Introduction to Information Security (Half Course: 2 Credits)

Dr. Ashok Kumar Das

IEEE Senior Member
Associate Professor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
Homepage: http://www.iiit.ac.in/people/faculty/ashokkdas
Personal Homepage: https://sites.google.com/view/iitkgpakdas/

Welcome to Introduction to Information Security

About me: Dr. Ashok Kumar Das

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

- **Education:** Ph.D. in Computer Science and Engineering, M.Tech. in Computer Science and M.Sc. in Mathematics (with specialization in Computer Science) from ***Indian Institute of Technology, Kharagpur (IIT Kharagpur)***, India
- **Research Interests:** Cryptography, system and network security, blockchain, security in Internet of Things (IoT), Internet of Vehicles (IoV), Internet of Drones (IoD), smart grids, smart city, cloud/fog computing and industrial wireless sensor networks, intrusion detection, AI/ML security
- **Research Highlights [Research Publications (Total: 292)]:**
  - Number of Journal Papers: 331
  - Number of Conference Papers: 36
  - Number of Book Chapters: 9
  - Number of Edited Books/Volumes: 2
  - Number of IEEE Transactions/IEEE Journal/IEEE Magazine Papers: 113

Research Contributions

- ***Total citations: 15,067, h-index: 73, i10-index: 207***
  (According to Google Scholar Citations as on January 8, 2023)
- **Published in top venues like**
  * **IEEE Transactions on Information Forensics and Security**
  * **IEEE Transactions on Dependable and Secure Computing**
  * **IEEE Transactions on Consumer Electronics**
  * **IEEE Transactions on Smart Grid**
  * **IEEE Transactions on Industrial Informatics**
  * **IEEE Transactions on Vehicular Technology**
  * **IEEE Internet of Things Journal**
  * **IEEE Consumer Electronics Magazine**
  * **IEEE Communications Magazine**
  * **IEEE Journal of Biomedical and Health Informatics**
  (Formerly, IEEE Transactions on Information Technology in Biomedicine)
  * **IEEE Transactions on Network Science and Engineering**
  * **IEEE Transactions on Intelligent Transportation Systems**
  * **IEEE Sensors Journal**

About me: Journal Editorial Board Members

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

- Associate Editor: **IEEE Systems Journal** (SCI Impact Factor: 3.987) [Duration: August 2020 onwards]
- Editor: **Journal of Network and Computer Applications (Elsevier)** (SCI Impact Factor: 5.570) [Duration: October 2020 onwards]
- Editor (Technical Committee): **Computer Communications (Elsevier)** journal (SCI Impact Factor: 2.816) [Duration: August 2020 onwards]
- Associate Editor: **Journal of Cloud Computing (Springer)** (SCI Impact Factor: 5.71) [Duration: August 2021 onwards]
- Associate Editor: **IET Communications** journal (SCI Impact Factor: 1.443) [Duration: February 2019 - February 2022]
- Associate Editor: **Cyber Security and Applications (Elsevier)** [Duration: September 2021 onwards]
- Editor: **KSII Transactions on Internet and Information Systems** (SCI Indexed Journal) (2016 -)
- Editor: **International Journal of Internet Technology and Secured Transactions (Inderscience)** (2016 -)
- Guest Editor: **Computers & Electrical Engineering (Elsevier)** (SCI Indexed Journal), Special issue on Big data and IoT in e-healthcare, 2016
- Guest Editor: **ICT Express (Elsevier)** (ESCI Indexed Journal), Special Issue on Blockchain Technologies and Applications for 5G Enabled IoT, 2019
- Guest Editor: **Wireless Communications and Mobile Computing** (SCI Indexed Journal), Special issue on “Security and Privacy for Smart Mobile Devices: Attacks, Challenges, and New Designs” 2020

About me

- Listed in the **Web of Science (Clarivate^{TM}) Highly Cited Researcher 2022 in recognition of exceptional research performance demonstrated by production of multiple highly cited papers that rank in the top** 1% **for field and year.**
- Media coverage on **"China way ahead in blockchain, India needs to catch up" in the Times of India newspaper on 6 May 2021** (please see at: `https://timesofindia.indiatimes.com/business/india-business/china-way-ahead-in-blockchain-india-needs-to-catch-up/articleshow/82428430.cms`)
- Included in the **Stanford University's Top** 2% **Most Influential Scientists List with Subject Rank (World): 179 for the year: 2021.**
  The ranking is based on an independent study done by the Stanford University, USA. For detailed information, please visit: `https://elsevier.digitalcommonsdata.com/datasets/btchxktzyw/3`
- Listed in the **Best Computer Science Scientists database maintained by Research.com, a leading academic platform for researchers, with World Ranking: 1650 and National Ranking (India): 8.**
- **Visiting Faculty at the Old Dominion University (ODU), Suffolk, VA 23435, USA, with the Virginia Modeling, Analysis, and Simulation Center (VMASC)** [Duration: 1 May 2022 - 30 June 2022].

- More detailed information at: `https://sites.google.com/view/iitkgpakdas/`

International Research Collaborations

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- **Sajal K. Das**, IEEE Fellow, Professor and Daniel St. Clair Endowed Chair, Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA
- **Mauro Conti**, IEEE Fellow, Head of SPRITZ Security and Privacy Research Group, Director of UniPD node of CINI Cybersecurity National Lab, EU Marie Curie Fellow Alumni, CEO and co-funder of CHISITO, and Co-funder of DYALOGHI, University of Padua, Italy
- **Willy Susilo**, IEEE Fellow, ARC Future Fellow, Co-Director, Centre for Computer and Information Security Research, University of Wollongong, AUSTRALIA
- **Sherali Zeadally**, Fellow of the British Computer Society and the Institution of Engineering Technology, Stevenage, U.K., University of Kentucky, Lexington, KY 405 06 USA
- **Kim-Kwang Raymond Choo**, Fellow, Australian Computer Society, Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249 USA
- **Xinyi Huang**, Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, Fujian, China
- **Alexey Vinel**, Halmstad University, Halmstad, Sweden
- **Muhammad Khurram Khan**, FIET (UK), FBCS (UK), Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

International Research Collaborations

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- **Athanasios V. Vasilakos**, Lulea University of Technology, Sweden
- **Minho Jo**, Department of Computer and Information Science, Korea University, Seoul, South Korea
- **Laurence T. Yang**, St. Francis Xavier University, Canada
- **Joel J. P. C. Rodrigues**, IEEE Fellow, National Institute of Telecommunications - Inatel, Brazil
- **Debiao He**, School of Cyber Science and Engineering, Wuhan University, Wuhan 430 072, China
- **Jong-Hyouk Lee**, Sangmyung University, Republic of Korea
- **Kee-Young Yoo**, Kyungpook National University, Daegu, Korea
- **Qi Jiang**, Xidian University, China
- **Sachin Shetty**,Old Dominion University, USA
- **Xiong Li**, Hunan University of Science and Technology, China
- **Mamoun Alazab**, Charles Darwin University, Australia
- **Mohammad S. Obaidat**, IEEE Fellow, University of Sharjah
- **YoungHo Park**, School of Electronics Engineering, Kyungpook National University, South Korea
- **Mohsen Guizani**, IEEE Fellow, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI)

Course Contents

- **Part 1: Basics of Cryptography**
  - Cryptographic goals and objectives
  - Types of attacks, passive and active attacks
  - Introduction to Number Theory
  - Complexity Theoretic Connections
  - Overview of symmetric and public key cryptography
  - Digital Signatures

Course Contents (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- **Part 2: Basics of System Security**
  - Intruders: Intruders, Intrusion detection, Intrusion prevention.
  - Software Vulnerabilities: Phishing, Buffer overflow (BOF), Heap overflow, Format string attacks, Cross-site scripting (XSS), SQL Injection.
- **Part 3: Basics of Network Security**
  - Overview of encrypting communication channels
  - Overview of various security protocols at OSI layers
  - **Introduction to IoT security:** IoT architecture, various IoT applications, security requirements, security attacks, threat model for the IoT ecosystem, taxonomy of security protocols
  - **Introduction to Blockchain technology:** Various applications of blockchain of Things (BCoT), centralized versus decentralized models, types of blockchain, brief overview of various consensus algorithms, block formation and addition in a blockchain
  - **AI/ML Security**

Preferred Textbooks and References

- William Stallings, “Cryptography and Network Security: Principles and Practices,” Pearson Education, 6th Edition, 2014.
- Bernard Menezes, “Network Security and Cryptography,” Cengage Learning, 2010.
- Behrouz A. Forouzan, “Cryptography and Network Security,” Special Indian Edition, 2010.

Grading and Examinations Policy

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Grading Method:: Relative

- Quiz: 30
- End Sem Exam: 50
- Assignments: 20

Thank You!!!

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Introduction to Cryptography

Dr. Ashok Kumar Das

Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

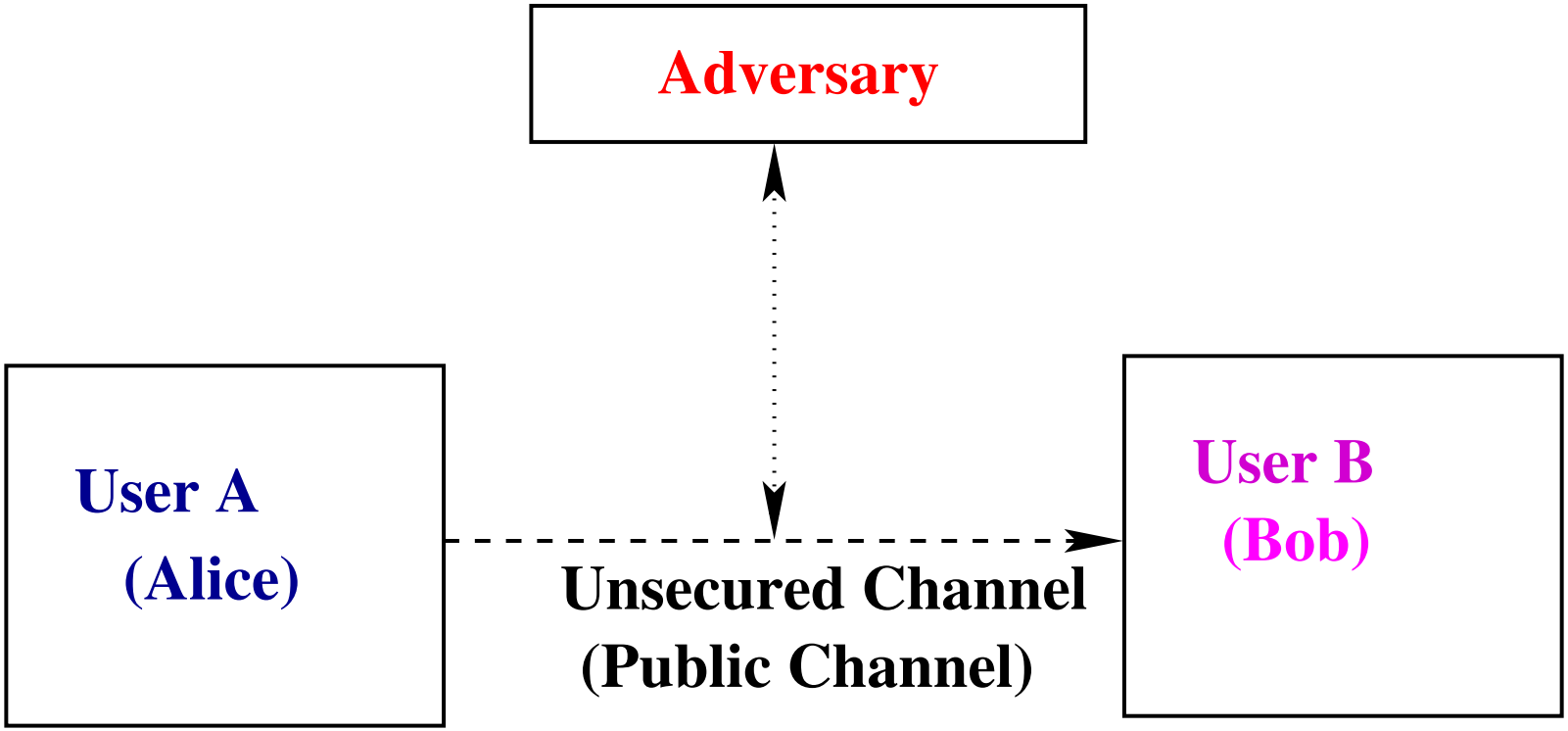
E-mail: *ashok.das@iiit.ac.in*
URL: http://www.iiit.ac.in/people/faculty/ashokkdas
https://sites.google.com/view/iitkgpakdas/

Overview of Cryptography

What is Cryptography?

- Cryptography is the study of **mathematical techniques** related to aspects of information security such as confidentiality, data integrity, entity authentication, message authentication (data origin authentication) and non-repudiation.
- Cryptography is not the only means of providing information security, but rather one set of techniques.
- Now-a-days, cryptography has moved from an art to a science. Thus, cryptography is the science of keeping secrets secret.

Consider the following simple two-party communication model:



- An "**adversary**" is an entity in a two-party communication which is neither the sender nor the receiver, and which tries to defeat the information security service being provided between the sender and the receiver.
- A "**channel**" is a means of conveying information from one entity to another entity.
- An "**unsecured channel**" is one from which parties other than the sender and the receiver can reorder, delete, insert, or read the data being transmitted.
- A "**secured channel**" is one from which an adversary does not have the ability to reorder, delete, insert, or read the data being transmitted.

Types of adversary

- A "**passive adversary**" is an adversary who is only capable of reading information from an unsecured channel.
- An "**active adversary**" is an adversary who is capable to transmit, alter, or delete information on an unsecured channel.

Cryptographic goals (objectives)

- **Confidentiality:** Privacy (confidentiality) is a service of keeping information secret from all but those who are authorized to see it.
- **Data integrity:** ensuring information has not been altered by unauthorized or unknown means.
- **Entity authentication or identification:** Corroboration of the identity of an entity (i.e., a person, a computer terminal, a credit card, etc.).
- **Message or data origin authentication:** Corroborating the source of information.
- **Non-repudiation:** Preventing the denial of the previous session (preventing the malicious nodes to hide their activities).

Cryptographic goals (objectives)

- **Authorization:** Conveyance to another entity such as a person or group of users. It ensures that the nodes (users) those who are authorized can be involved in providing information to network services.
- **Signature:** a means to bind information to an entity.
- **Access control:** restricting access to resources to privileged entity.
- **Certification:** endorsement of information by a trusted entity.

Cryptographic goals (objectives)

We need also to consider the forward and backward secrecy when new nodes join in the network and existing nodes depart from the network.

