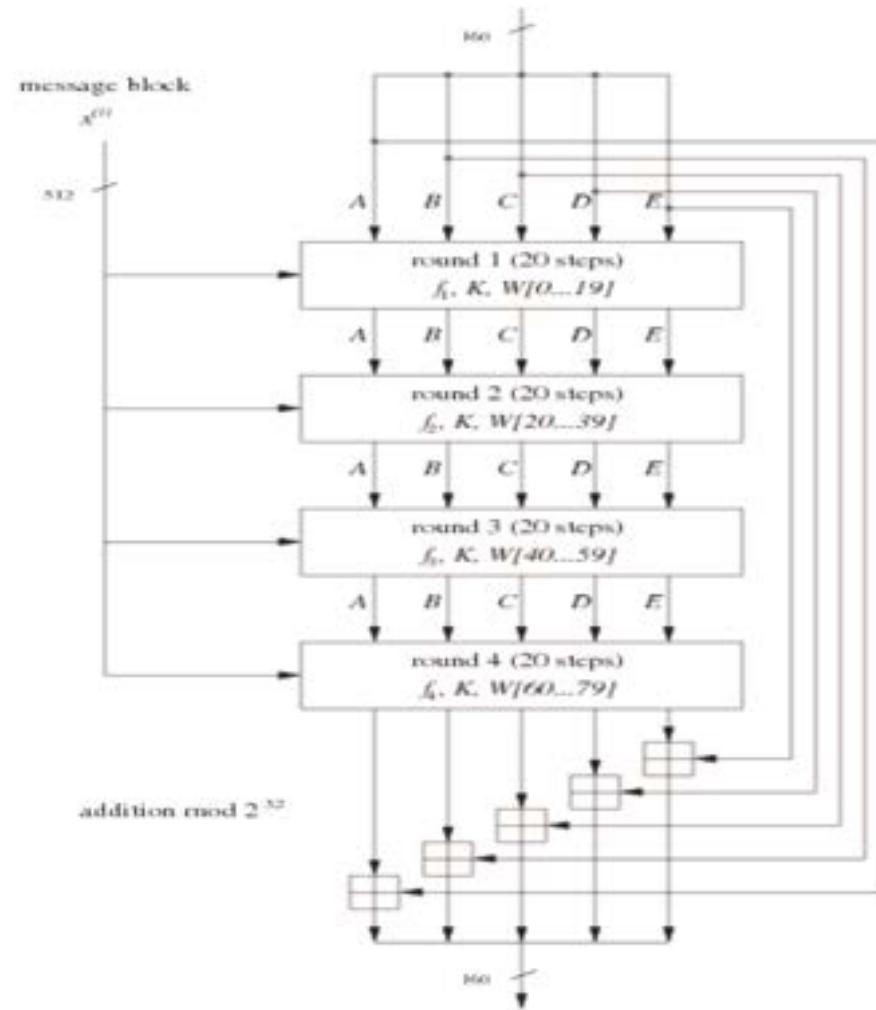
Recall Block Cipher

SHA-1 (Block Diagram)

SHA-1

Stage t	Round j
1	0 ... 19
2	20 ... 39
3	40 ... 59
4	60 ... 79

SHA-1 (Block Diagram)



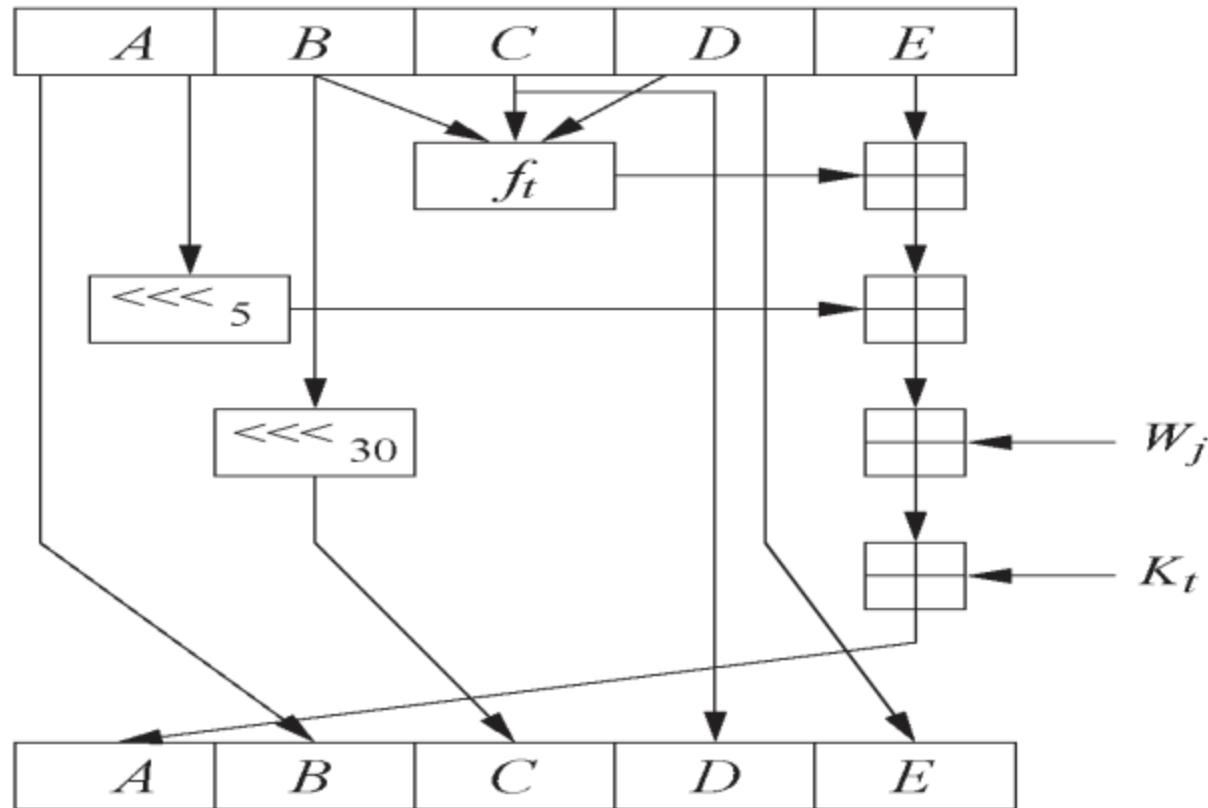
SHA-I

- Preprocessing (Padding) of the message.
- Fiestel block cipher, based on Merkel-Damgard construction.
- Message block acts as a key and input is the previous hash value (H_{i-1}).
- Message is divided into 512 bit blocks.
- Each 512-bit blocks are further subdivided into 16 words of size of 32 bits.

SHA-1 (Round Function)

$$A, B, C, D, E = (E + f_t(B, C, D) + (A)_{\ll 5} + W_j + K_t), A, (B)_{\ll 30}, C, D$$

SHA-1 (Round Function)



$$A, B, C, D, E = (E + f_t(B, C, D) + (A) \lll 5 + W_j + K_t), A, (B) \lll 30, C, D$$

SHA-1 (Round Function)

- Initial H_0 :

$$A = H_0^{(0)} = 67452301$$

$$B = H_0^{(1)} = \text{EFCDAB89}$$

$$C = H_0^{(2)} = 98\text{BADCFE}$$

$$D = H_0^{(3)} = 10325476$$

$$E = H_0^{(4)} = \text{C3D2E1F0}.$$

Stage t	Round j	Constant K_t	Function f_t
1	0 ... 19	$K_1 = 5\text{A827999}$	$f_1(B,C,D) = (B \wedge C) \vee (\bar{B} \wedge D)$
2	20 ... 39	$K_2 = 6\text{ED9EBA1}$	$f_2(B,C,D) = B \oplus C \oplus D$
3	40 ... 59	$K_3 = 8\text{F1BBCDC}$	$f_3(B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
4	60 ... 79	$K_4 = \text{CA62C1D6}$	$f_4(B,C,D) = B \oplus C \oplus D$

SHA-1 (Message Schedule)

- 32-bit 80 words W_j are computed from the 512-bit message block for each round as follows:

$$W_j = \begin{cases} x_i^{(j)} & 0 \leq j \leq 15 \\ (W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3}) \lll 1 & 16 \leq j \leq 79, \end{cases}$$

SHA-1 (Message Schedule)

- 32-bit 80 words W_j are computed from the 512-bit message block for each round as follows:

$$W_j = \begin{cases} x_i^{(j)} & 0 \leq j \leq 15 \\ (W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3}) \lll 1 & 16 \leq j \leq 79, \end{cases}$$

Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

Today's Agenda

- Message Authentication code (MAC)
- MAC's from hash function (HMAC)

NS Goals

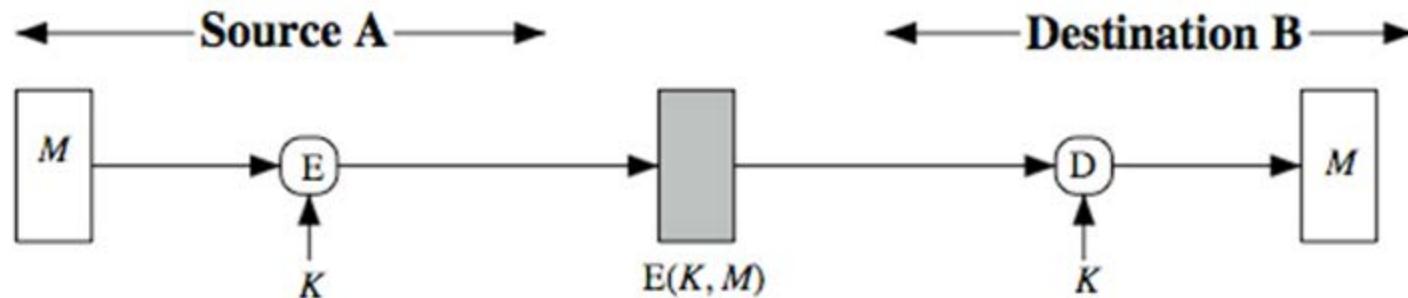
- Information security is an important area of information technology and this course on Network Security help audience to understand the three important security goals in the networks –
 - Confidentiality
 - Integrity
 - Availability
 - Also Authenticity and Non-repudiationand cryptographic techniques to implement these security goals.

Message Authentication

- Message authentication is concerned with:
 - validating identity of originator
- Two Solution
 - message encryption using Block Cipher
 - message authentication code (MAC)

Symmetric Message Encryption

- Encryption can also provides authentication



Properties of MAC

Message Authentication Code

Principle:

- MAC is realized with cryptographic hash functions
 - (e.g., SHA-1) SHA-1 HMAC (O/P 160 bits)
 - (e.g., SHA-256) SHA-256 HMAC (O/P 256 bits)
- HMAC is such a MAC built from hash functions
- Basic idea: Key is hashed together with the message

- Secret prefix MAC: $m = \text{MAC}_k(x) = h(k||x)$.
- Secret suffix MAC: $m = \text{MAC}_k(x) = h(x||k)$.
- Have strong one-wayness, hence strong checksum.
- Weakness in secret prefix and suffix MACs.



Secret Prefix MAC

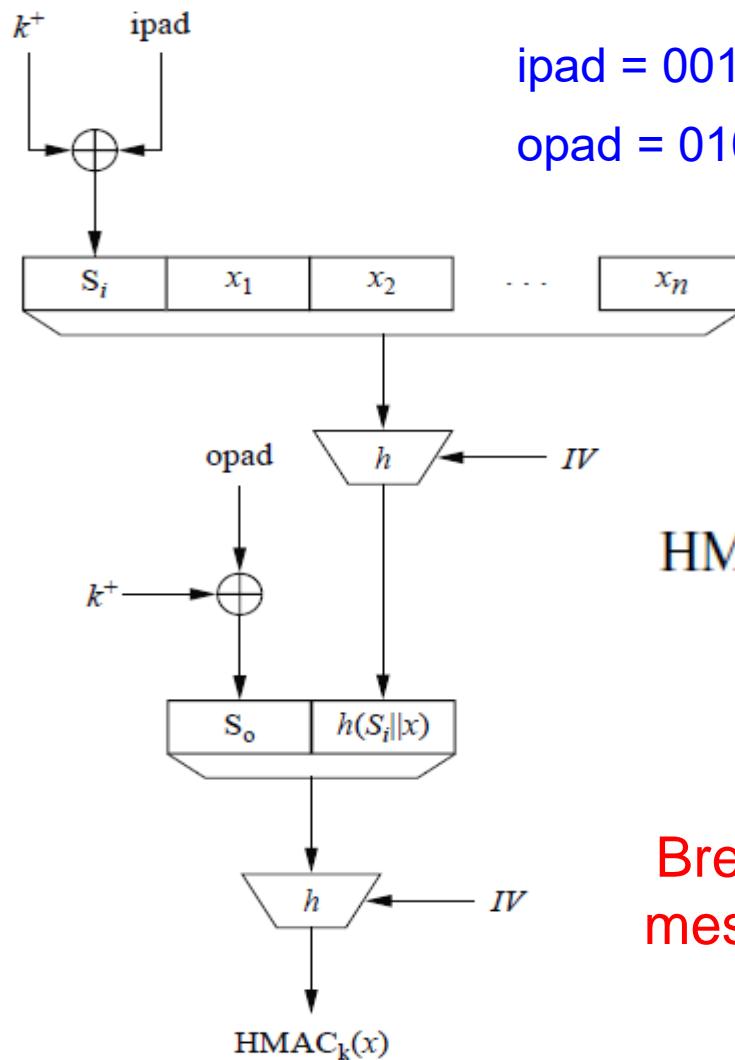


Secret Suffix MAC



HMAC

HMAC



ipad = 0011 0110, 0011 0110, , 0011 0110, 0011 0110
 opad = 0101 1100, 0101 1100, , 0101 1100, 0101 1100

$$\text{HMAC}_k(x) = h [(k^+ \oplus \text{opad}) || h [(k^+ \oplus \text{ipad}) || x]]$$

Breaking the HMAC means constructing valid message tag without knowledge of key.

Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

NS (Encryption)

Symmetric Cryptography



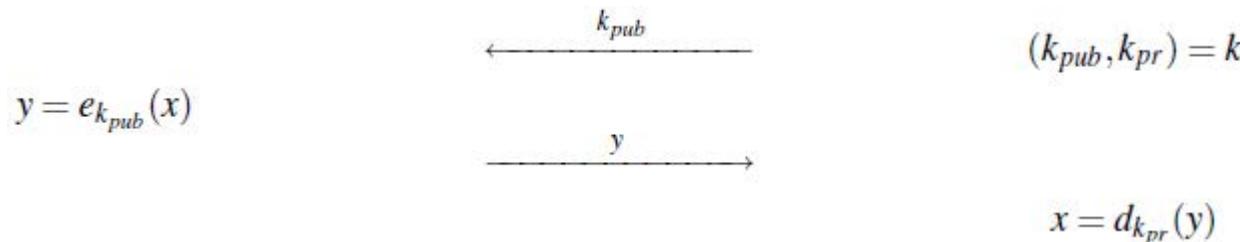
Asymmetric Cryptography

Need of Asymmetric Cryptography

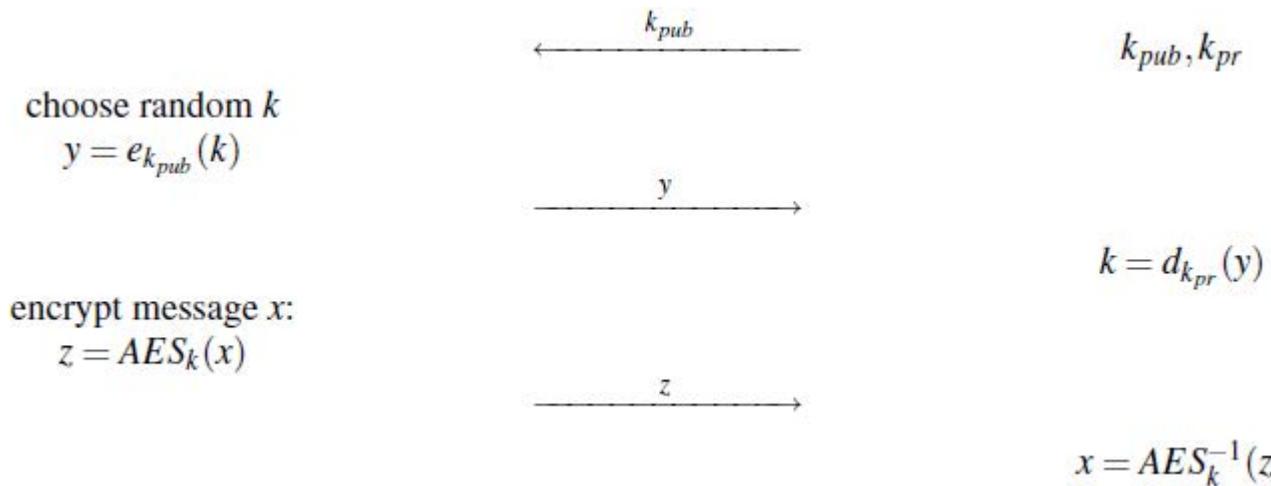
- Symmetric Cryptography: Same key, encryption and decryption are similar, efficient and very secure.
- Short comings of Symmetric Cryptography: Key distribution problem, Number of keys $n(n-1)/2$, single point failure, No protection against cheating (non-repudiation).
- Asymmetric cryptography based on number theory was introduced by Whitfield Diffie, Martin E. Hellman and Ralph Mebleam.

Principle of PKC

- Encryption key is not required to keep secret, only decryption key has to keep secret.



- Protocol can be modified for the key exchange.



Principle of PKC

- Public key algorithms can be built from the concept of one way function, defined as

A function $f()$ is a one-way function if:

1. $y = f(x)$ is computationally easy, and
2. $x = f^{-1}(y)$ is computationally infeasible.

- Popular one way function are integer factorization, DLP, EC methods.

- Eg : $n = p \times q$
 - Given p, q and performing ($n = p \times q$) is computationally easy
 - But, given n , computing p and q is computationally infeasible

Security Mechanism of PKC

- Encryption (RSA, DLP, ECC, etc.)
- Key establishment (DHKE, RSA, etc.)
- Message integrity and non-repudiation with digital signature.
- Identification using challenge-response and digital signature.
- Problem with PKC is efficiency, long key and authenticity of public key.

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

- Complexity grows roughly with cube of the bit size i.e. 1024 to 3072, will 27 times slower.

RSA

- Not a replacement of Symmetric Cipher.
- Designed by Ronald Rivest, Adi Shamir Leonared Adleman in 1997.
- Based on integer factorization.
- Encryption of small size data, key transport, digital signature, exchange of symmetric key.

Introduction

- **Encryption:**
- Given: k_{pub} , = e, n, message=x and encryption function

$$y = e_{k_{pub}}(x) \equiv x^e \pmod{n}$$

- **Decryption:**
- $x = d_{k_{pr}}(y) \equiv y^d \pmod{n}$

$$\in \mathbb{Z}_n$$

- n, e, d, x, y

RSA Requirements

- Computationally in-feasible to compute $K_{\text{pvt}} = d$ from $K_{\text{pub}} = n$ and e
- $x \leq n$.
- Efficient method to compute $x^e \bmod n$ and $y^d \bmod n$.
- Multiple pairs should exist for the given n .

Key Generation

How to generate n, e and d ?

1. Choose two large primes p and q .
2. Compute $n = p \cdot q$.
3. Compute $\Phi(n) = (p - 1)(q - 1)$.
4. Select the public exponent $e \in \{1, 2, \dots, \Phi(n) - 1\}$ such that

$$\gcd(e, \Phi(n)) = 1.$$

5. Compute the private key d such that

$$d \cdot e \equiv 1 \pmod{\Phi(n)}$$

RSA Example (Key Generation)

1. Choose two large primes p and q .
2. Compute $n = p \cdot q$.
3. Compute $\Phi(n) = (p - 1)(q - 1)$.
4. Select the public exponent $e \in \{1, 2, \dots, \Phi(n) - 1\}$ such that

$$\gcd(e, \Phi(n)) = 1.$$

5. Compute the private key d such that

$$d \cdot e \equiv 1 \pmod{\Phi(n)}$$



RSA Example (E and D)

Fast Exponentiation

Fast Exponentiation

Fast Exponentiation



GCD





GCD





GCD





GCD



Thanks!!!
Queries?



Network Security

SS ZG513

BITS Pilani
K K Birla Goa Campus

Hemant Rathore
Department of Computer Science and Information Systems

RSA Key Generation

1. Choose two large primes p and q .
2. Compute $n = p \cdot q$.
3. Compute $\Phi(n) = (p-1)(q-1)$.
4. Select the public exponent $e \in \{1, 2, \dots, \Phi(n)-1\}$ such that

$$\gcd(e, \Phi(n)) = 1.$$

5. Compute the private key d such that

$$d \cdot e \equiv 1 \pmod{\Phi(n)}$$

RSA Encryption / Decryption

- Encryption:

$$y = e_{k_{pub}}(x) \equiv x^e \pmod{n}$$

- Decryption:

$$x = d_{k_{pr}}(y) \equiv y^d \pmod{n}$$

- $n, e, d, x, y \in \mathbb{Z}_n$

Square and Multiply Algo

Square and Multiply Algo

Square and Multiply Algo

Algorithm: Square-and-Multiply for $x^H \bmod n$

Input: Exponent H , base element x , Modulus n

Output: $y = x^H \bmod n$

1. Determine binary representation $H = (h_t, h_{t-1}, \dots, h_0)_2$
2. **FOR** $i = t-1$ **TO** 0
3. $y = y^2 \bmod n$
4. **IF** $h_i = 1$ **THEN**
5. $y = y * x \bmod n$
6. **RETURN** y



GCD





GCD



Diffe-Hellman Key Exchange

- Based on the property that exponentiation is commutative.

$$k = (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

- Set-up protocol:
 - Choose a large prime p .
 - Choose integer α element of $\{2, 3, \dots, p-2\}$.
 - Publish p and α .



Diffe-Hellman Key Exchange



Diffe-Hellman Key Exchange

Diffe-Hellman Key Exchange

A

choose $a = k_{pr,A} \in \{2, \dots, p-2\}$
 compute $A = k_{pub,A} \equiv \alpha^a \pmod{p}$

$k_{pub,A}=A$

$k_{pub,B}=B$

$k_{AB} = k_{pub,B}^{k_{pr,A}} \equiv B^a \pmod{p}$

choose $b = k_{pr,B} \in \{2, \dots, p-2\}$
 compute $B = k_{pub,B} \equiv \alpha^b \pmod{p}$

$k_{AB} = k_{pub,A}^{k_{pr,B}} \equiv A^b \pmod{p}$

Thanks!!!
Queries?



BITS Pilani
KK Birla Goa Campus

Network Security

SS ZG513

S.K. Sahay
Department of Computer Science and Information Systems

Recent Malware Attacks (1)

Android security: Flaw in an audio codec left two-thirds of smartphones at risk of snooping, say researchers

Written by [Liam Tung](#), Contributor
on April 22, 2022 | Topic: Security

New Android malware steals millions after infecting 10M phones

By [Sergiu Gatlan](#)

September 29, 2021 10:45 AM 1

1,446,336 views | Aug 10, 2019, 06:38am

Google Warning: Tens Of Millions Of Android Phones Come Preloaded With Dangerous Malware

Google confirms some Android devices were infected with malware even before they shipped

Over 500,000 Huawei phones found infected with Joker malware

April 14, 2021

The Joker malware, known for tricking smartphone users into downloading fake apps and covertly registering them to premium-rate services, recently infiltrated over 500,000 Huawei phones via ten apps using which the malware communicated with a command-and-control server.

October 4, 2019 1:28 PM IST

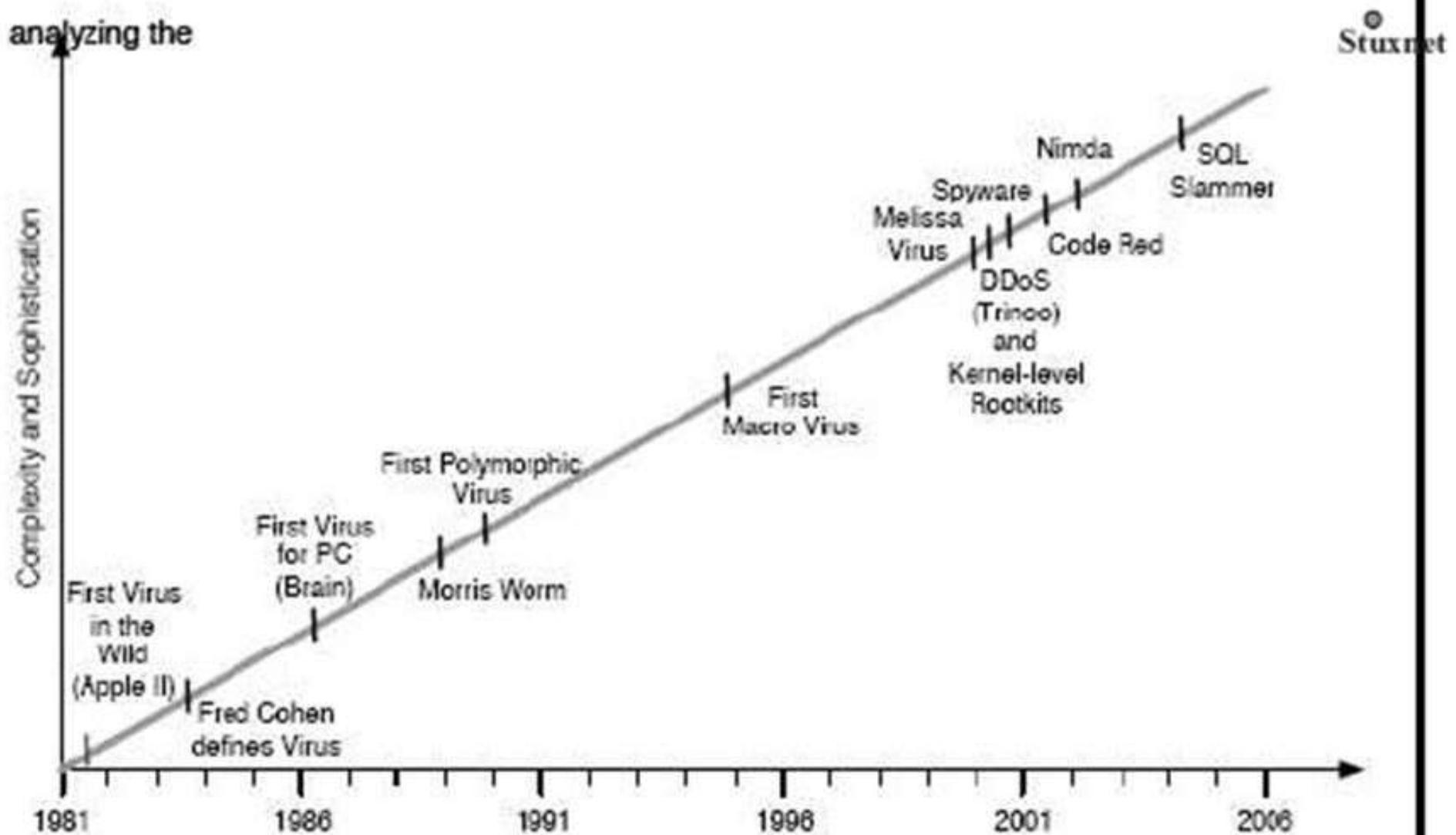
172 malicious apps with 335M+ installs found on Google Play

Harmful app type	Number of apps	Number of installs
Adware	48	300,600,000+
Subscription Scam	15	20,000,000+
Hidden Ads	57	14,550,000+
SMS Premium Subscription	24	472,000+
Hidden App	7	310,000+
Banking Trojan	1	10,000+
Stalkware	1	10,000+
Fake Antivirus	1	10,000+
Credit Card Phishing	2	200+
Fake Cryptocurrency Exchanges	1	100+
Fake App	15	100+
sum	172	335,962,400+

Malicious Software

- **Malware:** A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim. --- NIST
- **Some Actions:** Gathering sensitive information, gaining access to private data, deleting files, etc.
- **Propagation:** Viruses, Worms, Social Engineering.
- **Payload Actions:** System corruption, Zombies and bots, Information theft, stealthing.
- **Countermeasures:** Anti-Malware.

History of Malware



Classification of Malwares

- Basic Malware
- Encrypted Malware
- Oligormorphic Malware
- Polymorphic Malware
- Metamorphic Malware

Static Malware (1)

Introduction

- **Virus** is a piece of code that attaches itself to host (benign) program and is activated by host program. [Spafford 1989]
- **Worm** is a standalone malicious program which can run independently. [Spafford 1989]
- **Trojan** is a software program that pretends to be useful but performs malicious actions in the background with user's knowledge or consent. [Schultz et al. 2001]
- **Bots** is a malicious application that allows the bot-master to remotely control the infected system. [Stinson and Mitchell 2007]

Static Malware (2)

Introduction

- **Spyware** is a type of malicious program that spies on user activities without the users' knowledge or consent. [Borders and Prakash 2004]
- Spam-ware, Adware are on same lines.
- **Ransomware** installs covertly on a victim's computer and executes a crypto-virology attack that adversely affects it.
- Hybrid malware combines two or more other forms of malicious codes into a new type to achieve more powerful attack functionalities.

Obfuscation Techniques

Introduction

- Garbage Code Inserter
 - “nop” instruction
- Equivalent Instruction Inserter
 - Register assignment -> Replace all EAX to EBX
- Jump Instruction Inserter
 - abc jmp1 jmp2 xyz jmp1 nop nop nop jmp2
- Control / Data Flow Permutation
 - Subroutine Reordering, 10 routines all does same thing, randomly change the order while spreading / execution
- Compression
 - while storing the malware, compress it

Malware Examples

Introduction

- Worm: Android.Obad.OS
- Trojan: FakeNetflix, Fakeplayer
- Backdoor: Basebridge, Kmin
- Botnet: Geinimi, Beanbot
- Spyware: Nickspy, GPSSpy
- Adware: Plankton
- Ransomware: FakeDefender.B

Source: <http://openaccess.city.ac.uk/12200/1/comsecreview%28Ra%29.pdf>

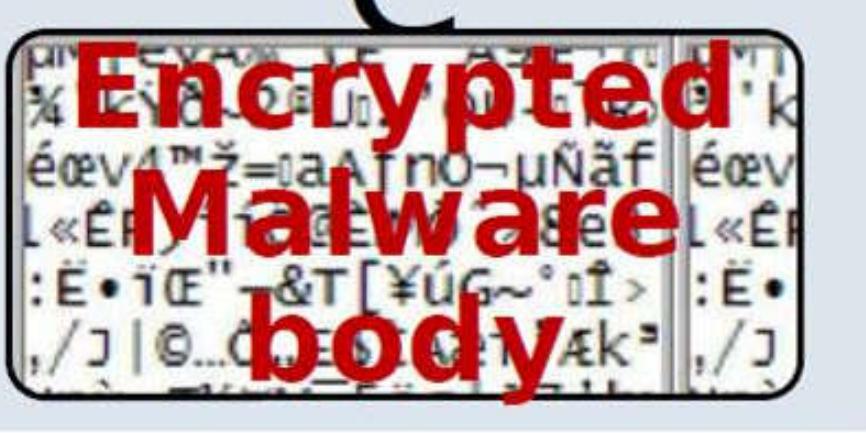
Virus Phases

- **Dormant Phase:** Idle, however eventually will be activated by some action viz. Date, HDD space, presence of some program etc..
- **Propagation Phase:** The virus places identical copy of itself into other program or into certain system areas on the disk.
- **Triggering Phase:** The virus is activated to perform the function for which was intended, can be caused by variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
- **Execution Phase:** The function is performed, may be harmless such as a message on the screen, or damaging, such as the destruction of programs and data files.

Encrypted Malware

Decryptor

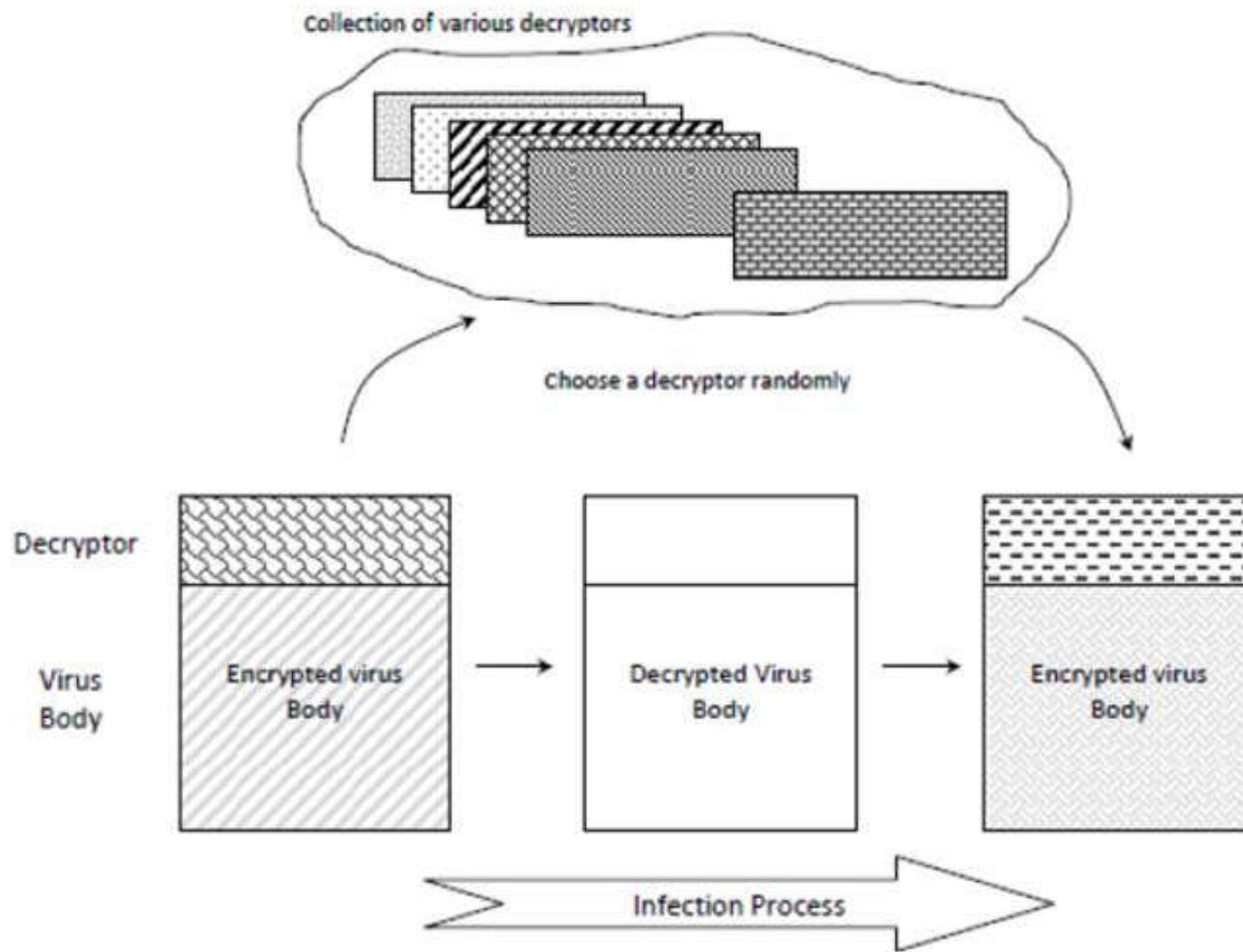
Encrypted
Malware
body



Decryptor

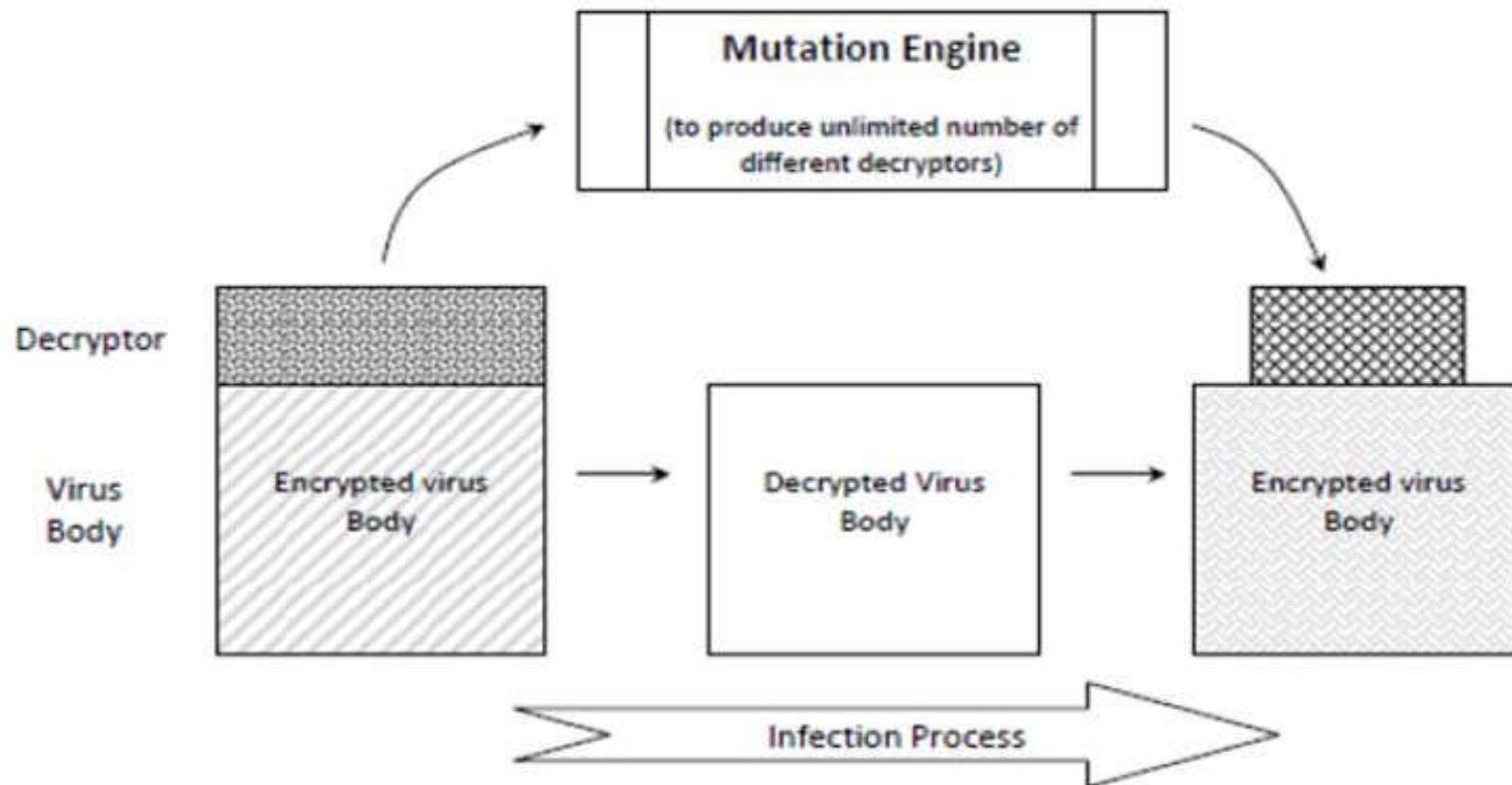
Decrypted
malware
body

Oligomorphic Malware



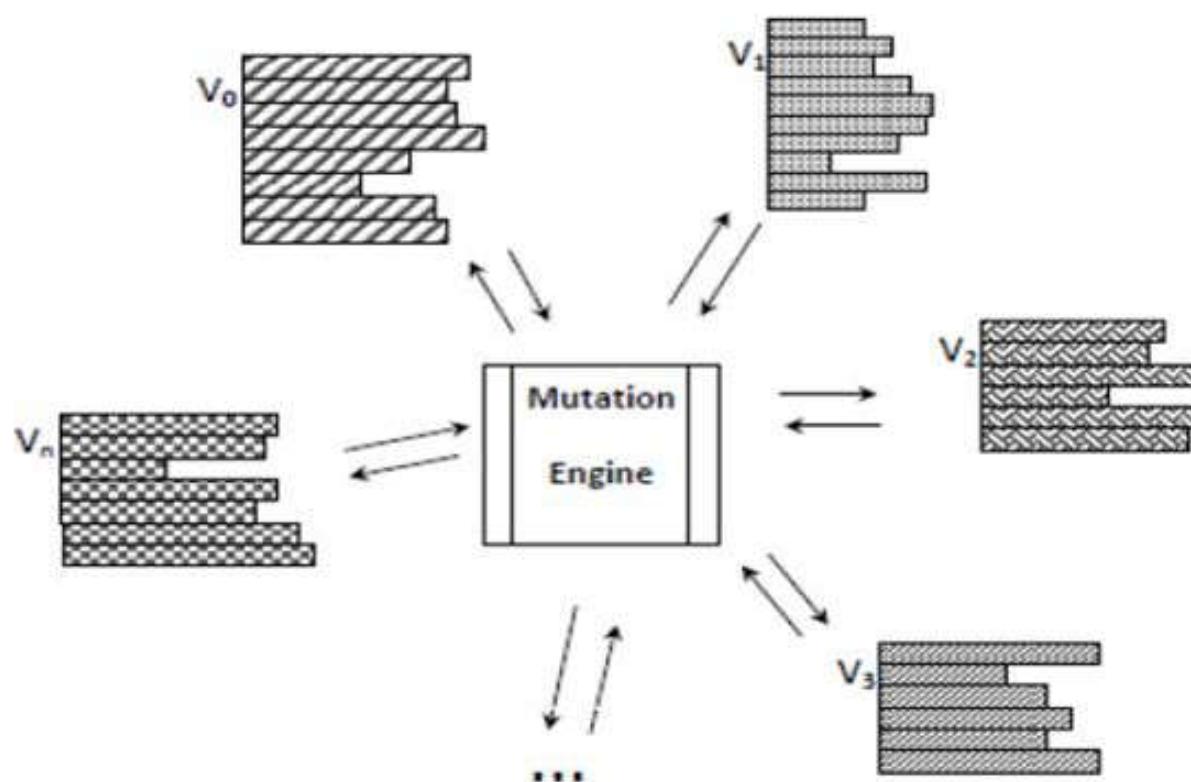
Polymorphic Malware

Contain a mutation engine which changes the decryptor randomly.



