Research in Information Security

Instructors:
Dr. Ashok Kumar Das and **Dr. Kannan Srinathan**

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

Contact Information:
Dr. Ashok Kumar Das: ashok.das@iiit.ac.in
Dr. Kannan Srinathan: srinathan@iiit.ac.in

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 and Computing from ***IIT Kharagpur***, India
- **Visiting Research Professor** at the ***Old Dominion University (ODU), Virginia, USA***, with the Virginia Modeling, Analysis, and Simulation Center (2022, 2024)
- **Adjunct Professor** at the ***Korea University, Seoul, South Korea*** from July 2024 onwards [2025 QS World Rank: 67]
- **Research Interests:** Network and system security, blockchain, security in Internet of Things (IoT), Internet of Vehicles (IoV), Internet of Drones (IoD), cloud/fog computing, intrusion detection, AI/ML security, and post-quantum cryptography
- **Research Highlights [Research Publications (Total: 426)]:**
  - Number of Journal Papers: 361
  - Number of Conference Papers: 52
  - Number of Book Chapters: 10
  - Number of Edited Books/Volumes: 3
  - Number of IEEE Transactions/IEEE Journal/IEEE Magazine Papers: 164

Research Contributions

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- ***Total citations: 22,488, h-index: 87, i10-index: 267***
  (According to Google Scholar Citations as on July 29, 2024)
- **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**
  * **IEEE Transactions on Network Science and Engineering**
  * **IEEE Transactions on Intelligent Transportation Systems**
  * **IEEE Sensors Journal**
  * **IEEE Transactions on Network and Service Management**
  * **IEEE Communications Surveys and Tutorials**

About me: Journal Editorial Board Members

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- Associate Editor: **IEEE Transactions on Information Forensics and Security** (SCI Impact Factor: 6.8)
- Associate Editor: **IEEE Systems Journal** (SCI Impact Factor: 4.4)
- Editor: **Journal of Network and Computer Applications (Elsevier)** (SCI Impact Factor: 8.7)
- Editor (Technical Committee): **Computer Communications (Elsevier)** journal (SCI Impact Factor: 6)
- Associate Editor: **Journal of Cloud Computing (Springer)** (SCI Impact Factor: 4)
- Associate Editor: **Alexandria Engineering Journal (Elsevier)** (SCI Impact Factor: 6.8)
- Associate Editor: **IET Communications** journal (SCI Impact Factor: 1.6)
- Associate Editor: **Cyber Security and Applications (Elsevier)**
- Editor: **KSII Transactions on Internet and Information Systems** (SCI Indexed Journal)
- Editor: **International Journal of Internet Technology and Secured Transactions (Inderscience)** (2016 -)
- Guest Editor: **Computers & Electrical Engineering (Elsevier)** (SCI Indexed Journal), 2016
- Guest Editor: **ICT Express (Elsevier)** (SCI Indexed Journal), 2019
- Guest Editor: **Wireless Communications and Mobile Computing** (SCI Indexed Journal), 2020

About me

- Visiting Research Professor at the **Old Dominion University (ODU), Suffolk, VA 23435, USA**, with the Virginia Modeling, Analysis, and Simulation Center (VMASC) [2022, 2024]
- **Adjunct Professor** at the ***Korea University, Seoul, South Korea*** [2025 QS World Rank: 67], for two-year term starting from July 2024
- Listed in the **Web of Science (Clarivate^{TM}) Highly Cited Researcher 2022, 2023** 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 **top 2% scientists world-wide (all fields) and also in the area of Networking & Telecommunications, with Rank world-wide (by subject area): 39 for the year: 2022 (October 2023)**.
- 2023 - Research.com Computer Science in India Leader Award; 2022 - Research.com Computer Science in India Leader Award
- Listed in the Top Computer Science Scientists in the World database maintained by Research.com with World Ranking: 1160 and Top Computer Science Scientists in India (National Ranking): 8 (https://research.com/scientists-rankings/computer-science/in)
- More detailed information at: `https://sites.google.com/view/iitkgpakdas/`

International Research Collaborations

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- **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
- **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
- **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
- **Shantanu Pal**, School of Information Technology, Deakin University, Melbourne, Australia

Welcome to Research in Information Security

Why Research?

- **"If we knew what it was we were doing, it would not be called research, would it?" – Albert Einstein**
- **"If I had an hour to solve a problem I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions." – Albert Einstein**
- **"Successful people are not gifted; they just work hard, then succeed on purpose." – G. K. Nielson**
- **"The three great essentials to achieve anything worthwhile are, first, hard work; second, stick-to-itiveness; third, common sense." – Thomas A. Edison**

What is Research?