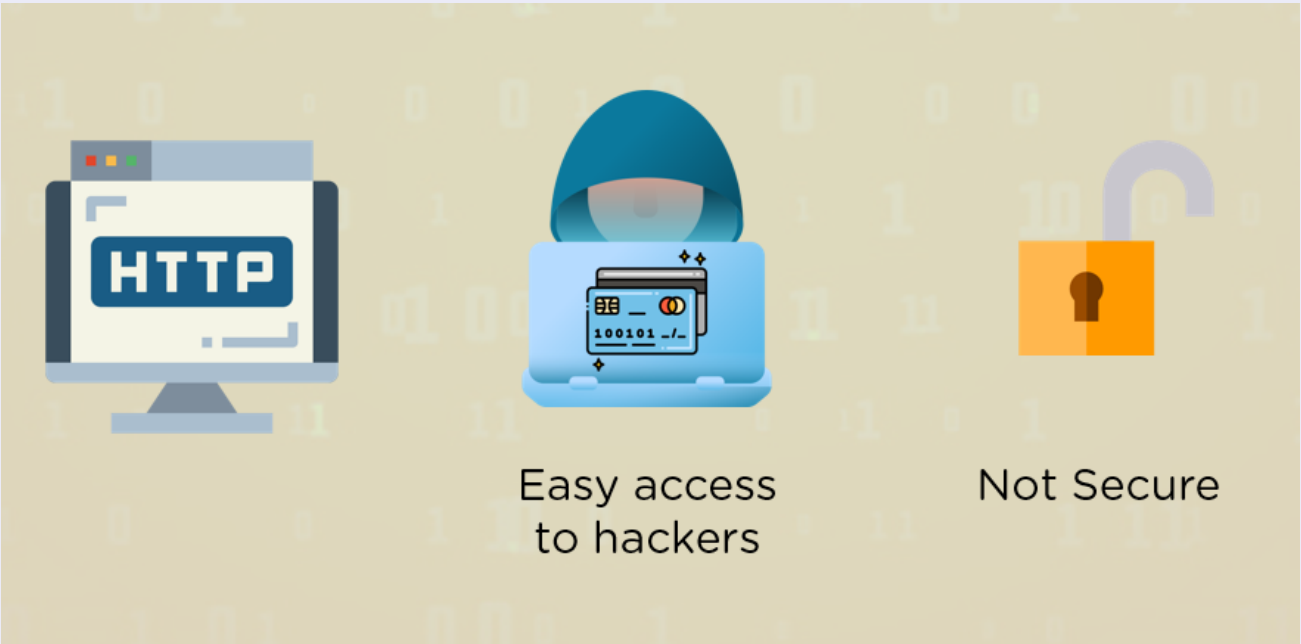
- ***Forward secrecy:*** When a node (user) leaves the network, it must not read any future messages after its departure.
- ***Backward secrecy:*** When a new node (user) joins in the network, it must not read any previously transmitted message.

What is the Need for Cryptography?

- Suppose Alice wants to look for a discount on the latest iPhone. After browsing the internet, she comes across a questionable website willing to offer a 50% discount on the first purchase.
- A few moments after she provides her payment details, the website withdraws a huge chunk of money from her account.
- Alice then wonders how she had failed in realizing that the website was a scam.

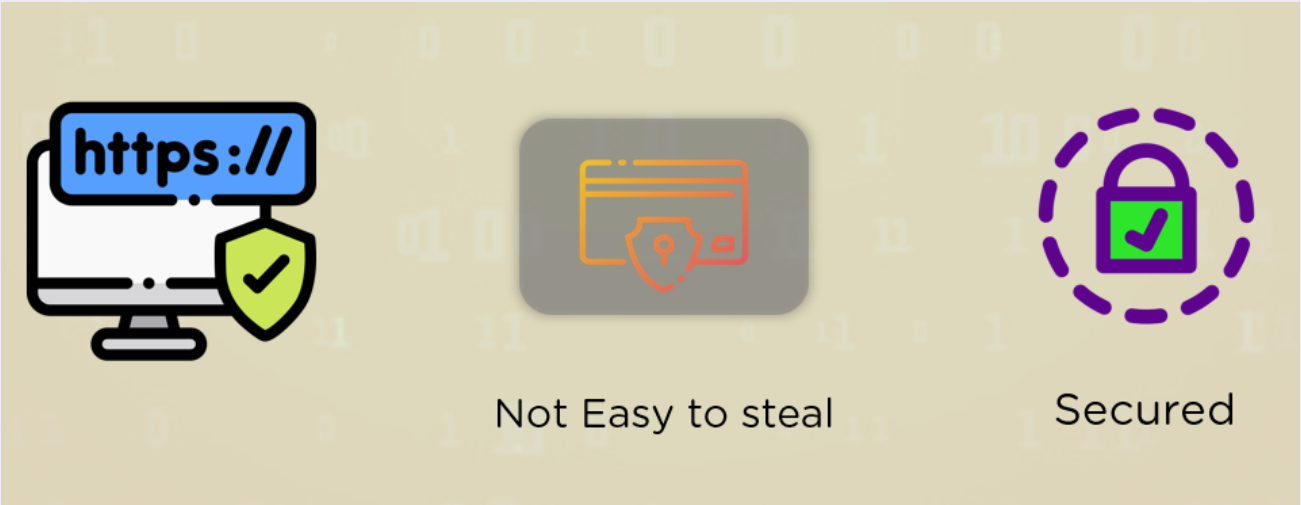
What is the Need for Cryptography?

- She then notices that the website is an HTTP webpage instead of HTTPS.
- Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes.
- The payment information submitted was not encrypted and visible to anyone keeping an eye, including the website owner.



What is the Need for Cryptography?

- Now, if she had chosen to use a reputed website, which has encrypted transactions and employs cryptography, this iPhone enthusiast could have avoided this particular incident. This is why it's never recommended to visit unknown websites or share any personal information on them.
- This is where Cryptography comes to play, and is so essential.



Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet.

Introduction to Cryptography

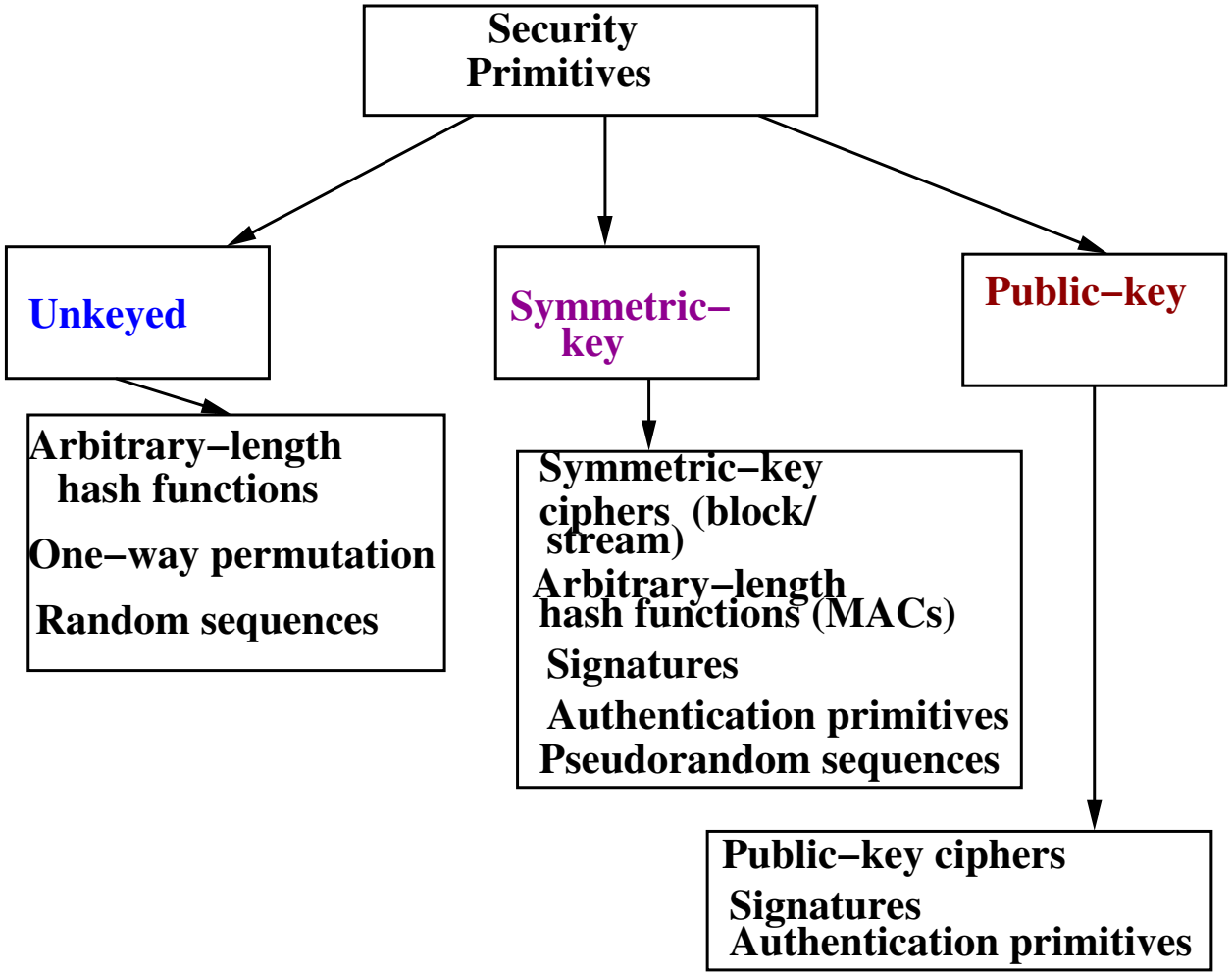


Figure: A taxonomy of cryptographic primitives

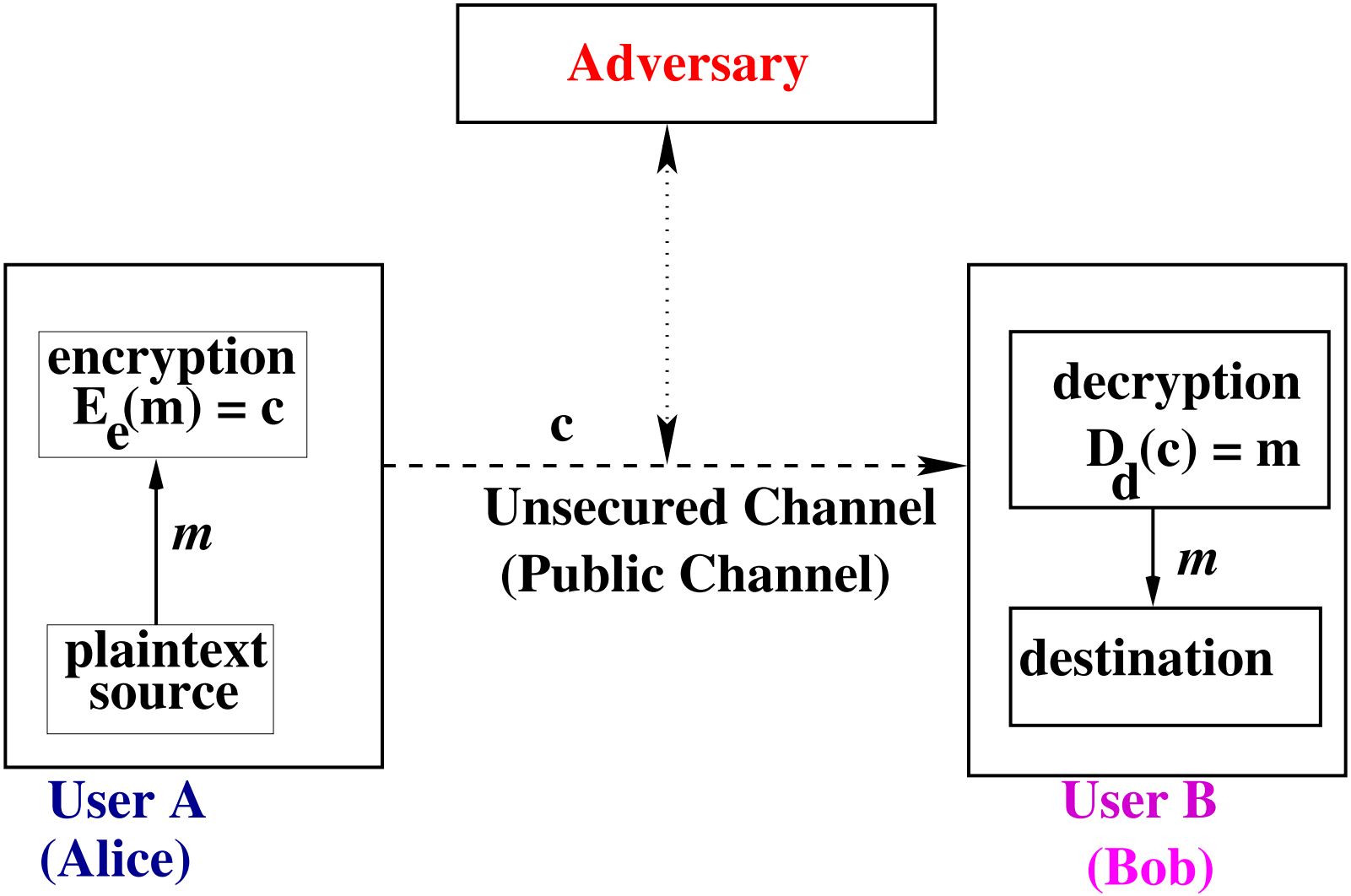
Evaluation criteria for the primitives

- ***Level of security:*** This is usually difficult to quantify.
- ***Functionality:*** Primitives will need to be combined to meet various information security objectives.
- ***Methods of operation:*** One primitive could provide very different functionality depending on its mode of operation or usage.
- ***Performance:*** This refers to the efficiency of a primitive in a particular mode of operation (For example, an encryption algorithm may be rated by the number of bits per second which it can encrypt).
- ***Easy of implementation:*** This might include the complexity of implementing the primitives in either a software or hardware environment.

Note that the relative importance of various criteria is very much dependent on the application and resources availability.

Introduction to Cryptography

Consider the following simple two-party communication model with encryption:



- **Encryption scheme 1:** Have only the encryption and decryption functions and these are kept secret to the sender and receiver only. No key is used in this method.
- **Encryption scheme 2:** Key is being used. However, the encryption and decryption functions are made public.

Quiz: Why keys are necessary? Why not just choose one encryption function and its corresponding decryption function?

- **Security of the scheme**
  - Depends entirely on the secrecy of the key
  - Does not depend on the secrecy of the algorithm (Needs to be public for criticism!)
- Hence, we make the **assumptions** as follows:
  - Algorithms for encryption/decryption are known to the public
  - Keys used are kept secret

P, NP, NP-Hard and NP-Completeness

P, NP, NP-Hard and NP-Completeness

Decision Problem

- Any problem for which the answer is either zero (0) or one (1) is called a decision problem.
- An algorithm for a decision problem is termed as a decision algorithm.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Time complexity classes

- Let $t : \mathcal{N} \rightarrow \mathcal{N}$ be a function, where $\mathcal{N}$ is the set of natural numbers.
- Define the time complexity class, $TIME(t(n))$, to be $TIME(t(n)) = \{L | L$ is a language decided by an $O(t(n))$-time deterministic Turing machine (DTM) $\}$.
- Define the time complexity class, $NTIME(t(n))$, to be $NTIME(t(n)) = \{L | L$ is a language decided by an $O(t(n))$-time non-deterministic Turing machine (NTM) $\}$.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

The Class P

- $P = \{L | L$ is a language or problem decided by a deterministic Turing machine (DTM) in polynomial time $\}$. In other words, $P = \cup_{k \in \mathcal{N}} TIME(n^k)$, where $\mathcal{N}$ is the set of natural numbers.
- P is the class of languages that can be decided quickly on a DTM.
- P is the set of all decision problems solvable by deterministic algorithms in poly-time.
- P is the class of all "practically" solvable problems.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Path problem in a graph

- Let $G = (V, E)$ be a (directed/undirected) graph, where V be the set of vertices (nodes) and E the set of edges of G. Then the adjacency matrix or link list representation $\langle G \rangle$ is an encoding (reasonable encoding) of the graph G.
- Define a formal problem as follows:
  $PATH := \{\langle G, s, t \rangle \rangle | G$ is a directed graph with a path from node s to node t in $G\}$.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Example of the class P:

Theorem

PATH is in P.

- Input: $\langle G, s, t \rangle$, for all valid encoding of graph G.
- Output: Accept, if there is a path from s to t in G; reject, otherwise.

Stages (Steps or Iterations):

1. Place a mark on the node s,
2. Repeat the following until one fails to mark an additional node:
3. Scan all the edges of G. It there is an edge $(u, v) \in E$ from marked node u to an unmarked node v, then mark v.
4. If t is marked, then “accept”; else “reject”.

Example of the class P (Continued...)