Metamorphic Malware

- Metamorphism is the process that can create an entirely new variant of a code which do same task after reproduction but is no-way similar to the original variant.
- There are a variety of code obfuscation techniques namely Garbage Code Insertion, Register Renaming, Subroutine Permutation, Code reordering and Equivalent code substitution.



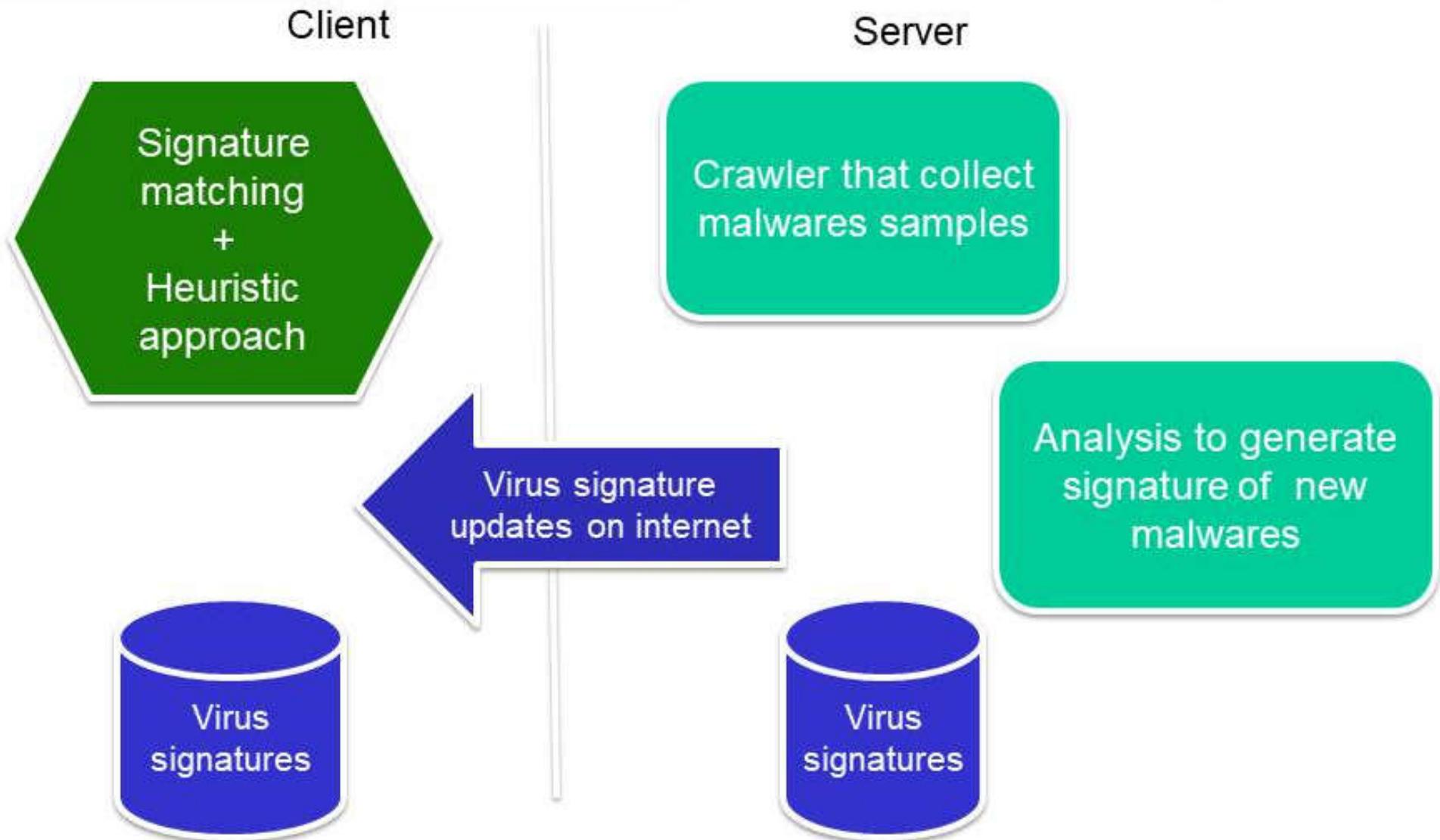
Signature-based Malware Detection Technique

Detection System

- Signature-based method identifies unique strings from the binary code. [Moskovich et al. 2009]
- Signatures are often manually generated, disseminated, and maintained by domain experts.
- Typical time window between a malware's release and its detection by anti-malware software is about 54 days. 15% of samples are still undetected even after 180 days. [Hu 2011]
- Malware can easily bypass signature-based identification by changing small pieces of its code without affecting the semantics. [Rastogi et al. 2013]
- **Reactive Approach** (no savior against zero day attack)

Traditional Detection System

Detection System



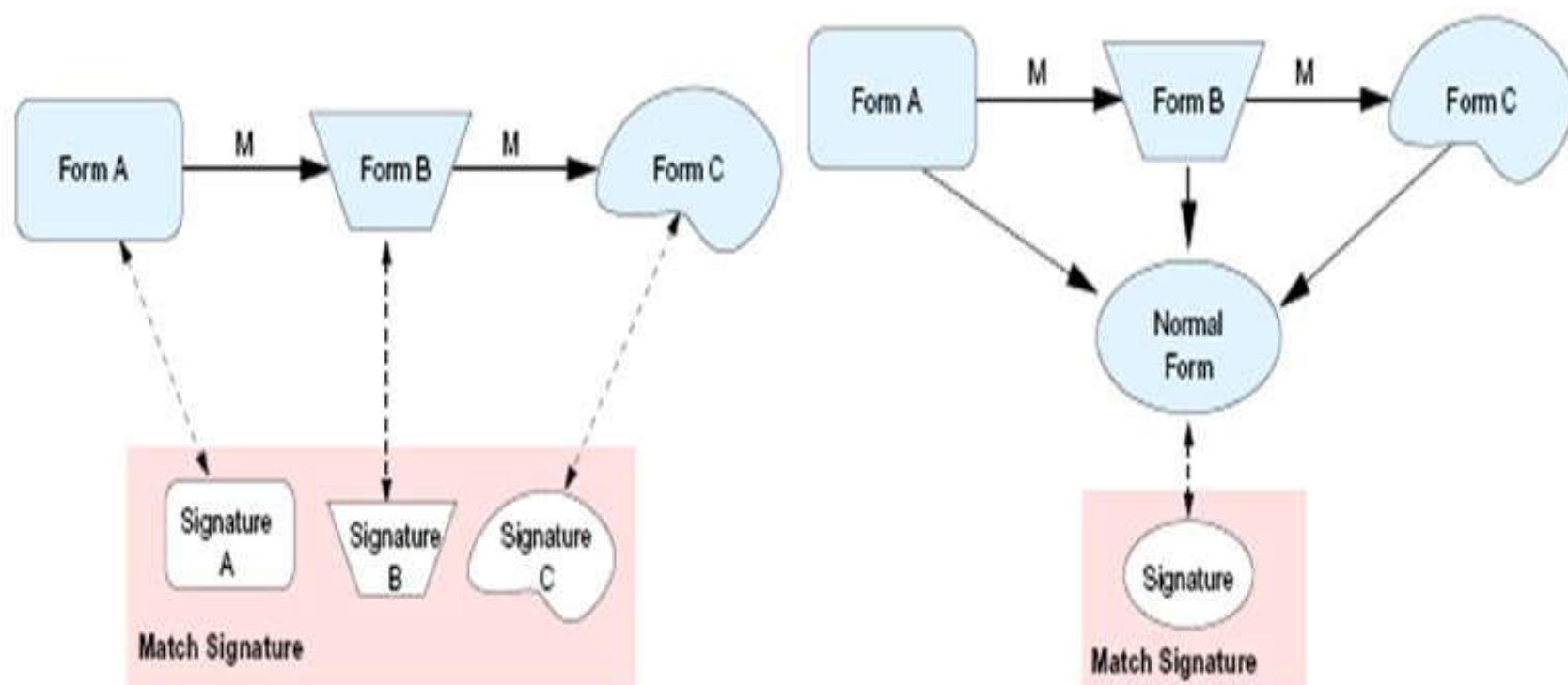
Heuristic-based Malware Detection Technique

Detection System

- Encrypted, polymorphic, and metamorphic malware can easily bypass the signature-based detection.
- So domain experts make **rules/patterns** to discriminate malware and benign files.
- Rules/patterns should be **generic** to detect variants of the same malware family, but not benign files. [Egele et al. 2012]
- Automated malware development toolkits like **Zeus** can mutate thousands of malicious codes per day. [TrendMicro 2010]
- Can be a **Proactive Approach** but need domain knowledge.

Detection Techniques (Countermeasures)

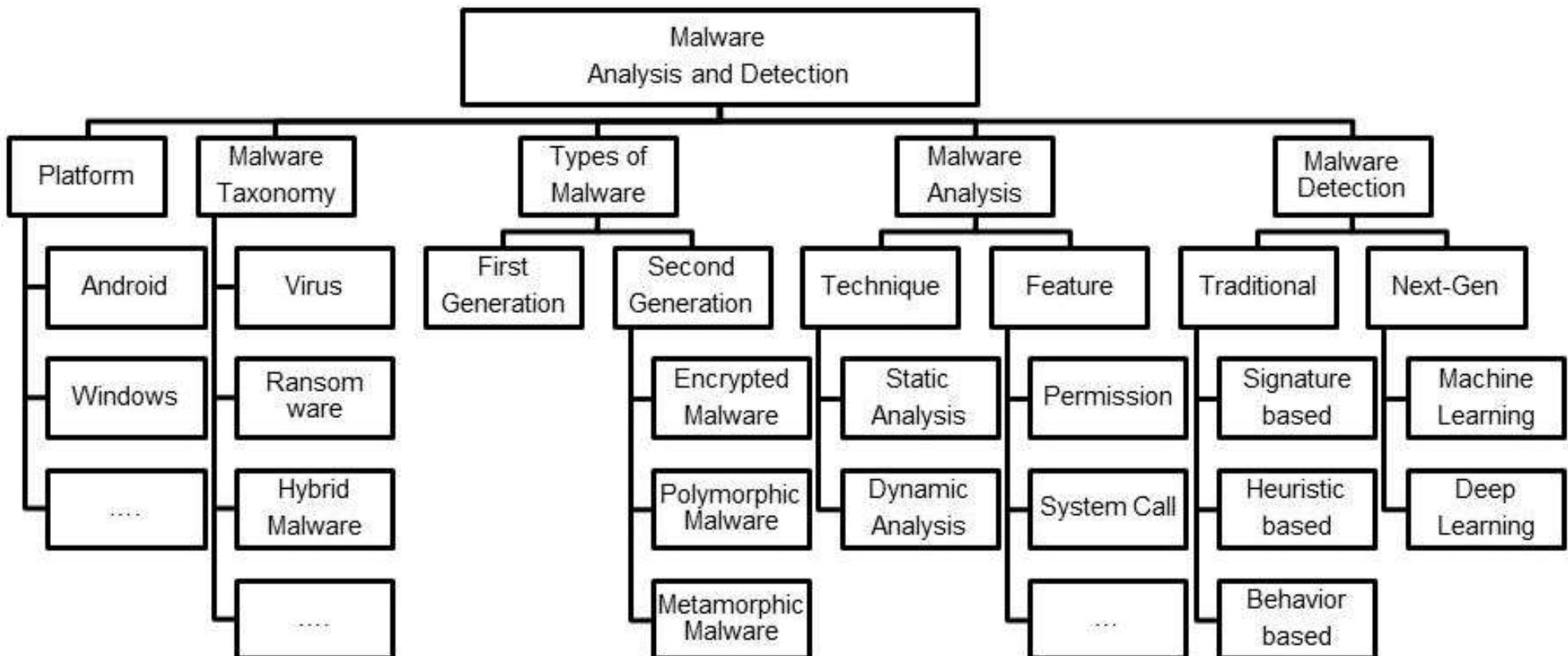
- Malware Normalization Technique:
 - In this technique system takes an obfuscated executable, undoes the obfuscation and outputs a normalized executable.



Detection Techniques (Countermeasures)

- Machine Learning:
 - The study of computers algorithm that is improved through past malware analysis and observation.
 - However its not a deterministic method to identify the file is a virus or not.
 - Not effective for the malwares embedded in the programs.

Literature Review



Malware Detection Motivation

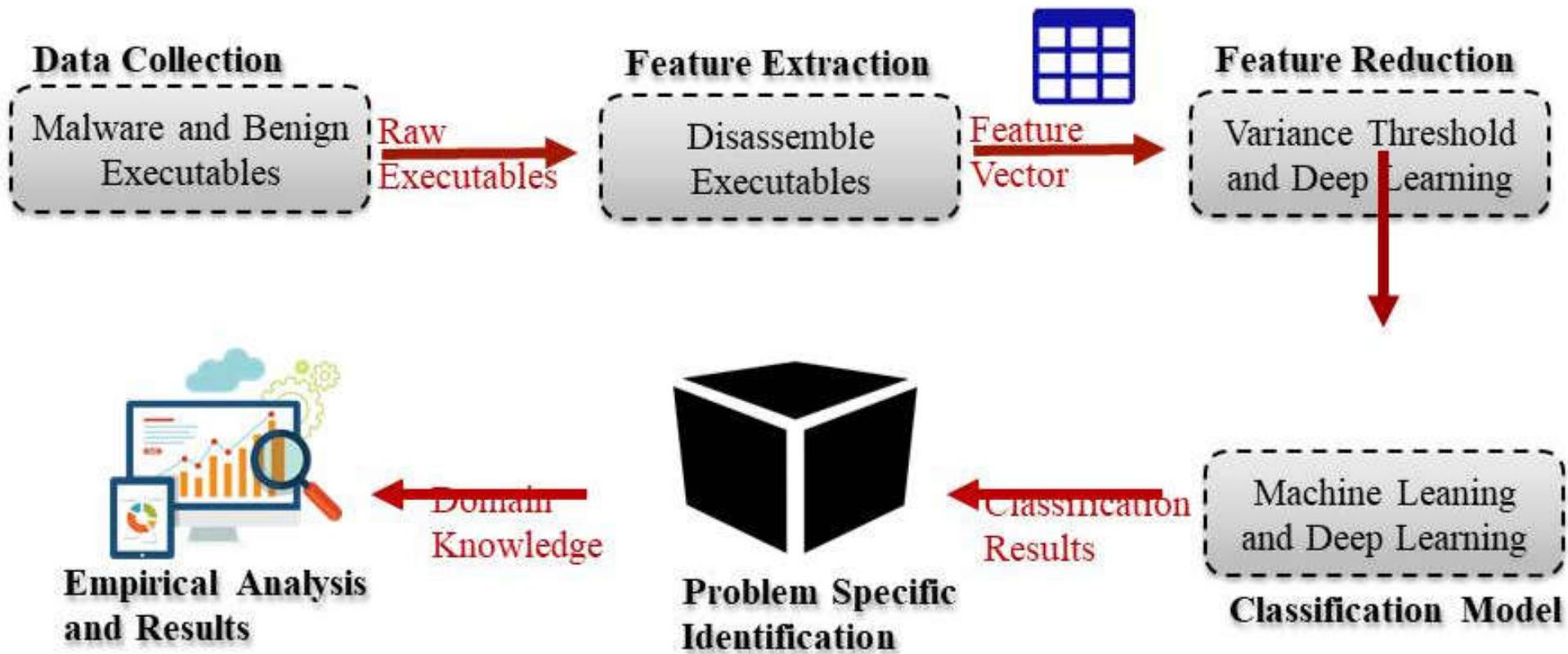
Research Motivation

- It is impossible to develop a generic algorithm to detect all possible malware [Cohen, 1987]
- Signature matching approach are not capable to detect continuously growing zero-day malware attack.
- Rat race
 - Malware designers:
 - more .. more malware
 - complex/sophisticated malware
 - AV designers: to prevent against both old and new malware attack (especially zero day attacks)

Proposed Framework @ Malware Detection Models



Proposed Solution

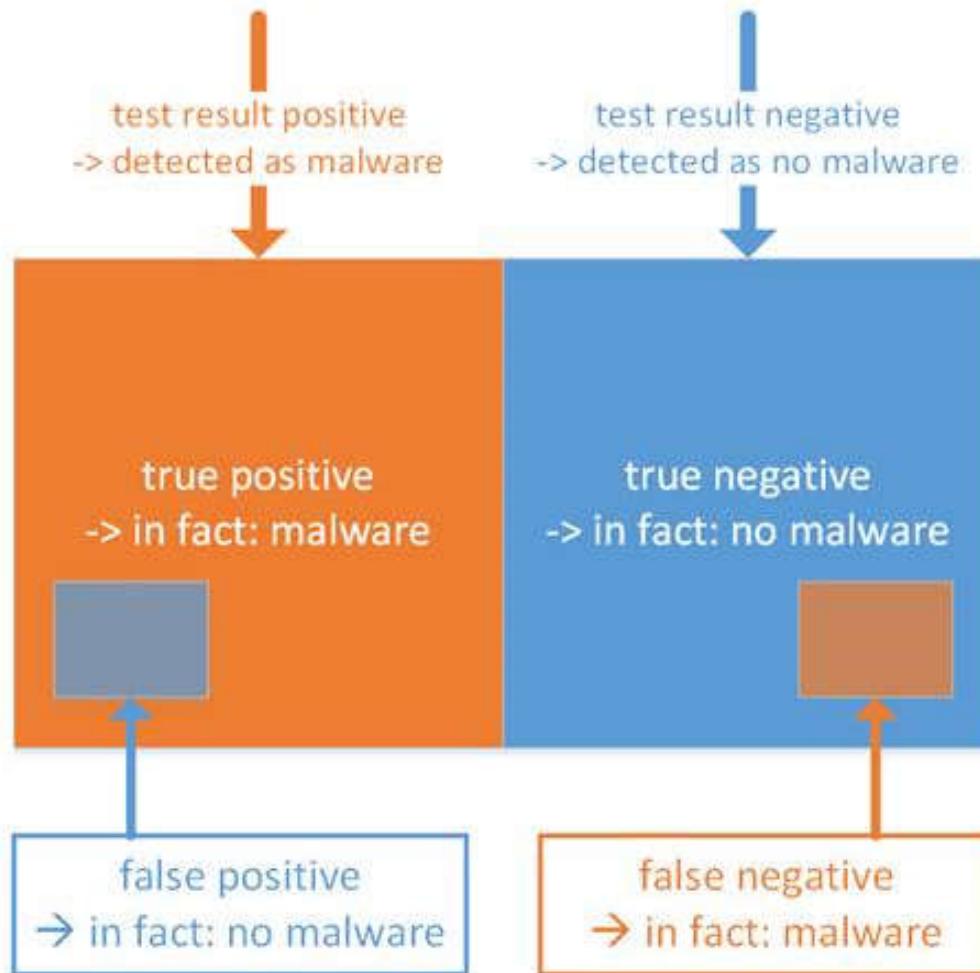


TP, FP, FN

Introduction

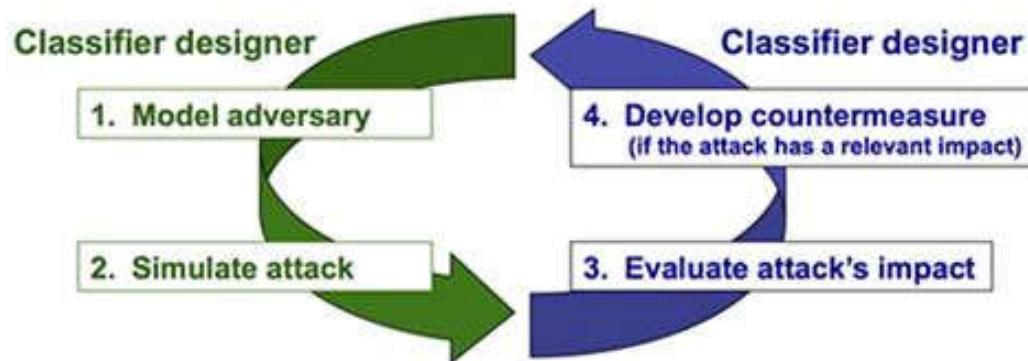
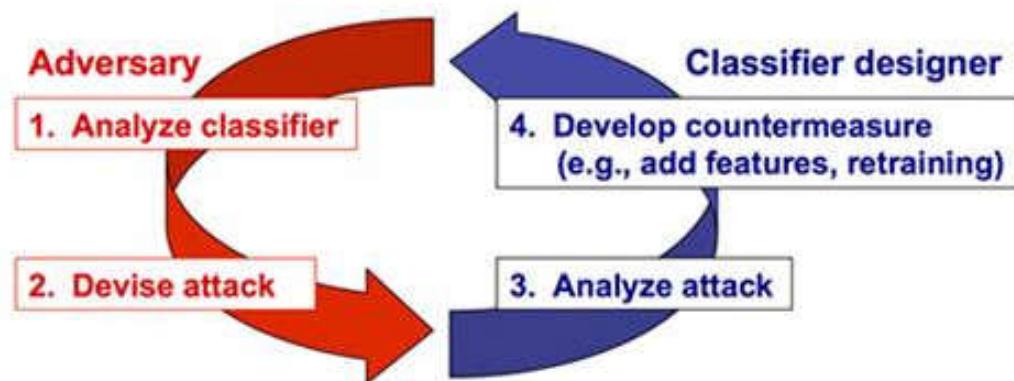
- **True Positive**
 - correct situation
 - if real malware was detected as malware
- **False Positive**
 - if the test of malware was positive
 - detected malware but the real file is NOT a malware.
 - i.e. the (positive) test result was false.
- **True Negative**
 - correct situation -> “benign was detected as benign”
 - “no malware” was detected as “no malware”.
- **False Negative**
 - like a “Missed SPAM”
 - “malware” came in but was not recognized as that.

Example: Malware Test



Rat Race

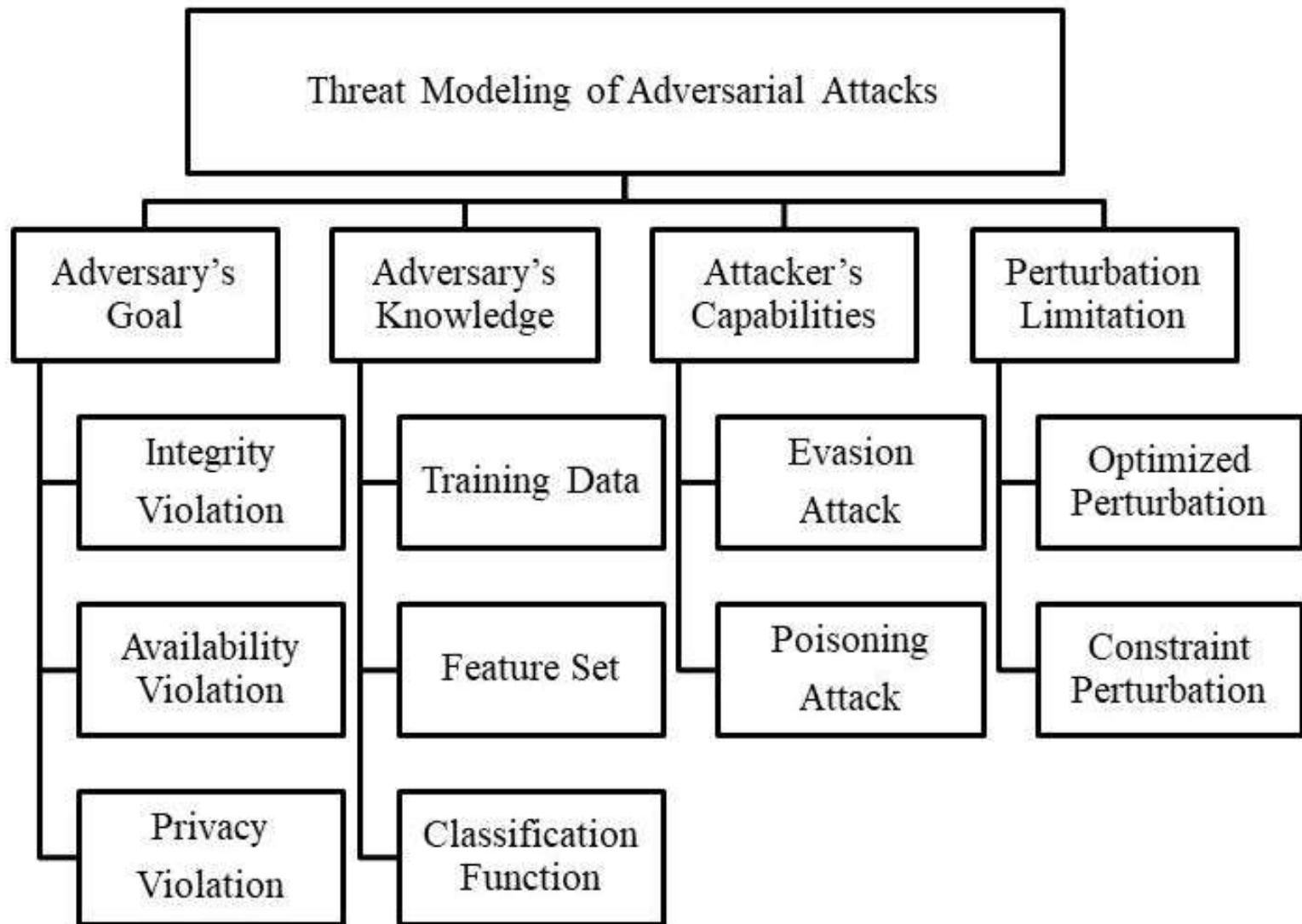
Proposed Solution



https://en.wikipedia.org/wiki/Adversarial_machine_learning

Threat Modeling

Proposed Solution





Thanks!!!
Queries?



SS ZG513

Network Security

Reference Model – Asymmetric (Public) Key Cryptography

Revision 1.0

BITS Pilani

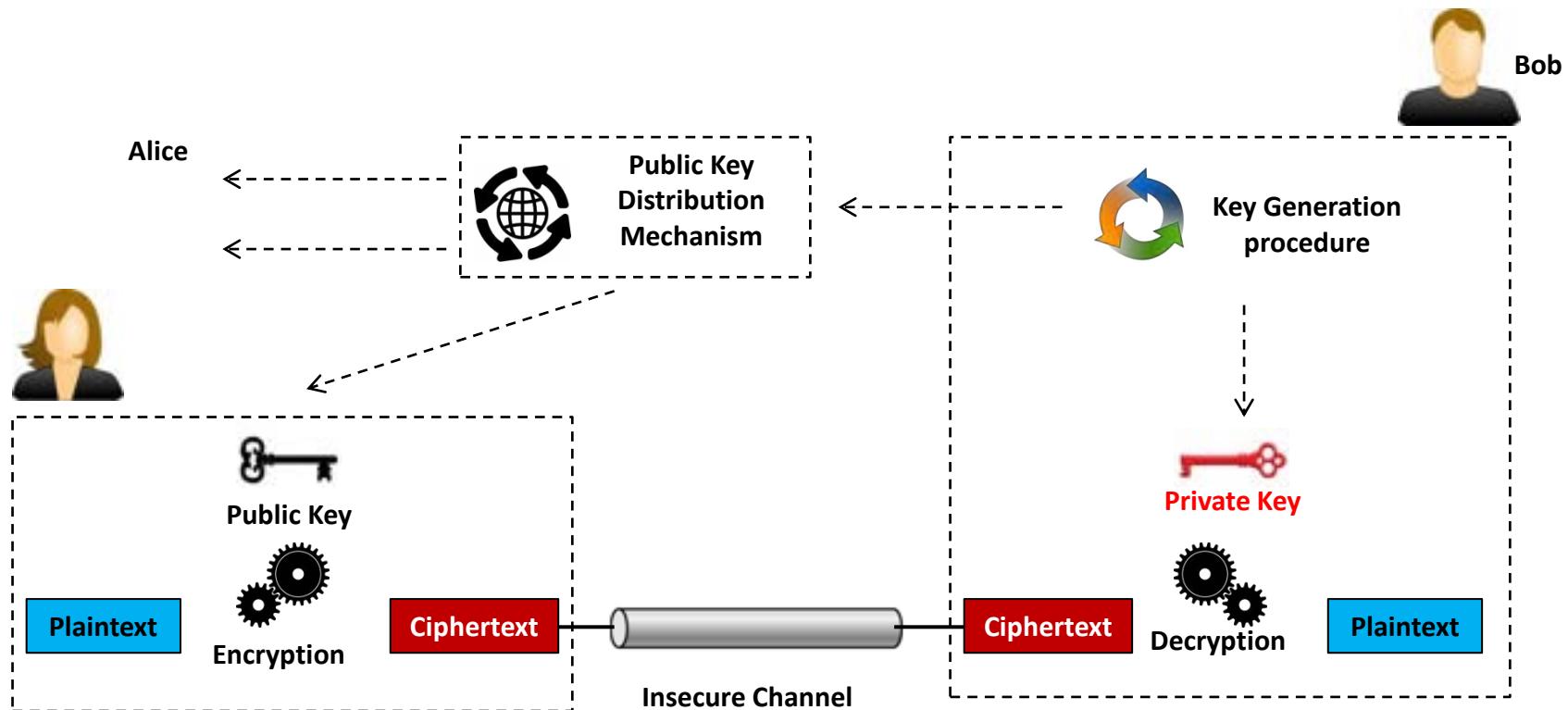
Work Integrated Learning Programmes



Asymmetric Key Cryptography



A Reference Network Model





Thank You



SS ZG513

Network Security

Public Key Cryptosystems - RSA

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



The RSA Algorithm



Introduction

- ❑ Developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT.
- ❑ The trio later founded a company RSA Security Inc in 1982 which is acquired by EMC Corporation in 2006.
- ❑ It is considered as the most widely accepted and implemented general-purpose approach to public-key encryption.
- ❑ It uses the Euler's Totient Function and Euler's Theorem.



Ron Rivest



Adi Shamir



Len Adleman

The RSA Algorithm



Formulation

- Plaintext is encrypted in blocks. Each plaintext and ciphertext block is treated as integers.
 - E.g. if the plaintext is ABC it is treated as 000102. That is, each character in two digit integers ranging from 00 to 25 for English language.
- If there is some plaintext block P and its corresponding ciphertext block is C then,

$$C = P^e \text{ mod } n$$

$$P = C^d \text{ mod } n, \text{ or}$$

$$P = (P^e \text{ mod } n)^d \text{ mod } n = P^{ed} \text{ mod } n$$

$$\text{E.g. } (2^3 \text{ mod } 5)^2 \text{ mod } 5 = 2^6 \text{ mod } 5 = 4$$

Where, both sender and receiver know the value of n , and
the sender knows the value of e , and
only the receiver knows the value of d

- This is a public-key (or asymmetric-key) encryption algorithm where **public-key** is PU = {e, n} and **private key** PR = {d, n}

The RSA Algorithm



Mathematics

- $P = P^{ed} \text{ mod } n$, for this equation to be true, e and d are selected from $Z_{\phi(n)}^*$ in such a way so that e and d are multiplicative inverse of each other in modulo $\phi(n)$:

$$ed \equiv 1 \pmod{\phi(n)},$$

$$\text{or, } d \equiv e^{-1} \pmod{\phi(n)}$$

- In other terms, d (or e) is relatively prime to $\phi(n)$. That is $\gcd(\phi(n), d) = 1$ or $\gcd(\phi(n), e) = 1$

The RSA Algorithm



Procedure

1. Select two distinct prime numbers p and q
2. Calculate $n = p \times q$
3. Calculate $\phi(n) = (p-1)(q-1)$
4. Select integer e such that $\gcd(\phi(n), e) = 1$
5. Calculate d such that $d \equiv e^{-1} \pmod{\phi(n)}$
6. Public Key (PU) = $\{e, n\}$
7. Private Key (PR) = $\{d, n\}$
8. For encryption, ciphertext $C = P^e \pmod{n}$
9. For decryption, plaintext $P = C^d \pmod{n}$

Note: All plaintext blocks P need to be selected so that $P < n$.

The RSA Algorithm



Example

- Let us say $p = 7$ and $q = 11$, the two distinct prime numbers.
- $n = p \times q = 77$
- $\phi(n) = (7 - 1)(11 - 1) = 60$
- Two values (e, d) are to be chosen from Z_{60}^* which are multiplicative inverse of each other.
 - $e = 13$
 - $d = 37$
- Let us say sender wants to send 'F' which can be represented as 05 as an integer.
- Sender will encrypt it as $P^e \bmod n = (05)^{13} \bmod 77 = 26$
- Receiver will decrypt it as $C^d \bmod n = (26)^{37} \bmod 77 = 5$



Thank You



SS ZG513

Network Security

Public Key Cryptosystems - ElGamal

Revision 1.0

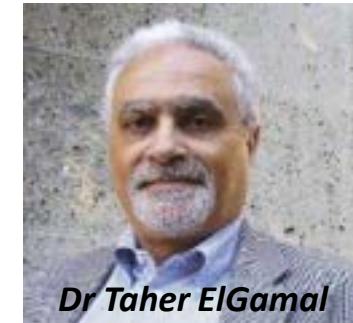
BITS Pilani

Work Integrated Learning Programmes



ElGamal Cryptographic System

- ElGamal Cryptosystems was developed in 1985 by *Dr. Taher ElGamal*, during his HP tenure. He is the present security CTO of salesforce.com.
- It is a public-key scheme based on discrete logarithms.
- The ElGamal Cryptosystem is used in many network security implementations like Digital Signature Standard (DSS) and the S/MIME e-mail.