- Correctness: Clear.
- Complexity (Running time):
  – Stage 1 requires only one step, that is, Stage 1 is executed only once.
  – Stage 4 is executed only once.
  – Stages 2 and 3 are executed at most $n(n-1) = O(n^2)$ time, where $n = |V|$ times.
- Therefore, the total number of stages is polynomial in the input size (n).
- Again, each stage can be implemented on a DTM using polynomially many steps (transitions) on the DTM. This means that the entire algorithm takes polynomially many steps in the input size.
- Thus, we have a deterministic algorithm for PATH decidable in polynomial time.
- Consequently, $PATH$ is in P, that is, $PATH \in P$.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

The Class NP

- $NP = \{L | L$ is a language or problem decided by a non-deterministic Turing machine (NTM) in polynomial time $\}$. In other words, $NP = \cup_{k \in \mathcal{N}} NTIME(n^k)$, where $\mathcal{N}$ is the set of natural numbers.
- NP is the class of languages that can be verified quickly on a DTM.
- P is the set of all decision problems solvable by non-deterministic algorithms in poly-time.
- Is it practical? Not known till date.

The Class NP (Continued...)

Verifier of a language L:

Definition

Let L be a language (problem). A verifier for L is a DTM V such that $L = \{\alpha | \langle \alpha, \beta \rangle$ is accepted by V for some string $\beta\}$.

In other words, for every $\alpha \in L$ there exists a certificate β such that V accepts $\langle \alpha, \beta \rangle$. That is, every $\alpha \in L$ is a certificate (prescription) for membership.
If V runs in poly-time (in $|\alpha|, |\alpha|$ is the length of α), then V is called poly-time verifier for L.

The Class NP (Continued...)

Theorem

$L \in NP$ if and only if L has a poly-time verifier.

To prove a problem L is in NP, we should have any one of the following methods:

1. A poly-time non-deterministic algorithm (NTM) for L.
2. A certificate (i.e., verifier) for L, that is, a poly-time verifier for L.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

CLIQUE in a graph

- A k-clique G' in an undirected graph $G = (V, E)$ is a subgraph G' of G isomorphic to the complete graph K_k.
- A maximal subgraph of an undirected graph $G = (V, E)$ is a clique.
- If the size of the clique is k (the number of vertices in $G' \subseteq G$), then it is known as k-clique.

CLIQUE in a graph (Continued...)

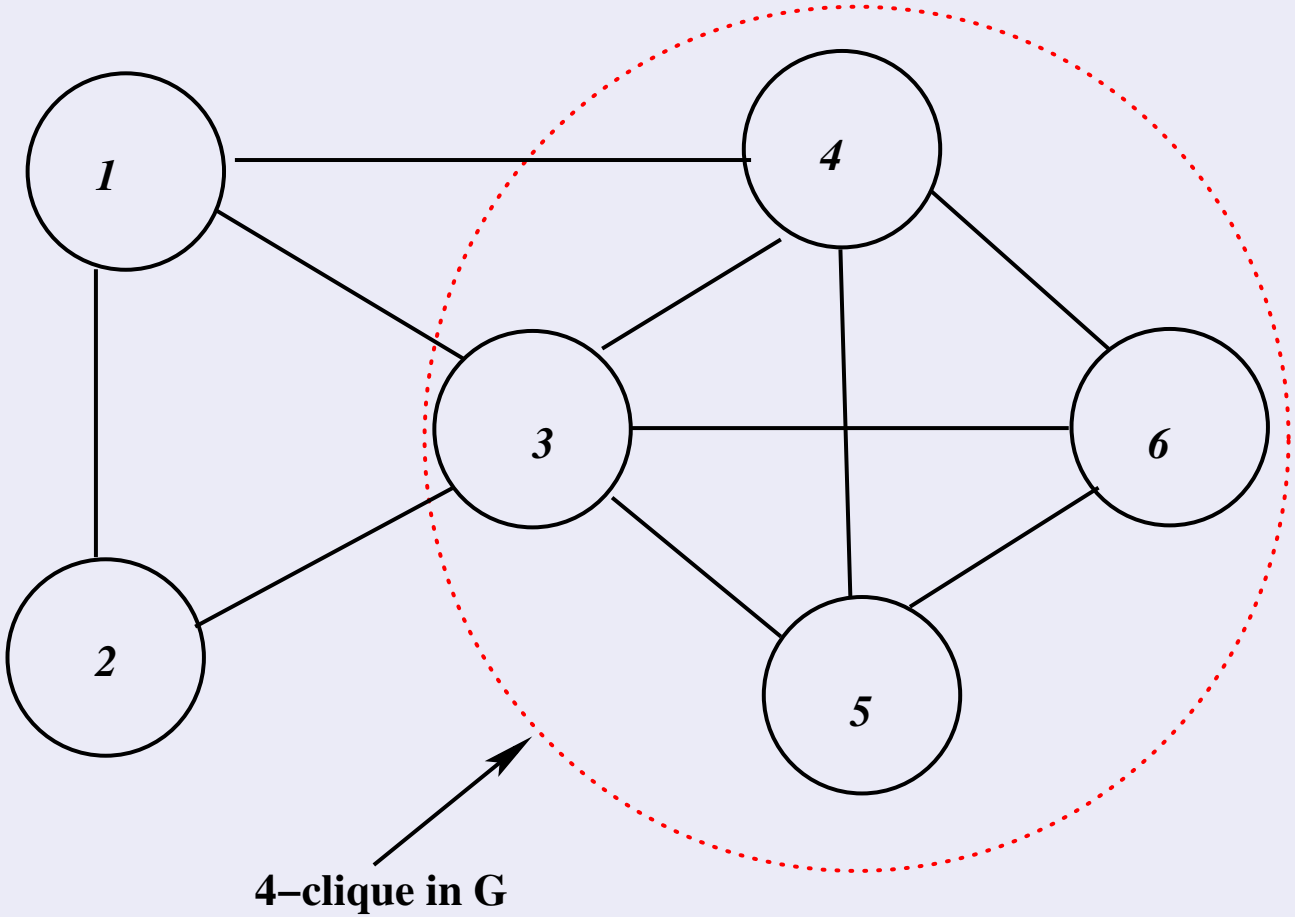


Figure: A 4-clique in a graph G.

CLIQUE in a graph (Continued...)

Consider the following problem:
$CLIQUE := \{\langle G, k\rangle | G$ is an undirected graph with a k-clique, where $k > 0$ is an integer constant $\}$.

Theorem

CLIQUE is in NP.

Proof: **Poly-time verifier for CLIQUE**
Input: $\langle\langle G, k\rangle, T\rangle$, where T is the certificate for $\langle G, k\rangle$, that is, T is the subset of $G = (V, E)$.
Output: Accept, if T is k-clique in G; reject, otherwise.
Stages:

1. If T does not contain k nodes, then "reject".
2. IF G does not contain an edge (u, v) with $u, v \in T$, then "reject".
3. Otherwise, "accept".

CLIQUE in a graph (Continued...)

Time complexity analysis:

- Stage 1 runs in $O(k)$ time.
- Stage 2 requires at most $k(k-1)/2 = O(k^2)$ time to check whether T is a complete graph or not.
- Thus, T is the verifier for CLIQUE runs in poly-time ($O(k^2)$).
- Hence, $CLIQUE \in NP$.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Polynomial-time reduction

- Let L_1 and L_2 be two problems (languages).
- Problem L_1 reduces to another problem L_2 in polynomial time, denoted by $L_1 \leq_p L_2$, if and only if there is a way to solve L_1 by a deterministic poly-time algorithm and using that deterministic algorithm we can solve L_2 also in poly-time.
- This definition implies that if we have a poly-time deterministic algorithm for L_1, then we can solve L_2 in poly-time.
- Moreover, if $L_1 \leq_p L_2$ and $L_2 \leq_p L_3$, then $L_1 \leq_p L_3$. In other words, the relation (reduction function) $\leq_p$ is transitive.

NP-Hard and NP-Completeness

Definition

A problem B is called **NP-hard** if every problem $A \in NP$ reduces to B in poly-time, that is, $A \leq_p B$, for all $A \in NP$.

Definition

A problem B is called **NP-complete** if
(i) $B \in NP$, and
(ii) B is NP-hard.

Note: All NP-complete problems are NP-hard. However, all NP-hard problems may or may not be NP-complete! (why?)

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

NP-Hard and NP-Completeness

Theorem

Let A and B be two problems. If $A \leq_p B$ and $B \in P$, then $A \in P$.

Theorem

Let A and B be two problems. If $A \leq_p B$ and $B \in NP$, then $A \in NP$.

Theorem

If a problem B is NP-complete and $B \leq_p C$ for another problem C in NP, then C is NP-complete.

Example of NP-Hard and NP-Complete problem

Hamiltonian Cycle Problem

- A directed Hamiltonian cycle in a directed graph $G = (V, E)$ is a directed cycle of length $n = |V|$. So, the cycle goes through every vertex (node) exactly once and then returns back to the starting vertex.

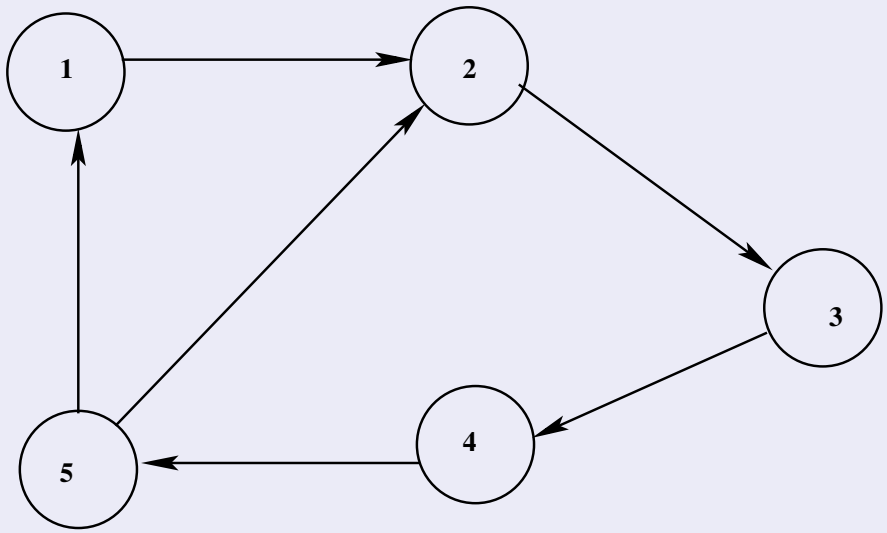


Figure: A directed graph G with a Hamiltonian cycle 1, 2, 3, 4, 5, 1.

P, NP, NP-Hard and NP-Completeness

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Hamiltonian Cycle Problem

- Consider the following problem formally as $HAMCYCLE := \{\langle G \rangle |$ there is a Hamiltonian cycle in the directed graph $G\}$.

Theorem

HAMPATH is NP-complete.

P, NP, NP-Hard and NP-Completeness

Hamiltonian Cycle Problem

- Consider the following problem formally as $HAMCYCLE := \{\langle G \rangle |$ there is a Hamiltonian cycle in the directed graph $G\}$.

Theorem

Using the fact that if HAMPATH is NP-complete, HAMCYCLE is also NP-complete.

Proof: To show HAMCYCLE is NP-complete, we must demonstrate two things:
(1) that HAMCYCLE is in NP; and
(2) that every language $A \in NP$ is poly-time reducible to HAMCYCLE.
That is, to prove HAMCYCLE is NP-hard, we take a poly-time reduction HAMPATH $\leq_p$ HAMCYCLE.

Hamiltonian Cycle Problem

Part 1. $HAMCYCLE \in NP$

- We need to construct a poly-time NTM for HAMCYCLE.
- **NTM for HAMCYCLE:**
  Input: $\langle G \rangle$.
  Output: Accept, if there is a Hamiltonian cycle; reject, otherwise.

Stages:

1. Non-deterministically generate a sequence of $n+1$ nodes, where $n = |V|$ is the number of nodes in G, say $p_1, p_2, p_3, \ldots, p_n, p_{n+1}$ from the set $\{1, 2, 3, \ldots, n\}$.
2. If $p_1 \neq p_{n+1}$, then "reject".
3. If there is a repetition in $p_1, p_2, \ldots, p_n, p_{n+1}$, then "reject".

Hamiltonian Cycle Problem

Part 1. *HAMCYCLE* $\in$ *NP* (Continued...)

4. If for some $i = 1, 2, \ldots, n-1$, the edge (p_i, p_{i+1}) is not an edge of G, then "reject".
5. If (p_n, p_{n+1}) is not an edge of G, then "reject".
6. "Accept".

Obviously, HAMCYCLE runs in poly-time by the NTM. Hence, *HAMCYCLE* $\in$ *NP*.

Hamiltonian Cycle Problem

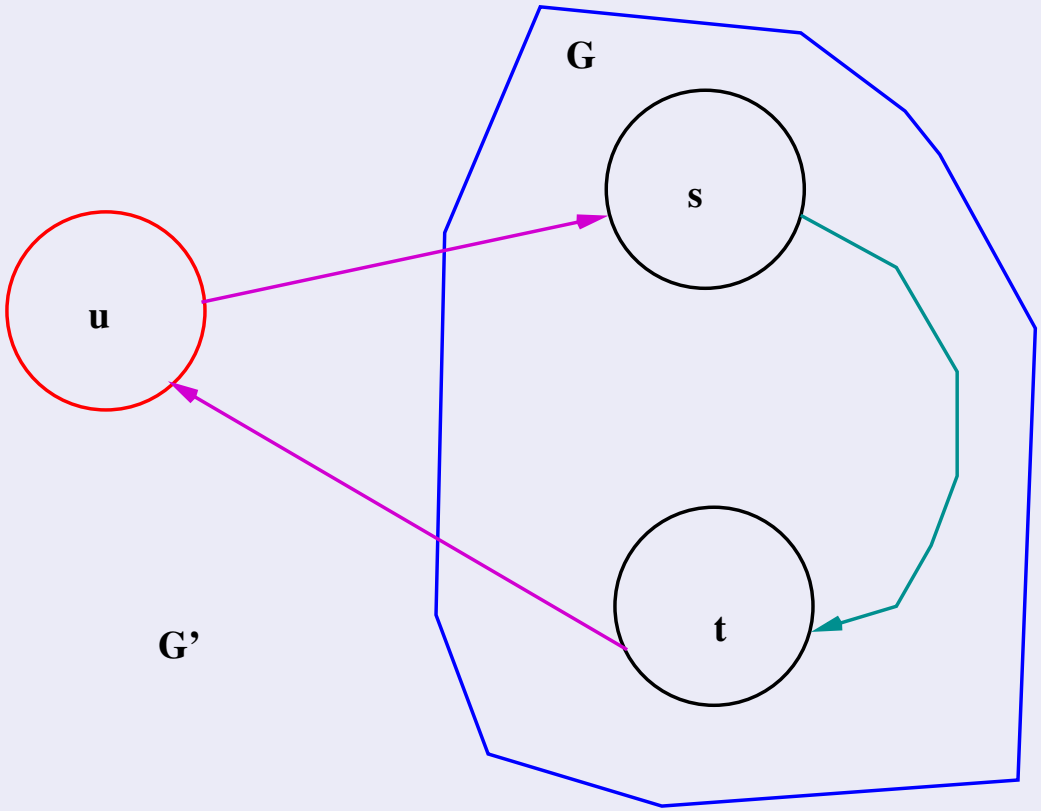
Part 2. $HAMPATH \leq_p HAMCYCLE$

- We have, $HAMPATH := \{\langle G, s, t\rangle |$ there is a (directed) Hamiltonian path from node s to node t in the directed graph $G\}$.
- Let $G = (V, E)$ be a directed graph with two vertices s and t.
- We plan to convert $\langle G, s, t\rangle$ to another directed graph $G' = (V', E')$ such that G has an (s, t)-Hamiltonian path if and only if G' has a Hamiltonian cycle.
- **Construction of** $G' = (V', E')$:
  1. $V' := V \cup \{u\}$.
  2. $E' := E \cup \{(u, s), (t, u)\}$.

Hamiltonian Cycle Problem

Part 2. *HAMPATH* $\leq_p$ *HAMCYCLE*

Construction of $G' = (V', E')$ (Continued...):



Hamiltonian Cycle Problem

Part 2. $HAMPATH \leq_p HAMCYCLE$
Construction of $G' = (V', E')$ (Continued...):

- Suppose that G has an s, t-Hamiltonian path $s, u_1, u_2, \ldots, u_m, t$. Then, $u, s, u_1, u_2, \ldots, u_m, t, u$ becomes a Hamiltonian cycle in G'.
- Conversely, let G' have a Hamiltonian cycle. If we traverse around the cycle starting from u, we must first reach s after leaving u. In order to complete the cycle we must take the edge (t, u). Between s and t the cycle visits every other node of G exactly once, that is, this cycle must be of the form $u, s, v_1, v_2, \ldots, v_m, t, u$. But then $s, v_1, v_2, \ldots, v_m, t$ is an s, t-Hamiltonian path in G.
- Clearly, this reduction runs in poly-time.

Cryptology = Cryptography + Cryptanalysis

Definition

An encryption scheme (cipher or cryptosystem) is said to be ***breakable*** if a third party, without prior knowledge of the key pair (e, d) where e is the encryption key and d is the corresponding decryption key, can systematically recover plaintext from corresponding ciphertext within some appropriate time frame.

Goal: We want this problem for an adversary (attacker) to be NP-hard (computationally infeasible).

Definition (Brute-force attack)

An encryption scheme can be broken by trying all possible keys to see which one the communicating parties are using (assuming that the class of encryption functions is public knowledge).
This is called an exhaustive search of the key space.

What is meant by "Security lies in the keys" (using brute-force attack)

Key size (bits)	**Number of alternative keys**	**Time required at 10^6 decryptions per microsecond**
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

Definition (Unconditionally secure scheme)

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 many ciphertexts are available. That is, no matter how much time an opponent has, it is impossible for him/her to decrypt the ciphertext, simply because the required information is not there.

Definition (Computationally secure scheme)

An encryption scheme is said to be “computationally secure” if the following two criteria are met:

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

Complexity Theoretic Connections

- We want **encryption and decryption** must be done in polynomial time (Class: P problems) to encrypt and decrypt the messages, respectively
- We want **cryptanalysis part** by an adversary must be computationally infeasible (Class: NP-hard problem) to break the cryptosystem

Types of Attacks on Encrypted Messages:

Type of Attack	Known to Cryptanalyst
1. Ciphertext only (COA)	• Encryption algorithm • Ciphertext to be decoded
2. Known plaintext (KPA)	• Encryption algorithm • Ciphertext to be decoded • One or more plaintext-ciphertext pairs formed with the secret key
3. Chosen plaintext (CPA)	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
4. Chosen ciphertext (CCA)	• Encryption algorithm • Ciphertext to be decoded • Purported (~~incorrect~~) ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
5. Chosen text (CTA)	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Purported ciphertext chosen by cryptanalyst together with its corresponding decrypted plaintext generated with the secret key

● Note:- CTA < CCA < CPA < KPA < COA (according to hardness of attacks)

COA is the most difficult attack,
CTA is the most easy attack..

Ciphertext-Only Attack (COA)

(i)

In a ciphertext-only attack, the attacker (Eve) has access to only some ciphertext. She tries to find the corresponding key and the plaintext.

The assumption is that Eve knows the algorithm and can intercept the ciphertext.

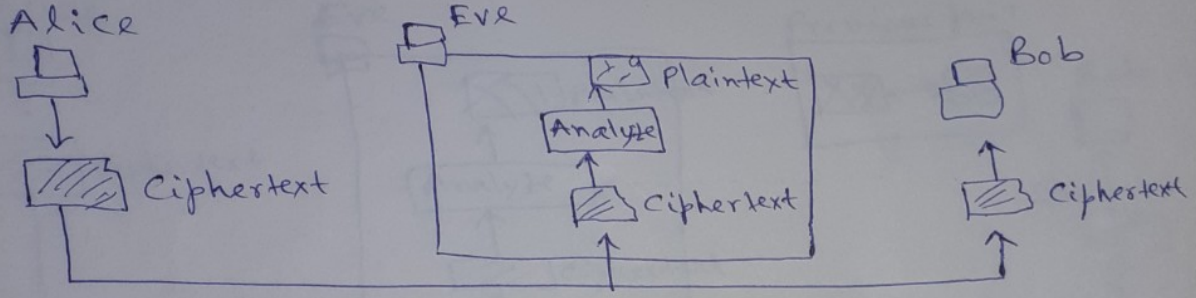
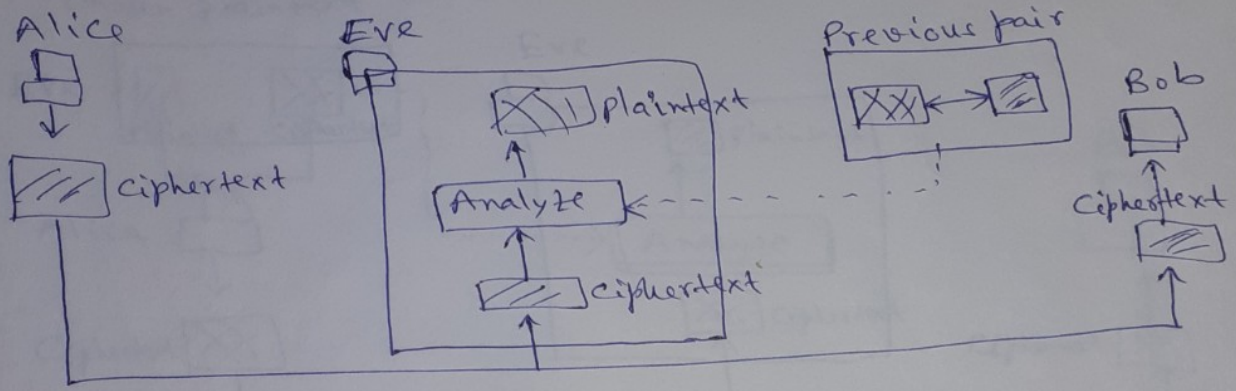


Fig. Ciphertext-only attack

Known-Plaintext Attack (KPA) (ii)

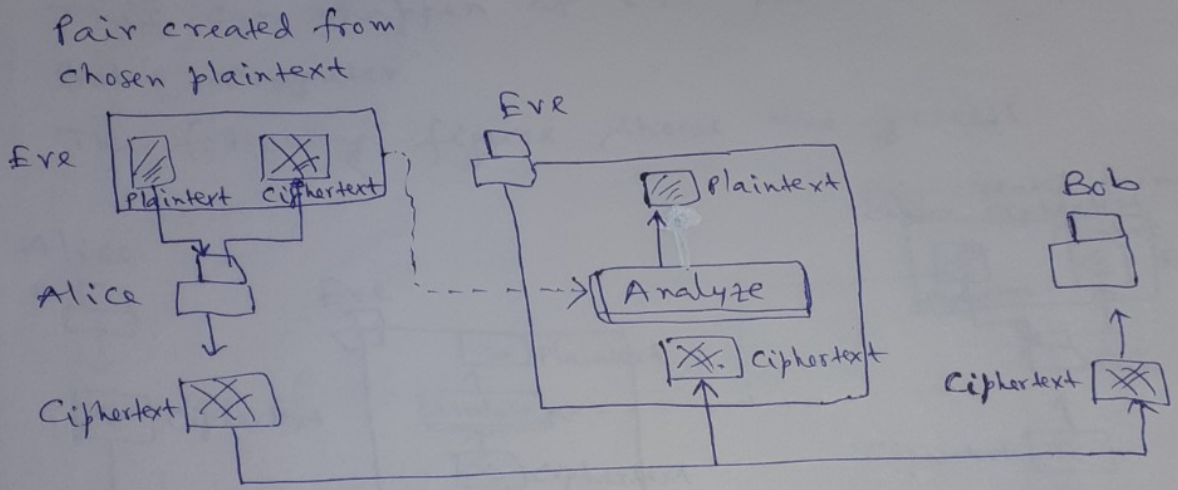
In a known-plaintext attack, Eve has access to some plaintext-ciphertext pairs in addition to the intercepted ciphertext that she wants to break, as shown in the following figure.



In this attack, the plaintext/ciphertext pairs have been collected earlier. For example, Alice has sent a secret message to Bob, but she has later made the contents of the message public. Eve has kept both the ciphertext and the plaintext to use them to break the next secret message from Alice to Bob, assuming that Alice has not changed her key.

Chosen-plaintext Attack (CPA)

The chosen-plaintext attack is similar to the known-plaintext attack, but the plaintext/ciphertext pairs have been chosen by the attacker (Eve) herself. The following figure shows the process.



This situation can happen, for example, if Eve has access to Alice's computer. She can choose some plaintext and intercept the created ciphertext.

Of course, she does not have the key because the key is normally embedded in the software used by the sender.

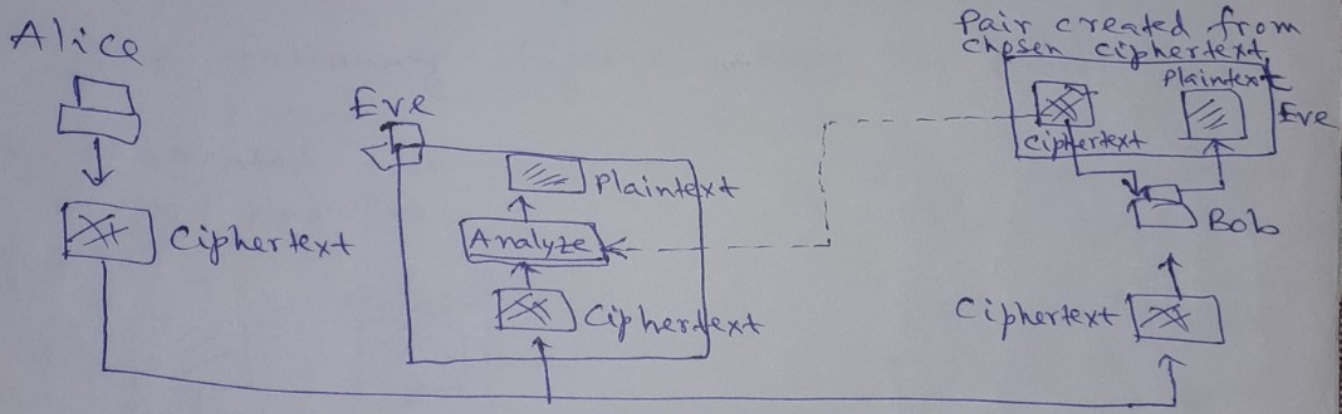
This type of attack is much easier to implement, but it is much less likely to happen.

Chosen-ciphertext Attack (CCA) (iv)

The chosen-ciphertext attack is similar to the chosen-plaintext attack, except that Eve chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.

This can happen if Eve has access to Bob's computer.

The following figure shows the process.

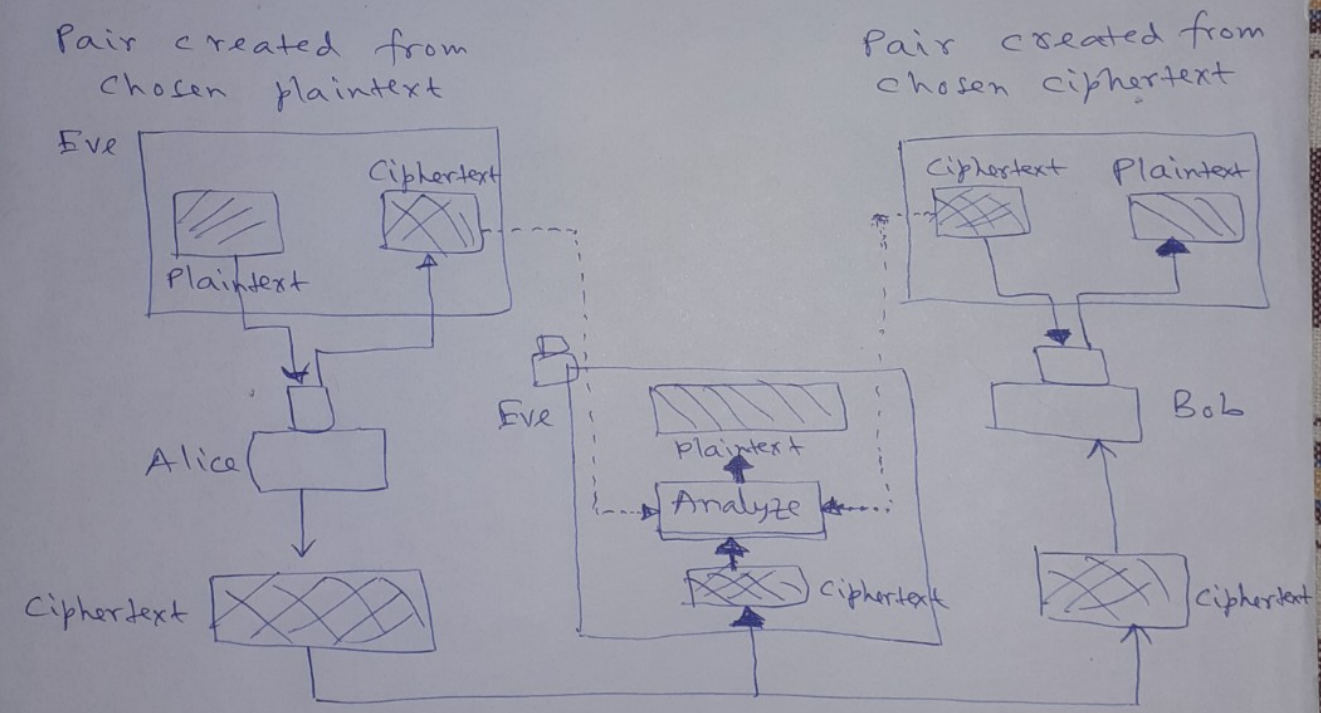


Chosen-text Attack (CTA)

The chosen text attack is a combination of both the chosen-plaintext and chosen-ciphertext attacks. In this attack, an adversary, Eve chooses some plaintext and encrypts it to form a plaintext/ciphertext pair; and also chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.

This can happen if Eve has access to both Alice's computer and Bob's computer.

The following figure shows the process.