ElGamal Cryptographic System



Formulation

- Receiver of the message (**Key Generation**):
 - Selects a large prime number p and e_1 a primitive root of it.
 - Generates a random number d such that $1 < d \leq p-1$
 - Calculates $e_2 = e_1^d \text{ mod } p$
 - The private key of the receiver $PR = \{p, e_1, e_2, d\}$
 - The public key for the sender is $PU = \{p, e_1, e_2\}$
- Sender of the message (**Encryption**):
 - Represents the message as an integer M in the range $0 \leq M \leq (p - 1)$
 - Longer messages are sent as a sequence of blocks, with each block being an integer less than p .
 - Generates a random number r such that $1 < r \leq p-1$
 - Calculates two ciphertexts $C1$ and $C2$ as:
 - i. $C1 = e_1^r \text{ mod } p$
 - ii. $C2 = (M \times e_2^r) \text{ mod } p$
 - Sends $C1$ and $C2$ to the receiver.
- Receiver of the message (**Decryption**):
 - Restores the plaintext as $M = [C2 \times (C1^d)^{-1}] \text{ mod } p$

ElGamal Cryptographic System



Example

- Receiver selects a prime number $p = 19$
- Primitive roots[#] of 19 = {2, 3, 10, 13, 14, 15}
- Receiver selects $e_1 = 10$ (one of the primitive roots) and a random number $d = 5$
- Then, $e_2 = e_1^d \bmod p = 10^5 \bmod 19 = 3$
- The private key for the receiver PR = {19, 10, 3, 5}
- The public key for the sender PU = {19, 10, 3}
- The sender wants to send $M = 17$ and selects random number $r = 6$
- Sender then calculates C1 and C2 as below:

$$C1 = e_1^r \bmod p = 10^6 \bmod 19 = 11$$

$$C2 = (M \times e_2^r) \bmod p = (17 \times 3^6) \bmod 19 = 5$$

- The receiver decrypts C1 and C2 as:

$$\begin{aligned} M &= [C2 \times (C1^d)^{-1}] \bmod p \\ &= [5 \times (11^5)^{-1}] \bmod 19 \\ &= [5 \times 11] \bmod 19 \\ &= 17 \quad (\text{the original plaintext}) \end{aligned}$$

Working:

$$\begin{aligned} &(11^5)^{-1} \bmod 19 \\ &= (11^5 \bmod 19)^{-1} \bmod 19 \\ &= (7)^{-1} \bmod 19 \\ &= 11 \end{aligned}$$

Review primitive roots concepts from Mathematics for Asymmetric Key Cryptography session slides.



Thank You



SS ZG513

Network Security

Diffie-Hellman Key Exchange

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Diffie-Hellman Key Exchange



Formulation

- Developed by **W. Diffie** and **M. E. Hellman** professors of Stanford university in 1976.
- There are two publically known numbers: a prime number p and e a primitive root of it.
- Users A and B want to create a shared secret key.
- User A selects a random number $X_A < p$ and calculates $Y_A = e^{XA} \text{ mod } p$.
- Similarly, user B selects a random number $X_B < p$ and calculates $Y_B = e^{XB} \text{ mod } p$.
- Each side keeps the value of X secret and make the value of Y public to the other side. So X_A is the private key for A and Y_A is its corresponding public key. The same logic stands true for the values of X_B and Y_B for B.
- Now A computes a key K_A as:

$$K_A = Y_B^{XA} \text{ mod } p$$

- B also computes a key K_B as:

$$K_B = Y_A^{XB} \text{ mod } p$$

- Both the keys K_A and K_B are same as proved below:

$$\begin{aligned} K_A &= Y_B^{XA} \text{ mod } p \\ &= (e^{XB} \text{ mod } p)^{XA} \text{ mod } p \\ &= e^{XB.XA} \text{ mod } p \\ &= (e^{XA} \text{ mod } p)^{XB} \text{ mod } p \\ &= Y_A^{XB} \text{ mod } p \\ &= K_B \end{aligned}$$



W. Diffie



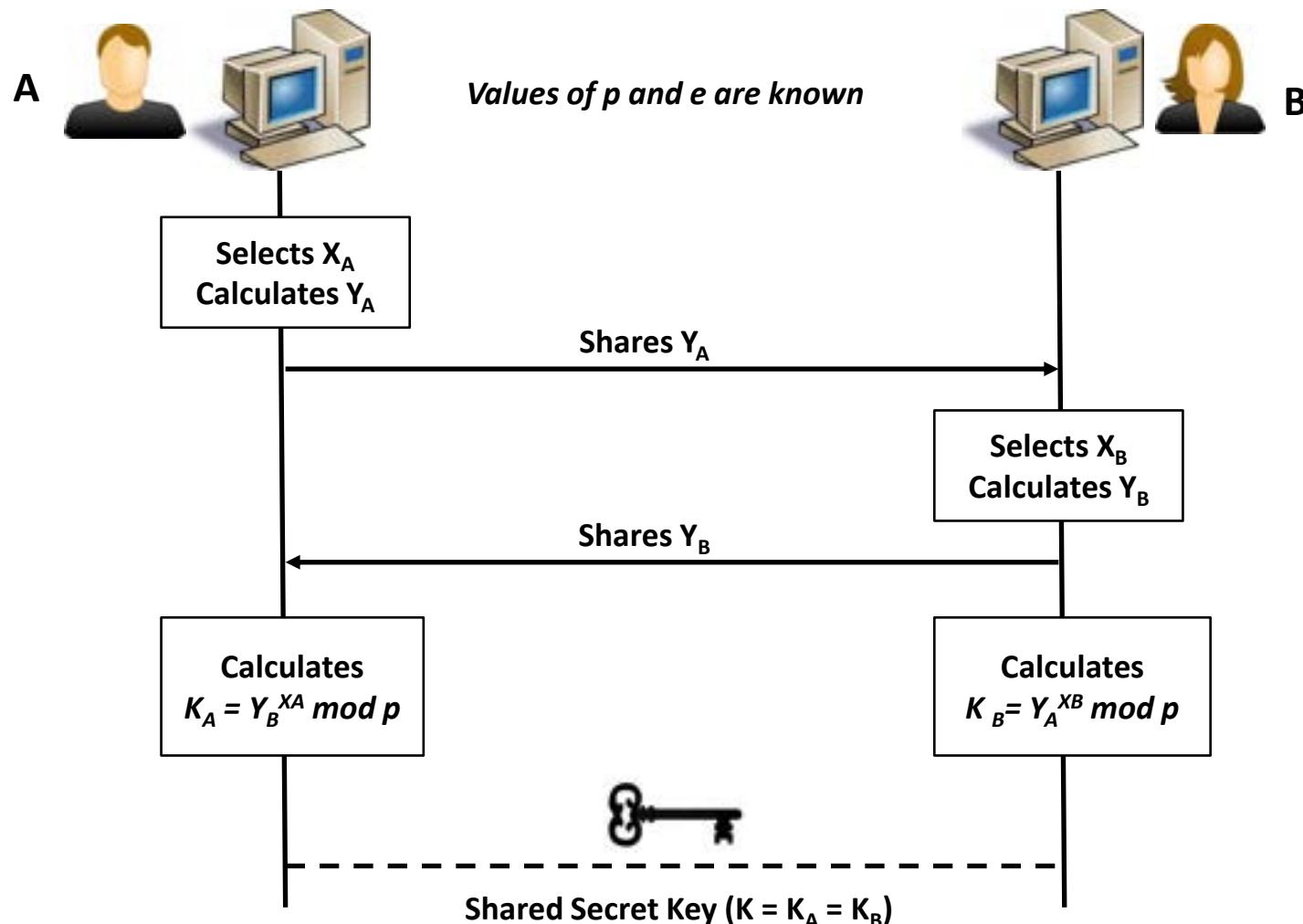
M. E. Hellman

- The key K ($= K_A = K_B$ is the shared secret key) which is generated without sharing the individual random numbers X_A and X_B and its value is $K = e^{XA.XB} \text{ mod } p$

Diffie-Hellman Key Exchange



Flow of Events



Diffie-Hellman Key Exchange



Example

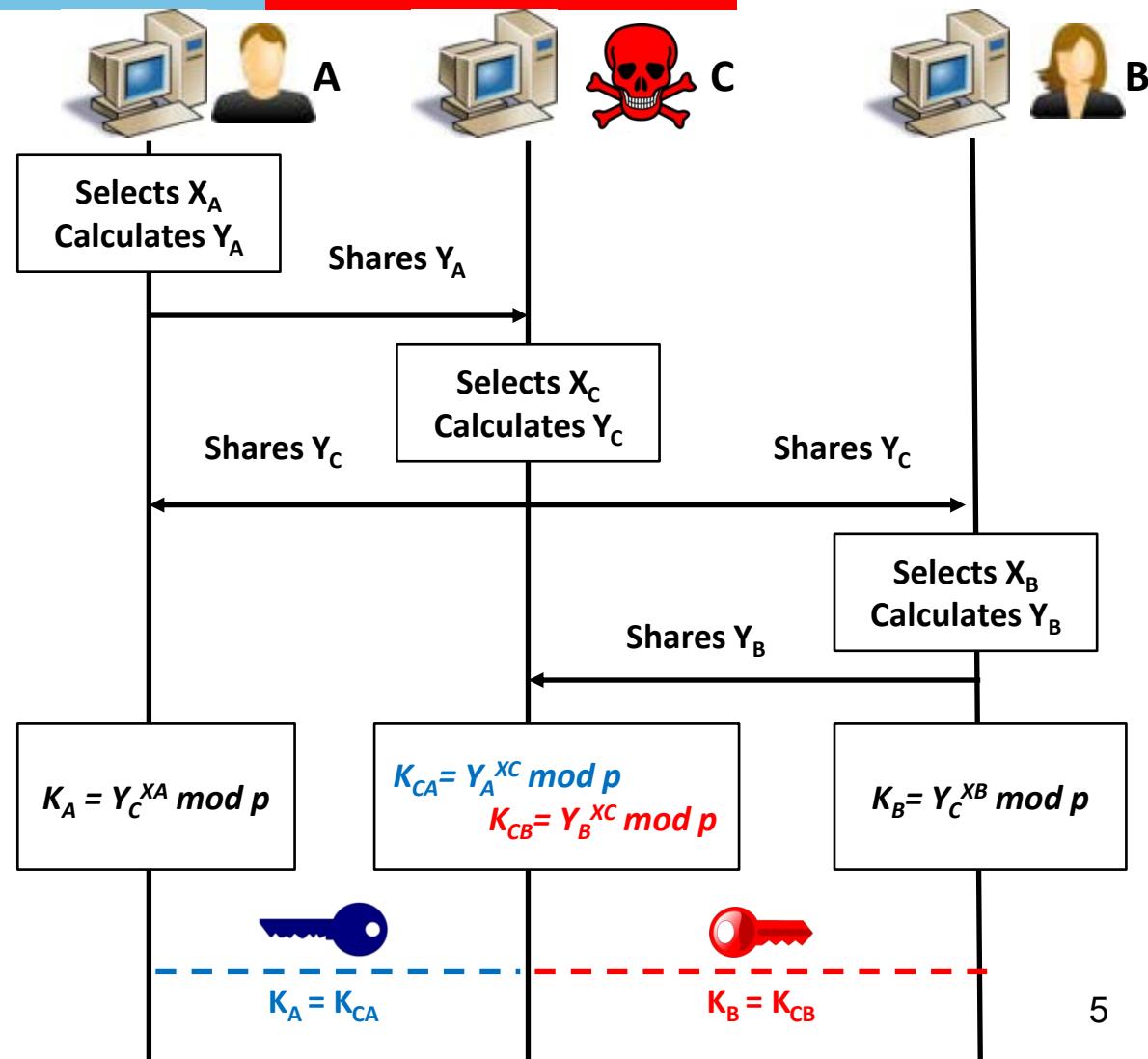
- Let us say $p = 23$ and one of its primitive roots $e = 7$.
- User A selects $X_A = 3$ and calculates $Y_A = e^{X_A} \bmod p = 7^3 \bmod 23 = 21$.
- User B selects $X_B = 6$ and calculates $Y_B = e^{X_B} \bmod p = 7^6 \bmod 23 = 4$.
- User A sends the number 21 to B.
- User B sends the number 4 to A.
- A calculates the shared secret as $K = Y_B^{X_A} \bmod p = 4^3 \bmod 23 = 18$.
- B calculates the shared secret as $K = Y_A^{X_B} \bmod p = 21^6 \bmod 23 = 18$

Both A and B calculate the same shared key (K) as 18.

Potential Attacks on the Diffie-Hellman Key Exchange



- There is an attacker C who poses a risk as man-in-the middle.
- Attacker intercepts Y_A from user A, calculates Y_C and shares it with both A and B.
- User B shares Y_B with C.
- A, B and C calculates shared keys and ***the system end up having two sets of shared keys*** – one between A and C and another between C and B.
- Attacker C is controlling the communication and legitimate users A and B and are not aware of this attack.
- It is also known as ***Bucket Brigade Attack*** (volunteers passing the buckets of water hop by hop). To avoid this attack, legitimate users can use ***authentication techniques***.





Thank You



SS ZG/WTZG 513
Network Security
Cryptographic Data Integrity Algorithms
Revision 1.0

BITS Pilani
Work Integrated Learning Programmes

Prof Vineet Garg
Bangalore Professional Development Center

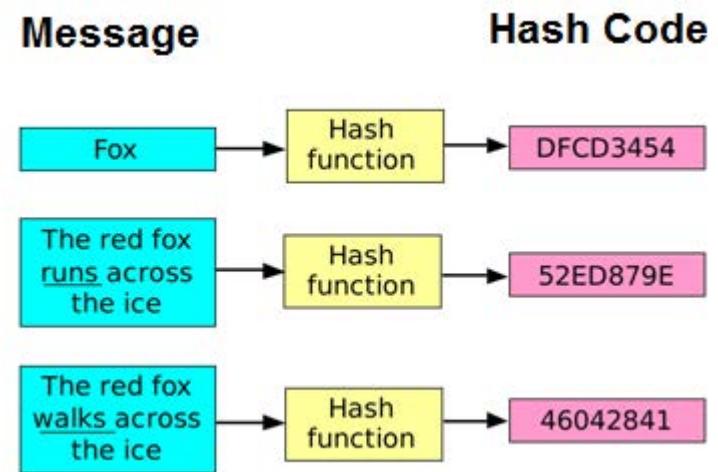
Integrity Attacks

Integrity Attacks	Modification	Some portion of a legitimate message is altered or the message is delayed.
	Masquerading	One entity pretends to be a different entity. E.g. Hoax bank sites.
	Replaying	Subsequent retransmission of a captured message to produce an unauthorized effect. E.g. Bill payment fake reminders.
	Repudiation	Sender denies that it sent the message or the receiver denies that it received the message.

In this session, we are primarily discussing about security from the first two types of Integrity attacks: ***Modification*** and ***Masquerading***.

Hash Function

- In Cryptography, a hash function (H) accepts a variable length message (M) and produces a fixed size hash value (h). Mathematically, $h = H(M)$. Here, h is called the hash code, hash digest, hash sum etc.
- A good hash function is expected to produce random and evenly distributed hash code but of same size.
- It is an important concept for Network Security because:
 - a. Irrespective of the length of the input message, hash code length is always same. Storage and transmission overhead can be estimated.
 - b. Even for a small change in the contents of the message, the hash code will turn out to be different. So, it can detect if the message was subject to ***modification attack***.



Different but same size hash codes for different messages having big or small differences.

Hash Function Properties

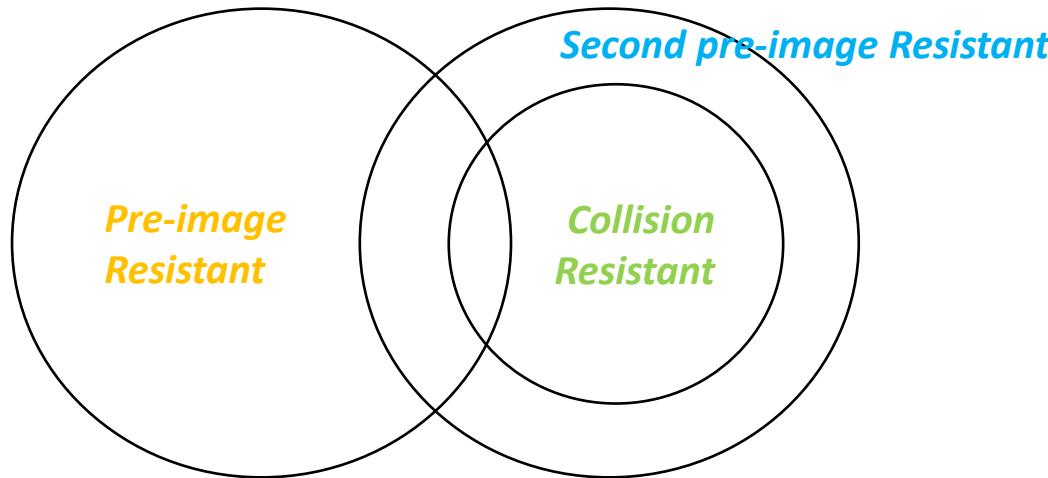
1. A hash function (H) can be applied to a block of message (data) of any size.
2. H produces a fixed-length output irrespective of the length of the message.
3. $H(x)$ is relatively easy to compute for any given message x , making both hardware and software implementations practical.
4. For any given hash code h , it is computationally infeasible to find x such that $H(x) = h$. A hash function with this property is referred to as ***one-way or pre-image resistant***.
5. For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$. A hash function with this property is referred to as ***second pre-image resistant***.
6. It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$. A hash function with this property is referred to as ***(strong) collision resistant***.

A hash function that follows first five properties - ***weak hash function***
all six properties - ***strong hash function***

Relationship



Among Hash Function Properties



Scenario	Hash Function Weakness
An attacker when gets a hash code is able to generate the original message.	Pre-image Weakness
An attacker intercepts a message and its hash code. Selects another message which gives the same hash code. Takes this new message to the intended receiver and commits fraud.	Second Pre-image Weakness
An attacker selects a message and gets the hash code of it from a sender. Finds another message which generates the same hash code. Commits fraud using the message that yields him more benefit.	Collision Weakness

Birthday Paradox



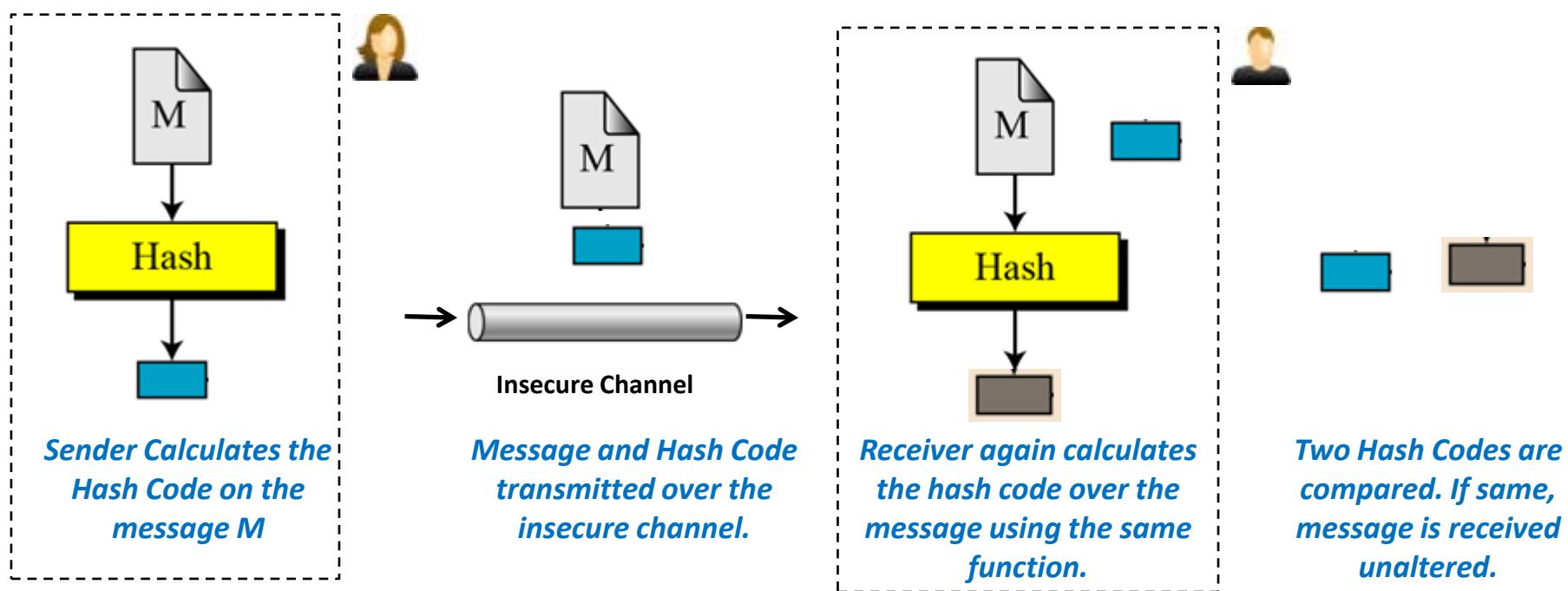
Some Statistical Results

- ❑ How many people must be in a room for the chance to be greater than 0.5 that one of them shares your birthday?
 - Answer is 253.
 - Analogous to the 5th property.
- ❑ How many people must be in a room for the chance to be greater than 0.5 that two of them shares the same birthday?
 - Answer is 23.
 - Popularly known as Birthday Attack.
 - Analogous to the 6th property
- ❑ The properties 5 and 6 are most crucial properties of a hash function for it to be useful for network security.
- ❑ Finding a message that hashes to a given hash value would require hashing 2^m random messages. Where m is the size of hash-code.
- ❑ Finding two messages that hash to the same value would only require hashing $2^{m/2}$ random messages.
- ❑ A machine that hashes one million messages per second would take 600,000 years to find a second message that matched a given 64-bit hash.
- ❑ The same machine could find a pair of messages that hashed to the same value in about an hour.
- ❑ **Moral of the story:** If you want to drop the odds of someone breaking your system to less than 1 in 2^{80} , use a 160-bit one-way hash function.

Hashing Technique



General Idea



Secure Hash Algorithm (SHA)

- ❑ Most widely used hash function.
- ❑ Developed by NIST in 1993 as part of FIPS-180 and later known as SHA-0.
- ❑ Iterative improvements:
 1. SHA-1 in 1995 ([RFC-3174](#))
 2. SHA-2 in 2002 (FIPS-180-4 and [RFC-4634](#))
 - SHA-2 has variants of producing hash codes of 256, 384 and 512 bits, which are known as SHA-256, SHA-384 and SHA-512 respectively.
 - 224 bits version published in 2008.
 3. SHA-3 in 2012
- ❑ Many commercial implementations still use SHA-512.
- ❑ Many later SHA versions are adapted variations of Message Digests (MD) algorithms developed by Ron Rivest of MIT.

SHA-512



Step-1: Append Padding Bits & Step-2: Append Length

- It takes a message which is $< 2^{128}$ bits and produces a 512-bit message digest. *Do not worry it would be Yottabytes!*
- The input is processed in the blocks of 1024 bits.
- The message is first padded so that its length $\equiv 896 \pmod{1024}$.
- Padding is always added, even if the message is already of the desired length. The number of padding bits is in the range of 1 to 1024.
- The padding consists of a single 1 bit followed by the necessary number of 0 bits.
- Now to this message, length of the original message before padding in unsigned 128-bit integer is appended keeping most significant byte first.
- The message is now a multiple of 1024 bits ($896+128=1024$) which is treated as blocks of 1024 bits each (M_1, M_2, \dots, M_N).

Example

Example: Initially a message is 2400 bits long. How many bits need to be added to it so that its length $\equiv 896 \pmod{1024}$?

Solution: Let us say m bits need to be added. So that:

$$2400+m \equiv 896 \pmod{1024}$$

$$\text{So, } m = [896 + (1024 \times \lfloor 2400/1024 \rfloor - 2400)] \pmod{1024}$$

$$= 544 \pmod{1024}$$

$$= 544$$

Example: Initially a message is 900 bits long. How many bits need to be added to it so that its length $\equiv 896 \pmod{1024}$?

Solution: Let us say m bits need to be added. So that:

$$900+m \equiv 896 \pmod{1024}$$

$$\text{So, } m = [896 + (1024 \times \lfloor 900/1024 \rfloor - 900)] \pmod{1024}$$

$$= -4 \pmod{1024}$$

$$= 1020$$

Example: Initially a message is 2590 bits long. How many bits need to be added to it so that its length $\equiv 896 \pmod{1024}$?

Solution: Let us say m bits need to be added. So that:

$$2590+m \equiv 896 \pmod{1024}$$

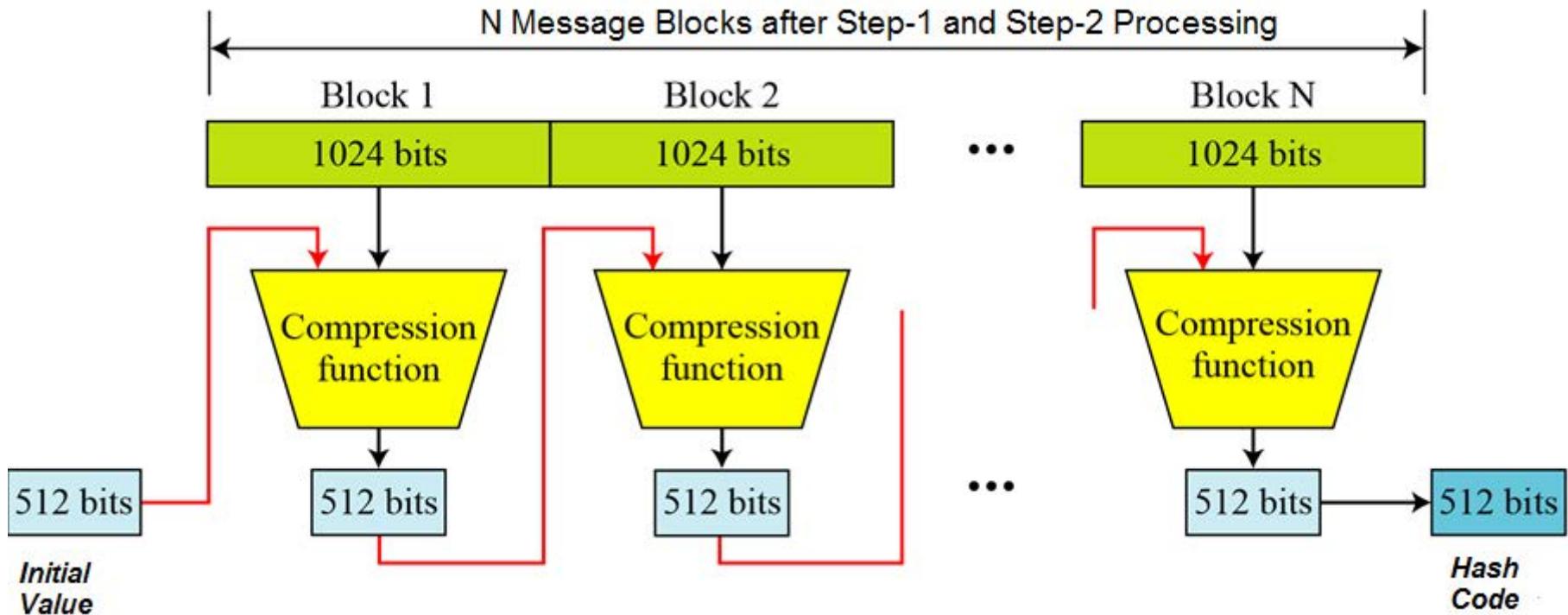
$$\text{So, } m = [896 + (1024 \times \lfloor 2590/1024 \rfloor - 2590)] \pmod{1024}$$

$$= 354 \pmod{1024}$$

$$= 354$$

Will this method work if the initial message is 896 or its multiple bits long ?

Top Level View



Observations:

- ✓ There is some initial value of 512 bits which is an input to the first block compression function.
- ✓ Each block of 1024 bits produces 512 bits after compression function.
- ✓ Output of one compression function is fed to the compression function of the next block.
- ✓ Compression function of the N th block produces the final 512 bits hash code.

SHA-512



Step-3: Initial Value

- A 512-bit buffer is used to hold the intermediate and the final results of the hash function.
- The buffer can be represented as eight 64-bit registers (A, B, C, D, E, F, G, H).
- These registers are initialized to the following 64-bit integers in hexadecimal:

$A = 6A\ 09\ E6\ 67\ F3\ BC\ C9\ 08$

$B = BB\ 67\ AE\ 85\ 84\ CA\ A7\ 3B$

$C = 3C\ 6E\ F3\ 72\ FE\ 94\ F8\ 2B$

$D = A5\ 4F\ F5\ 3A\ 5F\ 1D\ 36\ F1$

$E = 51\ 0E\ 52\ 7F\ AD\ E6\ 82\ D1$

$F = 9B\ 05\ 68\ 8C\ 2B\ 3E\ 6C\ 1F$

$G = 1F\ 83\ D9\ AB\ FB\ 41\ BD\ 6B$

$H = 5B\ E0\ CD\ 19\ 13\ 7E\ 21\ 79$

Total 8x64 bits = 512 bits

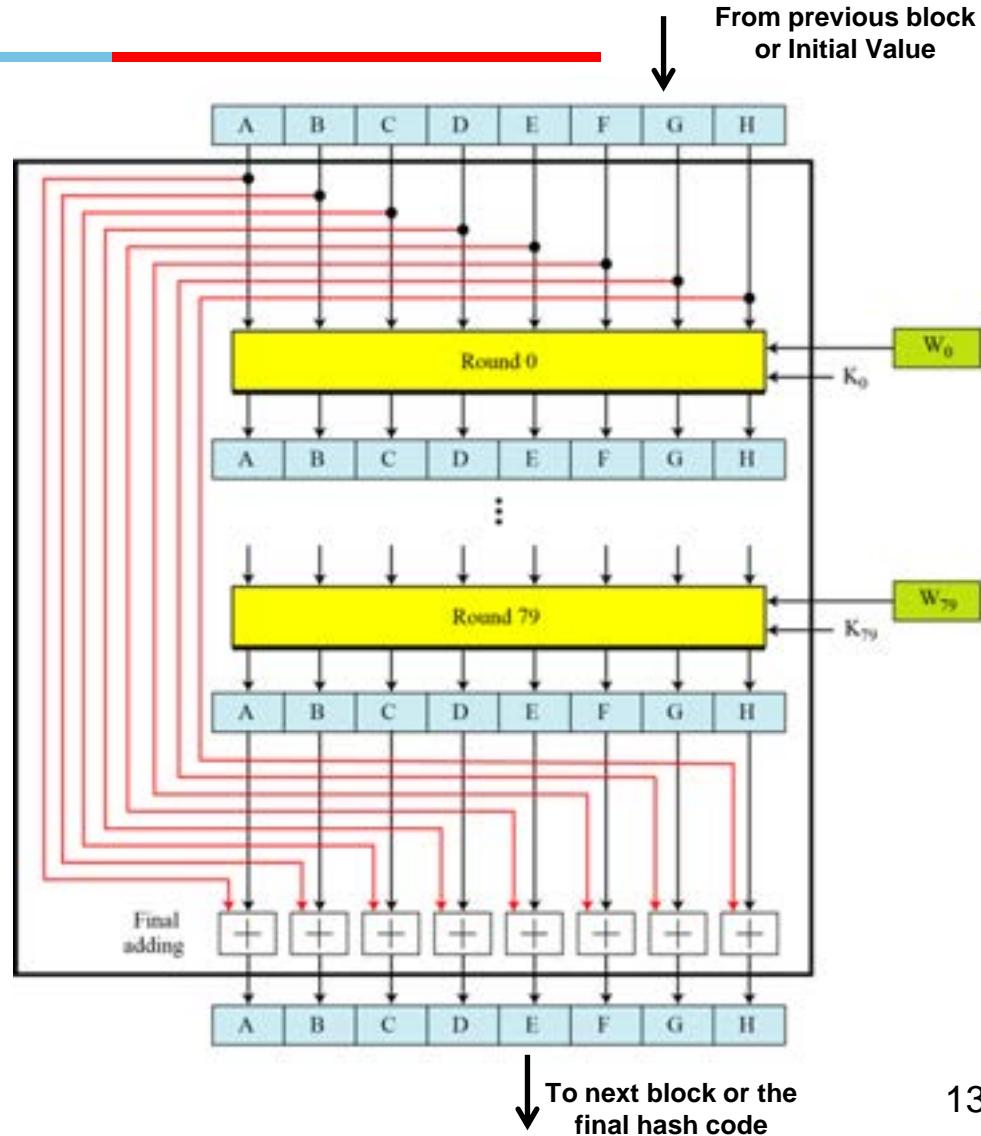
- These words were obtained by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers.

SHA-512



Step-4: Compression Function

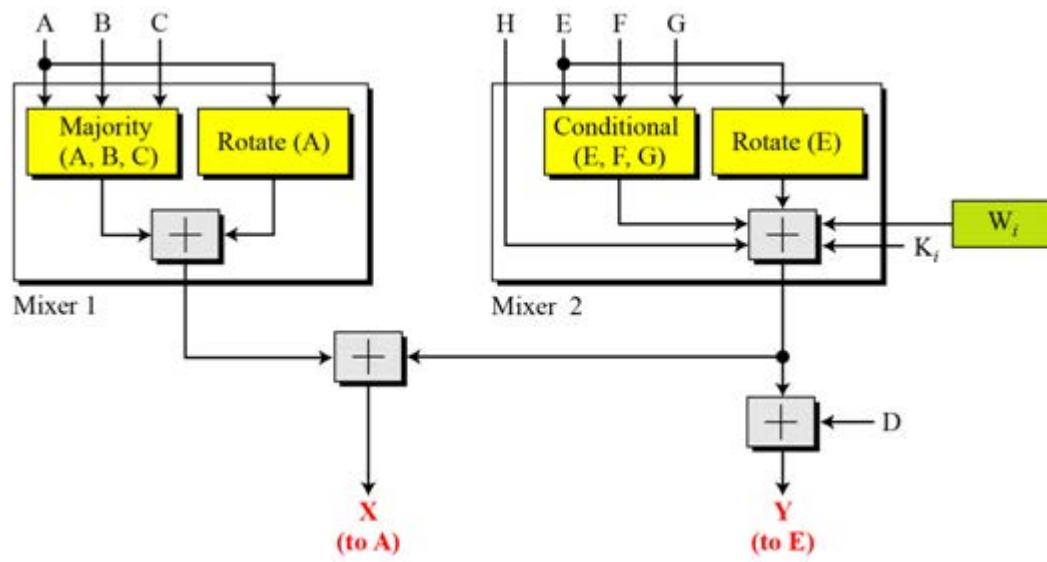
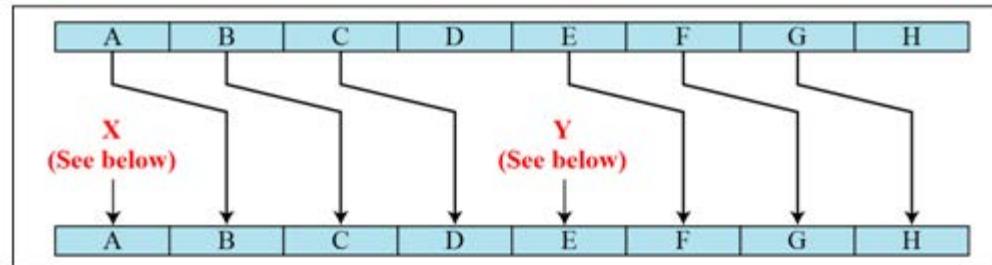
- Each compression function consists of 80 rounds 0 to 79.
- Input is the 512 bits of Initial Value or the output of the previous block.
- Input to each round is the output of the previous round.
- Each round also takes constants K as an input (K_0 to K_{79}). [Appendix-A](#).
- Each round also takes a value of W as an input (W_0 to W_{79}).
- Output of the round-79 is added with the input of the round-0, each of the 8 words independently in modulo 2^{64} arithmetic.
- The final output is the input to the next block (or the final hash code after N blocks).



SHA-512

Inside Each Round of Compression Function

Round



Majority (A, B, C)

$$= (A \text{ AND } B) \oplus (B \text{ AND } C) \oplus (C \text{ AND } A)$$

Conditional (E, F, G)

$$= (E \text{ AND } F) \oplus (\text{NOT } E \text{ AND } G)$$

Rotate (A)

$$= \text{ROTR}^{28}(A) \oplus \text{ROTR}^{34}(A) \oplus \text{ROTR}^{39}(A)$$

Rotate (E)

$$= \text{ROTR}^{14}(E) \oplus \text{ROTR}^{18}(E) \oplus \text{ROTR}^{41}(E)$$

$\text{ROTR}^X(Y)$ = Circular Right Shift of Y by X bits

K_i = SHA-12 constant for round i from Appendix-A

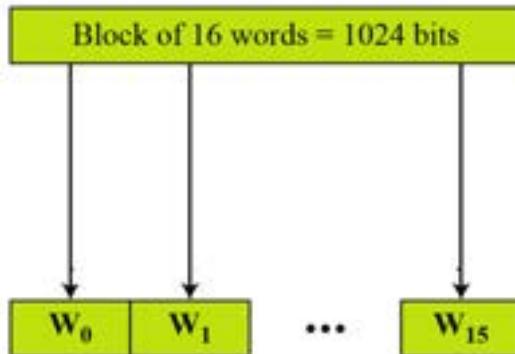
W_i = A 64 bit word derived from current 1024 bit block of the message (refer to next slide).