System and Network Security

Dr. Ashok Kumar Das

Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
URL: http://www.iiit.ac.in/people/faculty/ashokkdas
Personal Home Page: http://sites.google.com/site/iitkgpakdas/

Protocol Hierarchies

- To reduce design complexity, most networks are organized as a stack of “layers” or “levels”, each one is built upon the one below it.
- The number of layers, the name of the layer, the contents of each layer, and the function of each layer differ from network to network.
- In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

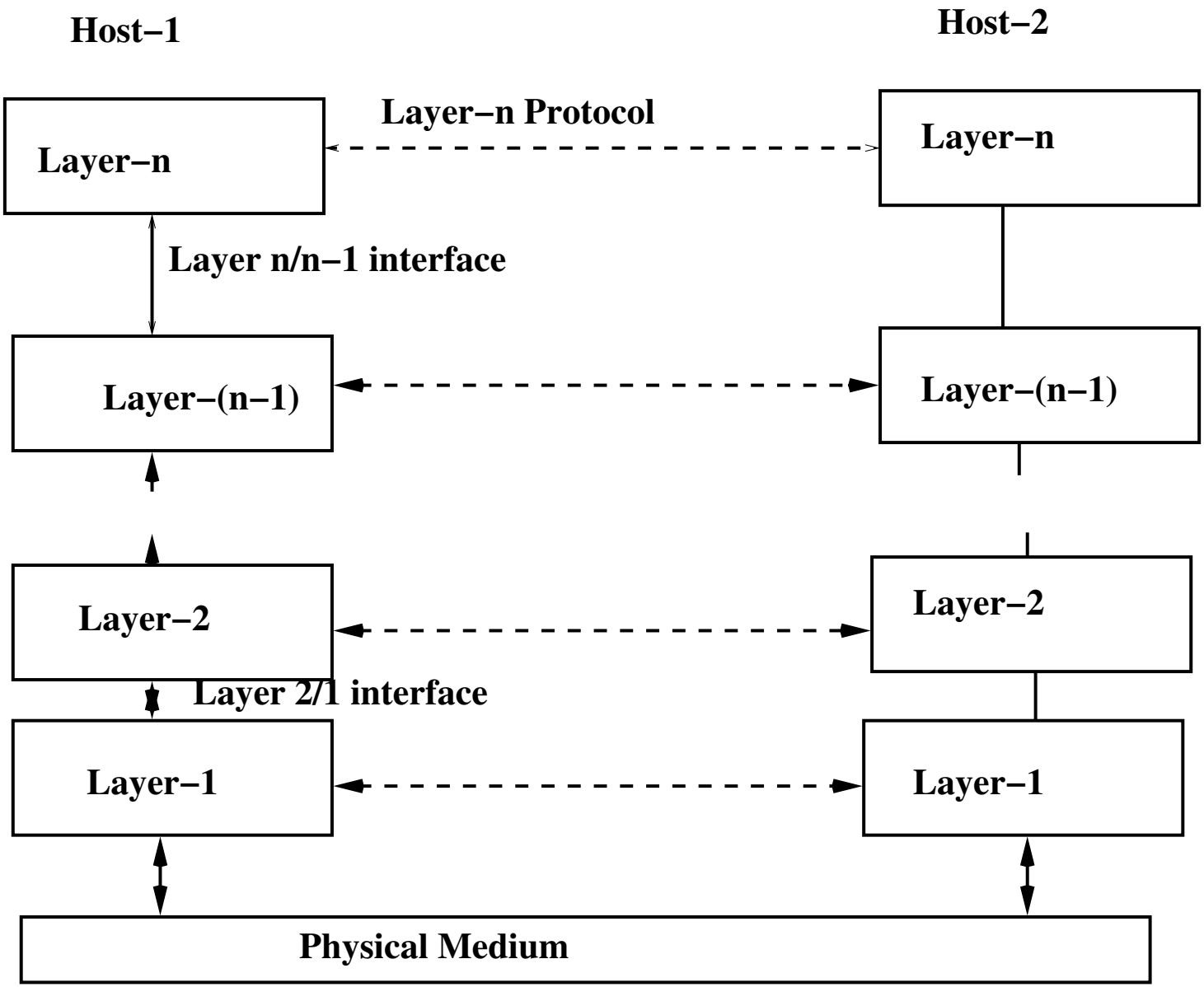
Why layering is needed?

- To provide well-defined interfaces between adjacent layers.
  - A change in one layer does not affect the other layer.
  - Interface must remain the same. [Interface defines which primitive operations and services the lower layer makes available to the upper layer.]
- Allows a structured development of network software.

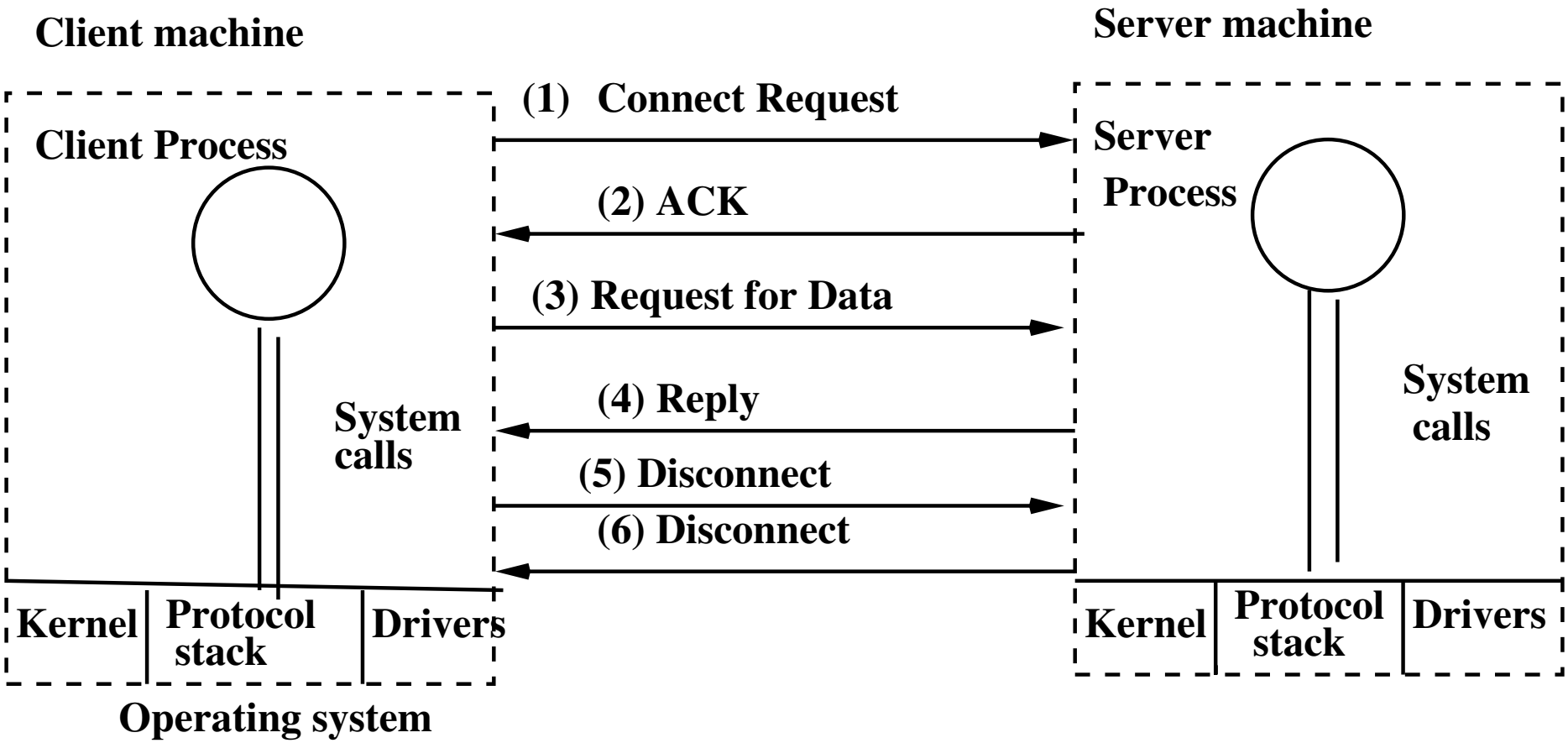
Protocol Hierarchies

- A set of layers and protocols is called a “network architecture”.
- A list of protocols used by a certain system, one protocol per layer, is called a “protocol stack”.

Layered Network Architecture



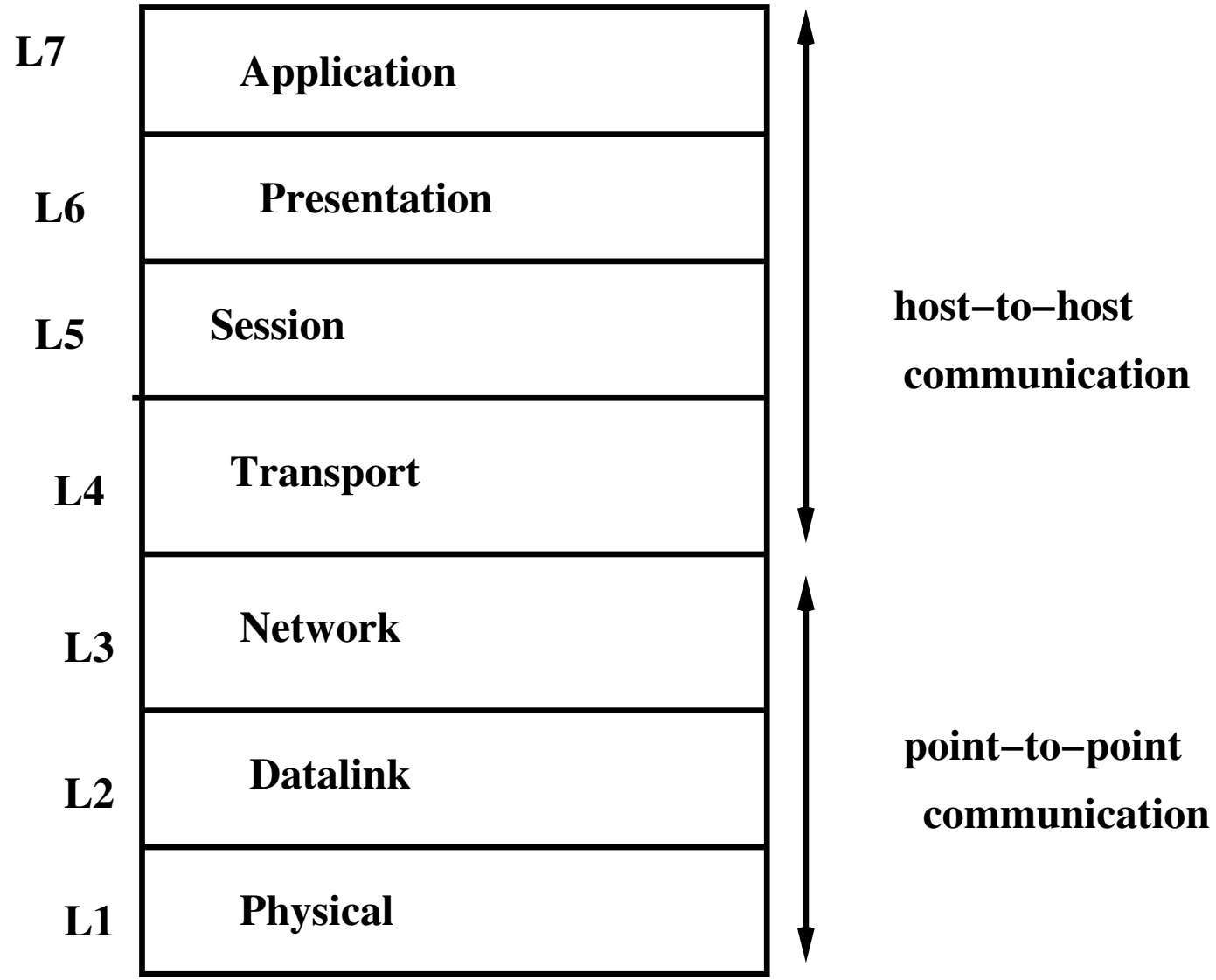
A simple client-server interaction on a connection-oriented network



The OSI Reference Model

- In 1978, International Standards Organization (OSI) proposed a 7-layer reference model for network services and protocols, known as the OSI model.
- The main objective of the OSI model as
  (1) Systematic approach to design.
  (2) Changes in one layer should not require changes in other layers.

The OSI Reference Model



Layer functions

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Physical Layer:

- Transmits raw bit stream over a physical medium.
- The design issues have to do making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not a 0 bit.
- The design issues largely deal with mechanical, electrical, and timing interfaces, and the physical transmission medium, which lies below the physical layer.
- Network components: Repeater, Multiplexer, Hubs, Amplifier.

Layer functions

Datalink Layer:

- Reliable transfer of frames (data) over a point-to-point link.
- Responsible for flow control, error control (error detection/correction), congestion control.
- Network components: Bridge, Switch, NIC, Advanced Cable Tester.

Layer functions

Network Layer:

- Establishing, maintaining and terminating connections.
- Routes packets (messages) through point-to-point link.
- Network components: Router, Frame Relay Device, ATM Switch.

Layer functions

Transport Layer:

- End-to-end reliable data transfer, with error recovery and flow control.
- Network components: Gateway.

Layer functions

Session Layer:

- Allows users on different machines (hosts) to establish sessions between them.
- Session offer various services, including
  - Dialog Control: Keeping track of whose turn it is to transmit.
  - Token Management: Preventing two parties from attempting the same critical operation at the same time.
  - Synchronization: Checkpointing long transmissions to allow them to continue from where they were after a crash.
- Network components: Gateway.

Layer functions

Presentation Layer:

- Translates data from application to network format, and vice-versa.
- All different formats from all sources are made into a common uniform format that the rest of the OSI model can understand.
- Network components: Gateway.

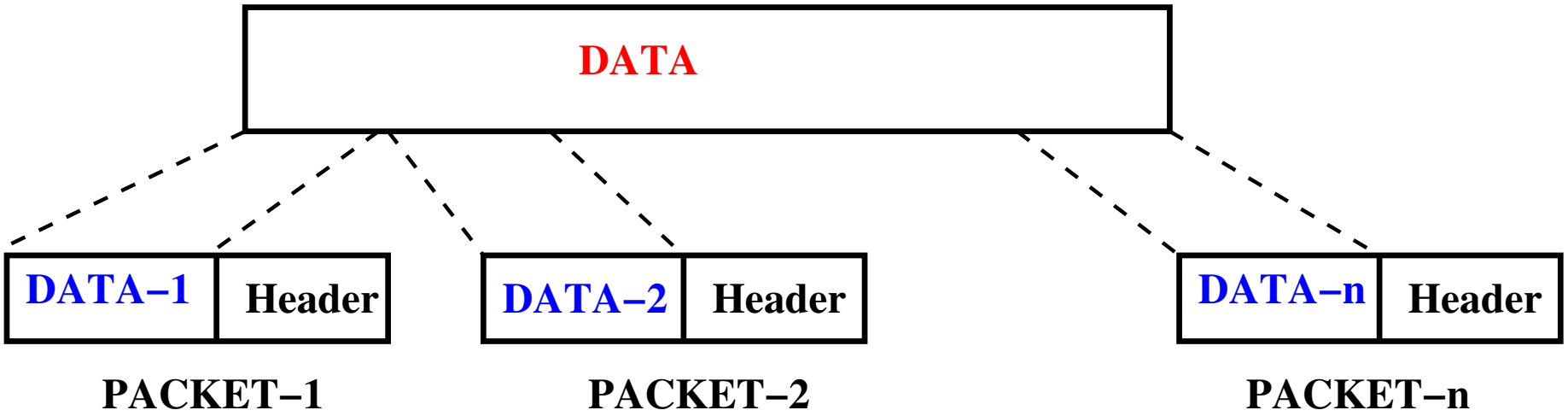
Layer functions

Application Layer:

- Interface point for user applications.
- Network components: Gateway.

Layer functions

Data handled in a particular layer:



System and Network Security

Dr. Ashok Kumar Das

Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
URL: http://www.iiit.ac.in/people/faculty/ashokkdas
Personal Home Page: http://sites.google.com/site/iitkgpakdas/

Encrypting Communications Channels

Encrypting Communications Channels

- This is the classical Alice and Bob problem:
  Alice wants to send Bob a secure message.
- What does she do?
- She encrypts the message.
- In theory, this encryption can take place at any layer in the OSI (Open Systems Interconnect) communication model.

Encrypting Communications Channels

- In practice, it takes place either at the lowest layers (one and two) or at the higher layers.
- If it takes place at the lowest layers, it is called ***link-by-link encryption (LLE)***; everything going through a particular data link is encrypted.
- If it takes place at higher layers, it is called ***end-to-end encryption (EEE)***; the data are encrypted selectively and stay encrypted until they are decrypted by the intended final recipient.

Layer-wise Network Security Protocols

Datalink Layer: Security protocols for proving the error detection mechanisms due to bit transmission error and bits changed by an adversary (intruder) in between the communication

- Internal Error Control
- External Error Control

Layer-wise Network Security Protocols

Network Layer: Internet Security Protocol (IPSec)

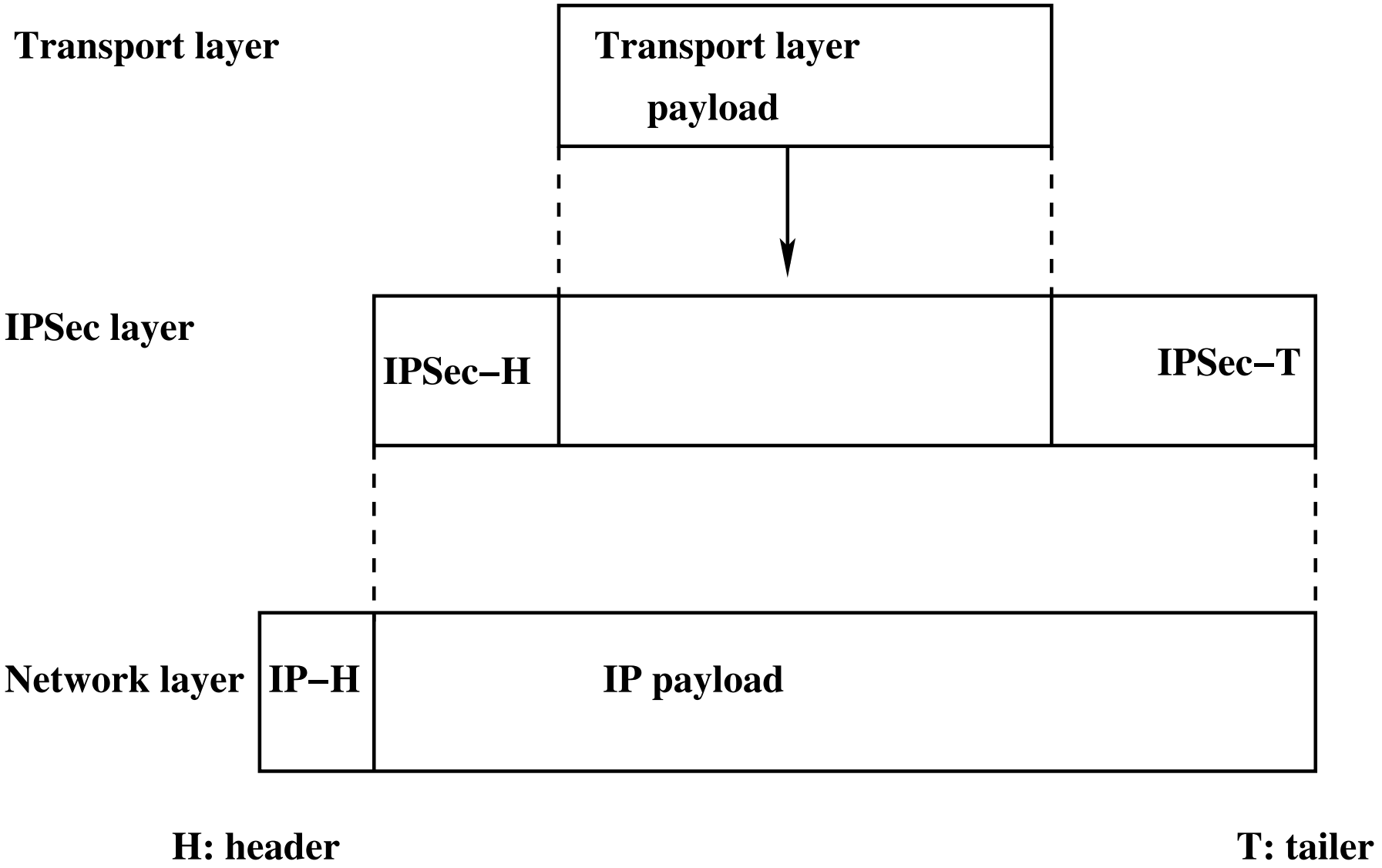
- IPSec is a collection of protocols designed by the Internet Engineering Task Force (IETF) to provide security for a packet at the network layer.
- IPSec helps create authenticated and credential packets for the IP layer.

Different Modes

Transport mode

- In this mode, IPSec protects what is delivered from the transport layer to the network layer.
- In other words, transport mode protects the network layer payload, the payload to be encapsulated in the network layer.
- Note that the transport mode does not protect the IP header.

IPSec in transport mode



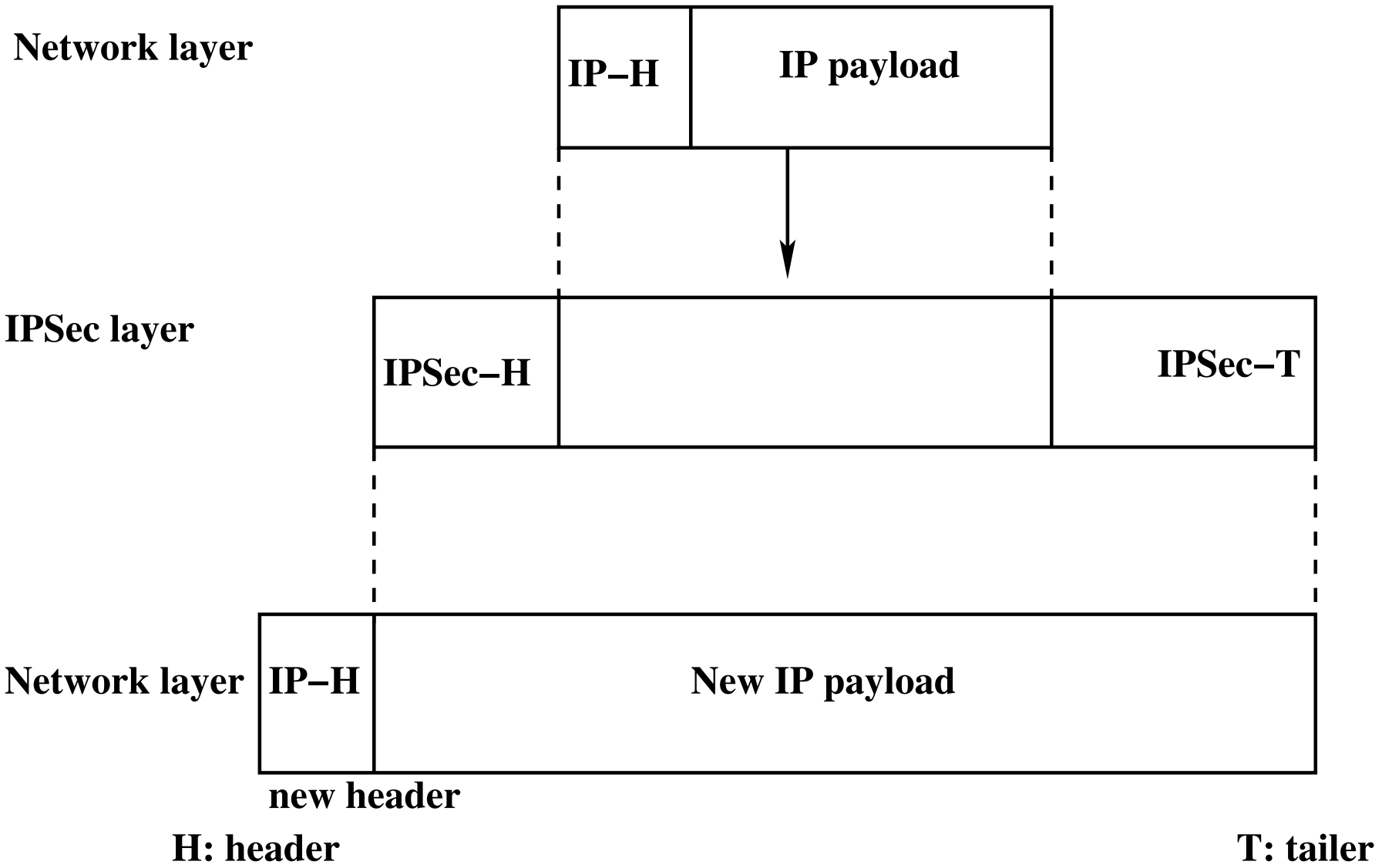
Transport mode

- This mode is normally used when we need host-to-host (end-to-end) protection of data.
- The sending host uses IPSec to authenticate and/or encrypt the payload delivered from the transport layer.
- The receiving host uses IPSec to check the authentication and/or decrypt the IP packet and deliver it to the transport layer.

Tunnel mode

- In this mode, IPSec protects the entire IP packet.
- It takes an IP packet, including the header, applies IPSec security methods to the entire packet, and then adds a new IP header.

IPSec in tunnel mode



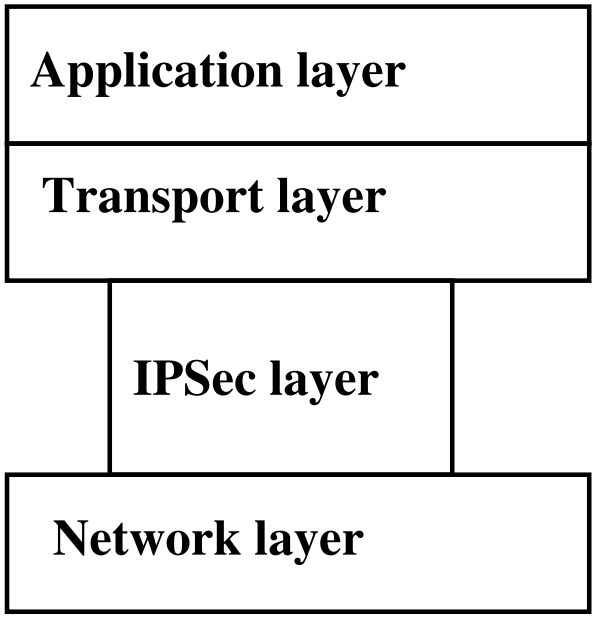
Tunnel mode

- The new IP header has different information than the original IP header.
- Tunnel mode is normally used between two routers, between a host and a router, or between a router and a host.
- In other words, tunnel mode is used when either the sender or the receiver is not a host.
- The entire original packet is protected from intrusion between the sender and the receiver, as if the whole packet goes through an imaginary tunnel.

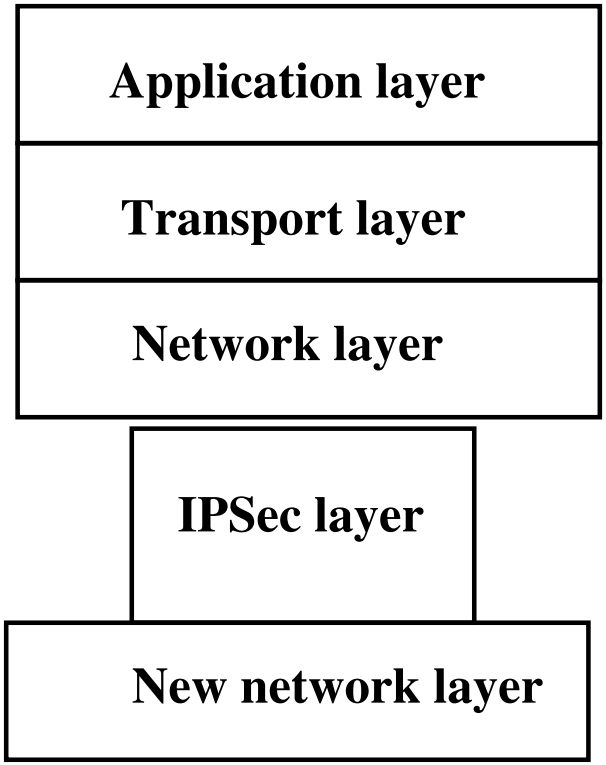
Transport mode *versus* Tunnel mode

- In transport mode, the IPSec layer comes between the transport layer and the network layer.
- In tunnel mode, the flow is from the network layer to the IPSec layer and then back to the network layer again.

Transport mode *versus* Tunnel mode



(a) Transport mode



(b) Tunnel mode

Layer-wise Network Security Protocols

Transport Layer

- Secure Socket Layer (SSL) provides security services between TCP and applications that use TCP. The Internet standard version is called Transport Layer Service (TLS).
- SSL/TLS provides confidentiality using symmetric encryption and message integrity using a message authentication code.
- SSL/TLS includes protocol mechanisms to enable two TCP users to deter- mine the security mechanisms and services they will use.
- HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server.
- Secure Shell (SSH) provides secure remote logon and other secure client/server facilities.

Layer-wise Network Security Protocols

Application Layer

- **Pretty Good Privacy (PGP)**
  - PGP is an open-source, freely available software package for e-mail security. It provides authentication through the use of digital signature, confidentiality through the use of symmetric block encryption, compression using the ZIP algorithm, and e-mail compatibility using the radix-64 encoding scheme.
  - PGP incorporates tools for developing a public-key trust model and public-key certificate management.
- **Secure/Multipurpose Internet Mail Extension (S/MIME)**
  - S/MIME is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.
  - S/MIME is an Internet standard approach to e-mail security that incorporates the same functionality as PGP.

Layer-wise Network Security Protocols

Application Layer

- **Secure Electronic Transaction (SET)**
  - SET is an open encryption and security specification designed to protect credit card transactions on the Internet.
  - SET emerged from a call for security standards by MasterCard and Visa in February 1996.
  - A wide range of companies were involved in developing the initial specification, including IBM, Microsoft, Netscape, RSA, Terisa, and Verisign.
  - Beginning in 1996, there have been numerous tests of the concept, and by 1998 the first wave of SET-compliant products was available.

SET Transaction

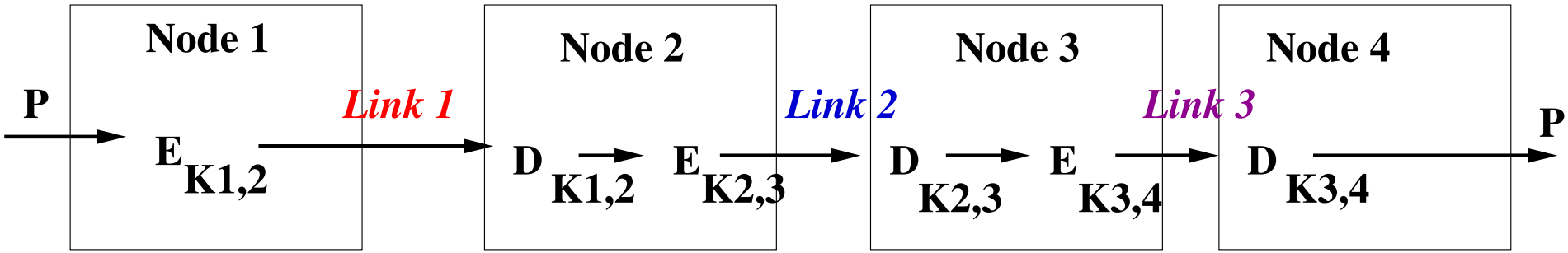
1. customer opens account
2. customer receives a certificate
3. merchants have their own certificates
4. customer places an order
5. merchant is verified
6. order and payment are sent
7. merchant requests payment authorization
8. merchant confirms order
9. merchant provides goods or service
10. merchant requests payment

Encrypting Communications Channels

Link-by-link encryption

- The easiest place to add encryption is at the physical layer.
- The interfaces to the physical layer are generally standardized, and it is easy to connect hardware encryption devices at this point.
- These devices encrypt all data passing through them, including data, routing information, and protocol information.
- They can be used on any type of digital communication link.
- On the other hand, any intelligent switching or storing nodes between the sender and the receiver need to decrypt the data stream before processing it.

Link-by-link encryption



P: plaintext message;
$K_{1,2}$: key shared between nodes 1 and 2;
$K_{2,3}$: key shared between nodes 2 and 3;
$K_{3,4}$: key shared between nodes 3 and 4;
$E_K(\cdot)$: encryption using the key K;
$D_K(\cdot)$: decryption using the key K.