\oplus = Exclusive OR Operation

$+$ = Addition in modulo 2^{64}

Derivation of W_i

When $i = 0$ to 15 , W_i is 64 bit words of 1024 bit current block of the message as below:



For $i = 16$ to 79 the following formula is used:

$$W_i = W_{(i-16)} \oplus \text{RotShift}_{1-8-7}[W_{(i-15)}] \oplus W_{(i-7)} \oplus \text{RotShift}_{19-61-6}[W_{(i-2)}]$$

Where:

$$\text{RotShift}_{x-y-z}[P] = \text{ROTR}^X(P) \oplus \text{ROTR}^Y(P) \oplus \text{SHL}^Z(P)$$

$\text{ROTR}^X(Y)$ = Circular Right Shift of Y by X bits

$\text{SHL}^X(Y)$ = Shift Left Y by X bits

\oplus = Exclusive OR Operation and $+$ Addition in modulo 2^{64}

Exercise



-
1. When applied the ***Majority*** function on buffers A, B, and C. If the leftmost hexadecimal digits of these buffers are 0x7, 0xA, and 0xE, respectively, what is the leftmost digit of the result? (Answer: 0xE)

 2. When applied the ***Conditional*** function on E, F, and G buffers. If the leftmost hexadecimal digits of these buffers are 0x9, 0xA, and 0xF respectively, what is the leftmost digit of the result? (Answer: 0xE)

 3. Expand the formula to calculate W60 in SHA-512.



Message Digests (MD)

- Message Digest Algorithms (MD2, MD4, MD5 and MD6) are different hash functions designed by **Ron Rivest**, professor of MIT from 1989 onwards. They are standardized by IETF in the form of RFC and also adapted by NIST for SHA:
 - MD4: [RFC-1320](#)
 - MD5: [RFC-1321](#)
 - MD-6 proposed to [NIST for RSA-3](#)
- The operational structure of MD is similar to SHA.

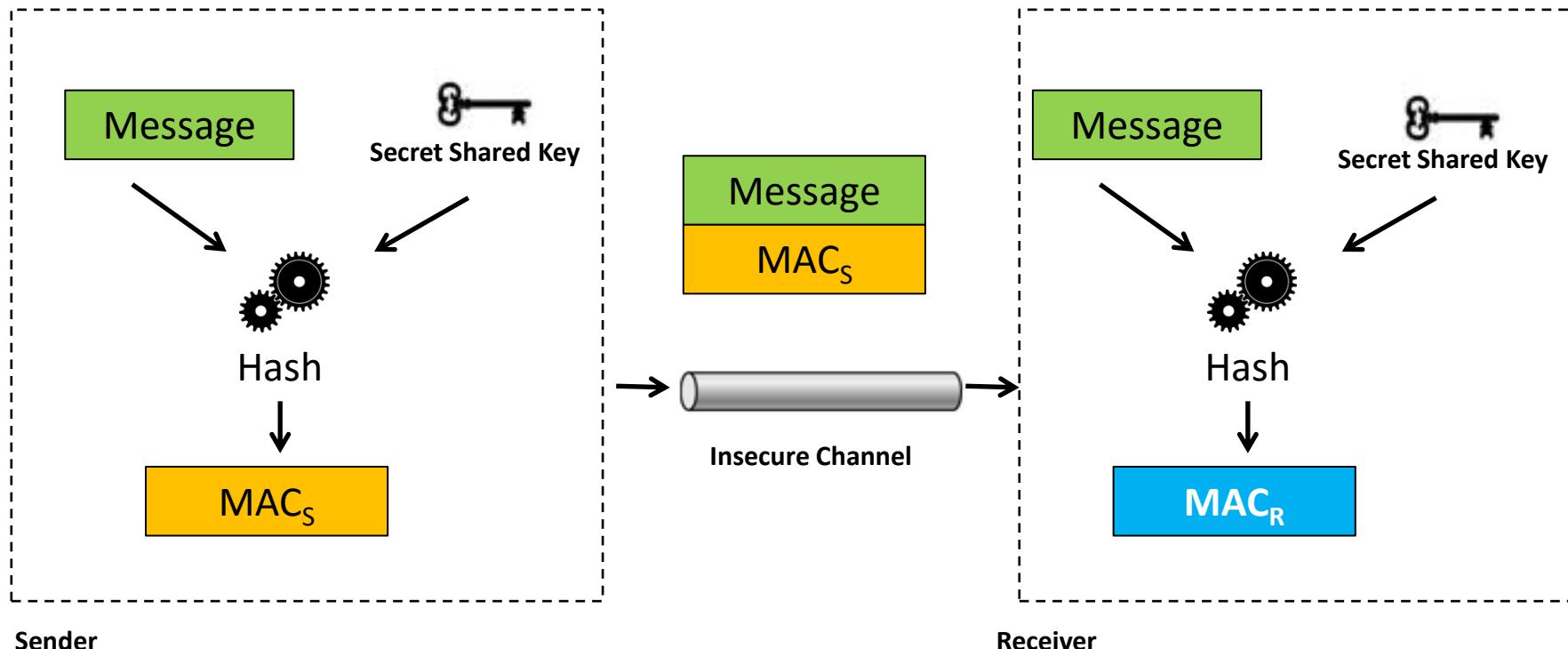
Message Authentication

- ❑ A hash code does not authenticate the sender of the message.
- ❑ To provide message authentication, sender needs to provide proof that it is “the sender” sending the message and not an impostor.
- ❑ The hash code created by a cryptographic hash function (e.g. SHA) is normally called a Modification Detection Code (MDC).
- ❑ What we need for message authentication is a Message Authentication Code (MAC).
- ❑ ***Seeding Thought - can we embed the identity of the sender in the hash code that the receiver can verify?***

Message Authentication Code (MAC)



General Idea



Sender

Receiver

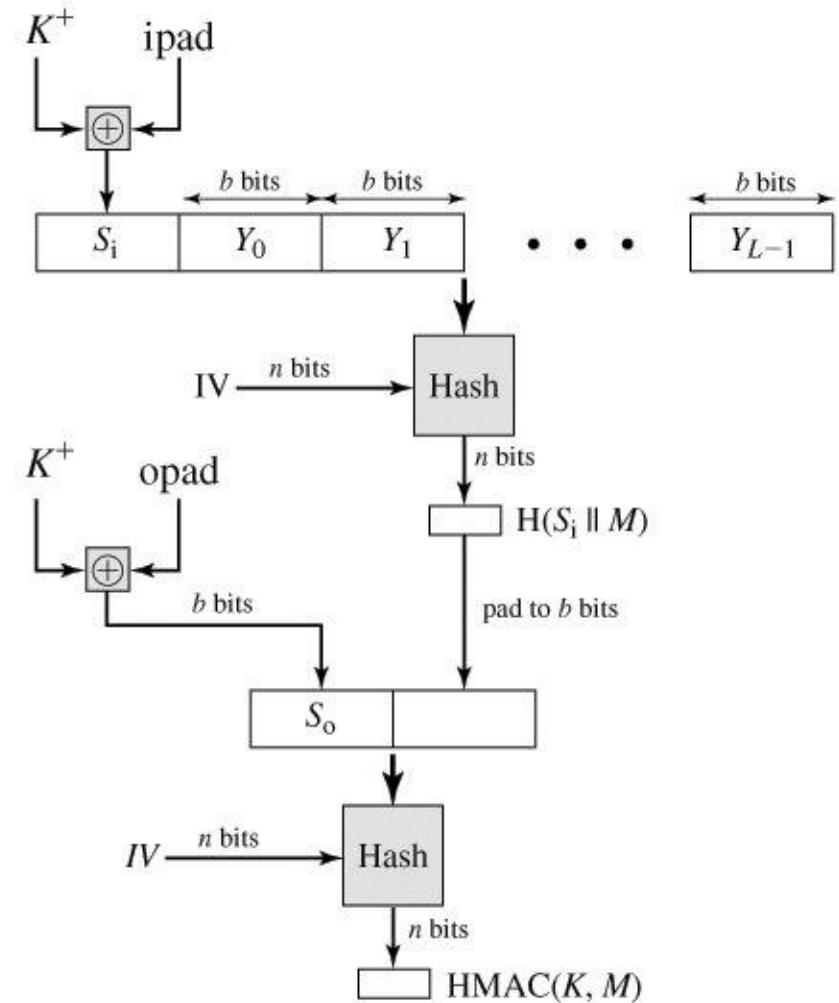
- ✓ MAC_S is the MAC calculated by the sender.
- ✓ MAC_R is the MAC calculated by the receiver.
- ✓ **MAC_S and MAC_R are to be same to declare that message is authenticated.**
(Content is unaltered and sender is who it is expected to be)
- ✓ **Can we use public/private keys, in place of secret shared key?**

Hashed MAC (HMAC) Structure

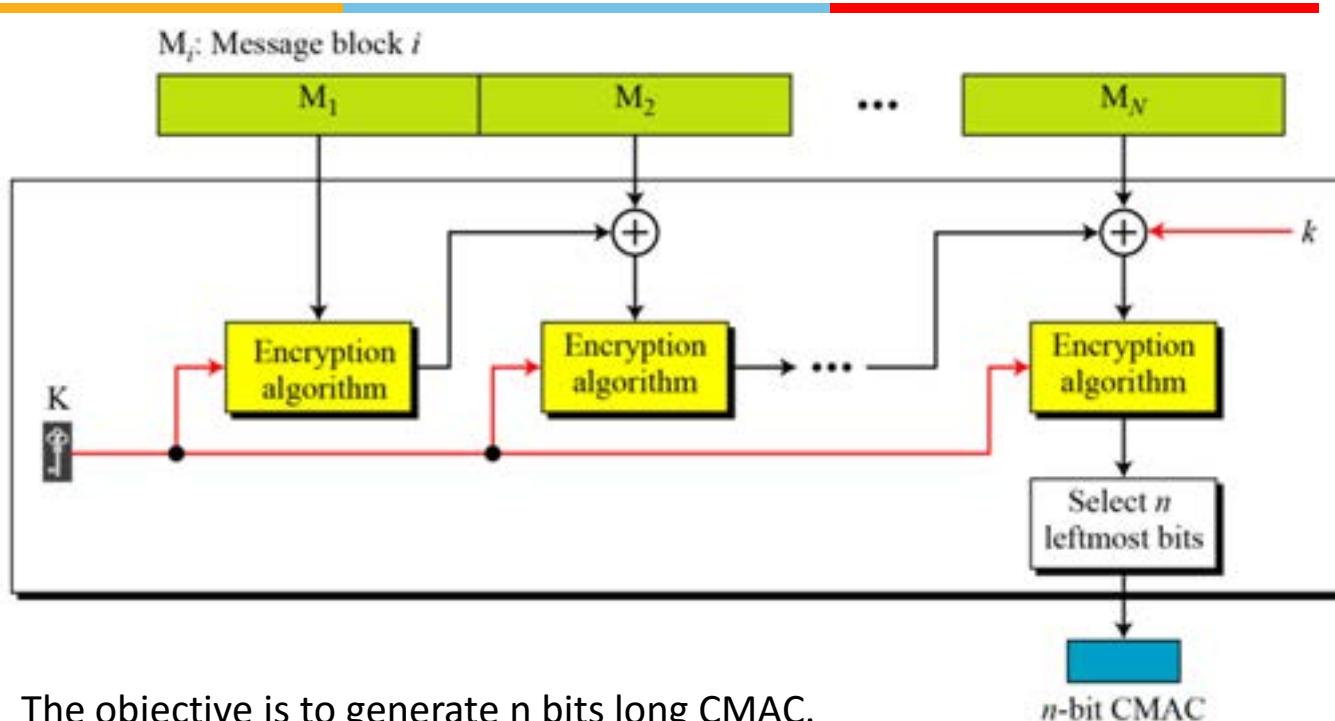


FIPS-198 and RFC-2104

- The Hash function produces n bits of hash code, which is also the size of HMAC.
- The message (M) is considered as blocks of b bits (Y_0 to Y_{L-1}).
- K is the shared secret key $\geq n$.
- K is appended on the left side with 0s (if required) so that its length becomes b . It is now called K^+ .
- ipad is 0x36 repeated $b/8$ times, so it is b bits long.
- K^+ is XORed with ipad and the output S_i is prepended with the message blocks.
- Using the Initial Value (IV) a hash is calculated and 0s are added on the left side to make it b bits long.
- opad is 0x5C repeated $b/8$ times, so it is b bits long.
- K^+ is now XORed with opad and the output S_0 is prepended to padded hash code calculated earlier.
- Hash is calculated on this prepared block and the final hash value is desired HMAC of the message M using key K .

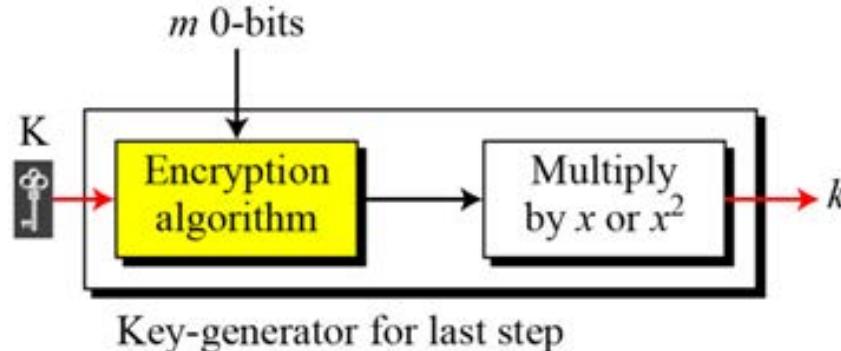


Cipher based MAC (CMAC) Structure



- The objective is to generate n bits long CMAC.
- The message M is divided into N blocks each having m bits. If the last block is not m bits long, it is padded with one 1 and required 0s.
- The first block is encrypted using symmetric key K and the output is XORed with the second block of the message. The result of XOR is again encrypted.
- The procedure continues until there is no more block left to process.
- In the last block processing one more key k is also used as an input for XOR.
- Output of the last block encryption is n bits CMAC.

CMAC: Additional Key Generation



- m bits all 0s are encrypted with the symmetric key K.
- The output is multiplied with x, if padding was not applied to the original message.
- Otherwise the output is multiplied with x^2 .
- The multiplication is in GF(2^m).
- Normally $m = 64$ or 128 so irreducible polynomials are used as $x^{64}+x^4+x^3+x+1$ and $x^{128}+x^7+x^2+x+1$ respectively.
- The output is k which is used in the last block processing in CMAC.

Exercise



A student tries to develop a new hashing algorithm. In it the size of hash code is one byte and its initial value is 0. The hash function works for English language only and it is case independent. It takes a character, adds it to the value in modulo-26 and then moves to the next character and so on.

1. Calculate the hash code for the word “CRYPT”.
2. Explain if you see any flaw in this hash function.



Thank You

Appendix-A

SHA-512 Constants (K_0 to K_{79} from left to right)

428A2F98D728AE22	7137449123EF65CD	B5C0FBCFEC4D3B2F	E9B5DBA58189DBBC
3956C25BF348B538	59F111F1B605D019	923F82A4AF194F9B	AB1C5ED5DA6D8118
D807AA98A3030242	12835B0145706FBE	243185BE4EE4B28C	550C7DC3D5FFB4E2
72BE5D74F27B896F	80DEB1FE3B1696B1	9BDC06A725C71235	C19BF174CF692694
E49B69C19EF14AD2	EFBE4786384F25E3	0FC19DC68B8CD5B5	240CA1CC77AC9C65
2DE92C6F592B0275	4A7484AA6EA6E483	5CB0A9DCBD41FBD4	76F988DA831153B5
983E5152EE66DFAB	A831C66D2DB43210	B00327C898FB213F	BF597FC7BEEF0EE4
C6E00BF33DA88FC2	D5A79147930AA725	06CA6351E003826F	142929670A0E6E70
27B70A8546D22FFC	2E1B21385C26C926	4D2C6DFC5AC42AED	53380D139D95B3DF
650A73548BAF63DE	766A0ABB3C77B2A8	81C2C92E47EDAEE6	92722C851482353B
A2BFE8A14CF10364	A81A664BBC423001	C24B8B70D0F89791	C76C51A30654BE30
D192E819D6EF5218	D69906245565A910	F40E35855771202A	106AA07032BBD1B8
19A4C116B8D2D0C8	1E376C085141AB53	2748774CDF8EEB99	34B0BCB5E19B48A8
391C0CB3C5C95A63	4ED8AA4AE3418ACB	5B9CCA4F7763E373	682E6FF3D6B2B8A3
748F82EE5DEFB2FC	78A5636F43172F60	84C87814A1FOAB72	8CC702081A6439EC
90BEFFFA23631E28	A4506CEBDE82BDE9	BEF9A3F7B2C67915	C67178F2E372532B
CA273ECEEA26619C	D186B8C721C0C207	EADA7DD6CDE0EB1E	F57D4F7FEE6ED178
06F067AA72176FBA	0A637DC5A2C898A6	113F9804BEF90DAE	1B710B35131C471B
28DB77F523047D84	32CAAB7B40C72493	3C9EBE0A15C9BEBC	431D67C49C100D4C
4CC5D4BECB3E42B6	4597F299CFC657E2	5FCB6FAB3AD6FAEC	6C44198C4A475817

These words were obtained by taking the first sixty-four bits of the fractional parts of the cubic roots of the first eighty prime numbers.



SS ZG513
Network Security
Layered Architecture and Networking Layers

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Need for a Layered Architecture



Service Model

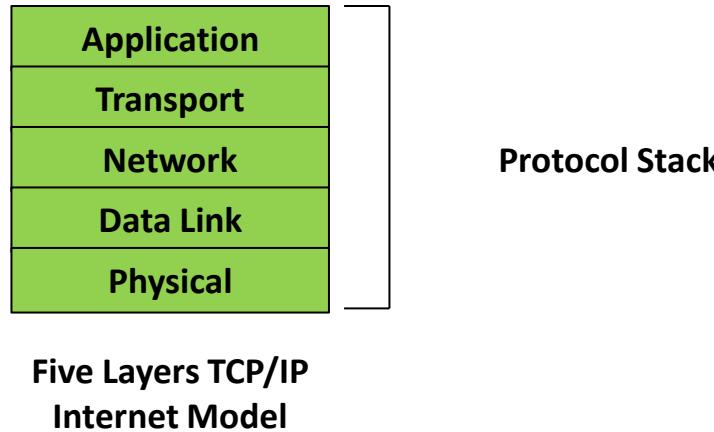


- In the figure above, the airline functionality is divided into layers where each layer is performing some functionality.
- Each layer provides some service to the layer above it and uses some service from the layer below it.
- A layered architecture allows us to discuss a well-defined, specific part of a large and complex system.
- Modularity is the key advantage. Any change in the layer is having only a local impact at that layer (provided the layer interfaces are kept intact).

Computer Networks



A Layered Architecture



- Similar to the airline functionality, a modern computer network can be designed in a layered architecture.
- A layer can be implemented in software, in hardware, or in a combination of the two. An application (e.g. HTTP) is usually implemented in software, whereas physical layer and data link layers are implemented in hardware (e.g. network interface cards). Network layer could be a mix of hardware and software.
- Rules for the two layers to communicate is called a *protocol*. When taken together, the protocols of the various layers are called the *protocol stack*.



Thank You



SS ZG513

Network Security

Security at the Application Layer

Revision 1.0

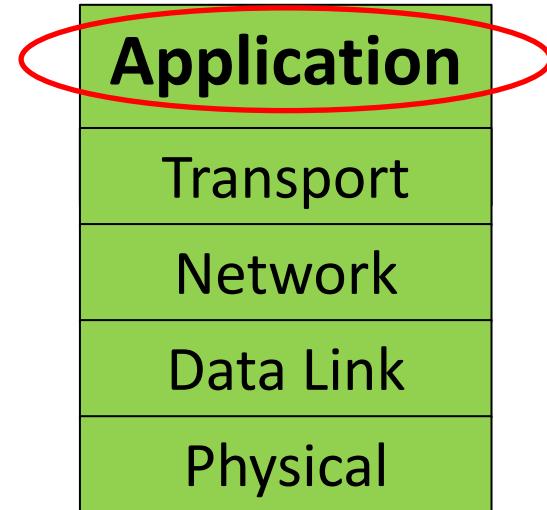
BITS Pilani

Work Integrated Learning Programmes



Security at Application Layer

- Application Layer provides services for an application to send and receive data over the network.
- Its interface to the transport layer is operating system dependent. E.g. Sockets.
- There are certain advantages providing security right at the application layer:
 - Executing in the context of the user. Easy access to user's credentials.
 - Complete access to the data. Easier to ensure nonrepudiation and small security granularity.
 - Application specific security.
- There are certain disadvantages also providing security at the application layer:
 - Need for each application specifically – expensive.
 - Many implementation variants – higher probability of errors and compatibility issues.





Thank You



SS ZG513
Network Security
E-Mail System Architecture

Revision 1.0

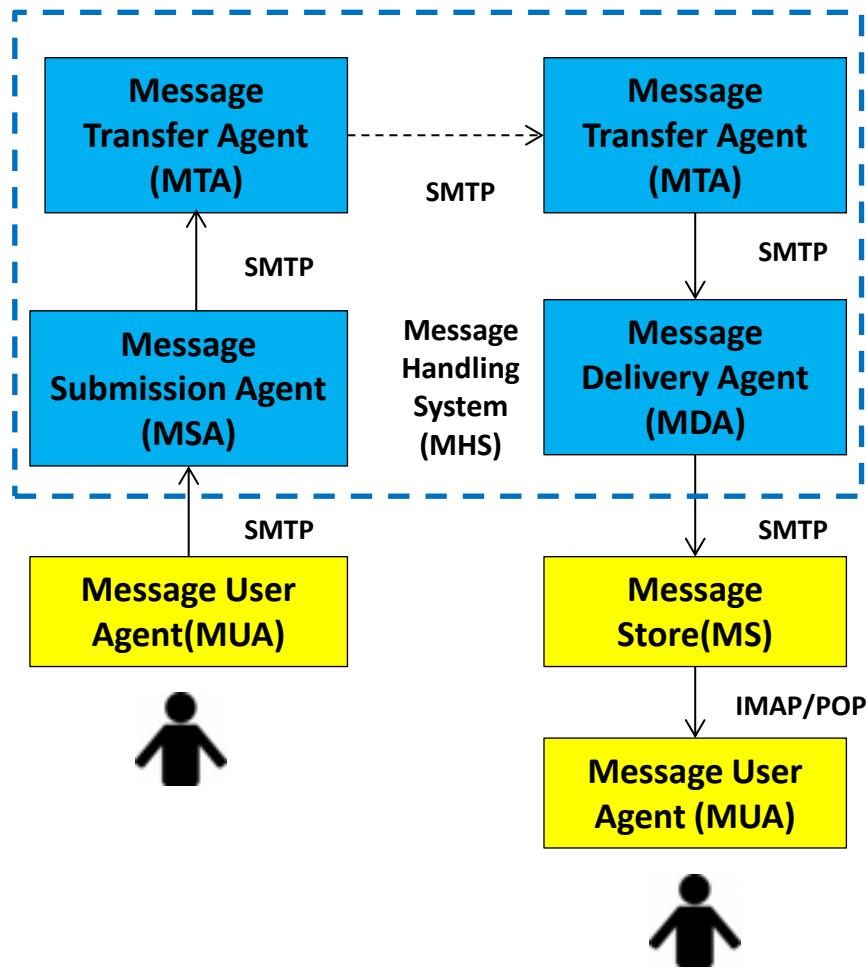
BITS Pilani

Work Integrated Learning Programmes

E-Mail System Architecture



Internet Based E-Mail Architecture - Reference IETF RFC -5598



- **Message User Agent (MUA):** A client user program for formatting, submission and reception of emails.
- **Message Submission Agent (MSA):** May be co-located with MUA. Enforces the policies of hosting domain and performs sanity checks.
- **Message Transfer Agent (MTA):** It is kind of a router/forwarder for the emails over the internet. Act as relays until an email reaches to the destination delivery agent.
- **Message Delivery Agent (MDA):** It delivers the email to the message store.
- **Message Store (MS):** A server which stores the emails for the users.
- **SMTP (Simple Mail Transfer Protocol):** The protocol used for email delivery. Defined in RFC-2821 (Obsoletes: 821).
- **IMAP (Internet Message Access Protocol)/ POP(Post Office Protocol):** MUA retrieves messages from MS using it.



Thank You



SS ZG513

Network Security

PGP - Introduction

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Pretty Good Privacy (PGP)

- It is created by Phillip R. Zimmermann. He is a member and leading advisor with many universities, Internet and security research groups.
- It provides protection from Confidentiality and Integrity attacks on E-Mails and file storage applications.
- First published on the Internet in 1991.
- PGP source code can be downloaded from <http://www.pgpi.org/> for various operating systems.
- Many products and web browser plug-ins use PGP for providing e-mails security.
- It is based on the cryptographic algorithms that are time tested, reviewed and considered extremely safe.
- Originally it was not created keeping any standardization in mind. But now it is on the standards track with IETF RFC-4880 and RFC-3156.





Thank You



SS ZG513

Network Security

PGP – Integrity Services

Revision 1.0

BITS Pilani

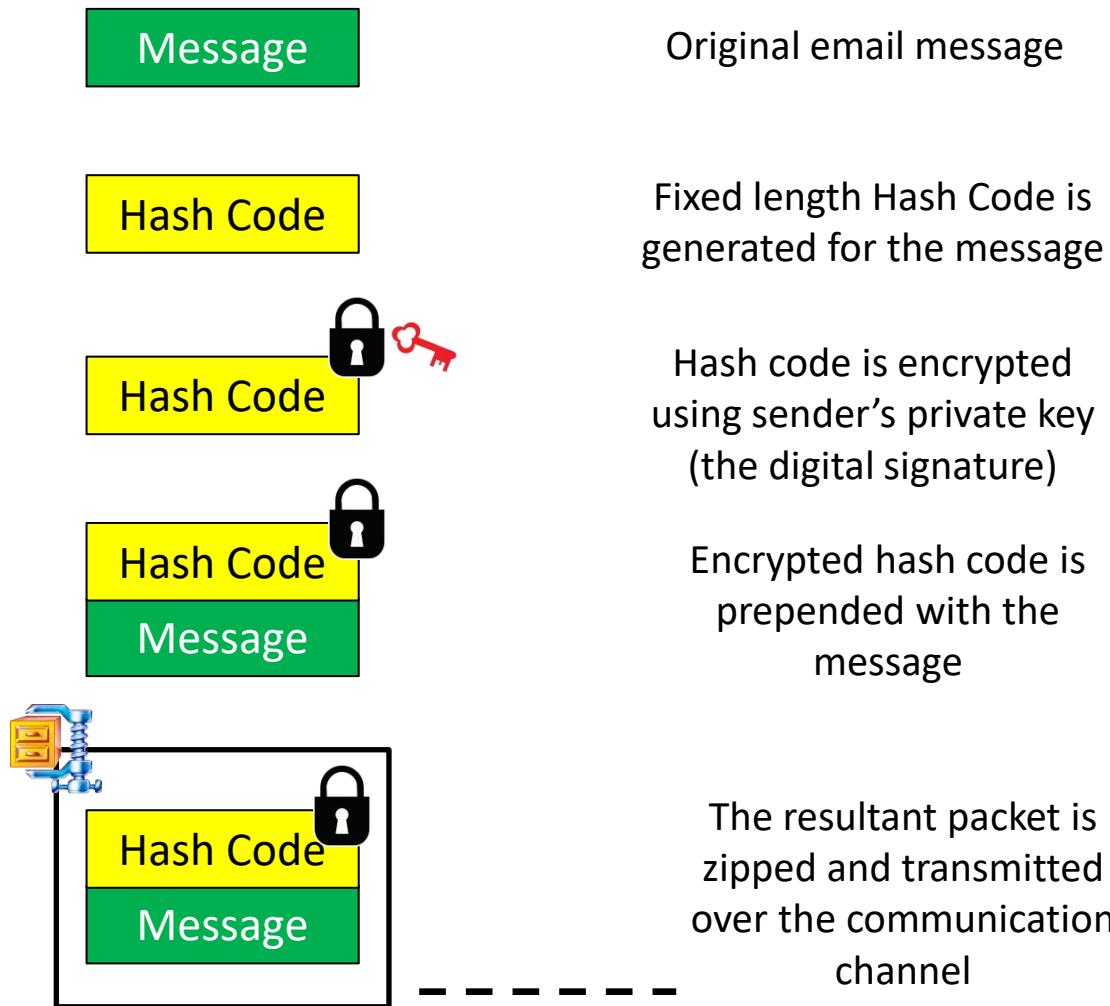
Work Integrated Learning Programmes



E-Mail Integrity using PGP



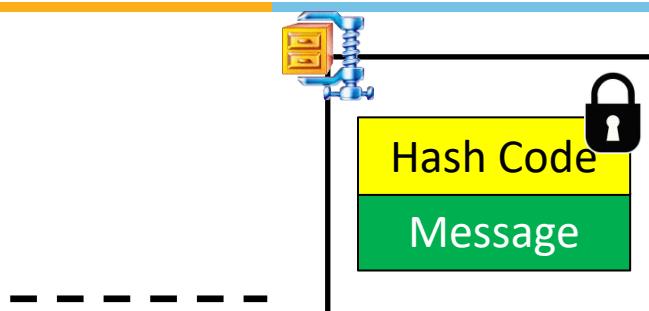
Sender Side Procedure



E-Mail Integrity using PGP



Receiver Side Procedure



The resultant packet is received from the communication channel



Packet is unzipped

Message

Hash Code for the message is calculated again by the receiver

Hash Code



Received Hash code is decrypted using the sender's public key

Hash Code

Decrypted Hash code

Hash Code

Resulting Hash code

No: Message Integrity Failed

Same ?

Yes: Message Integrity Verified



Thank You



SS ZG513

Network Security

PGP – Confidentiality Services

Revision 1.0

BITS Pilani

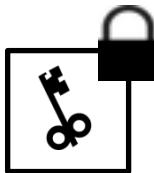
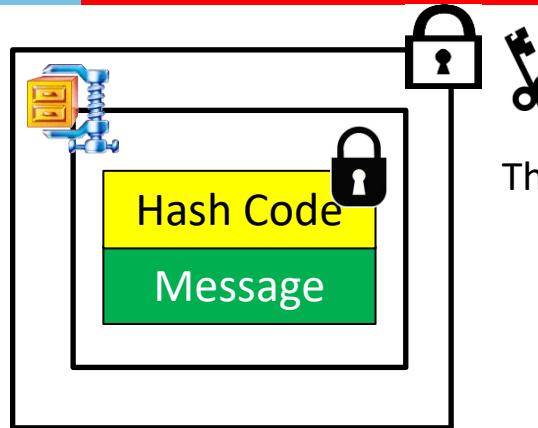
Work Integrated Learning Programmes



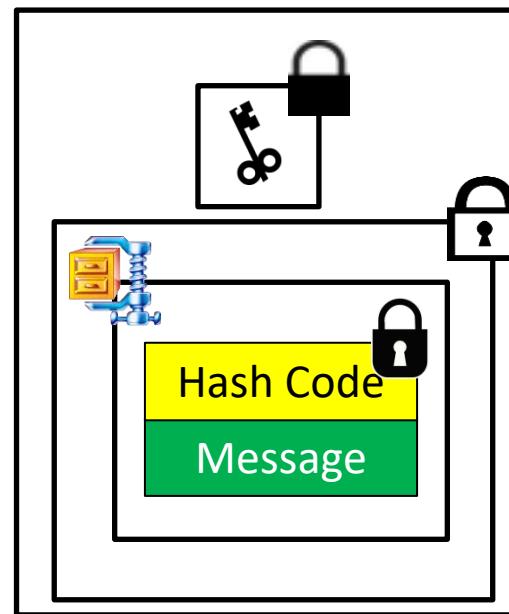
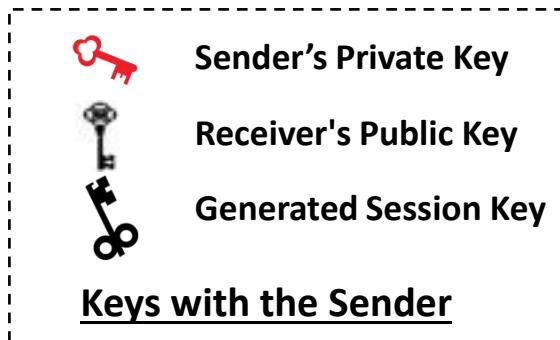
E-Mail Confidentiality using PGP



Sender Side



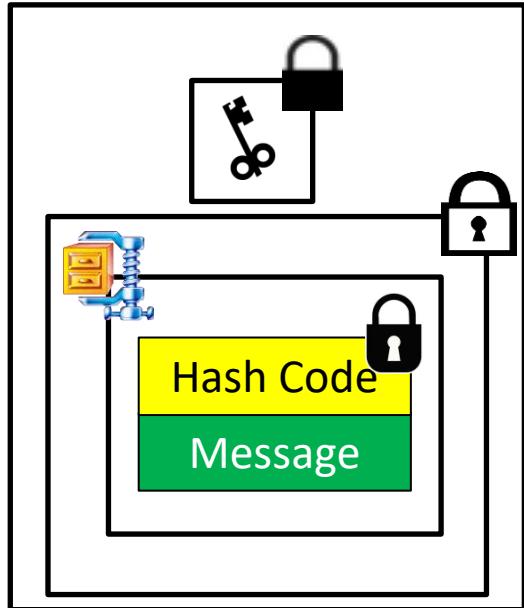
The session key used for symmetric encryption itself is encrypted using the public key of the receiver



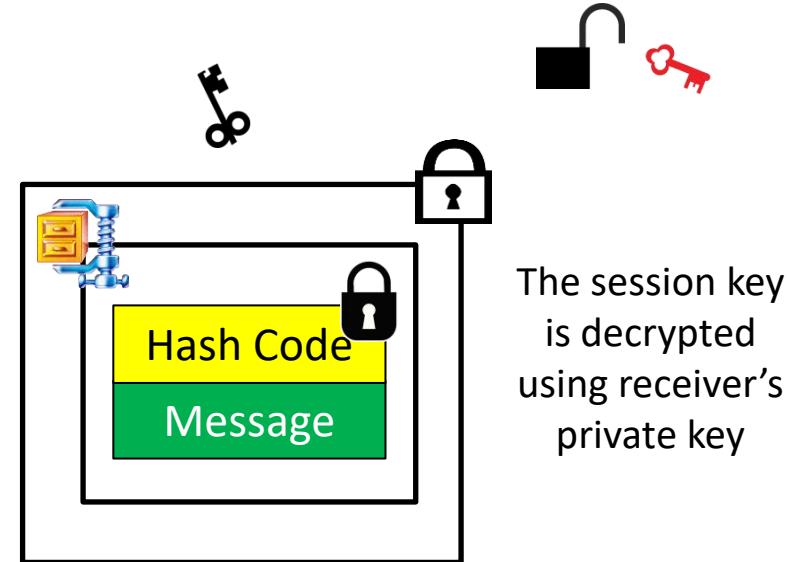
E-Mail Confidentiality using PGP



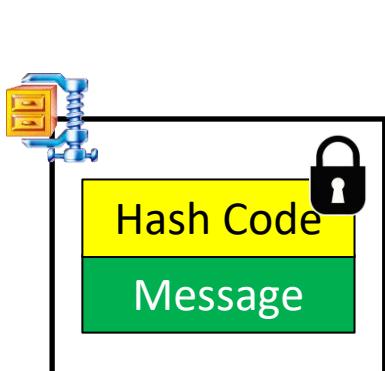
Receiver Side



The encrypted session key and the encrypted packet received from the communication channel



The session key is decrypted using receiver's private key



The packet is decrypted using the decrypted session key

Remaining procedure will remain same as in the previous PGP Integrity slides





Thank You

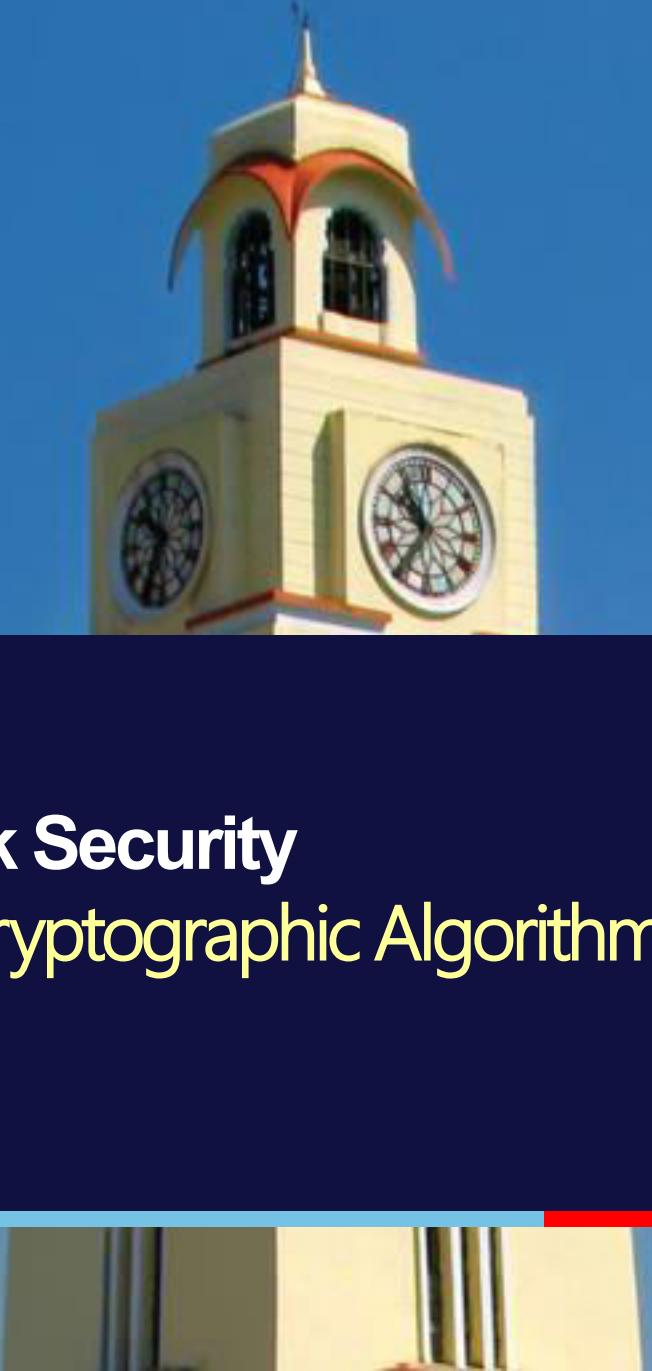


SS ZG513
Network Security
PGP – Cryptographic Algorithms

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



PGP: Different Cryptographic Algorithms



Functions	Algorithms Used	Description
Digest/Digital Signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 Conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

DSS: Digital Signature Standard

SHA: Secure Hash Algorithm

DES: Data Encryption Standard

RSA: Rivest Shamir Adleman **IDEA:** International Data Encryption Algorithm

CAST: Carlisle Adams and Stafford Tavares (a Symmetric Key Algorithm)



Thank You



SS ZG513

Network Security

MIME and MIME Headers

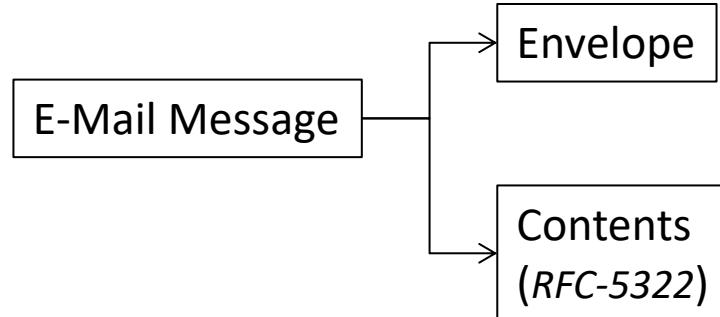
Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



E-Mail Message Structure



Date: July 18, 2015 9:00:00 AM IST

From: "Professor" <professor@bits-pilani.ac.in>

Subject: The E-Mail Format in RFC 5322

To: class@some_host.com

Cc: dean@bits-pilani.ac.in

Good Morning! We are committed to provide you world class education!