Link-by-link encryption

Advantages

- Easier operation, since it can be made transparent to the user. That is, everything is encrypted before being sent over the link.
- Only one set of keys per link is required.
- Provides traffic-flow security, since any routing information is encrypted.

Link-by-link encryption

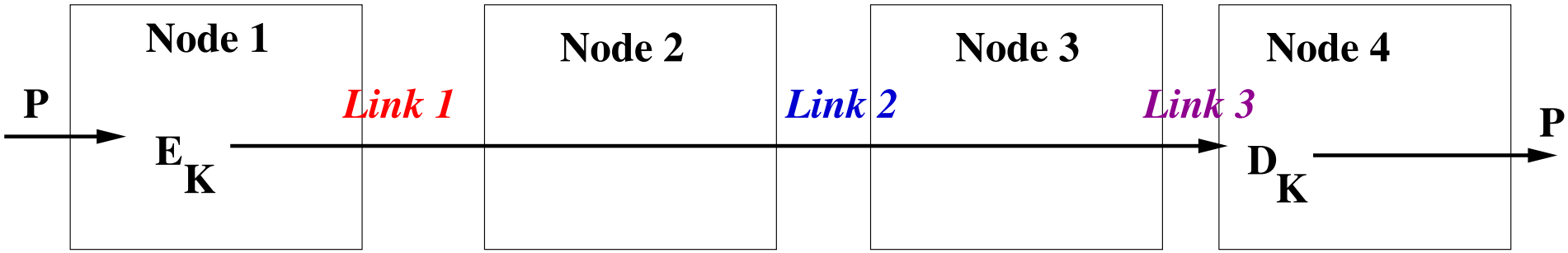
Disadvantages

- Data is exposed in the intermediate nodes.
- The biggest problem with encryption at the physical layer is that each physical link in the network needs to be encrypted: Leaving any link unencrypted reveals the security of the entire network.
  If the network is large, the cost may quickly become prohibitive for this kind of encryption.
- Additionally, every node in the network must be protected, since it processes unencrypted data.
  If all the network's users trust one another, and all nodes are in secure locations, this may be tolerable.

End-to-end encryption

- This approach is to put encryption equipment between the network layer and the transport layer.
- The encryption device must understand the data according to the protocols up to layer three and encrypt only the transport data units, which are then recombined with the un-encrypted routing information and sent to lower layers for transmission.
- This approach avoids the encryption/decryption problem at the physical layer.
- By providing EEE, the data remains encrypted until it reaches its final destination.

End-to-end encryption



P: plaintext message;
K: key shared between nodes 1 and 4;
$E_K(\cdot)$: encryption using the key K;
$D_K(\cdot)$: decryption using the key K.

End-to-end encryption

Advantages

- Higher secrecy level.

End-to-end encryption

Disadvantages

- The primary problem with EEE is that the routing information for the data is not encrypted; a good cryptanalyst can learn much from who is talking to whom, at what times and for how long, without ever knowing the contents of those conversations.
- Key management is also more difficult since individual users must make sure they have common keys.
- Traffic analysis is possible, since routing information is not encrypted.

Combining the Two: Link-by-link encryption and End-to-end encryption

- Combining the two, while most expensive, is the most effective way of securing a network.
- Encryption of each physical link makes any analysis of the routing information impossible, while end-to-end encryption reduces the threat of unencrypted data at the various nodes in the network.
- Key management for the two schemes can be completely separate:
  The network managers can take care of encryption at the physical level, while the individual users have responsibility for end-to-end encryption.

Comparing link-by-link encryption and end-to-end encryption

Link-by-link encryption	End-to-end encryption
Security within hosts	
1. Message exposed in sending host.	1. Message encrypted in sending host.
2. Message exposed in intermediate nodes.	2. Message remains encrypted in intermediate nodes.

Comparing link-by-link encryption and end-to-end encryption

Link-by-link encryption	End-to-end encryption
Role of user	
1. Applied by sending host.	1. Applied by sending process.
2. Invisible to user.	2. User applies encryption.
3. Host maintains encryption.	3. User must find algorithm.
4. One facility for all users.	4. User selects encryption.
5. Can be done in hardware.	5. More easily done in software.
6. All or no messages encrypted.	6. User chooses to encrypt or not, for each message.

Comparing link-by-link encryption and end-to-end encryption

Link-by-link encryption	End-to-end encryption
Implementation concerns	
1. Requires one key per host pair.	1. Requires one key per user pair.
2. Requires encryption hardware or software at each host.	2. Requires encryption hardware or software at each node.
3. Provides node authentication.	3. Provides user authentication.

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Introduction to Information Security

Dr. Ashok Kumar Das

Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
URL: http://www.iiit.ac.in/people/faculty/ashokkdas
https://sites.google.com/view/iitkgpakdas/

Symmetric-Key Encryption

Symmetric-Key Encryption

Model of conventional encryption

- Consider an encryption scheme consisting of
  - the set of encryption transformations $\{E_e : e \in K\}$
  - the set of corresponding decryption transformations $\{D_d : d \in K\}$, where K is the key space.
- The encryption scheme is said to be S-key or symmetric-key, if for each associated encryption/decryption key pair (e, d), it is computationally "easy" to determine d from e and to determine e from d.
- In most practical symmetric-key encryption schemes, $e = d$.
- Other terms used are single-key, one-key, private-key and conventional encryption.

Symmetric-Key Encryption

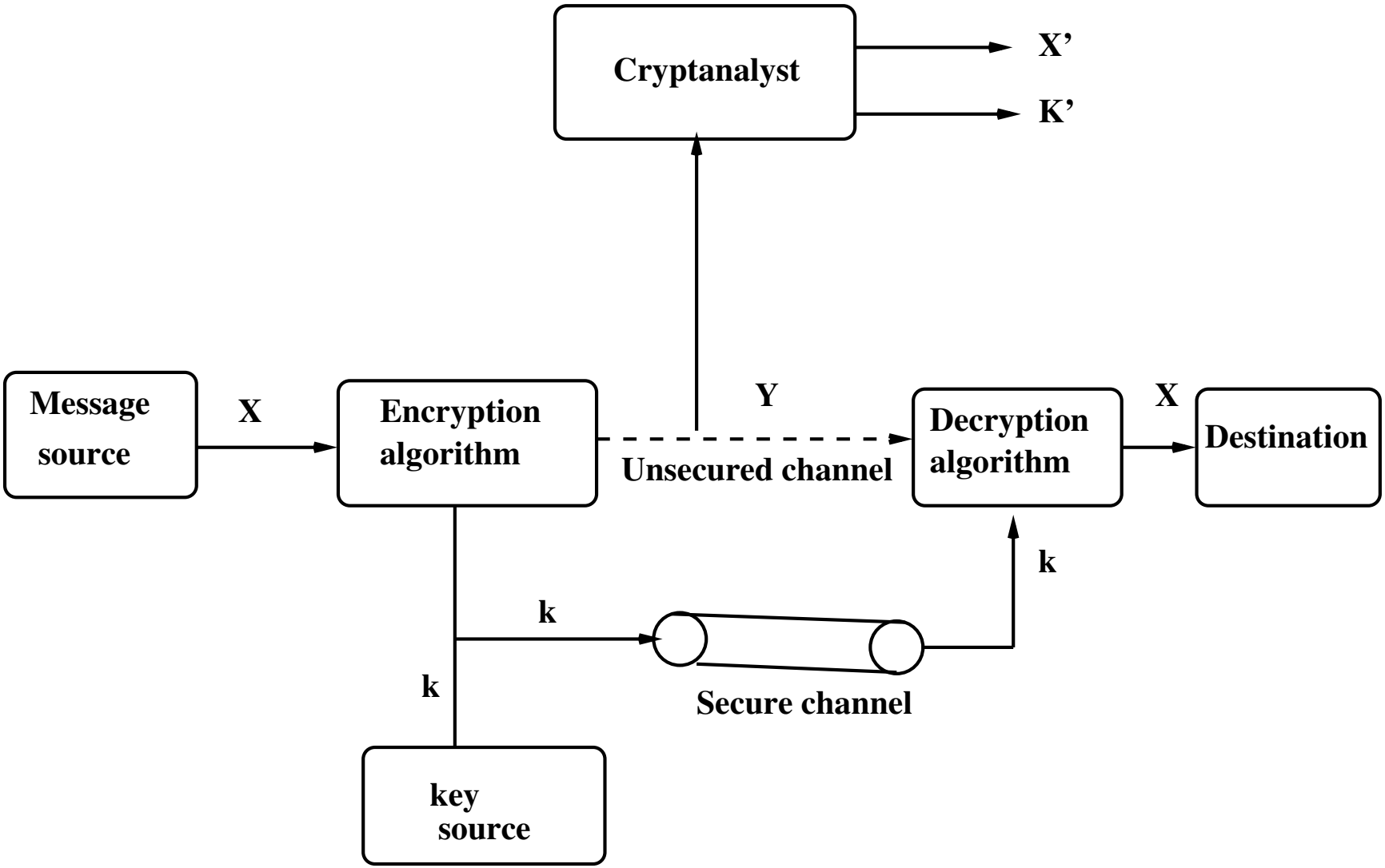


Figure: Model of conventional encryption

Symmetric-Key Encryption

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Model of conventional encryption

- With the message $X = [X_1, X_2, \ldots, X_n]$ and the encryption key k as input, the encryption algorithm forms the ciphertext $Y = [Y_1, Y_2, \ldots, Y_n]$.
- $Y = E_k[X]$
- $Y_i = E_k[X_i]$, for $i = 1, 2, \ldots, n$.
- $X = D_k[Y]$
- $X_i = D_k[Y_i]$, for $i = 1, 2, \ldots, n$.

Symmetric-Key Encryption

Classical Techniques

- There are two classical techniques in conventional or symmetric-key encryption scheme:
  - Substitution Techniques: Involve the substitution of a ciphertext symbol for a plaintext symbol.
  - Transposition Techniques: A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters.

Symmetric-Key Encryption

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Caesar Cipher

- It is the earliest known use of a substitution cipher, and the simplest, was by Julius Caesar.
- Each letter of the alphabet is replaced with the letter standing the three places further down the alphabet.
- For example,
  plaintext: meet me after the new year party
  ciphertext: PHHW PH DIWHU WKH QHZ BHDU SDUWB
- Each letter is wrapped around, so that the letter following Z is A. Define the transformation by listing all possibilities as follows.

plaintext:	a	b	c	...	v	w	x	y	z
ciphertext:	D	E	F	...	Y	Z	A	B	C

Symmetric-Key Encryption

Caesar Cipher

- Encoding technique: Let us assign a numerical equivalent to each letter:

a	b	c	...	v	w	x	y	z
0	1	2	...	21	22	23	24	25

- Mathematical model:
  - Encryption: For each plaintext letter p, substitute the ciphertext letter c: $c = E_k(p) = (p + 3) \pmod{26}$, where $k = 3$.
  - Decryption: For each ciphertext letter c, substitute the plaintext letter p: $p = D_k(c) = (c - 3) \pmod{26}$, where $k = 3$.

Symmetric-Key Encryption

The Generalized Caesar Cipher

- A shift may be of any amount, so that the general Caesar algorithm is as follows.
- Mathematical model
  - Encryption: For each plaintext letter p, substitute the ciphertext letter c: $c = E_k(p) = (p + k) \pmod{26}$, where $0 \leq k \leq 25$.
  - Decryption: For each ciphertext letter c, substitute the plaintext letter p: $p = D_k(c) = (c - k) \pmod{26}$, where $0 \leq k \leq 25$.

Symmetric-Key Encryption

Security issues of the Caesar cipher

- If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed.
- The key space K in this case contains 25 keys, that is $|K| = 25$.
- Attacker simply tries all the 25 possible keys.
- In this case, the attacker could be able to recover the plaintext as well as the encryption key k from the ciphertext easily (It is an example of Ciphertext-only attack (COA)).

Symmetric-Key Encryption

Characteristics of the Caesar cipher

- The encryption an decryption algorithms are known.
- There are only 25 keys to try.
- The language of the plaintext is known and easily recognizable.

Symmetric-Key Encryption

Vernam Cipher

- An encryption system was introduced by an AT& T engineer named Gilbert Vernam in 1918.
- He introduced a new parameter (keyword) which is as long as the plaintext and has no statistical relationship to it.
- **Encryption algorithm**
  The system can be expressed as follows:
  $c_i = p_i \oplus k_i$
  where $p_i = i^{th}$ binary digit of plaintext,
  $c_i = i^{th}$ binary digit of ciphertext,
  $k_i = i^{th}$ binary digit of key,
  $\oplus =$ bitwise exclusive-or (XOR) operator.
- **Decryption algorithm**
  Because of the properties of XOR, decryption simply involves the same bitwise operation: $p_i = c_i \oplus k_i$.

Symmetric-Key Encryption

Vernam Cipher

- **Construction of key:**
  - Keyword should be as long as the plaintext and can be repeating.
- Vernam cipher is an example of classical stream cipher.
- It is also called one-time pad, because each plaintext is appended with random key.
- It is proved in the literature that one-time pad is unbreakable (proof will be given mathematically later), since it produces random output that bears NO satistical relationship to the plaintext.

Symmetric-Key Encryption

Vernam Cipher

Problems with the one-time pad

- Generation of key.
- Problem of key distribution and protection.

Because of these difficulties, the one-time is of limited utility, and is used primarily for low-bandwidth channels requiring very high security.

Introduction to Information Security

Dr. Ashok Kumar Das

IEEE Senior Member
Associate Proofessor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
URL: http://www.iiit.ac.in/people/faculty/ashokkdas
https://sites.google.com/view/iitkgpakdas/

Symmetric-Key Encryption

Principle of Shannon (1945)

- **Diffusion:** The mechanism of diffusion seeks to make the statistical relationship between the plaintext and ciphertext as complex as possible in order to thwart attempts to deduce the key.
  - Diffusion can be achieved by repeatedly performing some permutation on the data followed by applying a function to that permutation.
- **Confusion:** It seeks to make the statistical relationship between the ciphertext and the value of encrypted key as complex as possible in order to thwart attempts to deduce the key.
  - Confusion can be achieved by the use of a complex substitution algorithm.

Symmetric-Key Encryption

The Fiestel Cipher

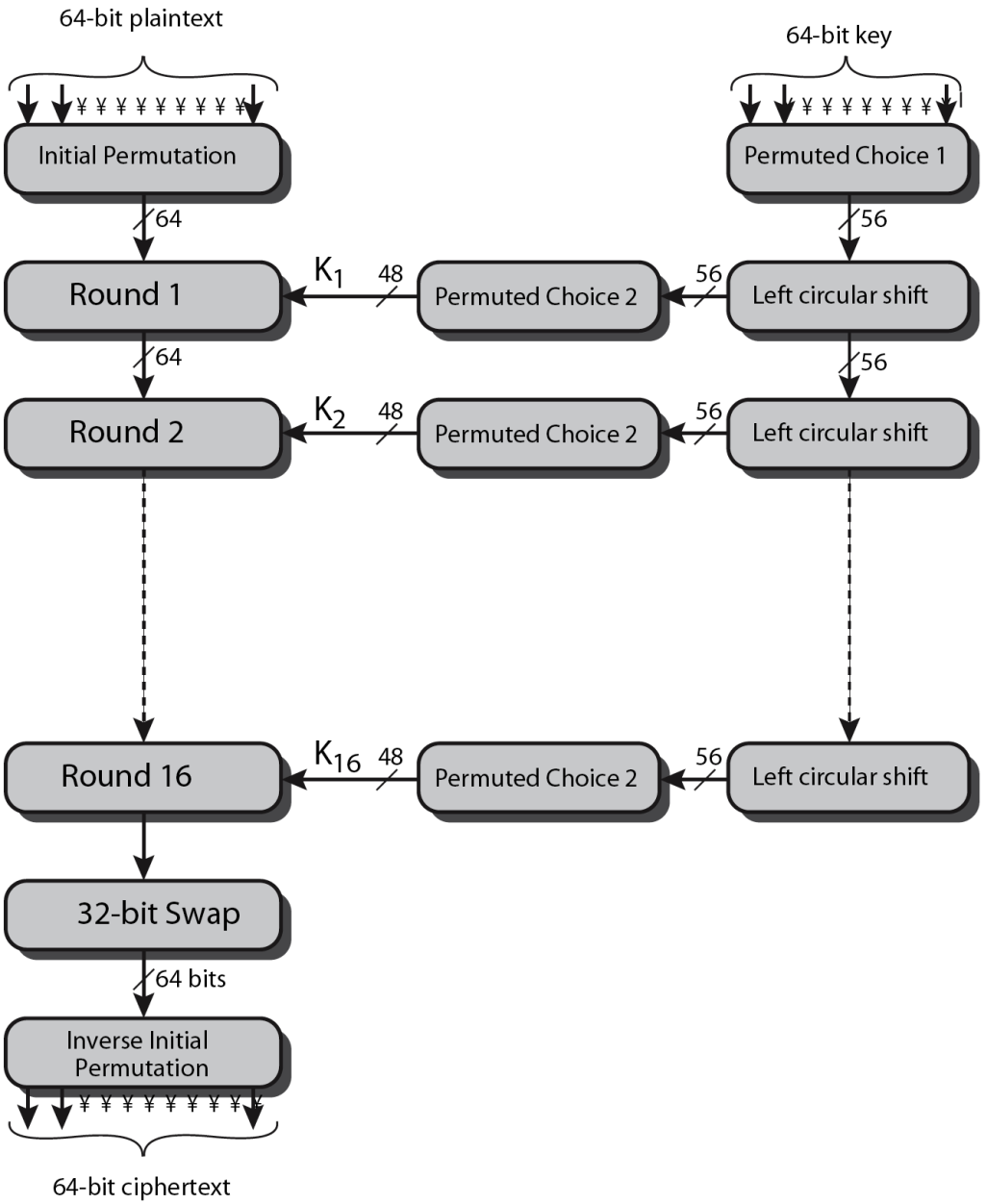
- All modern day block ciphers are based on Fiestel cipher structure.
- Fiestel structure is based on the principle of Shannon (1945): Diffusion and Confusion
- Fiestel structure is useful to construct a SPN (Substitution-Permutation Network) cipher

Symmetric-Key Encryption

Data Encryption Standard (DES)

- The most widely used encryption is based on the Data Encryption Standard (DES) adopted in 1977 by the National Institute of Standards and Technology (NIST), USA.
- For DES, data are encrypted in 64-bit blocks using a 56-bit key.
- The encryption algorithm transforms 64-bit input in a series of steps into a 64-bit output.
- The same steps, with the same key, are used to reverse the encryption (decryption).
- Mathematically, $DES : \{0,1\}^{64} \times \{0,1\}^{56} \longrightarrow \{0,1\}^{64}$ such that the ciphertext be $C = DES_K(P)$, where $K \in \{0,1\}^{56}$ is the 56-bit key, $P \in \{0,1\}^{64}$ is the plaintext message (block) and $C \in \{0,1\}^{64}$ is the ciphertext block.

Overview of Data Encryption Standard (DES)

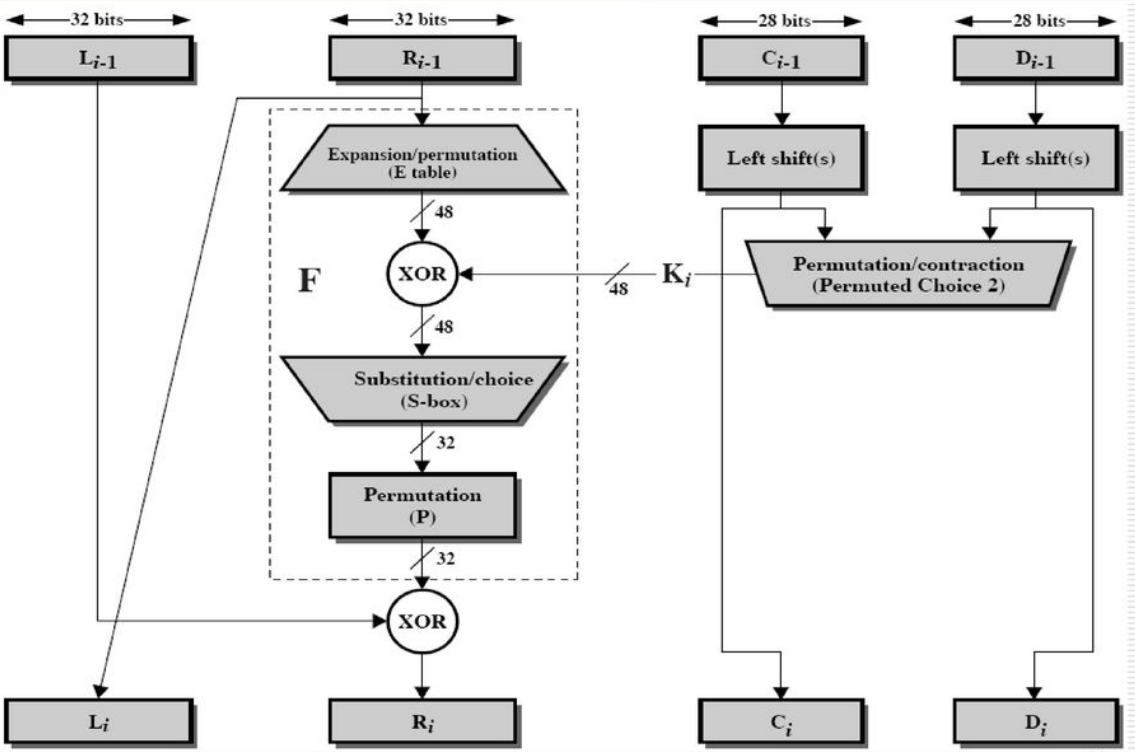


Data Encryption Standard (DES)

- K: given 56 bit key
- K is converted to 64 bit key packed with 8 bit parity: parity 8 bits at positions 8, 16, 24, 32, 40, 48, 56, and 64.
- $K_1, K_2, \cdots, K_{16}$: 16 round keys
- **Schedule of left circular shifts:**

  if (**round number = 1, 2, 9, 16**), then bits_rotated = 1
  else
  bits_rotated = 2

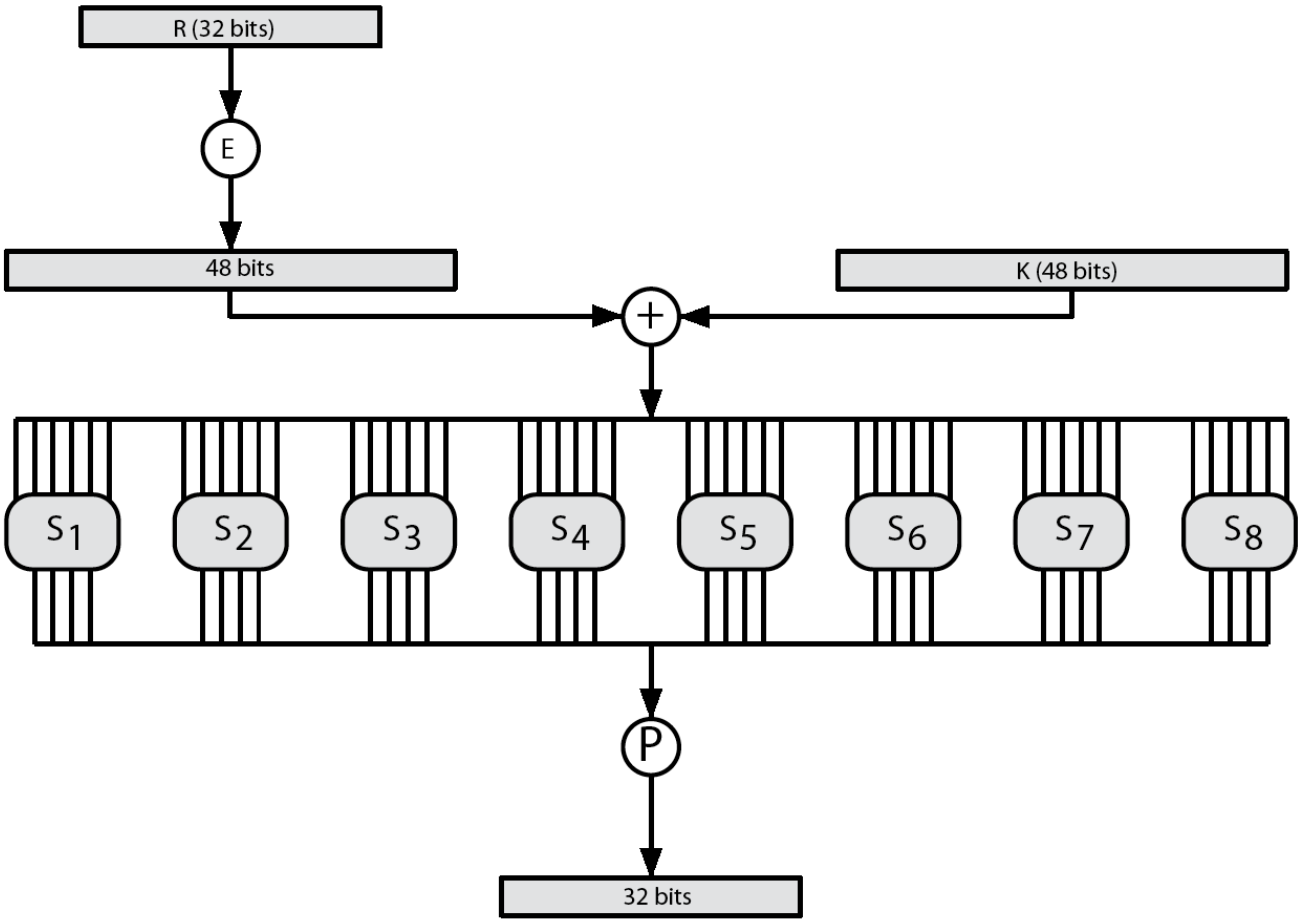
Single Round of DES



$L_i = R_{i-1}$; $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$, $\forall i = 1, 2, \cdots 16$

E: Expansion/permutation; S-Box (S_i)**: Substitution/choice; P: permutation;** L_i **: left half (32 bits) of message;** R_i **: right half (32 bits) of message;** C_i **: left half (28 bits) of key;** D_i **: left half (28 bits) of key.**

Calculation of function $F(R_i, K_i)$ in DES



$F(R_i, K_i) = P(S(E(R_i) \oplus K_i))$

E: Expansion/permutation; S: S-Box; L_i : left half (32 bits) of message; R_i : right half (32 bits) of message; K_i: i^{th} round key.

Initial Permutation (IP) and IP^{-1}

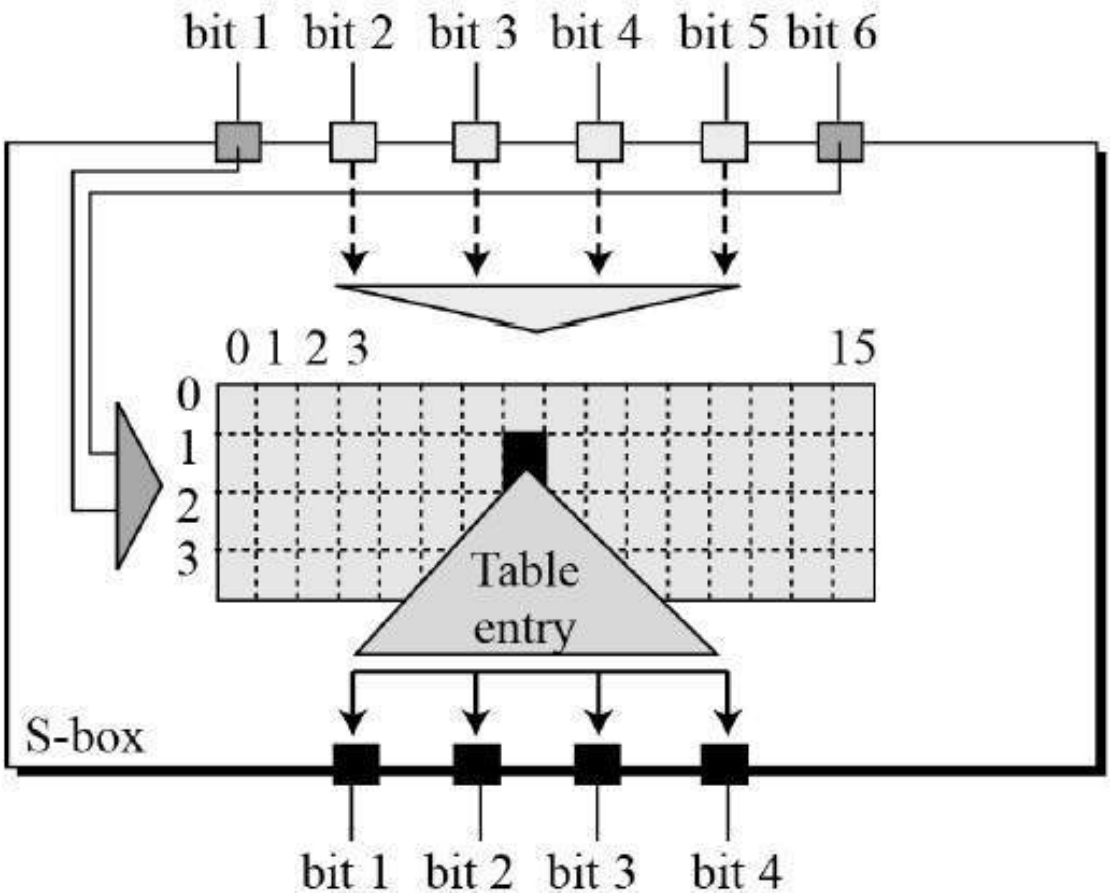
IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

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

E: Expansion/permutation

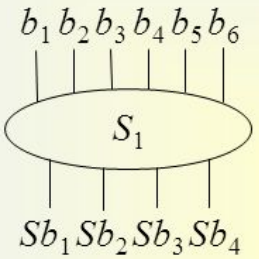
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	28
24	25	26	27	28	29
28	29	30	31	32	1

S-Box Rule


bit 1 bit 2 bit 3 bit 4 bit 5 bit 6
0 1 2 3
15
0
1
2
3
Table entry
S-box
bit 1 bit 2 bit 3 bit 4

S-Box (S_1) Example

S-box (substitution box)



Look-up a value from the table using
$b_1\,b_6$: row
$b_2\,b_3\,b_4\,b_5$: column

$b_1\,b_6$: row — S_1-box table

	Sb_1															
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

$b_2\,b_3\,b_4\,b_5$: column

8

Example: Input (6 bits) = 1 1 1 0 0 1; row-index $= b_1\,b_6 =$ **(1 1)**$_2 = 3$**;**
col-index $= b_2\,b_3\,b_4\,b_5 =$ **(1 1 0 0)**$_2 = 12$**;**
output = S_1**[row-index][col-index]** $= 10 =$ **(1 0 1 0)**$_2$

DES S-Box Tables

Substitution Boxes S-Boxes

Box	Row	Column 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1																	
	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2																	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3																	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4																	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5																	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6																	
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7																	
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8																	
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Data Encryption Standard (DES)

Theorem

Let $DES_{K_1 K_2 \cdots K_{16}}$ denote the DES encryption function, where $K_1, K_2, \ldots, K_{16}$ be the 16 round keys of a given 56-bit input key K. Then, for all plaintext messages $x \in \{0,1\}^{64}$, $DES_{K_{16} K_{15} \cdots K_1}(DES_{K_1 K_2 \cdots K_{16}}(x)) = x$, that is, $DES_{K_{16} K_{15} \cdots K_1}$ becomes the DES decryption function.