Header

Blank Line

Body

Multipurpose Internet Mail Extensions (MIME)



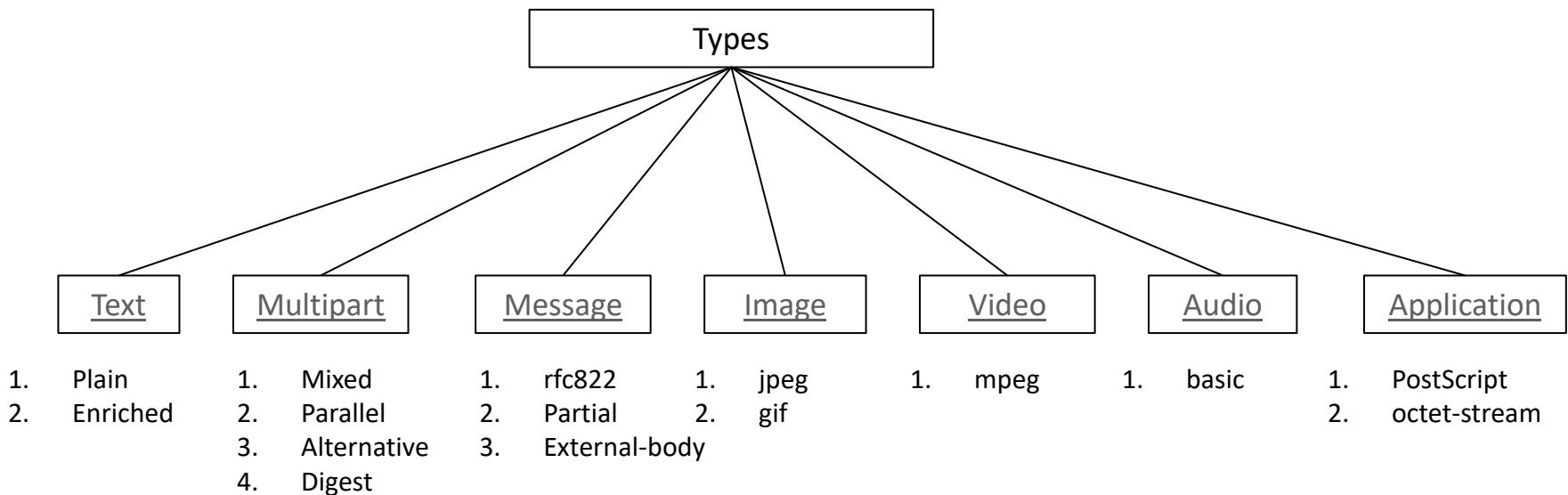
References: IETF [RFC-2045](#) and [RFC-2046](#)

- Five new message headers are defined which may be included in the RFC-5322 header.
 - I. **MIME-Version:** The present value is 1.0.
 - II. **Content-Type:** Description for the data contained in the body for user agent to pick up the appropriate mechanism to represent the data to the user.
 - III. **Content-Transfer-Encoding:** Transformation used in to make the E-Mail acceptable for transport.
 - IV. **Content-ID:** Identifications for MIME entities in multiple contexts.
 - V. **Content-Description:** A text description about the content which is useful to describe when the data is not readable (e.g. audio or video data)
- Any MIME compliant system must support the first 3 headers and other headers are optional.
- A later [RFC-2183](#) added a new optional header **Content-Disposition**, which is used to convey how the data should be handled. E.g. if there is any attachment with the email.

MIME Headers



Content Type



MIME Headers



Content Type – Example

```
From: Professor <prof@bits-pilani.ac.in>
To: Class <class@some_host.com>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="example boundary"
```

Any body part here will be ignored.

```
-- example boundary
```

Since there is no other header here, so it will be considered ASCII text implicitly.

```
-- example boundary
```

```
Content-type: text/plain; charset=us-ascii
```

Some text can go here too, which is explicitly mentioned as ASCII.

```
-- example boundary
```

```
Content-type: text/enriched
```

This is <i>rich text.</i>

```
-- example boundary --
```

Any body part here will be ignored.

Note: Two hyphen prefixes in the boundary delimiters. Two hyphen prefixes and suffixes in the final boundary delimiter.

MIME Headers



Content Type – Multipart example with parallel subtype

```
From: Professor <prof@bits-pilani.ac.in>
To: Class <class@some_host.com>
Subject: Sample message
```

```
MIME-Version: 1.0
```

```
Content-Type: multipart/parallel; boundary= "example boundary-1"
```

```
-- example boundary-1
```

```
Content-Type: audio/basic
```

```
Content-Transfer-Encoding: base64
```

```
... base64-encoded 8000 Hz single-channel mu-law-format audio data  
goes here....
```

```
-- example boundary-1
```

```
Content-Type: image/jpeg
```

```
Content-Transfer-Encoding: base64
```

```
... base64-encoded image data goes here....
```

```
-- example boundary-1 --
```

MIME Headers



Content-Type-Encoding

Syntax:

Content-Type-Encoding: **type**

Type	Description
7bit	All data is in ASCII characters with short lines
8bit	ASCII or non-ASCII characters with short lines
binary	ASCII/non-ASCII characters unlimited length
base64	6 bits blocks of data converted into 8 bits ASCII characters. Also called Radix-64 encoding
quoted-printable	A method to represent non-ASCII into ASCII
x-token	Non-standard encoding, <i>token</i> is to be replaced by its name.

MIME Headers



Content-ID and Content-Description

Content-ID: The Content-ID header associates a unique ID with a MIME body. The value of a Content-ID header has the format <xxxx@yyyy>. The left side of the value generally contains a serial number, an indication of the date and time the message was generated, or both. The right side of the value to indicate the system where the MIME part originated.

It is used where receiver is expected to receive multiple emails and wants to use the ID of each email to some specific context.

Example: **Content-ID: <1118072015@bits-pilani.ac.in>**

Content-Description: Some textual description about the E-Mail message.

Example:

Content-Description: This email contains the project status.



Thank You



SS ZG513

Network Security

S/MIME - Introduction

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes

Secure/MIME or S/MIME

- Security Enhancement to the MIME is based on the technology from RSA Data Security. RSA was named after the initials of its co-founders, *Rivest, Shamir* and *Adleman*. RSA is a division of EMC² Corporation now.
- Similar to PGP, S/MIME also offers ability to sign and/or encrypt messages.
- Standardization attempt through IETF [RFC-5751](#).
- S/MIME provides the following functions:
 - Enveloped Data
 - Signed Data
 - Clear-Signed Data
 - Signed and Enveloped data - combination of above three.
- S/MIME uses Content-Type header of MIME and introduces new types and subtypes.





Thank You



SS ZG513
Network Security
Security Services through S/MIME

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



S/MIME: Enveloped Data

Enveloped Data: Encrypted content and encrypted session key which was used to encrypt the content in base64 encoding.

Sender's Procedure:

- Generate a session key for a symmetric encryption algorithm.
- Encrypt the session key with the recipient's public key.
- Prepare a block known as ***RecipientInfo*** that contains: an identifier of the recipient's public-key certificate, an identifier of the algorithm used to encrypt the session key, and the encrypted session key.
- Encrypt the message content with the session key.
- ***RecipientInfo*** block followed by encrypted content is called the Enveloped Data.
- It is encoded into base-64 encoding.

Recipient's Procedure:

- Base-64 encoding is stripped off.
- Recipient's private key is used to decrypt the session key.
- Session key is used to decrypt the encrypted message.

S/MIME: Enveloped-Data



Example

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;  
name=smime.p7m
```

```
Content-Transfer-Encoding: base64
```

```
Content-Disposition: attachment; filename=smime.p7m
```

```
rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6  
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGrfvbnjT6jh7756tbB9H  
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4  
0GhIGfHfQbnj756YT64V
```

Observations:

- Content-Type is application and sub-type is pkcs7-mime.
- Encoding is done using base64.
- A new MIME header **Content-Disposition** specifying an attachment with a parameter of file name having p7m extension.
- Blue text is base64 encoded enveloped data.
- PKCS-7 = Public Key Cryptography Standards-7.

S/MIME: Signed Data

Signed Data: Encrypted message digest as signature and the content in base64 encoding.

Sender's Procedure:

- Sender selects the digest algorithm and a message digest is computed using the algorithm.
- The digest is encrypted using sender's private key.
- A block called ***SignerInfo*** is prepared that contains sender's public key certificate, message digest algorithm used, encryption algorithm used and then the encrypted message digest.
- ***SignerInfo*** block followed by content is called the Signed Data.
- It is encoded into base-64 encoding.

Recipient's Procedure:

- Base-64 encoding is stripped off.
- Sender's public key is used to decrypt the digest.
- Recipient also independently calculates the digest and compares it with the decrypted digest to verify the signature.

S/MIME: Signed-Data



Example

```
Content-Type: application/pkcs7-mime; smime-type=signed-data;  
name=smime.p7m
```

```
Content-Transfer-Encoding: base64
```

```
Content-Disposition: attachment; filename=smime.p7m
```

```
567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7  
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH  
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jh7756tbB9H7n8HHGghyHh  
6YT64V0GhIGfHfQbnj75
```

Observations:

- Content-Type is application and sub-type is pkcs7-mime.
- Encoding is done using base64.
- Content-Disposition specifying an attachment with a parameter of file name having p7m extension.
- Blue text is base64 encoded signed data.

S/MIME: Clear-Signed Data

Clear-Signed Data: Same as Signed-Data but base64 encoding is done only on the encrypted digest (digital signature). A recipient without S/MIME capability can read the message but cannot verify the signature.

Sender's Procedure:

- Sender selects the digest algorithm and a message digest is computed using the algorithm.
- The digest is encrypted using sender's private key.
- A block called ***SignerInfo*** is prepared that contains sender's public key certificate, message digest algorithm used, encryption algorithm used and then the encrypted message digest. It is encoded into base-64 encoding.
- Message followed by ***SignerInfo*** block is called the Signed Data.

Recipient's Procedure:

- Recipient can read the message without verifying the signature.
- Base-64 encoding is stripped off from the ***SignerInfo block***.
- Sender's public key is used to decrypt the digest.
- Recipient also independently calculates the digest and compares it with decrypted digest to verify the signature.

S/MIME: Clear-Signed Data



Example

```
Content-Type: multipart/signed; protocol="application/pkcs7-signature";
micalg=sha1; boundary=boundary

--boundary
Content-Type: text/plain

This is a clear-signed message.

--boundary
Content-Type: application/pkcs7-signature; smime-type=signed-data; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jh77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756
--boundary--
```

Observe the following:

- Content-Type is multipart and sub-type is signed. Protocol parameter tells the signature and micalg tells the digest algorithm.
- The message body is plain text. The digital signature is in the second part.
- Content-Disposition specifying an attachment with a parameter of file name having p7s extension.



Thank You



SS ZG513
Network Security
S/MIME - Cryptographic Algorithms

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



S/MIME: Cryptographic Algorithms



Quick Summary

- Hash Functions: SHA-1 and Message Digest (MD5)
- Digital Signatures: DSS and RSA
- Session Key Encryption: Diffie-Hellman/ElGamal and RSA
- Message Encryption: 3-DES, RC2/40

Note: There are few absolute (MUST) and few recommended (SHOULD) requirements for the sender and receiver to support the above algorithms. They are described in IETF [RFC-5751](#).



Thank You



SS ZG513

Network Security

Web Security - Threats, Challenges and Solutions.

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes

Web Security Threats



Scenarios

- Bob is surfing the Web and arrives at the Alice Inc. site, which is selling electronic goods. The Alice Inc. site displays a form in which Bob is supposed to enter the type of item and quantity desired, his address, and his payment card number. Bob enters this information, clicks on submit, and expects to receive the goods.
 - If no confidentiality (encryption) is used, an intruder could intercept Bob's order and obtain his payment card information. The intruder could then make purchases at Bob's expense. **Attack on Confidentiality.**
 - If no data integrity is used, an intruder could modify Bob's order, having him purchase ten times more items than desired. **Attack on Integrity.**
 - A competitor can flood bogus requests to bring Alice Inc's web server down. **Attack on Availability.**
 - If no server authentication is used, a fake server could display Alice Inc's famous logo when in actuality the site is maintained by crooks, who are masquerading as Alice Inc. **Attack on Authenticity.**

- *These are common day-to-day scenarios. How these attacks can be foiled?*
- *The above example is related to Web application that uses HTTP, but a similar situation can occur in any type of application that uses TCP/UDP transport service.*
- *Binding security to specific application (browser) is not a good idea. How can we achieve security with application independence? At least up to some extent.*

Secure Socket Layer (SSL)

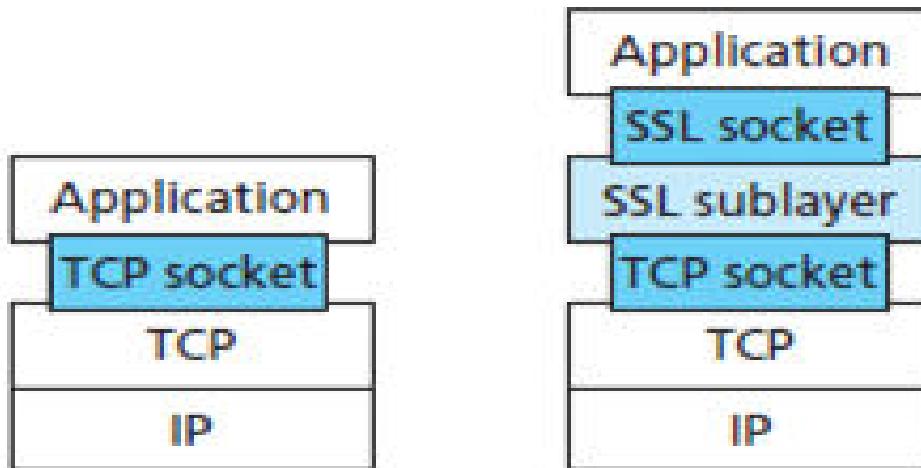


Introduction

- The purpose of SSL is to enhance the capability of TCP with confidentiality, data integrity, server authentication and client authentication features to protect from the security threats discussed in the previous slides.
- It was developed by ***Dr. Taher ElGamal***, present security CTO of salesforce.com during his Netscape tenure during 1995-98.
- SSL is often used to provide security to transactions that take place over HTTP. However, because SSL secures TCP, it can be employed by any application that runs over TCP. Security having application independence is the prime motivation behind SSL.
- SSL provides a simple Application Programmer Interface (API) with sockets, which is similar and analogous to TCP's API. When an application wants to employ SSL, the application includes SSL classes/libraries.
- Its follow-on standard known as Transport Layer Security (TLS) is defined in [IETF RFC-5246](#).



Position of SSL



**TCP/IP Protocol
stack without SSL**

**TCP/IP Protocol
stack with SSL**



Thank You



SS ZG513

Network Security

SSL Architecture – Introduction

Revision 1.0

BITS Pilani

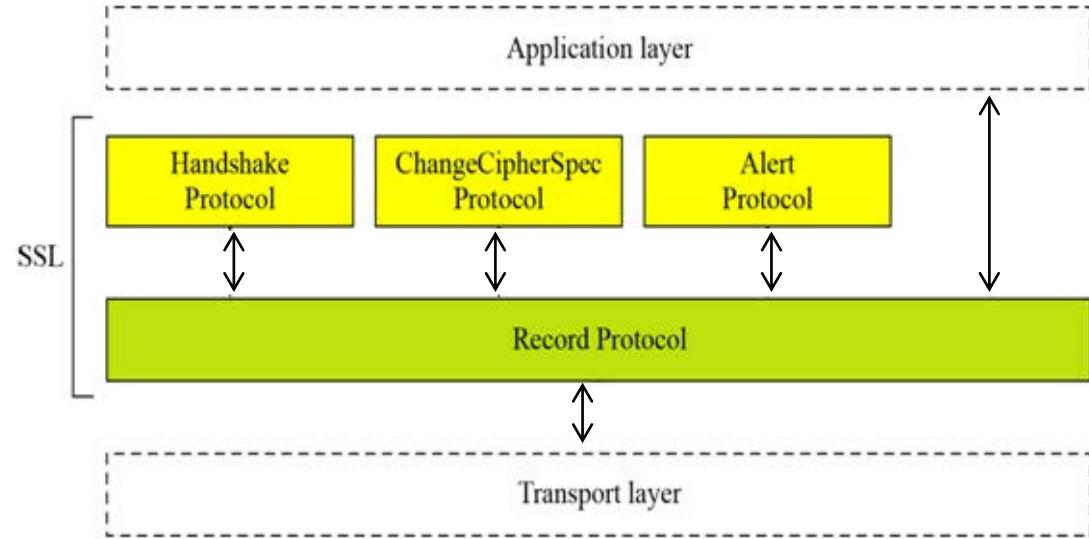
Work Integrated Learning Programmes



SSL Architecture - Introduction



- ❑ SSL is not a single protocol but rather two layers of protocols.
- ❑ **SSL Handshake Protocol** allows server and client to exchange different security parameters. This protocol performs its job before application data is transmitted.
- ❑ **SSL Change Cipher Spec Protocol** is used to update the cipher suite to be used for the connection.
- ❑ **SSL Alert Protocol** is used to convey SSL related alarms to the peer entities (client and server).
- ❑ **SSL Record Protocol** provides basic security services to the different application layer protocols. E.g. HTTP uses SSL to provide secure Web client/server interactions.





Thank You



SS ZG513

Network Security

SSL Handshake Protocol

Revision 1.0

BITS Pilani

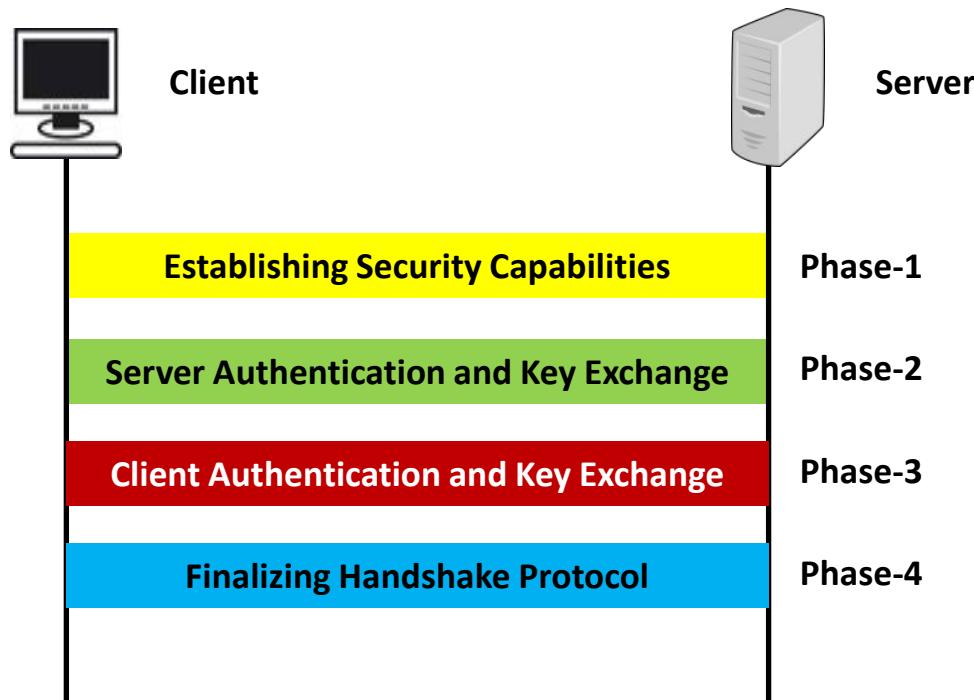
Work Integrated Learning Programmes



SSL Handshake Protocol



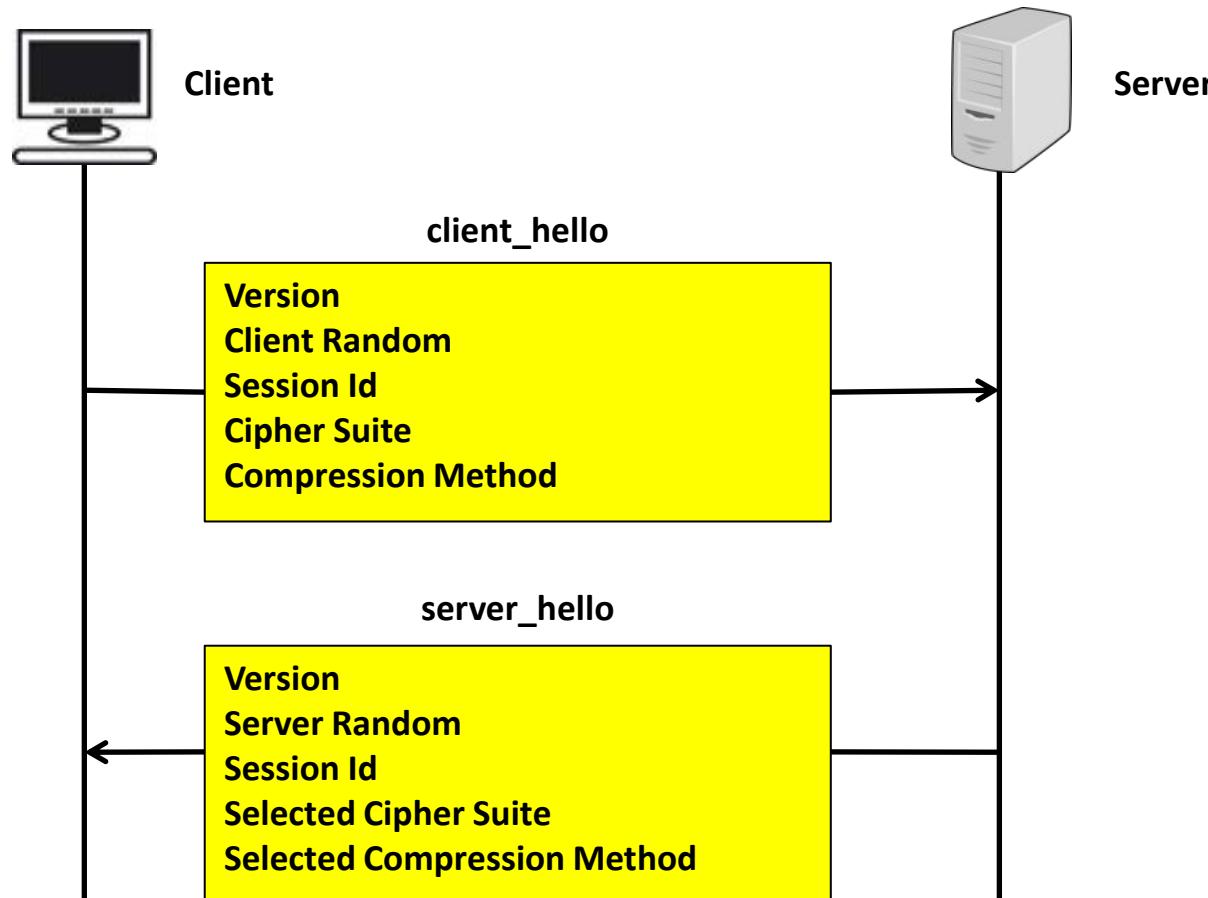
Introduction



SSL Handshake Protocol



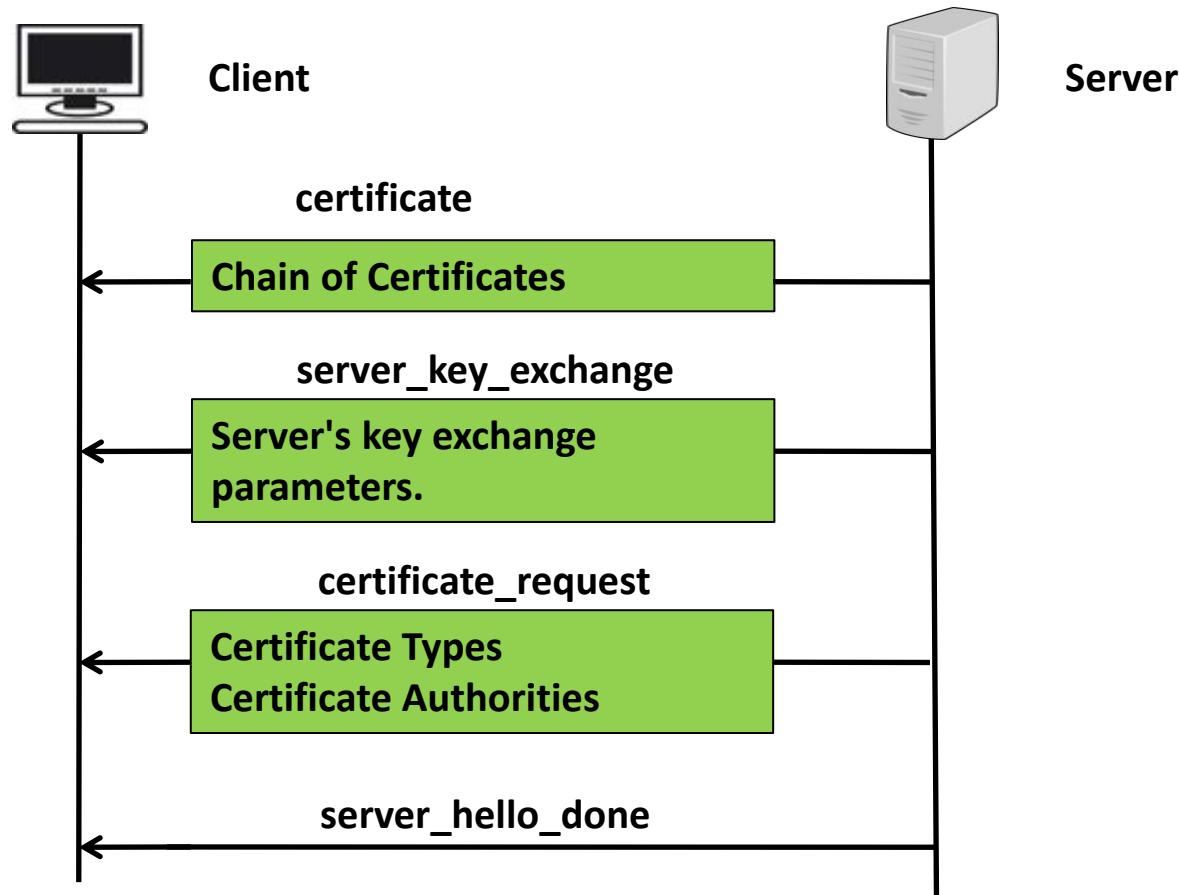
Phase-1: Establishing Securing Capabilities



SSL Handshake Protocol



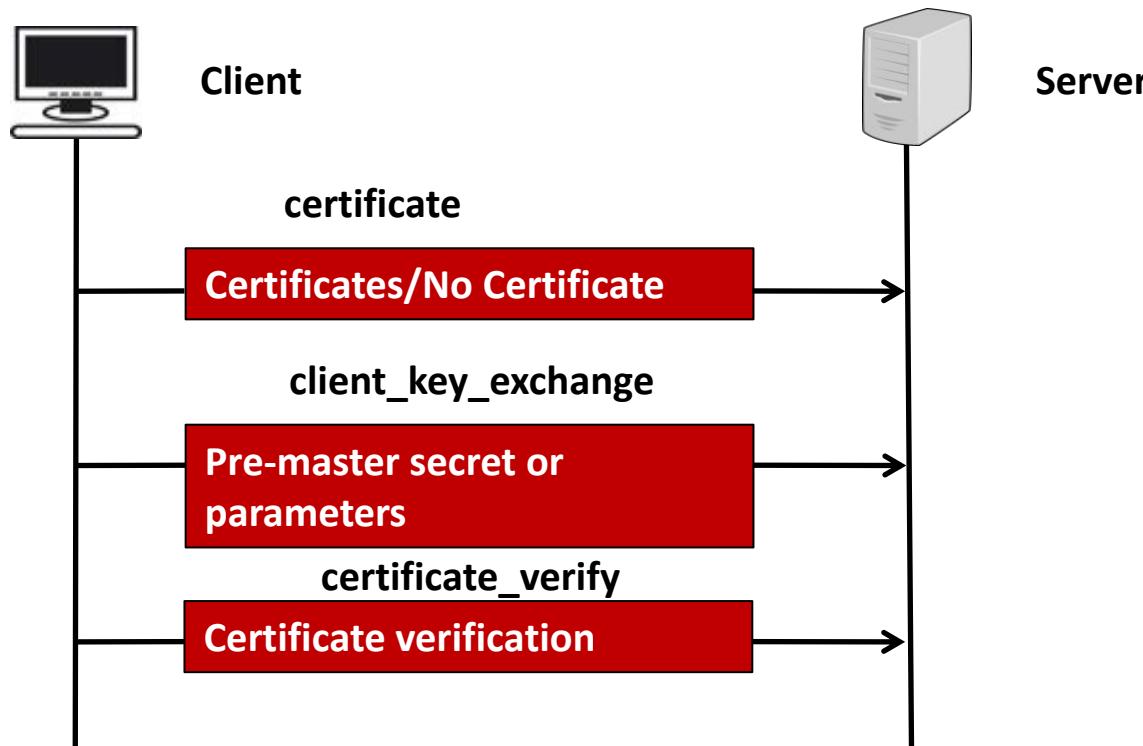
Phase-2: Server Authentication and Key Exchange



SSL Handshake Protocol



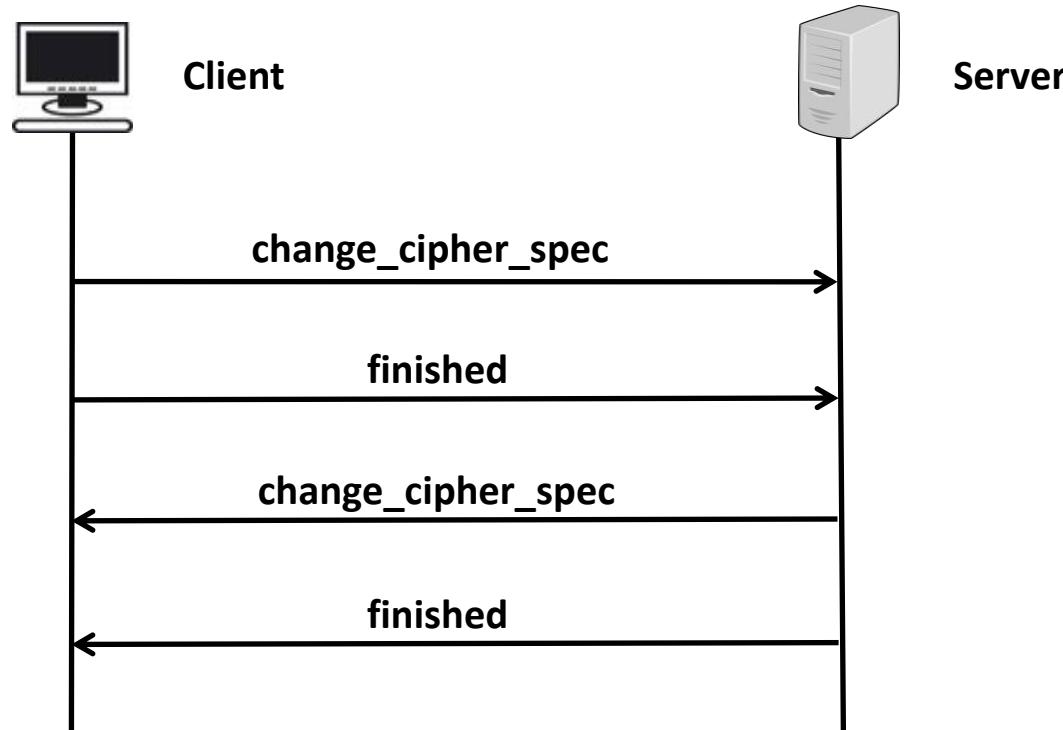
Phase-3: Client Authentication and Key Exchange



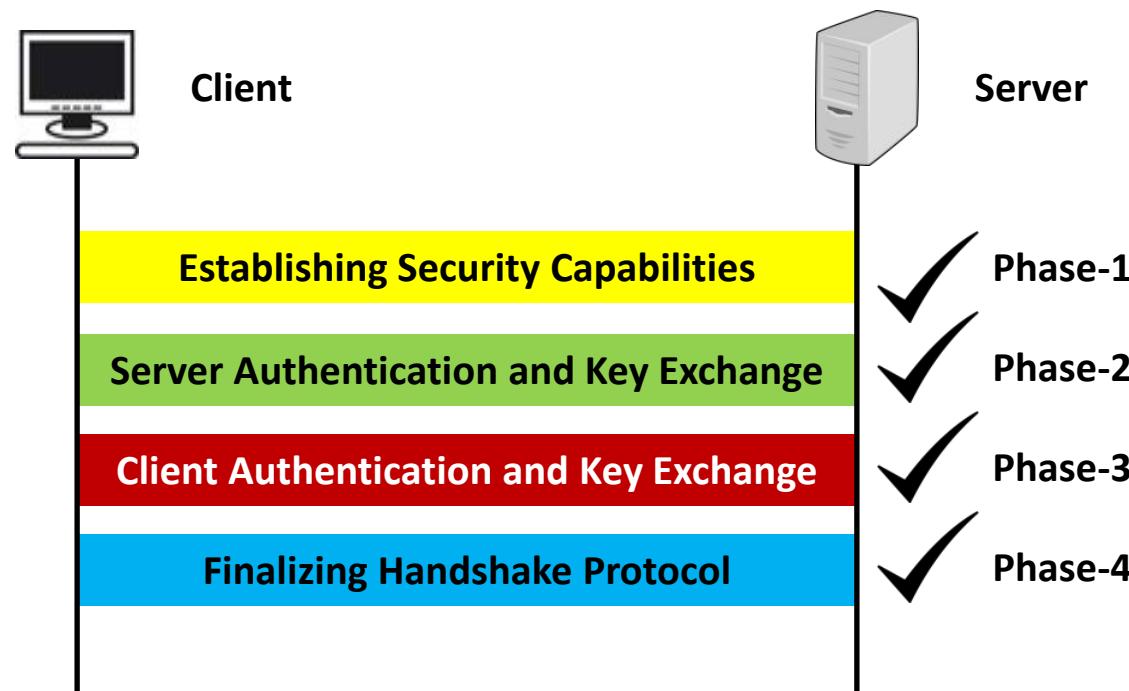
SSL Handshake Protocol



Phase-4: Finalizing Handshake Protocol



SSL Handshake Protocol





Thank You



SS ZG513

Network Security

SSL Change Cipher Spec Protocol

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes

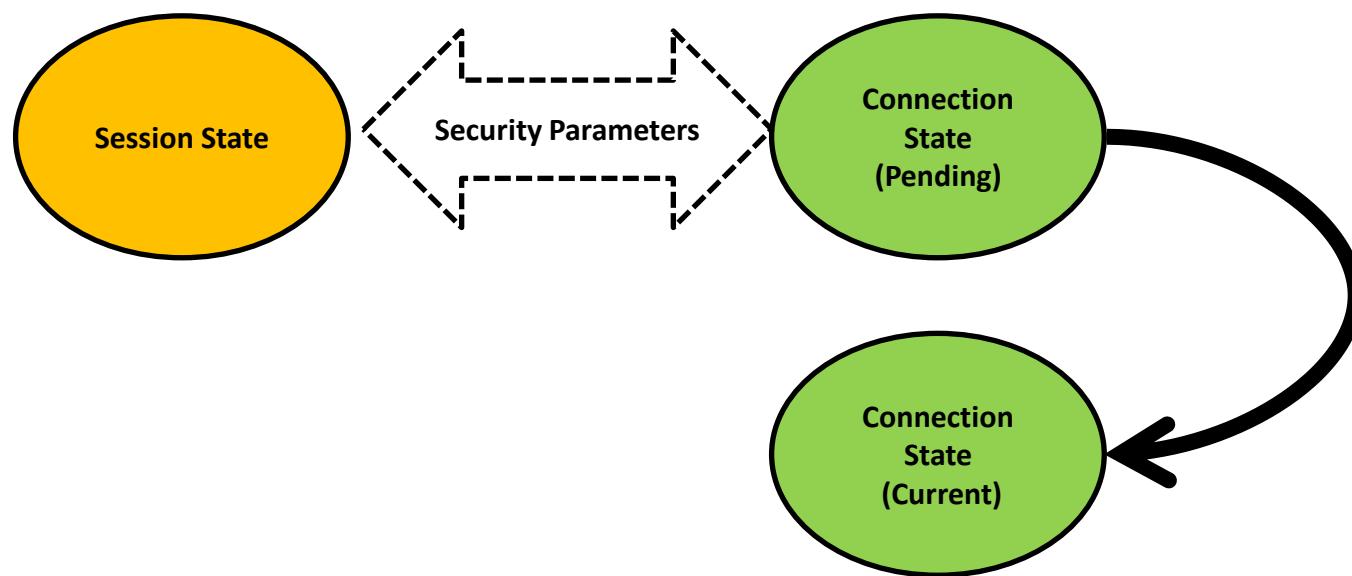


SSL Session and Connection

Session State Parameters
<ol style="list-style-type: none">1. Session Identifier2. Peer Certificate3. Compression Method4. Cipher Spec – algorithms for encryption and hash5. Master Secret – 48 bytes secret shared between client and server.6. Is resumable – can it be used to initiate new connections?

Connection State Parameters
<ol style="list-style-type: none">1. Server and client random numbers.2. Server write MAC key3. Client write MAC key4. Server write Encryption key5. Client write Encryption key6. Initialization Vectors – A value maintained if block ciphering cryptography is used.7. Sequence Numbers – Each peer maintains transmitted and received message sequence numbers. When change_cipher_spec message is received, it is set to 0.

SSL Change Cipher Spec Protocol





Thank You



SS ZG513

Network Security

SSL Alert Protocol

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



SSL Select Alert Messages

Alert Code	Alert Message	Description
10	unexpected_message	Received an inappropriate message. This alert should never be observed in communication between proper implementations. This message is always fatal.
20	bad_record_mac	Received a record with an incorrect MAC. This message is always fatal.
22	record_overflow	Received a SSL cipher text record which had a length more than $2^{14}+2048$ bytes, or a record decrypted to a SSL compressed record with more than $2^{14}+1024$ bytes. This message is always fatal.
30	decompression_failure	Received improper input, such as data that would expand to excessive length, from the decompression function. This message is always fatal.
40	handshake_failure	Indicates that the sender was unable to negotiate an acceptable set of security parameters given the options available. This is a fatal error.
47	illegal_parameter	Violated security parameters, such as a field in the handshake was out of range or inconsistent with other fields. This is always fatal.
48	unknown_ca	Received a valid certificate chain or partial chain, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA. This message is always fatal.
0	close_notify	Notifies the recipient that the sender will not send any more messages on this connection.
42	bad_certificate	There is a problem with the certificate, for example, a certificate is corrupt, or a certificate contains signatures that cannot be verified.
43	unsupported_certificate	Received an unsupported certificate type.
44	certificate_revoked	Received a certificate that was revoked by its signer.
45	certificate_expired	Received a certificate has expired or is not currently valid.
46	certificate_unknown	An unspecified issue took place while processing the certificate that made it unacceptable.



Thank You



SS ZG513

Network Security

SSL Record Protocol

Revision 1.0

BITS Pilani

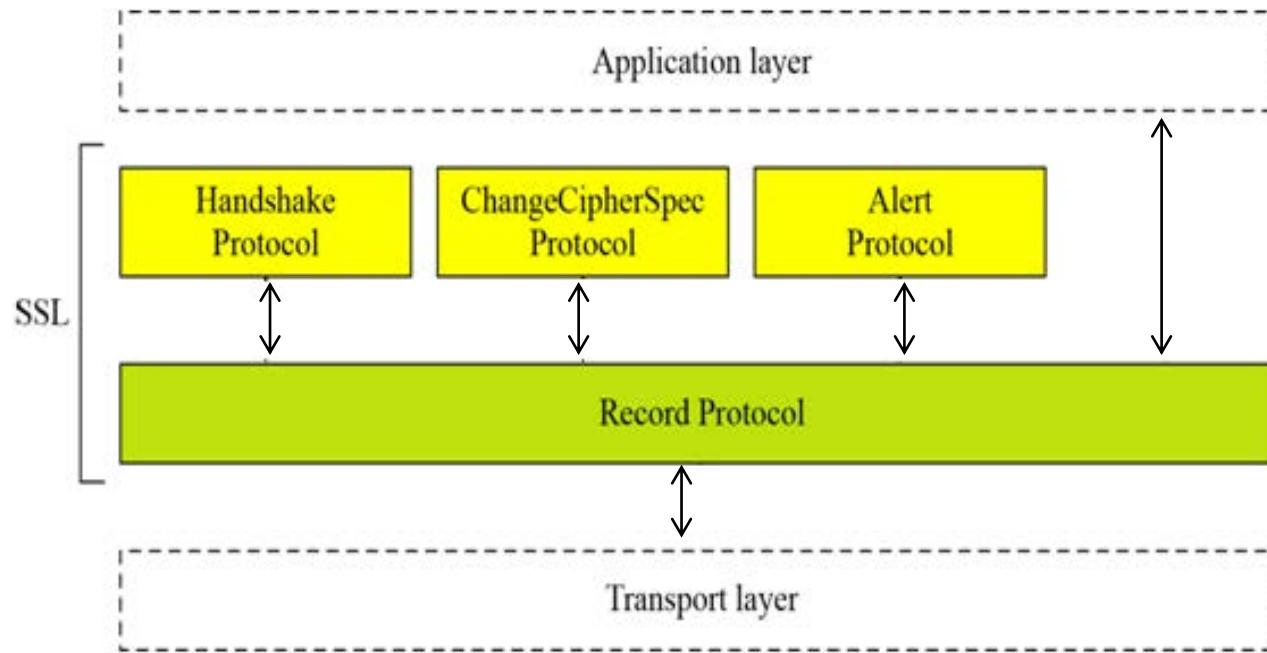
Work Integrated Learning Programmes



SSL Architecture



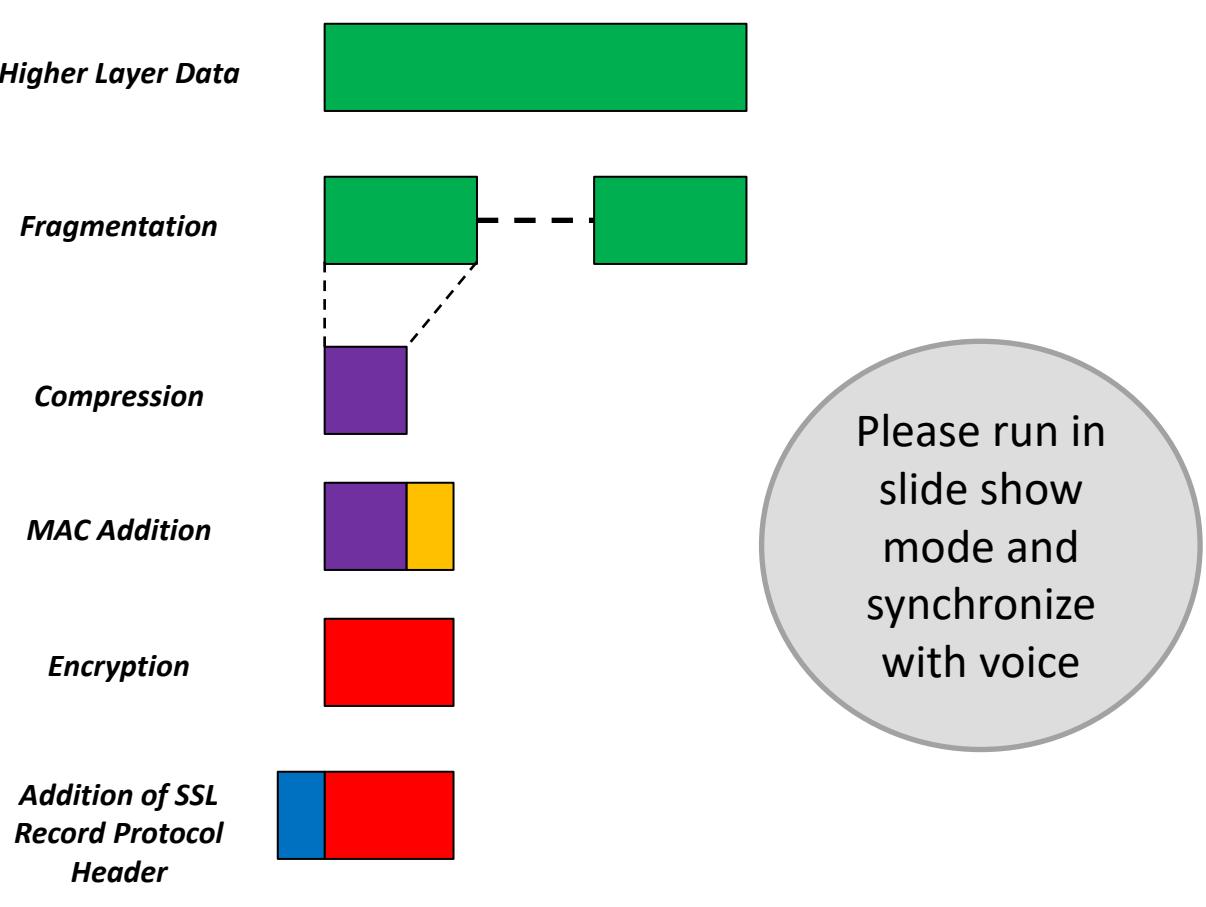
Detailed



SSL Record Protocol



Operation

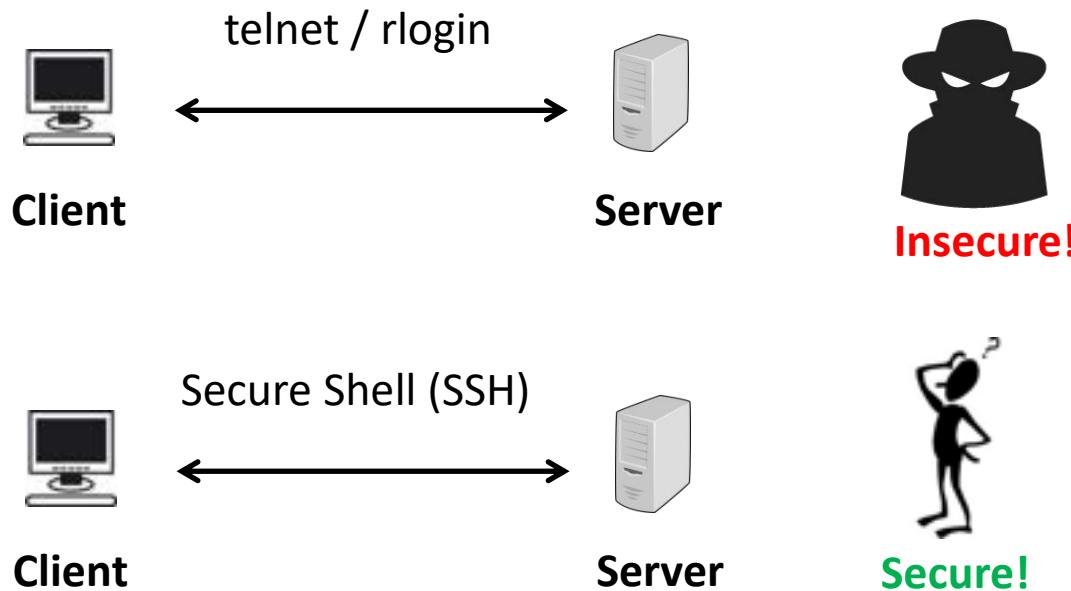


SSL Record Protocol Operation



Thank You

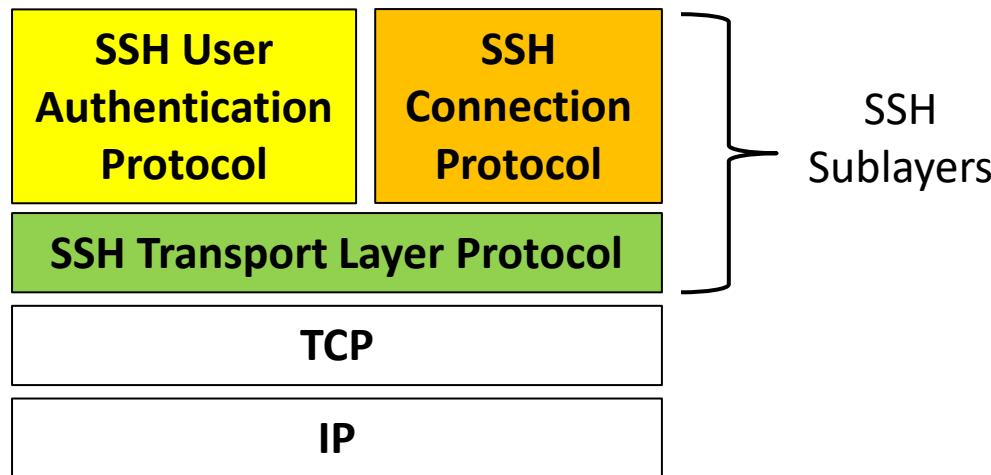
Remote Login and Security



SSH Protocol Stack



IETF RFC-4251

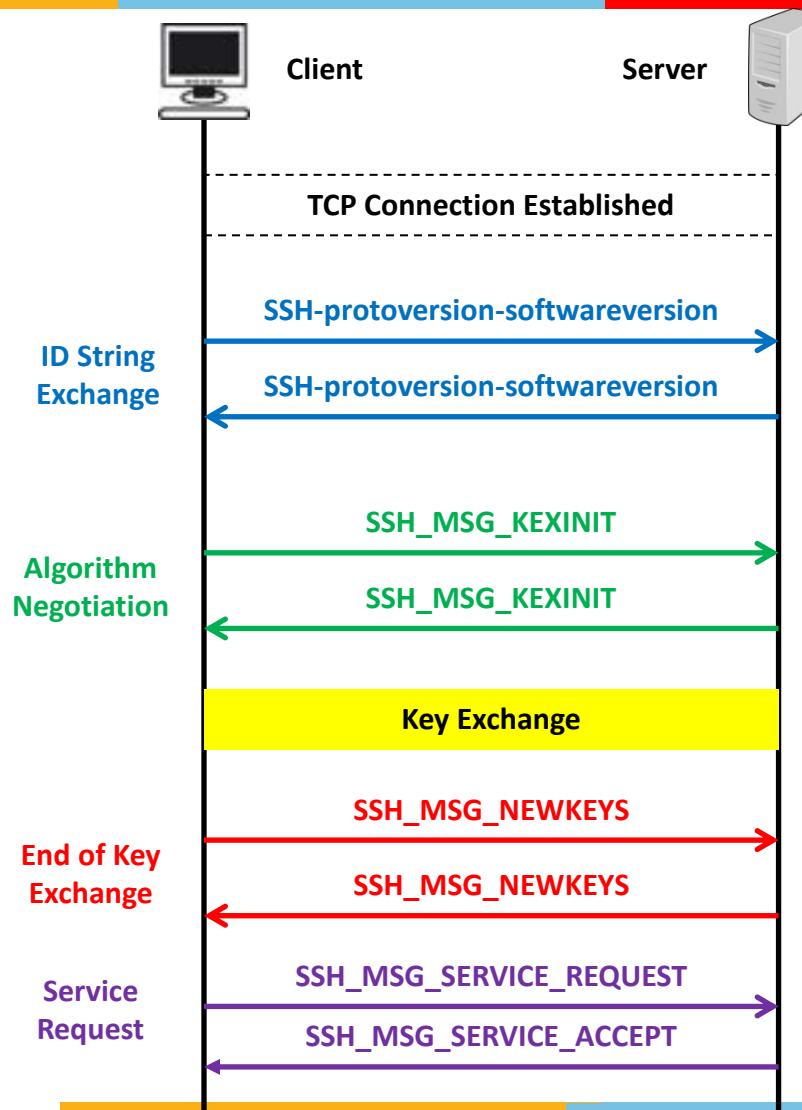


Described in
RFC-4251

SSH Transport Layer Protocol



IETF RFC-4253



Please run
in slide
show mode
and
synchronize
with voice

SSH User Authentication Protocol



Authentication Methods

- ❑ **Password:** The client sends a message containing a plaintext password, which is protected by encryption by the Transport Layer Protocol.
- ❑ **Public Key:** The client sends a message to the server that contains the client's public key signed by the client's private key. When the server receives this message, it checks whether the supplied key is acceptable for authentication and, if so, it checks whether the signature is correct.
- ❑ **Host Based:** Authentication is performed on the client's host rather than the client itself. Thus, a host that supports multiple clients would provide authentication for all its clients. This method works by having the client send a signature created with the private key of the client host. Thus, rather than directly verifying the user's identity, the SSH server verifies the identity of the client host.

SSH Connection Protocol



Channel Types

1. **session:** The remote execution of a program. The program may be a shell, an application such as file transfer or e-mail, a system command, or some built-in subsystem. Once a session channel is opened, subsequent requests are used to start the remote program.
2. **x11:** This refers to the X Window System, a computer software system and network protocol that provides a graphical user interface (GUI) for networked computers. X allows applications to run on a network server but to be displayed on a desktop machine.
3. **forwarded-tcpip:** This is remote port forwarding. Details follow.
4. **direct-tcpip:** This is local port forwarding. Details follow.



Thank You



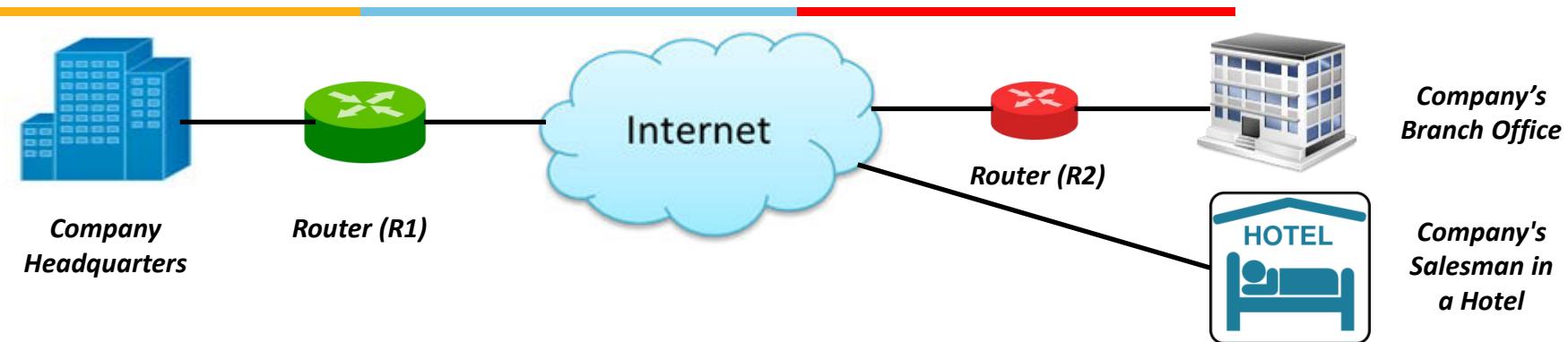
SS ZG513
Network Security
Security Challenges at the Network Layer
Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



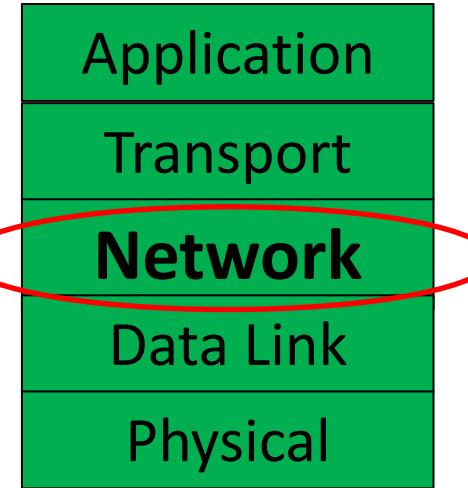
Need for the Network Layer Security



- A company's headquarters and branch offices are located in geographically distant areas. They exchange a lot of data through – emails, browser based applications etc.
- Company has few salespersons who keep travelling with their laptops (tablets or smart phones too). They also need to be in touch and work seamlessly.
- The company wants to have a *complete opaqueness* – outsiders must not know if emails or web browsing or remote logins or even ping are taking place among its employees.**
- How this company can provide required software and hardware infrastructure for a secure communication?
 - Custom made secured applications? *IP addresses, Ports, Protocol type will still be revealed.*
 - Dedicated hardware (its own routers and communication links everywhere)? *High Cost.*
 - As much as possible - off the shelf available applications and already available communication infrastructure (the Internet as shown above)?
- The last option sounds most feasible but in that case what will happen to the security of the 2 communication?

IP Security: Key Idea

- The layer-3 network layer (IP Layer in TCP/IP protocol stack) adds the ***IP Header*** to construct a layer-3 packet.
- As discussed in the previous slide, the important requirement is ***complete blanket security coverage***. So let us encrypt the whole layer-3 packet for confidentiality.
- If IP header is also encrypted how the IP datagram will be routed in the network then?
 - No network element will know what are source and destination IP addresses (part of the IP header).
 - There may be many intermediate network devices between source and destination. To who all the decryption key need to be given?
- What will happen to the authentication and integrity?
 - We are achieving confidentiality in the above point. But destination cannot check authentication and integrity.
- So what is the solution?***



Plain Text (un-encrypted) Layer-3 Packet

IP Header	Transport Header	Application Header and Data
-----------	------------------	-----------------------------

Encrypted Layer-3 Packet

IP Header	Transport Header	Application Header and Data
-----------	------------------	-----------------------------



Thank You



SS ZG513

Network Security

IP Security Overview

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



IP Security: Working

Run this in slide show and video should also show in the same manner and synchronize with voice



Plain Text (un-encrypted) Layer-3 packet



Plain Text Layer-3 packet



IP Security Header and Trailer are added

←----- *Encryption* -----→



Encryption is performed over the select portion

←----- *MAC* -----→

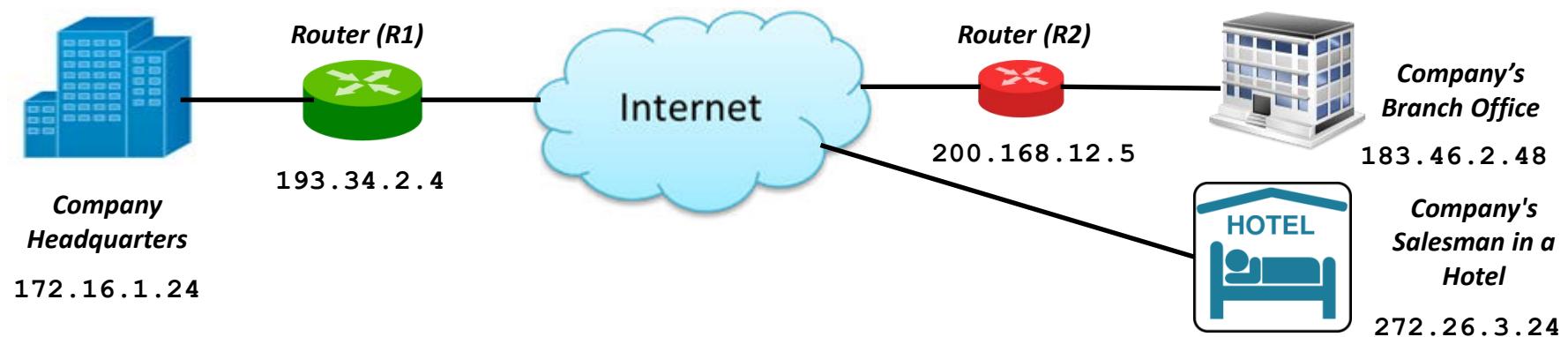


Message Authentication is appended for the select portion



New IP Header is added

Working of IP Security





Thank You



SS ZG513
Network Security
IP Security – Variations

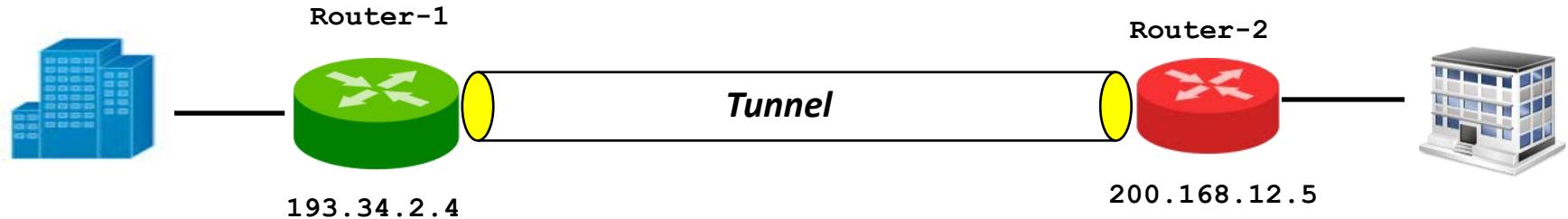
Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



IP Security – Tunnel Mode



IP Security - Transport Mode

Run this in slide show and video should also show in the same manner and synchronize with voice

IP Header is untouched all throughout.

Plain Text (un-encrypted) Layer-3 Packet



Plain Text Layer-3 Packet

IP Security Header and Trailer are added



←----- *Encryption* -----→

Encryption is performed over the select portion



←----- *MAC* -----→

Message Authentication is appended for the select portion



IP Security - Types

- Whatever has been discussed so far, encryption is mandatory but authentication (through authentication code) is optional. This IP Security protocol is called ***Encapsulating Security Payload (ESP)*** and standardized through IETF [RFC-4303](#).

- If encryption is never required, standardization allows it through ***Authentication Header (AH)*** IP Security Protocol with IETF [RFC-4302](#).



Thank You



SS ZG513
Network Security
IP Security Architecture
Revision 1.0

BITS Pilani

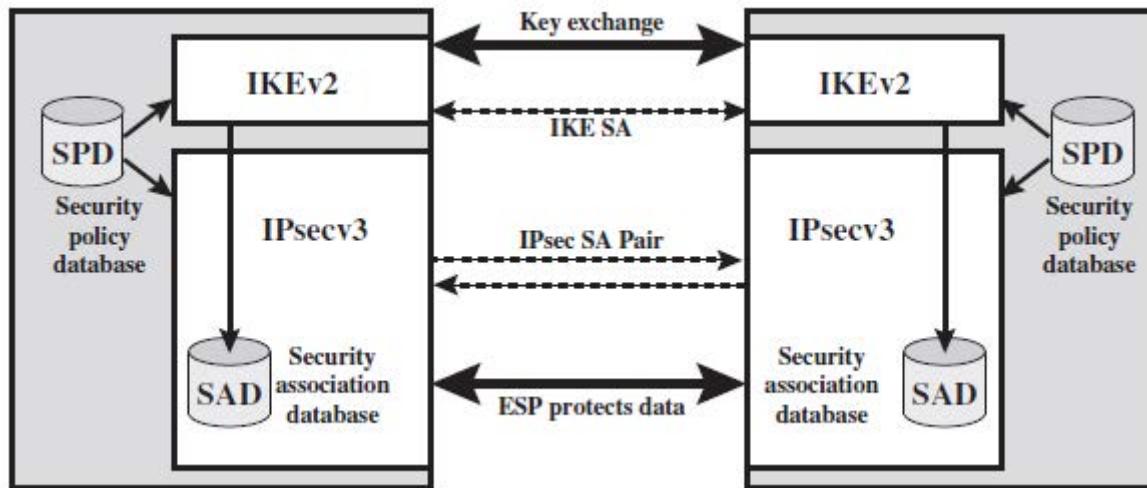
Work Integrated Learning Programmes



IP Security: Architecture (IPSec v3)



RFC-4301



Security Association (SA) and its Database (SAD)

- Before sending IPSec datagrams from the source to the destination entity, the source and destination entities create a network-layer logical connection. This logical connection is called a ***Security Association (SA)***.
- These Security Associations are stored in a database which is called ***Security Association Database (SAD)***.
- Important parameters are stored in this database:
 - Encryption Algorithms
 - Authentication Algorithms
 - Keys
 - IP Security Protocol – ESP or AH
 - IP Security Mode – Tunnel or Transport
 - Etc.

Security Policy (SP) and its Database (SPD)

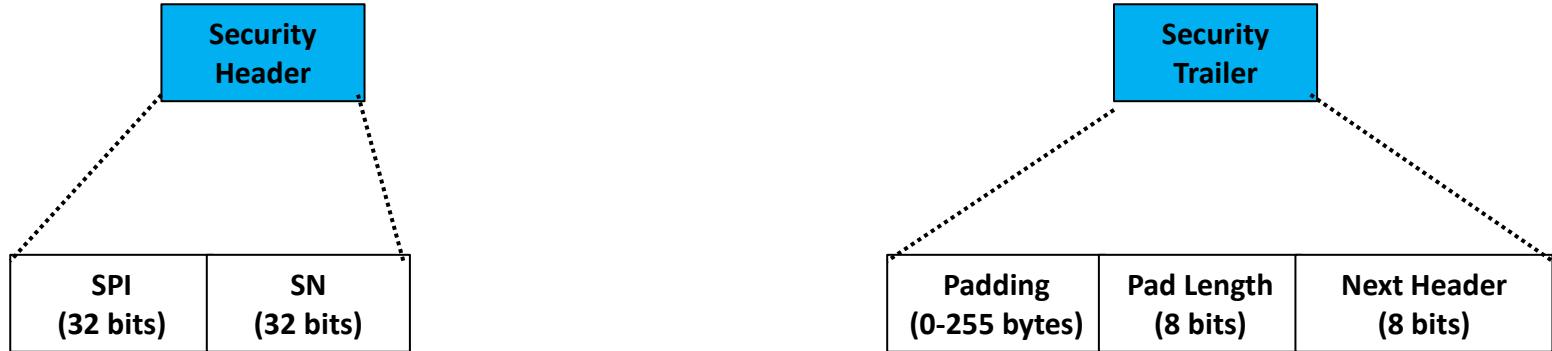


- In addition to the secure communication, a host in headquarters may want to access a web server (such as Amazon or Google) in the public Internet. It is not necessary to provide IP Security to the traffic of this type. So, router will transmit into the Internet both plain IP datagrams and secured IPSec datagrams based on some ***Security Policies***.
- These Security Policies are stored in a database which is called ***Security Policy Database (SPD)***.
- An entry within the SPD is identified with some keys (IP Addresses, Protocols, Ports) and then it decides what needs to be done for the traffic.

ESP: Security Header and Trailer



RFC-4303



As reviewed, there are two modes of ESP IPSec: Tunnel mode and Transport Mode. In both of these two modes, security header and security trailer are added. They are called ***ESP Header*** and ***ESP Trailer*** respectively. They consists of the following fields:

1. **SPI (Security Parameter Index, 32 bits):** Key identifier for an SA.
2. **SN (Sequence Number, 32 bits):** Monotonically increasing number. Assists in anti-replay.
3. **Padding (0-255 bytes):** Few encryption algorithms need plain text to be in a multiple of some bytes, so these dummy bytes may be added.
4. **Pad Length (8 bits):** The count of dummy padding bytes added.
5. **Next Header (8 bits):** Identifies the type of payload in application header and data field.



Thank You



SS ZG513
Network Security
IP Security - Packet Processing
Revision 1.0

BITS Pilani

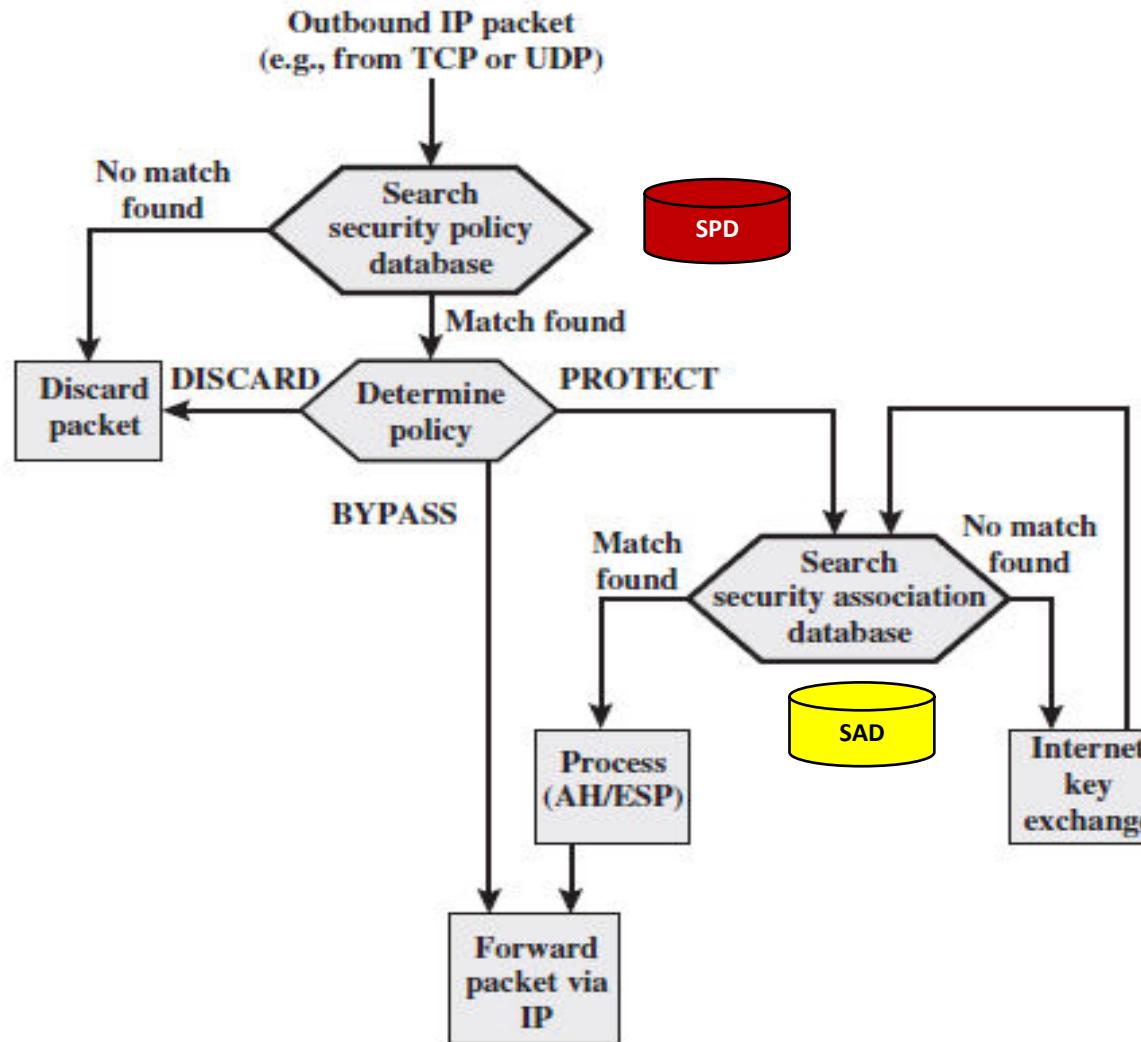
Work Integrated Learning Programmes



IP Security Policy



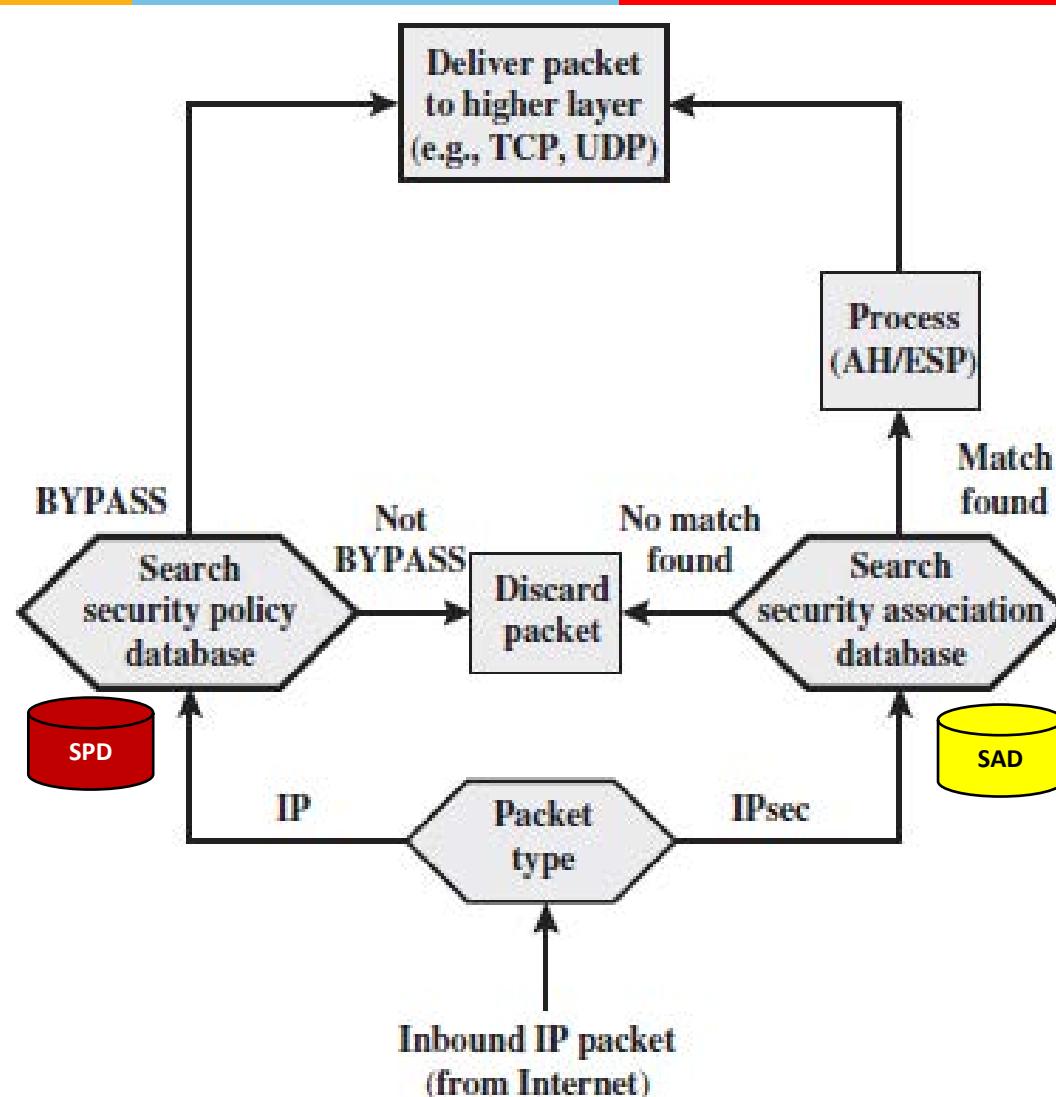
Model for Outbound Packets



IP Security Policy



Model for Inbound Packets





Thank You



SS ZG513

Network Security

Number Theory

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes

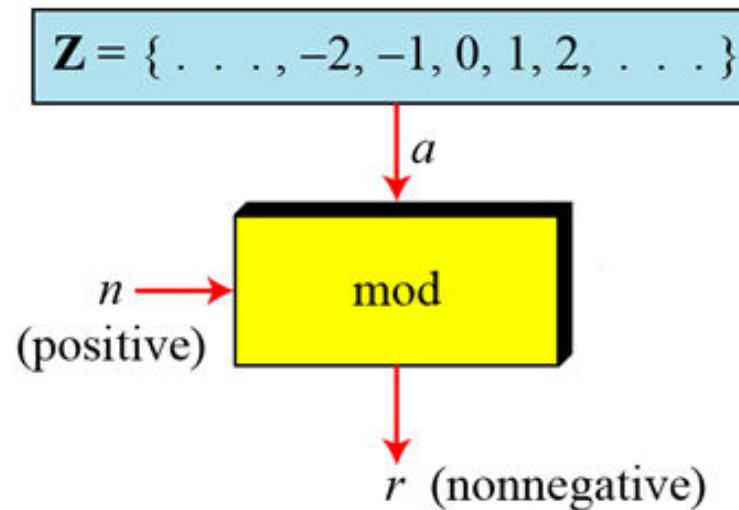


Numbers

- The ***set of integers***, denoted by Z , contains all integral numbers (with no fraction) from negative infinity to positive infinity.
 $Z = \{-\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty\}$
- A ***rational number*** is any number that can be expressed as p/q of two integers, p and q , with the denominator q not equal to zero. Since q may be equal to 1, every integer is a rational number. E.g. 23, 1.5, $22/7$, -7, 1.14141414.....
- An ***irrational number*** is any real number that cannot be expressed as p/q of two integers p and q . Irrational numbers cannot be represented as terminating or repeating decimals. E.g. $\pi = 3.1412857\dots$
- The ***real numbers*** include all rational and irrational numbers.
- ***Prime numbers*** has only two divisors 1 and itself.

Modular Arithmetic

- $a = q \times n + r$
- The input n is called the **modulus**.
- The output r is called the **remainder** or **residue**
- The operation is called **modulo** or just **mod**.
- The **mod** operator is defined in such a way that it gives the non-negative residue r as an output.
- If r is negative then n is added to it so that the final residue becomes positive.

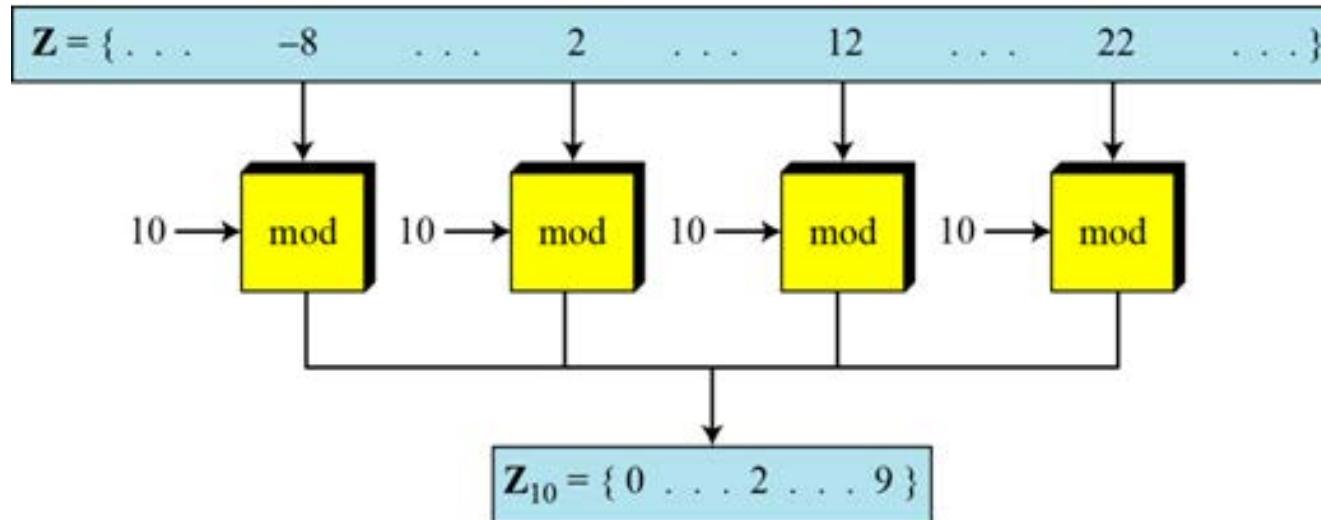


Examples

- $27 \bmod 5 = 2$
- $36 \bmod 12 = 0$
- $-18 \bmod 14 = -4 = -4+14 = 10$
- $-7 \bmod 10 = -7 = -7+10 = 3$

Congruence

- Two integers a and b are said to be congruent if $(a \bmod n) = (b \bmod n)$.
- It is written as $a \equiv b \pmod{n}$.



$$-8 \equiv 2 \equiv 12 \equiv 22 \pmod{10}$$

Congruence Relationship

Examples

- $2 \equiv 12 \pmod{10}$
- $3 \equiv 8 \pmod{5}$
- $8 \equiv 13 \pmod{5}$
- $-8 \equiv 12 \pmod{10}$



Thank You



SS ZG513
Network Security
GCD and Euclidean's Theorem

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Greatest Common Divisor (GCD)

- The greatest common divisor of a and b , $\gcd(a, b)$, is the largest integer that divides both a and b . Also it is defined that $\gcd(0, 0) = 0$.

- The greatest common divisor is to be positive, so
 - $\gcd(a, b) = \gcd(a, -b)$
 - $= \gcd(-a, b)$
 - $= \gcd(-a, -b)$

- In general, $\gcd(a, b) = \gcd(|a|, |b|)$

- a and b are **relatively prime** if $\gcd(a, b) = 1$

Examples

- GCD of 24 and 27 = 3
- GCD of 25, 34 = 1 and also 25 and 34 are relatively prime.
- GCD of -5, 30 = 5
- GCD of 16, -40 = 8

Euclidean Algorithm



To find out the GCD

q	r ₁	r ₂	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

GCD of 2740 and 1760 is 20



Thank You



SS ZG513

Network Security

Matrix Mathematics

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes

Matrix - Multiplication



Example-1

$$\begin{matrix} \text{C} & \text{A} & \text{B} \\ \left[\begin{matrix} 53 \end{matrix} \right] & = & \left[\begin{matrix} 5 & 2 & 1 \end{matrix} \right] \times \left[\begin{matrix} 7 \\ 8 \\ 2 \end{matrix} \right] \end{matrix}$$

→ ↓

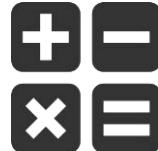
In which: $53 = 5 \times 7 + 2 \times 8 + 1 \times 2$

Example-2

$$\begin{matrix} \text{C} & \text{A} & \text{B} \\ \left[\begin{matrix} 52 \\ 41 \end{matrix} \right] & = & \left[\begin{matrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{matrix} \right] \times \left[\begin{matrix} 7 & 3 & 2 & 1 \\ 8 & 0 & 0 & 2 \\ 1 & 3 & 4 & 0 \end{matrix} \right] \end{matrix}$$

$$52 = 5 \times 7 + 2 \times 8 + 1 \times 1$$

Matrix - Determinant



1. If $m = 1$, $\det(\mathbf{A}) = a_{11}$
2. If $m > 1$, $\det(\mathbf{A}) = \sum_{i=1 \text{ and } j=1..m} (-1)^{i+j} \times a_{ij} \times \det(\mathbf{A}_{ij})$

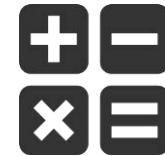
Where \mathbf{A}_{ij} is a matrix obtained from \mathbf{A} by deleting i^{th} row and j^{th} column.

$$\det \begin{bmatrix} 5 & 2 \\ 3 & 4 \end{bmatrix} = (-1)^{1+1} \times 5 \times \det [4] + (-1)^{1+2} \times 2 \times \det [3] \longrightarrow 5 \times 4 - 2 \times 3 = 14$$

Example

or
$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11} \times a_{22} - a_{12} \times a_{21}$$

Matrix – Inverse and Identity



Multiplicative Inverse (or just inverse) M^{-1} of a square matrix M is defined in such a way that $M \times M^{-1} = M^{-1} \times M = I$, where I is the identity matrix.

In an identity matrix I , all the elements are 0 except main diagonal elements from upper left to lower right which are all 1.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2x2 Identity Matrix

It is not always possible to have an integer multiplicative inverse of an integer matrix.

Residue Matrix

- $11 \times 19 = 209 = 1 \pmod{26}$
So, 11 and 19 are multiplicative inverse in modulo-26 arithmetic.
- Z is the set of integers. When a number is divided by n the remainder is always from 0 to $(n-1)$. Z_n represents this set of 0 to $(n-1)$ elements. E.g. $Z_5 = \{0, 1, 2, 3, 4\}$. **Z_n is the residue set.**

Cryptography uses residue matrices where all elements of a matrix are drawn from a set Z_n . If $n = 26$, it means all elements of a Z_{26} matrix will be drawn from $\{0, 1, 2, \dots, 25\}$.

- A residue matrix Z_n will have a multiplicative inverse matrix, if the determinant of that matrix has a multiplicative inverse in set Z_n .

Mathematically, if **$\text{GCD}(\det(A), n) = 1$** for a matrix A , it will have a multiplicative inverse. GCD is Greatest Common Divisor.

Multiplicative Inverse in Z_n

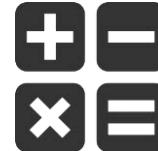


Extended Euclidean Method

q	n1	n2	r	t1	t2	t
2	26	11	4	0	1	-2
2	11	4	3	1	-2	5
1	4	3	1	-2	5	-7
3	3	1	0	5	-7	26
	1	0		-7	26	

- Let us find out if 11 (n2) has a multiplicative inverse in Z_{26} ($n1=26$) using **Extended Euclidean Method**. Here, q = quotient and r = remainder.
- In the beginning, the temporary numbers, $t1 = 0$ and $t2 = 1$ and $t = t1 - q.t2$ for all the steps.
- After all the operations, since n1 and n2 reduces to 1 and 0, so their GCD is 1 and hence there is a multiplicative inverse of 11 in Z_{26} .
- Multiplicative inverse = (last $t1$) mod 26 = -7 mod 26 = 19
- Verification: $(19 \times 11) \text{ mod } 26 = 209 \text{ mod } 26 = 1$.
- **Hence, 11 and 19 are multiplicative inverse in Z_{26} .**

Matrix - Multiplicative Inverse



If a square matrix A has a non-zero determinant then the multiplicative inverse of the matrix is calculated as:

$$[A^{-1}]_{ij} = (\det(A))^{-1} \cdot (-1)^{i+j} \cdot D_{ji}$$

Where:

$(\det(A))^{-1}$ = Multiplicative inverse of $\det(A)$ in Z_{26} . It means using extended Euclidean method the GCD of $\det(A)$ and 26 is to be found out. If it is 1, it means $(\det(A))^{-1}$ exists.

D_{ji} = Determinant of the matrix deleting j^{th} row and i^{th} column.

Refer to the worksheet for detailed example.



Thank You



SS ZG513

Network Security

Fermat's Theorem

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes





Fermat's Theorem

- Named after 17th century French mathematician **Pierre de Fermat**.
- Also called Fermat's Little Theorem.
- It states that if p is a prime number and a is a positive integer not divisible by p then:

$$a^{p-1} \equiv 1 \pmod{p}$$
 ----- 1st Version

$$a^p \equiv a \pmod{p}$$
 ----- 2nd Version

Fermat's Theorem



Examples

$$(1) 6^{10} \bmod 11$$

Comparing it with the 1st version of Fermat's theorem, $a^{p-1} \equiv 1 \pmod{p}$

$$6^{11-1} \equiv 1 \pmod{11}$$

$$\text{So, } 6^{10} \bmod 11 = 1$$

$$(2) 4^7 \bmod 7$$

Comparing it with the 2nd version of Fermat's theorem, $a^p \equiv a \pmod{p}$

$$4^7 \bmod 7 = 4$$

Fermat's Theorem



Application

Fermat's theorem eliminates the need for Extended Euclidean's Algorithm if modulus is prime by calculating multiplicative inverse in the following way:

$$a^{-1} \text{ mod } p = a^{p-2} \text{ mod } p$$

Example: $4^{-1} \text{ mod } 5 = 4^3 \text{ mod } 5 = 4 \text{ (mod } 5)$

How it eliminates the need of Extended Euclidean Algorithm:

Multiplying both the sides with a

$$\begin{aligned} a \times a^{-1} \text{ mod } p &= a \times a^{p-2} \text{ mod } p \\ &= a^{p-1} \text{ mod } p \\ &= 1 \text{ mod } p \text{ (from Fermat's Theorem 1st version)} \end{aligned}$$

Or, $a \times a^{-1} = 1$ in modulus p ; the same as Extended Euclidean's Algorithm.



Thank You



SS ZG513

Network Security

Euler's Theorem

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes

Set of Multiplicative Inverse

- The ***set of integers***, denoted by Z , contains all integral numbers (with no fraction) from negative infinity to positive infinity.
 $Z = \{-\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty\}$
- Z_n all the elements from 0 to $(n-1)$.
Example: $Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Z_n^* has only those elements drawn from 0 to $(n-1)$ which have multiplicative inverse in Z_n .
Example: $Z_{10}^* = \{1, 3, 7, 9\}$
 $1 \times 1 \equiv 1 \pmod{10}$
 $3 \times 7 \equiv 1 \pmod{10}$
 $9 \times 9 \equiv 1 \pmod{10}$
- There are no other integers in Z_{10} other than 1, 3, 7 and 9 meeting this quality of multiplicative inverse, so $\{1, 3, 7, 9\}$ are the only elements of Z_{10}^* .

Euler's Totient Function



Basics

- Named after 18th century Swiss mathematician **Leonhard Euler** who extensively worked in the area of prime numbers.
- Euler's Totient Function, $\phi(n)$, which is sometimes called the **Euler's Phi-Function** plays a very important role in cryptography.
- Euler's Totient Function, $\phi(n)$ calculates the count of elements in this set Z_n^* .
 1. $\phi(1) = 0$
 2. $\phi(p) = p-1$, if p is prime
 3. $\phi(m \times n) = \phi(m) \times \phi(n)$, if m and n are relatively prime
 4. $\phi(p^a) = p^a - p^{a-1}$, if p is prime

Combining 3rd and 4th rule, $\phi(q) = (p_1^{a1} - p_1^{a1-1}) \times (p_2^{a2} - p_2^{a2-1}) \times \dots \times (p_n^{an} - p_n^{an-1})$
as q can be factored as $p_1^{a1} \times p_2^{a2} \times p_3^{a3} \times \dots \times p_n^{an}$

Euler's Totient Function



Examples

Example-1:

$$\begin{aligned}\phi(13) \\ = 13 - 1 = 12\end{aligned}$$

Example-2:

$$\begin{aligned}\phi(10) \\ = \phi(2) \times \phi(5) = (2-1) \times (5-1) = 1 \times 4 = 4\end{aligned}$$

Example-3:

$$\begin{aligned}240 = 2^4 \times 3^1 \times 5^1 \\ \text{So, } \phi(240) = (2^4 - 2^3) \times (3^1 - 3^0) \times (5^1 - 5^0) = 8 \times 2 \times 4 = 64\end{aligned}$$

Example-4:

$$\begin{aligned}\text{The number of elements in } Z_{14}^* \\ \phi(14) \\ = \phi(2) \times \phi(7) = (2-1) \times (7-1) = 1 \times 6 = 6\end{aligned}$$

Euler's Theorem

For every a and n that are relatively prime, the following relationship exists:

$$\begin{aligned} a^{\phi(n)} &\equiv 1 \pmod{n} && 1^{st} \text{ Version} \\ a^{\phi(n)+1} &\equiv a \pmod{n} && 2^{nd} \text{ Version} \end{aligned}$$

Example-1:

Find out the value of $6^{24} \pmod{35}$.

Because, $\phi(35) = \phi(5) \times \phi(7) = 4 \times 6 = 24$

So, $6^{24} \pmod{35} = 6^{\phi(35)} \pmod{35} = 1$

Example-2:

Find out the value of $6^{61} \pmod{77}$.

Because, $\phi(77) = \phi(7) \times \phi(11) = 6 \times 10 = 60$

So, $6^{61} \pmod{77} = 6^{\phi(77)+1} \pmod{77} = 6$

Euler's Theorem



Application

Euler's theorem eliminates the need for Extended Euclidean's Algorithm by calculating multiplicative inverse in the following way:

$$a^{-1} \bmod n = a^{\phi(n)-1} \bmod n$$

Examples:

$$\begin{aligned} 4^{-1} \bmod 5 &= 4^{\phi(5)-1} \bmod 5 \\ &= 4^{4-1} \bmod 5 \\ &= 64 \bmod 5 \\ &= 4 \end{aligned} \quad \begin{aligned} 3^{-1} \bmod 11 &= 3^{\phi(11)-1} \bmod 11 \\ &= 3^{10-1} \bmod 11 \\ &= 3^9 \bmod 11 \\ &= 4 \end{aligned}$$



Thank You



SS ZG513

Network Security

Primitive Roots

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Primitive Roots

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}
1	1	1	1	1	1	1	1	1	1
2	4	8	5	10	9	7	3	6	1
3	9	5	4	1	3	9	5	4	1
4	5	9	3	1	4	5	9	3	1
5	3	4	9	1	5	3	4	9	1
6	3	7	9	10	5	8	4	2	1
7	5	2	3	10	4	6	9	8	1
8	9	6	4	10	3	2	5	7	1
9	4	3	5	1	9	4	3	5	1
10	1	10	1	10	1	10	1	10	1

Primitive Roots of 11 are 2, 6 , 7 and 8

 Full Length Sequence. Going from a to a^{10} without repetition.

 Partial Length Sequence. Sequence repeats before reaching a^{10}

Where we have Primitive Roots.

- Not all the integers have primitive roots.
- Integers in the form 2 , 4 , p^x and $2p^x$ have primitive roots.
Where p is any odd prime and x is a positive integer.
- The concept of Primitive Roots are used in Asymmetric Key Cryptography and Diffie-Hellman Key Exchange Algorithms.



Thank You



SS ZG513

Network Security

Distribution of Symmetric Keys

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Distribution of Symmetric Key



Using Symmetric Encryption

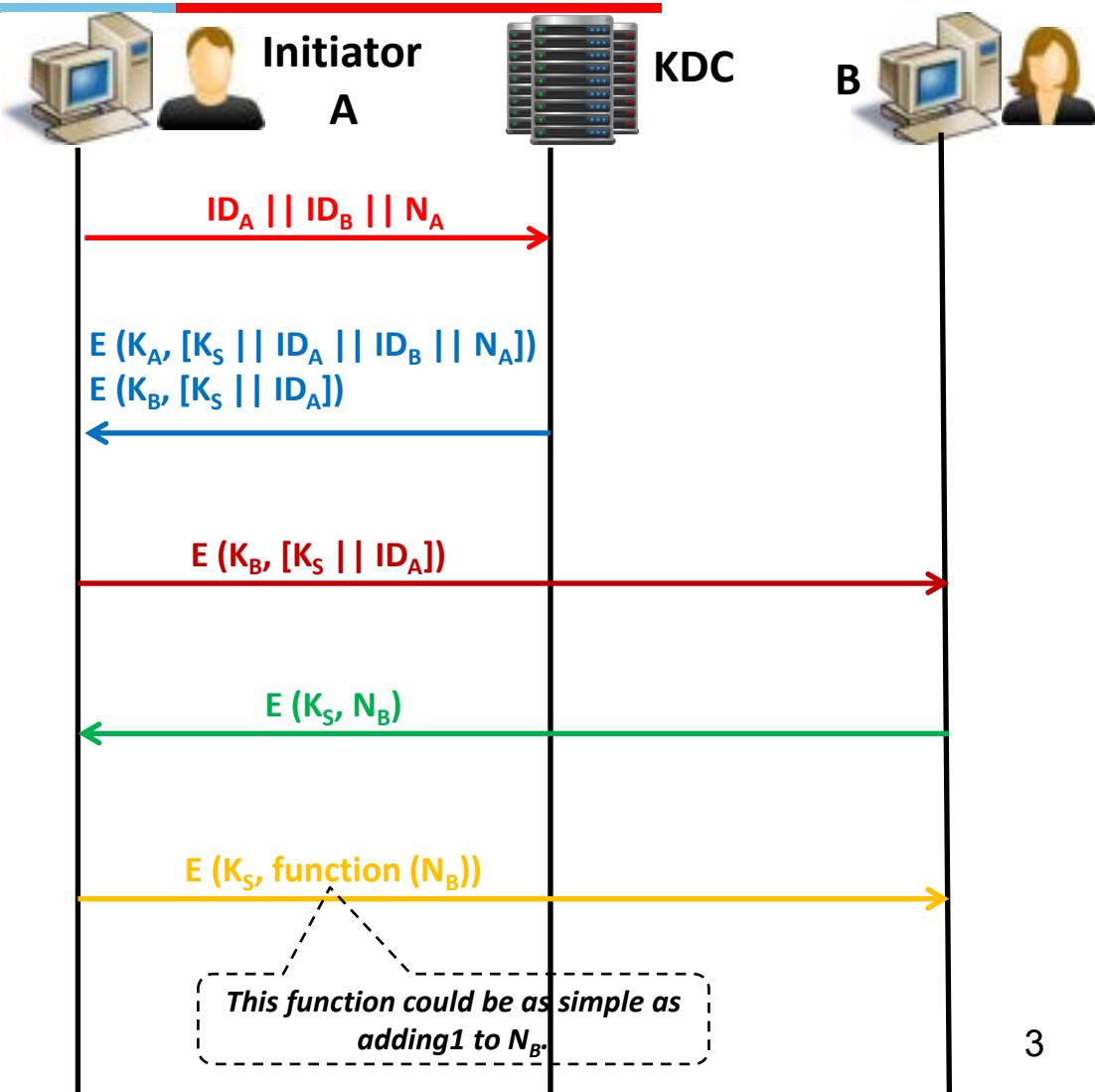
- ❑ For symmetric key encryption, sender and receiver must share the same key.
- ❑ Frequent changes (renewals) would be desirable in this shared key, to avoid attacks.
- ❑ So delivering the shared key to two parties is crucial for symmetric encryption to sustain successfully. This is called Key Management and Distribution for Symmetric Encryption.
- ❑ In a large network of N parties, it would need $[N(N-1)/2]$ keys for symmetric encryption. The count of keys grows rapidly as N becomes large.
- ❑ If the two parties are A and B, there are different possibilities to share the key:
 - i. A can select a key and physically deliver it to B.
 - ii. A third party can select the key and physically deliver it to A and B.
 - iii. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
 - iv. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B.
- ❑ The possibilities i and ii are not feasible in the modern communication world.
- ❑ Option iii is a possibility, but if an attacker gains access to one key, all other subsequent keys will be compromised. Initial distribution of the first key will still be a challenge.
- ❑ Option iv is most suitable and adopted in the communication networks using a Key Distribution Centre (KDC).

Key Distribution Scenario



With KDC in the loop

- Users A and B have master keys K_A and K_B respectively shared with only KDC.
- User A wants to establish the connection with B and contacts KDC with its own ID, B's ID and its nonce N_A .
- KDC responds with a message having two data items:
 - i. One time session key K_S and the original A's message encrypted with A's master key (K_A) known only to A and KDC.
 - ii. Same session key K_S and A's ID encrypted with B's master key (K_B) known only to B and KDC.
- A stores the session key K_S and forwards the second data item to B. Session Key K_S is delivered to both the parties.
- Using the new session key, B sends its encrypted nonce N_B to A.
- After receiving N_B , A performs some function on N_B and sends it to B encrypted with K_S .



Review Questions!

- i. Do you see any vulnerability because initiator did not encrypt the very first message?
- ii. Why initiator A included its own id and B's id in its original first message to KDC?
- iii. Why initiator A included a nonce N_A in its original first message to KDC?
- iv. Why KDC replied with the session key K_s in the first data element encrypted in A's master key?
- v. Why KDC included A's original message in its response to A in the first data item?
- vi. Why KDC prepared second data item encrypted with B's master key?
- vii. Why KDC included A's identity in the data item that was encrypted with B's master key?
- viii. Why B also sent it nonce N_B only to A and not to KDC?
- ix. Why A performed some function on N_B and sent it back to B?
- x. Why A and B used session key K_s for the last two messages?
- xi. Could A and B use their respective master keys (K_A and K_B) for the last two messages for encryption?
- xii. A and B are authenticated with KDC as they shared master keys (K_A and K_B) with KDC. Are A and B authenticating each other? If this is an issue, was it being taken care of? How?



Thank You



SS ZG513

Network Security

Distribution of Asymmetric (Public) Keys

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



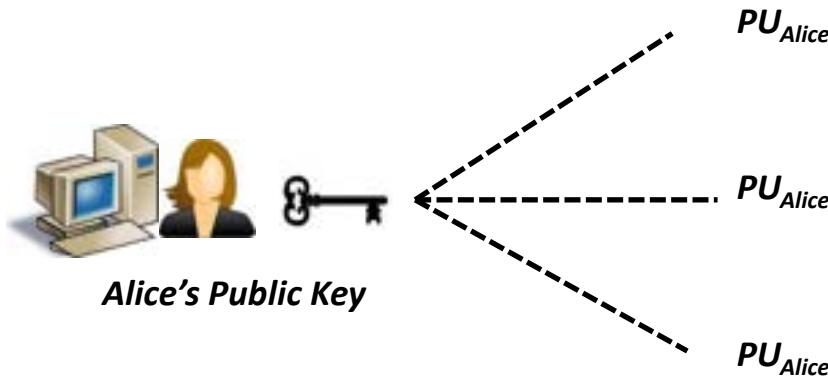
Distribution of Public Keys



Public Announcement

Any participant can send his or her Public Key (PU) to any other participant or broadcast the key to the community at large.

Drawback: Anyone can forge such a public announcement. E.g. some user could pretend to be user A and send a public key to another participant or broadcast such a public key. Until such time as user A discovers the forgery and alerts other participants, the forger is able to read all encrypted messages intended for A.

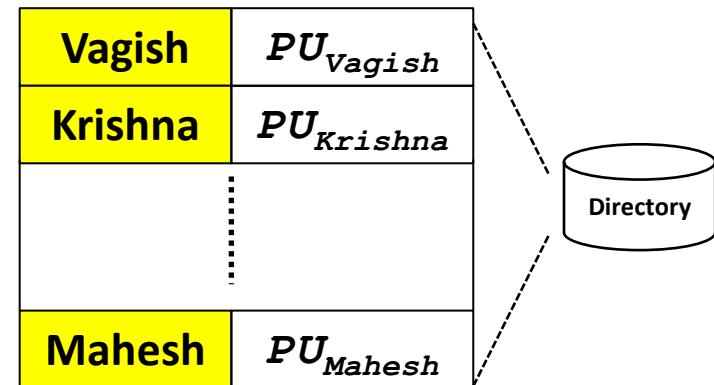


Distribution of Public Keys



Publicly Available Directory

1. A trusted organization or authority maintains a directory with a $\{Name, Public\ Key\}$ entry for each participant.
2. Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.
3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.
4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.
5. Safer than public announcement but confidentiality and integrity of the directory is crucial.

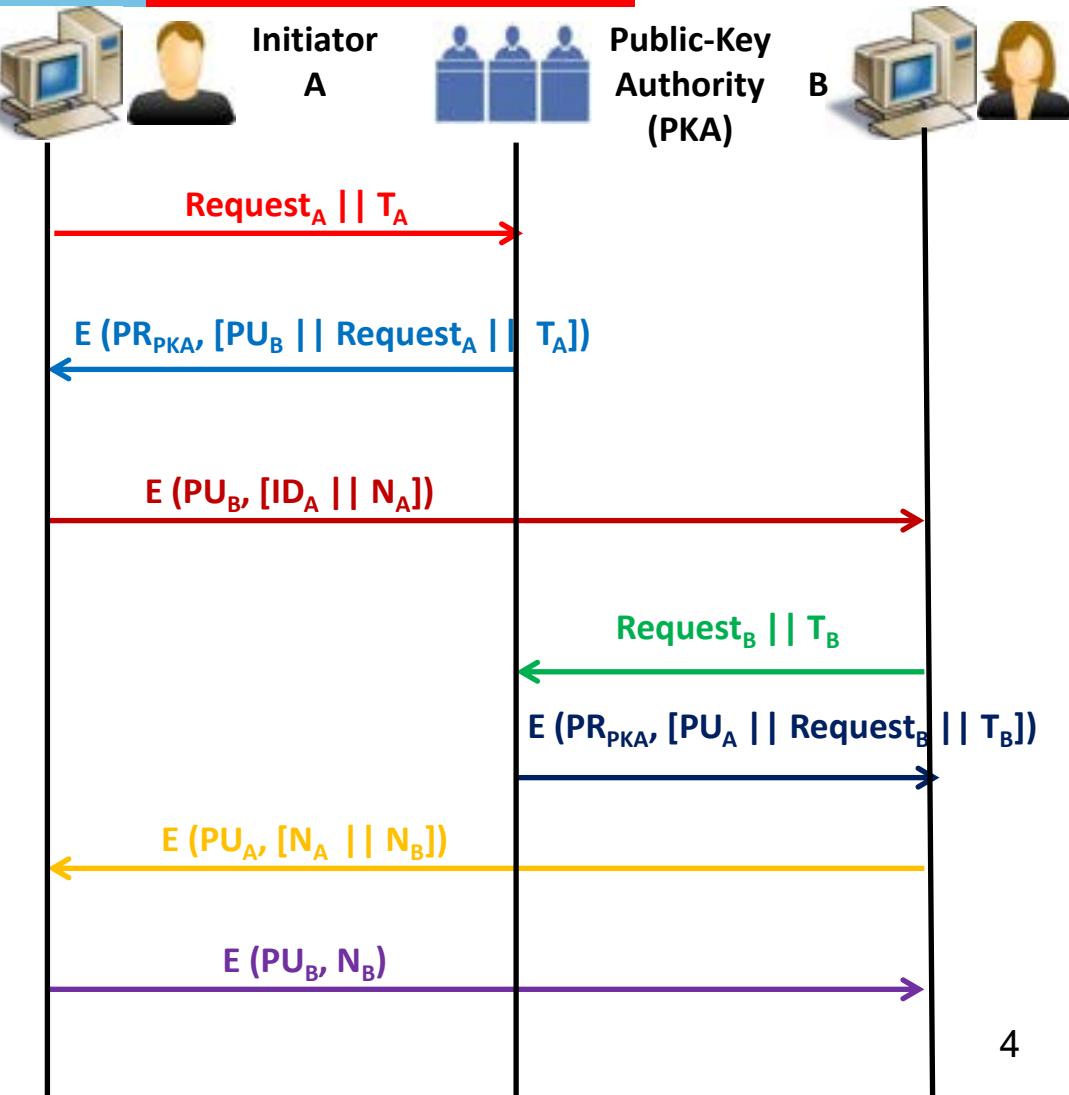


Distribution of Public Keys



Public Key Authority

- ❑ Initiator A requests the Public-Key Authority (PKA) the public key of B. The message is time stamped T_A .
- ❑ PKA sends an encrypted response with its private key. The message contains the public key of B (PU_B) and the original request and time stamp sent by A.
- ❑ A saves B's public key and sends an encrypted message with B's public key (PU_B). The message contains A's identity (ID_A) and a nonce generated by A (N_A).
- ❑ B also gets the public key of A (PU_A) in the same manner from PKA.
- ❑ At this point of time, both A and B have each other's public keys.
- ❑ B sends an encrypted message with the public key of A. The message contains the nonce generated by A and a new nonce generated by B.
- ❑ A responds with the nonce generated by B encrypted with the public key of B.



Review Questions!

- i. Why A time stamped his first message and why PKA included this time stamp in his response to A?
- ii. When PKA provided B's public key to A, the message was encrypted with the private key of the PKA itself. How A will decrypt it?
- iii. Why A contacted B with its id and its nonce in a message encrypted with B's public key?
- iv. Why B responded to A with A's nonce and its own nonce?
- v. Why A replied B with B's nonce and why the message was encrypted with the public key of B?
- vi. Can we conclude for PKA mechanism to work, PKA's public key must be known to the other users first?
- vii. What are the disadvantages of this scheme?



Thank You



SS ZG513

Network Security

Public Key Certificates

Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Distribution of Public Keys



Public-Key Certificates

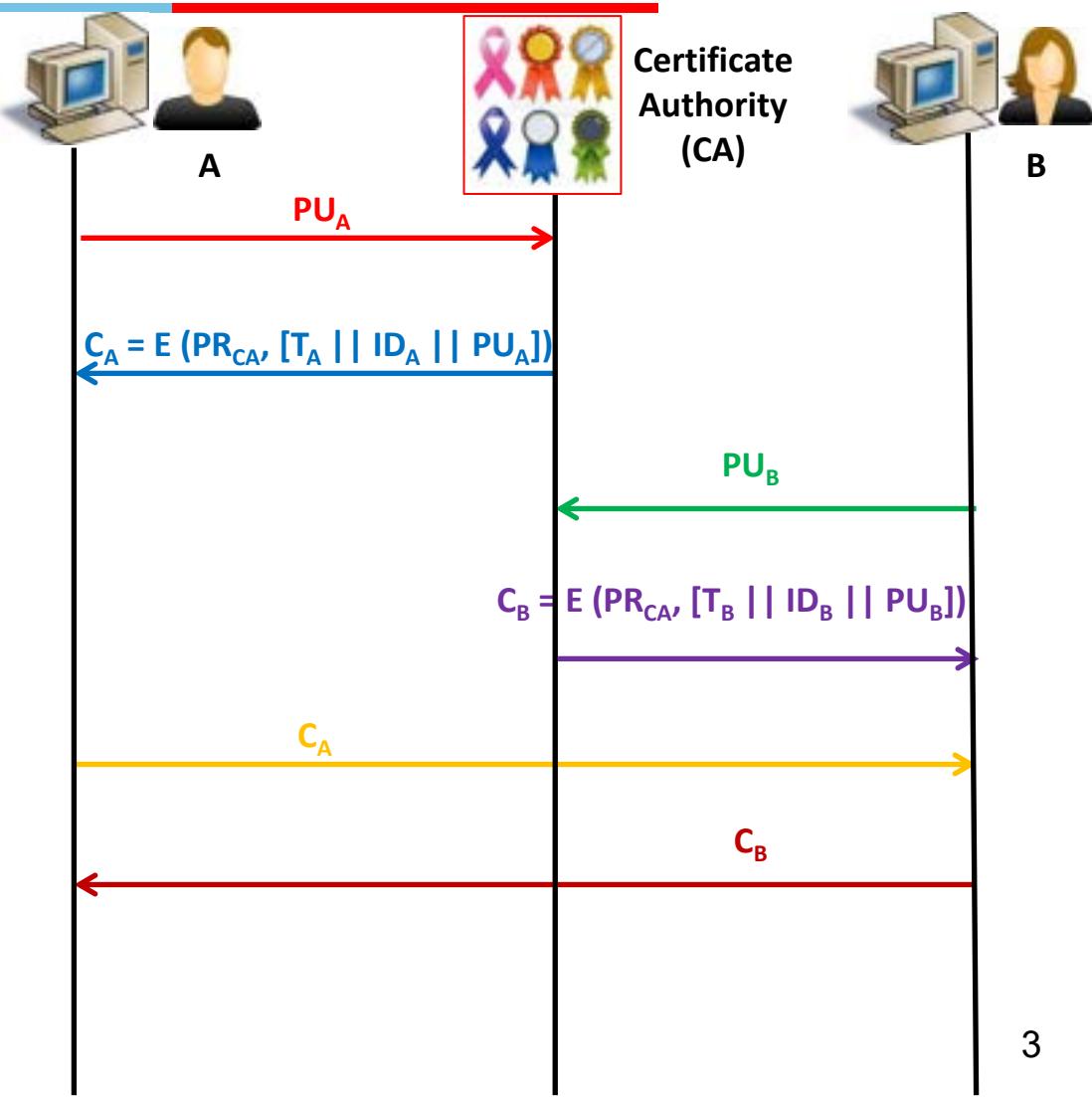
- Certificates can be used by participants to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority.
- A certificate consists of a public key, an identifier of the key owner, and the whole block signed by a trusted third party.
- A user can present his or her public key to the authority in a secure manner and obtain a certificate.
- The user can then publish the certificate. Anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature.
- Requirements of Public-Key Certificate:**
 1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
 2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
 3. Only the certificate authority can create and update certificates.
 4. Any participant can verify the currency (freshness) of the certificate.

Exchange of Public-Key Certificates



Basic Idea

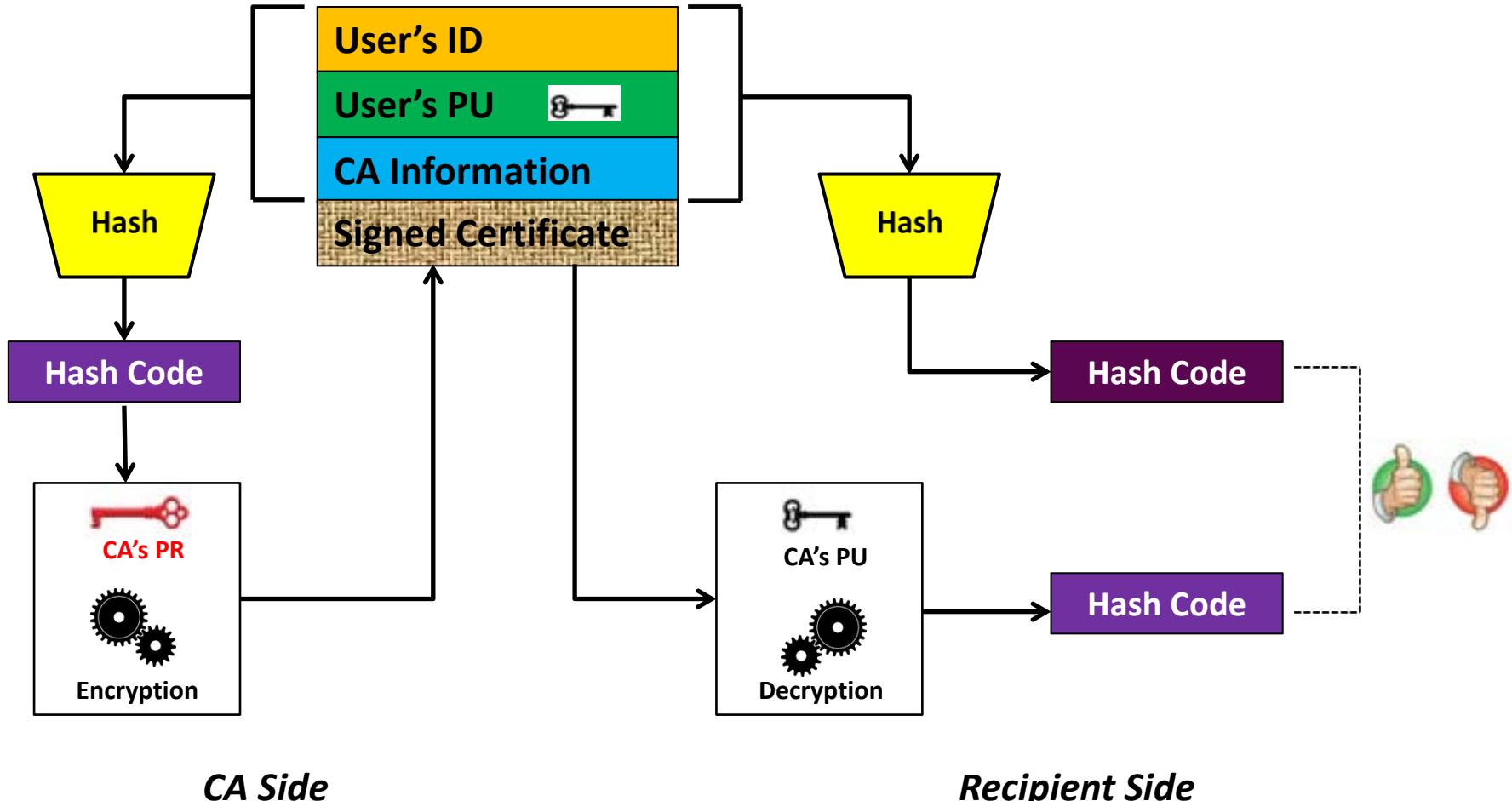
- A applies to the Certificate Authority (CA), supplying its public key (PU_A) and requesting a certificate.
- Certificate Authority (CA) prepares a certificate C_A for A which is an encrypted message of the following using its private key (PR_{CA}):
 - i. Time stamp (T_A)
 - ii. ID of A (ID_A) as known to CA.
 - iii. Public Key of A (PU_A) supplied by A.
- B also receives its certificate from CA in the same manner.
- A and B now can exchange their certificates directly without CA in between.
- B can decrypt the certificate of A using the public key of CA and retrieve T_A , ID_A and PU_A . In the same manner, A can also decrypt the certificate of B.
- If time stamp (T) is old the certificate must be considered expired.



Public Key Certificates



Basic Concept – Establishing CA's Authenticity





Thank You



SS ZG513
Network Security
X.509 Certificate Structure
Revision 1.0

BITS Pilani

Work Integrated Learning Programmes



Introduction

X.509 is part of the ITU-T X.500 series. It defines a directory service.

Directory is a server or distributed set of servers that maintains a database of information about users.

X.509 defines the certificate structure and other details to be used for public key distribution.

X.509 Certificate Structure



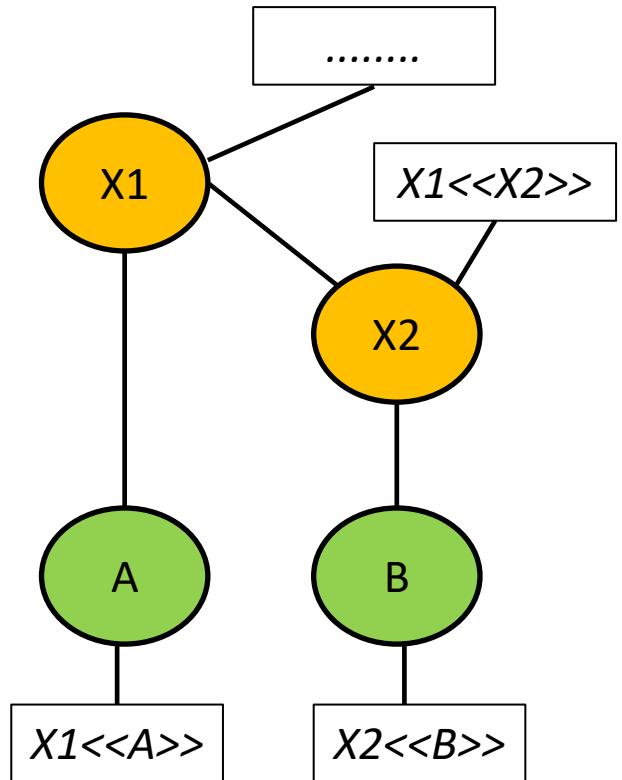
ITU-T X.509

Version	Version-1
Certificate Serial Number	Version-2
Algorithm	Version-3
Parameters	
Issuer Name	
Not Before	
Not After	
Subject Name	
Algorithm	
Public Key and Parameters	
Issuer's Unique Id	
Subject's Unique Id	
Extensions	
Algorithm	CA Sign All versions
Encrypted Hash + Parameters	

- Version:** default is 1, if issuer's and subject's ids are present then it is 2. With certain extensions it is 3.
- Certificate Serial Number:** A unique integer value from a range associated with the CA.
- Algorithms and Parameters:** Algorithm used to sign the certificate. Redundant because last fields also has it.
- Issuer Name:** X.500 format name of the CA.
- Period of Validity:** In terms of not before and not after.
- Subject Name:** Name of the user who holds the certificate.
- Public Key, Algorithm and Parameters:** Public key, algorithm where this key is to be used and parameters.
- Issuer's and Subject's Unique Id:** Unique identifiers if the same name exist for more than one CA and user.
- Extensions:** Version-3 extensions.
- Algorithm, Hash and Parameters:** Signature information used to sign the certificate by CA.

Obtaining and Verifying Certificates

- Representation: $M << N >>$ means certificate of N is issued by M .
- A has obtained a certificate from certification authority X_1 and B has obtained a certificate from CA X_2 . So, $X_1 << A >>$ and $X_2 << B >>$. X_1 and X_2 are CA, while A and B are users.
- The certificate of X_2 signed by X_1 . So $X_1 << X_2 >>$ but the reverse is not true.
- A obtains the certificate of X_2 from the X.500 directory. Since A securely knows X_1 's public key, A can obtain X_2 's public key from its certificate and verify it by means of X_1 's signature on the certificate.
- A then goes back to the directory and obtains the certificate of B signed by X_2 . Because A now has a trusted copy of X_2 's public key, A can verify the signature and securely obtain B's public key.
- Note that B cannot verify the certificate of A in the similar manner because X_1 's certificate is not signed by X_2 in the given example (though it is also possible to have a situation like this).



Revocation of X.509 Certificates



Structure of Certificate Revocation List (CRL)

Algorithm	
Parameters	
Issuer Name	
This Update Date	
Next Update Date	
User Certificate Serial Number	
Revocation Date	
User Certificate Serial Number	
Revocation Date	
Algorithm	
Encrypted Hash + Parameters	

- ❑ Each CA must maintain a list consisting of all revoked but not expired certificates issued by that CA, including both those issued to users and to other CAs. These lists should also be posted on the directory. The reasons could be many like:
 - User's private key is compromised.
 - The user is temporarily suspended.
 - The certificate was not issued conformed to the policies.
- ❑ Each certificate revocation list (CRL) posted to the directory is signed by the issuer CA and includes
 - The issuer's name.
 - The date the list was created.
 - The date the next CRL is scheduled to be issued.
 - An entry for each revoked certificate - consists of the serial number of a certificate and revocation date.
- ❑ When a user receives a certificate in a message, the user must determine whether the certificate has been revoked. The user could check the directory each time a certificate is received. User could also maintain a local cache of certificates and lists of revoked certificates to save delays.⁵



Thank You



SS ZG513
Network Security
Problem Statement of User Authentication
Revision 1.0

BITS Pilani

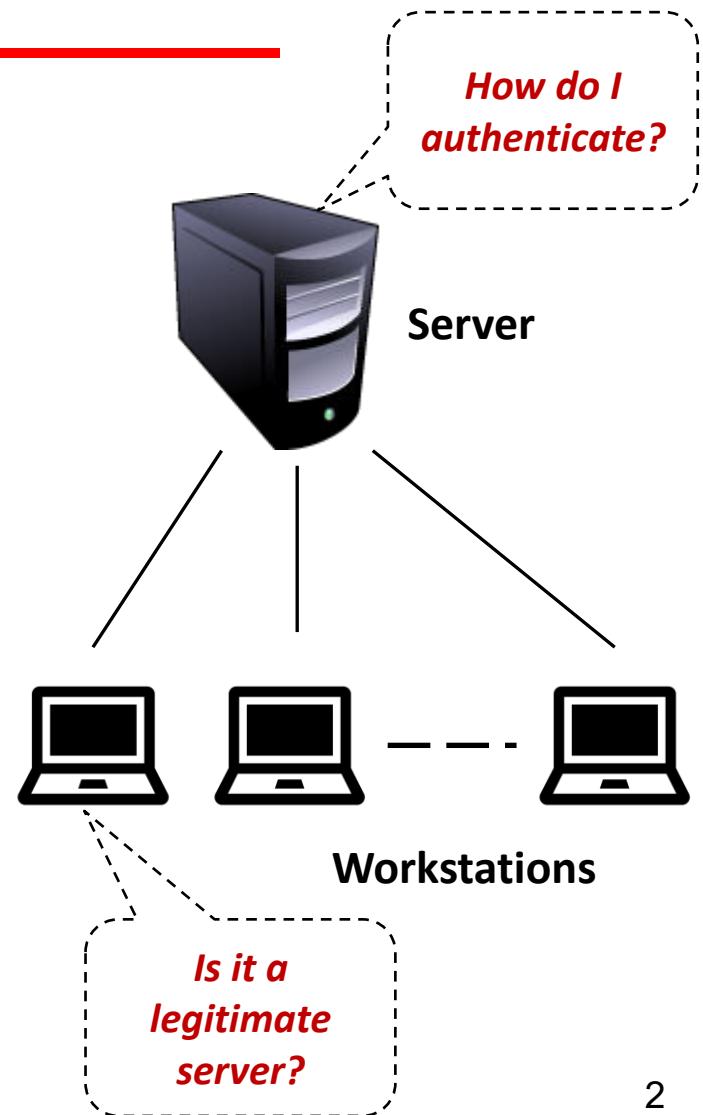
Work Integrated Learning Programmes



Need of Authentication

- Open Distributed Environment.
- Users at workstations wish to access services on servers distributed throughout the network.
- Servers to be able to restrict access to authorized users and to be able to authenticate requests for service.
- A user may gain access to a particular workstation and pretend to be another user.
- A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
- A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

How to provide authentication between the user and the server?



Kerberos

Kerberos is an authentication service developed as part of Project Athena at MIT attempted to address these issues and provide authentication service in a distributed environment.



Thank You



SS ZG513 **Network Security**

Kerberos-4.0

Revision 1.0

BITS Pilani

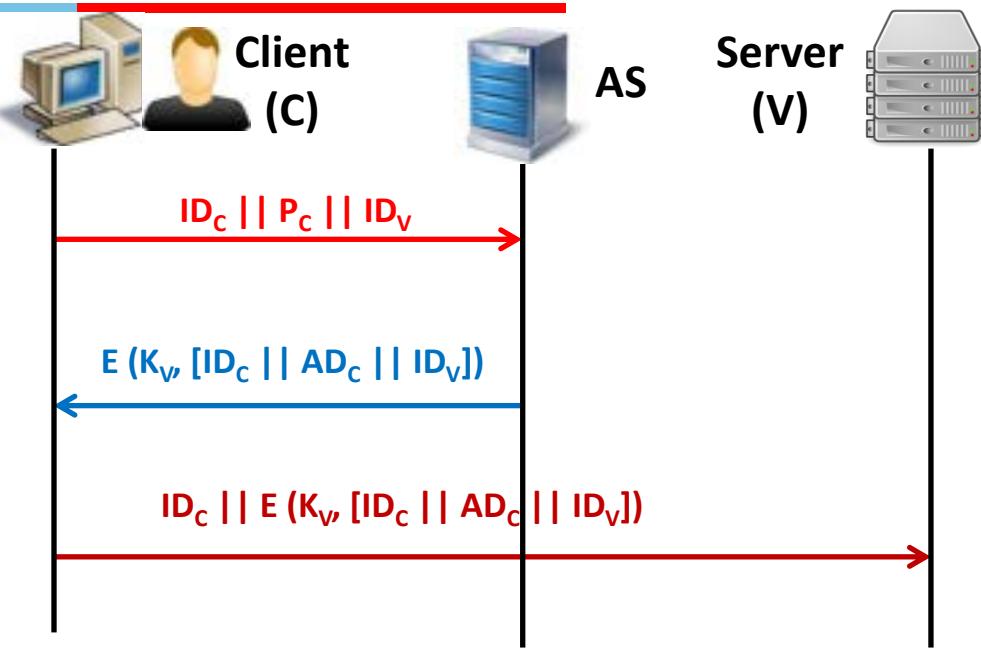
Work Integrated Learning Programmes

Kerberos



Seeding Thoughts – Simple Model

- The client (C) is looking for a service on the server (V).
- The user logs into a machine and client module in that machine requests user's password. Client enters the password (P_C).
- Client module sends a message to the Authentication Server (AS) that includes client id (ID_C), password (P_C) and the server's ID (ID_V).
- AS checks if password is valid and client C is permitted to access services on V. If yes, AS provides a ticket to C. This ticket contains client id (ID_C), network address of the client (AD_C) and the server's ID (ID_V). All of this is encrypted with the shared key (K_V) with the server V and AS.
- Client C can now apply to server V for the required service with its ID_C and the ticket.
- Server V can decrypt the ticket and compare the ID_C (from the ticket and message) before providing the service.



Observations!

- ✓ User needs to get the ticket every time, it needs to access the server. May be ticket can be made reusable.
- ✓ Different tickets for different servers.
- ✓ Password is transmitted in plaintext in the first message. An attacker can steal it easily.

Kerberos

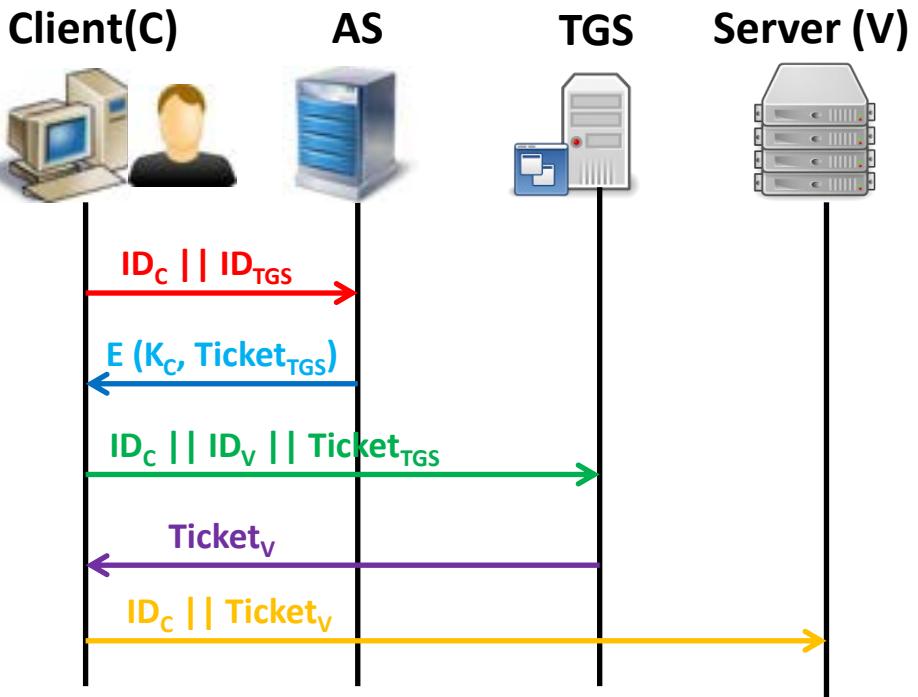


Seeding Thoughts – More Secure Model

- The client (C) requests a ticket granting ticket from AS providing its ID_C and Ticket Granting Server (TGS) id (ID_{TGS}).
- AS responds to C with a ticket ($Ticket_{TGS}$) encrypted with a key K_C . The key is derived from the user password stored in AS.
- When C receives this response, it prompts the user for the password, re-generates the key (K_C) and the message is decrypted and the ticket is retrieved.
- Now C requests TGS a ticket to access the services from the server V. The message contains client ID_C, server ID_V and the ticket received earlier from AS ($Ticket_{TGS}$).
- TGS can decrypt the $Ticket_{TGS}$ using a key (K_{TGS}) shared between AS and TGS and verify the following:
 - i. Presence of its ID (ID_{TGS}).
 - ii. Lifetime has not expired.
 - iii. Client ID (ID_C) and Client Address (AD_C) is correct from the received message.
- If client is permitted to use the server V, TGS grants a ticket ($Ticket_V$) to the client .

$$Ticket_{TGS} = E(K_{TGS}, [ID_C || AD_C || ID_{TGS} || TS_1 || Lifetime_1])$$

$$Ticket_V = E(K_V, [ID_C || AD_C || ID_V || TS_2 || Lifetime_2])$$



- The client requests server V for the service sending it a message with its ID_C and $Ticket_V$.
- Server V can decrypt the message with K_V that is known only to the server and TGS and verify other parameters and take the decision to grant the service accordingly.
- Timestamp (TS) and Lifetime tell the freshness of the tickets to avoid later replays.

Observations!

For More Secure Seeding Thoughts

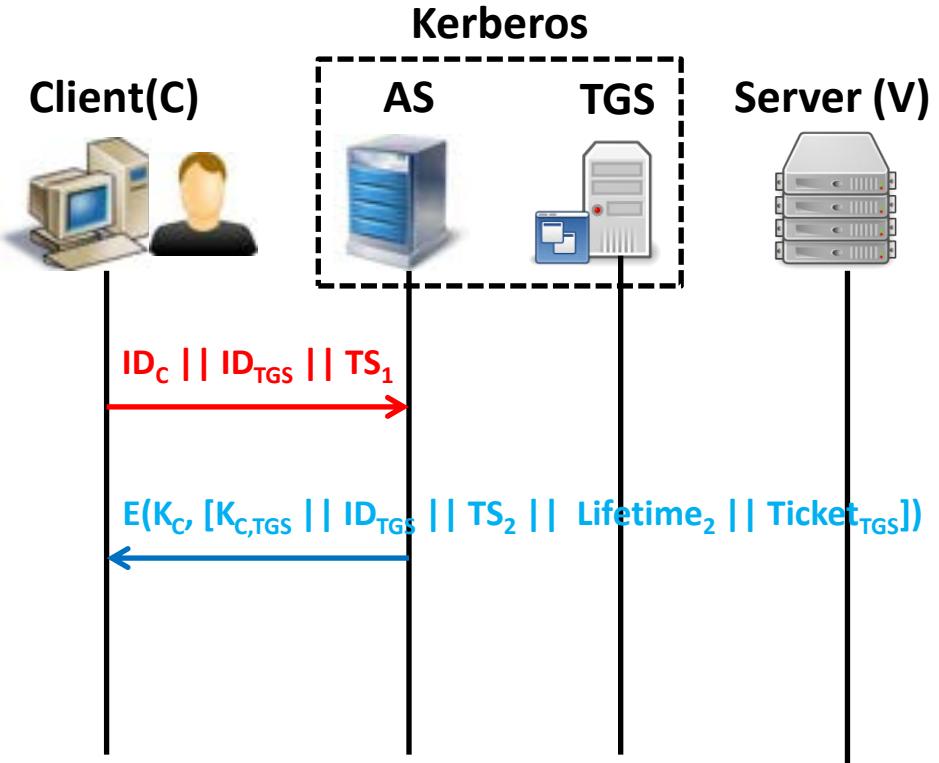
- ✓ Client password is not transmitted in plaintext.
- ✓ Ticket_{TGS} can be decrypted only by the legitimate user because key generation is password dependent on the client side.
- ✓ Using Ticket_{TGS}, the client can request TGS for tickets for multiple servers.
- ✓ Tickets contain time stamps (TS1 and TS2) with lifetimes. So that they cannot be replayed after lifetime expired.
- ✓ Short lifetime would generate frequent traffic overhead.
- ✓ Longer lifetime would lead to greater chances of replay (if an opponent could gain access to client's credentials).
- ✓ Servers are not authenticating themselves to the client. Masquerading could happen from the server side.

Kerberos Version-4



Authentication Dialogue

- The client (C) requests a ticket from AS providing its ID_C , Ticket Granting Server (TGS) id (ID_{TGS}) and Timestamp (TS_1).
- The AS responds with a message, encrypted with a key derived from the user's password (K_C), that contains the ticket.
- The encrypted message also contains a copy of the session key, $K_{C,TGS}$. This is a session key for C and TGS.
- Client can decrypt the message generating the key K_C with client password.
- Only client can read this session key with TGS because the message is encrypted with K_C .
- The same session key is included in the ticket, which can be read only by the TGS because ticket is encrypted with K_{TGS} , a key shared between AS and TGS.



End of this authentication dialogue, the session key ($K_{C,TGS}$) is securely available to C and can also be provided to TGS through ticket.

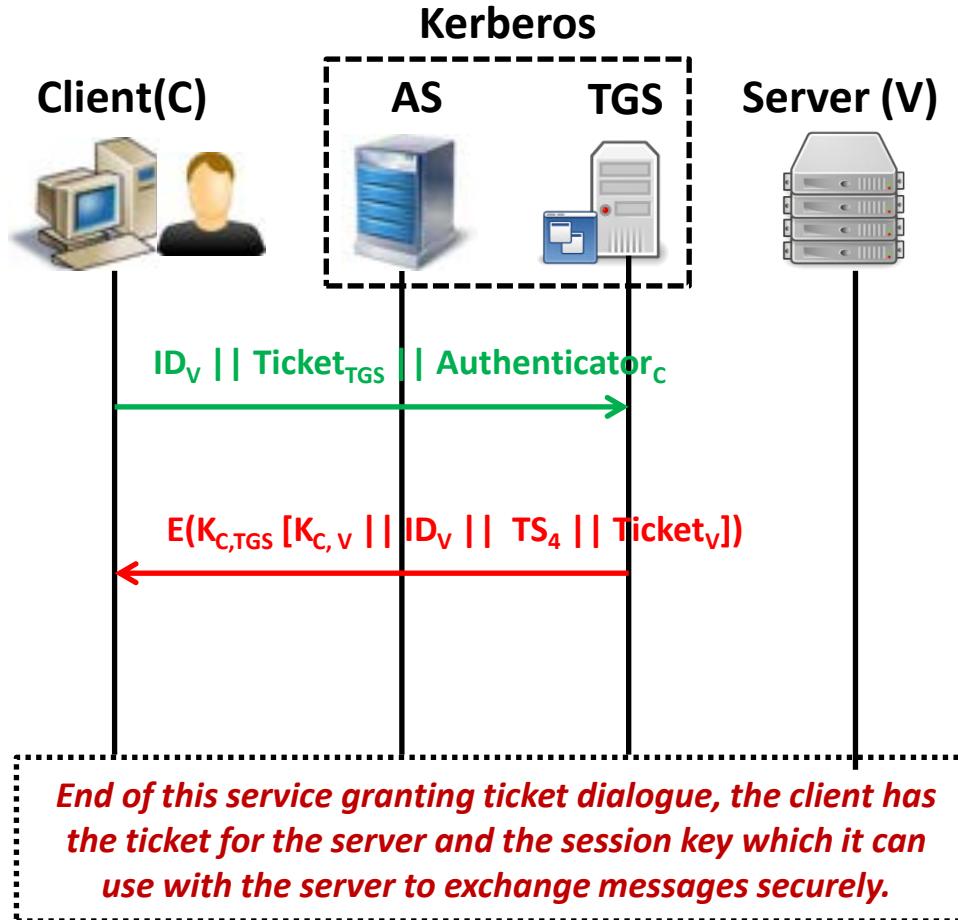
$$Ticket_{TGS} = E(K_{TGS}, [K_{C,TGS} || ID_C || AD_C || ID_{TGS} || TS_2 || Lifetime_2])$$

Kerberos Version-4



Service Granting Ticket

- The client (C) approaches TGS for a service granting ticket for server V. The message contains server id (ID_V), the ticket received from AS ($Ticket_{TGS}$) and an Authenticator_C.
- The Authenticator contains the ID and address information about the client and a timestamp encrypted with $K_{C,TGS}$.
- TGS decrypts the ticket with the key (K_{TGS}) and fetches $K_{C,TGS}$ to decrypt the authenticator to verify the client (ID and address).
- If everything is fine, TGS responds with a message containing server ID, time stamp and ticket for server. The message is encrypted with a key shared by client and TGS ($K_{C,TGS}$) and also contains a key to be shared by client and sever ($K_{C,V}$), so client can fetch the ticket for the server.



$$Ticket_{TGS} = E(K_{TGS}, [K_{C,TGS} || ID_C || AD_C || ID_{TGS} || TS_2 || Lifetime_2])$$

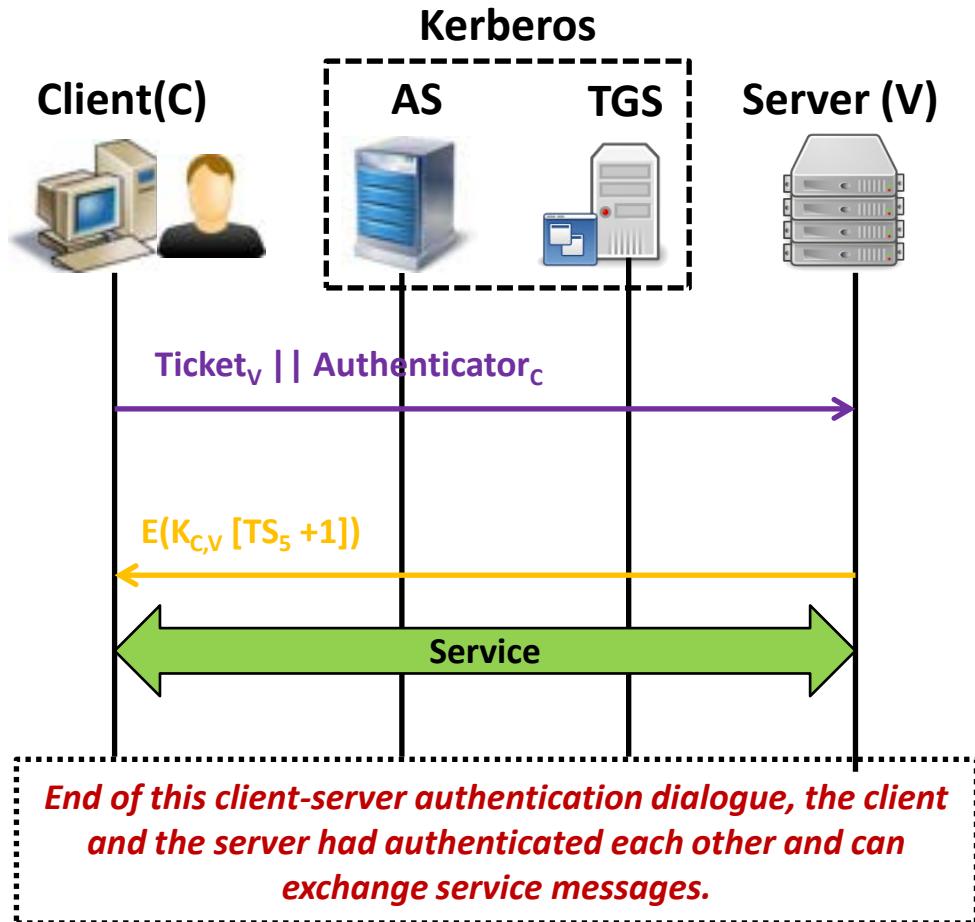
$$Authenticator_C = E(K_{C,TGS}, [ID_C || AD_C || TS_3])$$

Kerberos Version-4



Client Server Authentication

- C now has a reusable ticket for server V. It approaches to the server with this ticket and its authenticator.
- Server decrypts this ticket from K_V and fetches session key ($K_{C,V}$) with the client to decrypt the authenticator.
- Using authenticator, server verifies the credentials of the client.
- As a mutual authentication message, the server takes the time stamp TS_5 from the authenticator , adds 1 to it and sends it back to the client encrypted with the session key.
- After verifying the value of the returned time stamp, client can obtain the service from the server.



$$Ticket_V = E(K_V, [K_{C,V} || ID_C || AD_C || ID_V || TS_4 || Lifetime_4])$$

$$Authenticator_C = E(K_{C,V} [ID_C || AD_C || TS_5])$$



Thank You



SS ZG513
Network Security
Digital Signature
Revision 1.0

BITS Pilani

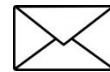
Work Integrated Learning Programmes



Integrity Attacks



Bob



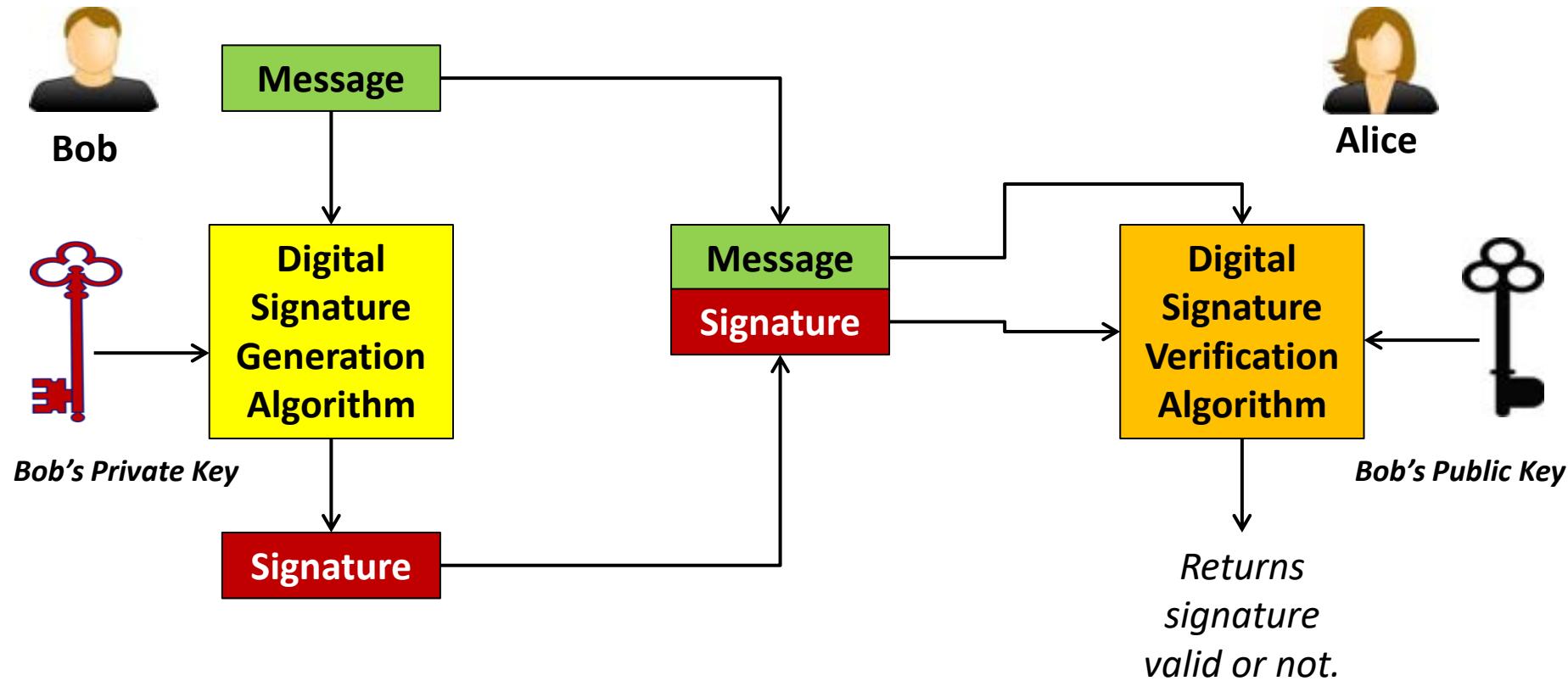
Message



Alice

- We have reviewed few important integrity services in the past: ***Message Authentication, Message Integrity*** and ***Non-repudiation***. Let us quickly review the corresponding attacks, where these services are required.
- Let us say, Bob sends a message to Alice.
- **Scenerio-1:** How Alice can ensure that the message really came from Bob? There could be authentication issues.
- **Scenerio-2:** Someone alters the message and claims that it was received from Bob. How Alice can be sure that message was actually received unaltered. There could be integrity attacks.
- **Scenerio-3:** After sometime if Bob denies that he sent the message, how Alice could prove it that it was actually received from Bob? There could be repudiation issues like this.

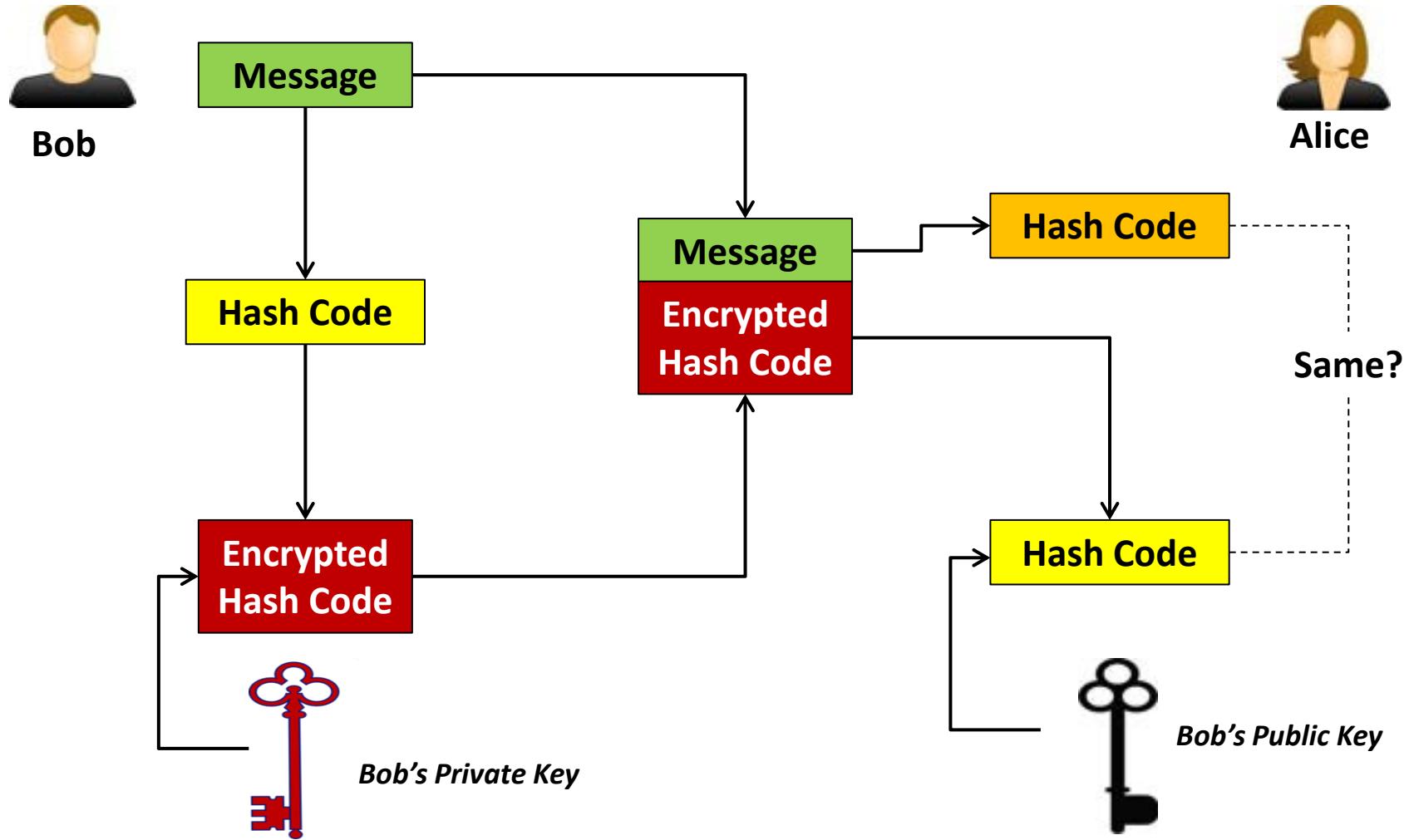
Digital Signature



How Digital Signature Helped?

- ✓ **Scenerio-1:** Since Alice used Bob's public key to verify the signature. The sender is authenticated.
- ✓ **Scenerio-2:** In case message is altered, the digital signature verification would fail and Alice would come to know that the message was altered.
- ✓ **Scenerio-3:** Alice can keep the signature received from Bob for later use. In case of repudiation, she can produce the signature to grievance authority.

Digital Signature based on RSA





Thank You