- Research means to carefully analyze the problems or to do the detailed study of the specific problems, by making use of special scientific methods.
- Research can be done on any topic, be it medical, non- medical, Information Technology (IT), or anything else.
- In order to do research, first of all, you need to have a topic or the problem on which you can do research. The topic must have relevant questions to answer.
  (**Formulation of Problem Statement**)
- For research, certain steps have to be followed like first observation, then background research then preparing of hypothesis, eventually conducting a simple experiment.
  (**Finding Solutions to the Defined Problems**)

Goals of Research

- Working on a research project will obviously be a challenging and rewarding experience, provided you put the best of your expertise and skill in it.
- It is an opportunity which helps you to pursue an in-depth or deep original study about any topic which interests you.
- The main aim of the goals is to provide the best of the solution to some of the world problems and also to enhance our knowledge.

Research Paper Structure

Sections	What to do?
Abstract	Write 150 words on the purpose of the study, research questions or suggestions, and main findings with conclusions.
Outline	Organize the map of the study.
Introduction	Provide the main information on the problem statement, the indication of methodology, important findings, and principal conclusion.
Literature Review	Analyze and incorporate scholarly sources on past studies.
Methodology or Materials and Methods	Explain the design of the research with techniques that are used for gathering information and other aspects related to the experiment.
Results	Present and illustrate the obtained findings.
Discussion	Review the information in the introduction part, evaluate their gained results, or compare it with past studies.
Recommendations	Propose potential solutions or new ideas based on the obtained results.
Limitations	Consider the weaknesses of the research and results to get new directions.
Conclusion	Provide final thoughts and the summary of the whole work.
Acknowledgments or Appendix	Include additional information on the research paper if it is necessary.
References	Provide and cite all used sources in the study.

How to Measure Impact of Research?

- The **Science Citation Index (SCI)** is a citation index originally produced by the Institute for Scientific Information and created by Eugene Garfield. It was officially launched in 1964. It is now owned by Clarivate Analytics.
- **Impact Factor:** The impact factor (IF) is a measure of the frequency in which the average article in a journal is cited in a particular year. Impact factors measure the impact of a journal, not the impact of individual articles.
- The **h-index** is an index to quantify an individual's scientific research output. The h-index is an index that attempts to measure both the scientific productivity and the apparent scientific impact of a scientist. The index is based on the set of the researcher's most cited papers and the number of citations that they have received in other people's publications. **h-index is the largest number** h **such that** h **publications have at least** h **citations.**
- The **i10-index** is the number of publications with at least 10 citations.

How to Measure Impact of Research?

- **DBLP:** https://dblp.org/pid/39/871.html
- **Google Scholar:** https://scholar.google.com/citations?user=bToAUHMAAAAJ&hl=en
- **Web of Science (Clarivate™):** https://www.webofscience.com/wos/author/record/U-2790-2019
- **Scopus:** https://www.scopus.com/authid/detail.uri?authorId=55450732800
- My more details on research publications: https://sites.google.com/view/iitkgpakdas/research-publications

Course Contents

- Digital Signatures and Its Variants: Proxy Signature, Multi-Signature and Aggregate Signature, and their applications in Internet of Things (IoT)-enabled environments
- Key management in hierarchical access control
- Key management, user authentication and access control in wireless sensor networks/IoT
- Security in Internet of Vehicles (IoV)
- Intrusion detection
- Blockchain and Its Security and Privacy Issues
- AI/ML Security
- Post-Quantum Cryptographic Protocols
- Multiparty computation (MPC): It allows for multiple parties to share data for computing tasks without revealing each other's data
- Quantum Cryptography, QKD, Quantum Teleportation (QT)

Preferred Textbooks and References

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- Books
  - William Stallings, “Cryptography and Network Security: Principles and Practices,” Pearson Education, 6th Edition.
  - Bernard Menezes, “Network Security and Cryptography,” Cengage Learning.
  - Behrouz A. Forouzan, “Cryptography and Network Security,” Special Indian Edition.
- Research papers
  - IEEE Transactions
  - ACM Transactions
  - Elsevier
  - Springer
  - Wiley

Grading Policy

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Grading Method:: Relative

- Mid Sem: 25% (Closed books and notes exam)
- End Sem: 35% (Closed books and notes exam)
- Project: 40%

Thank You!!!

Research in Information Security

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
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/`

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.

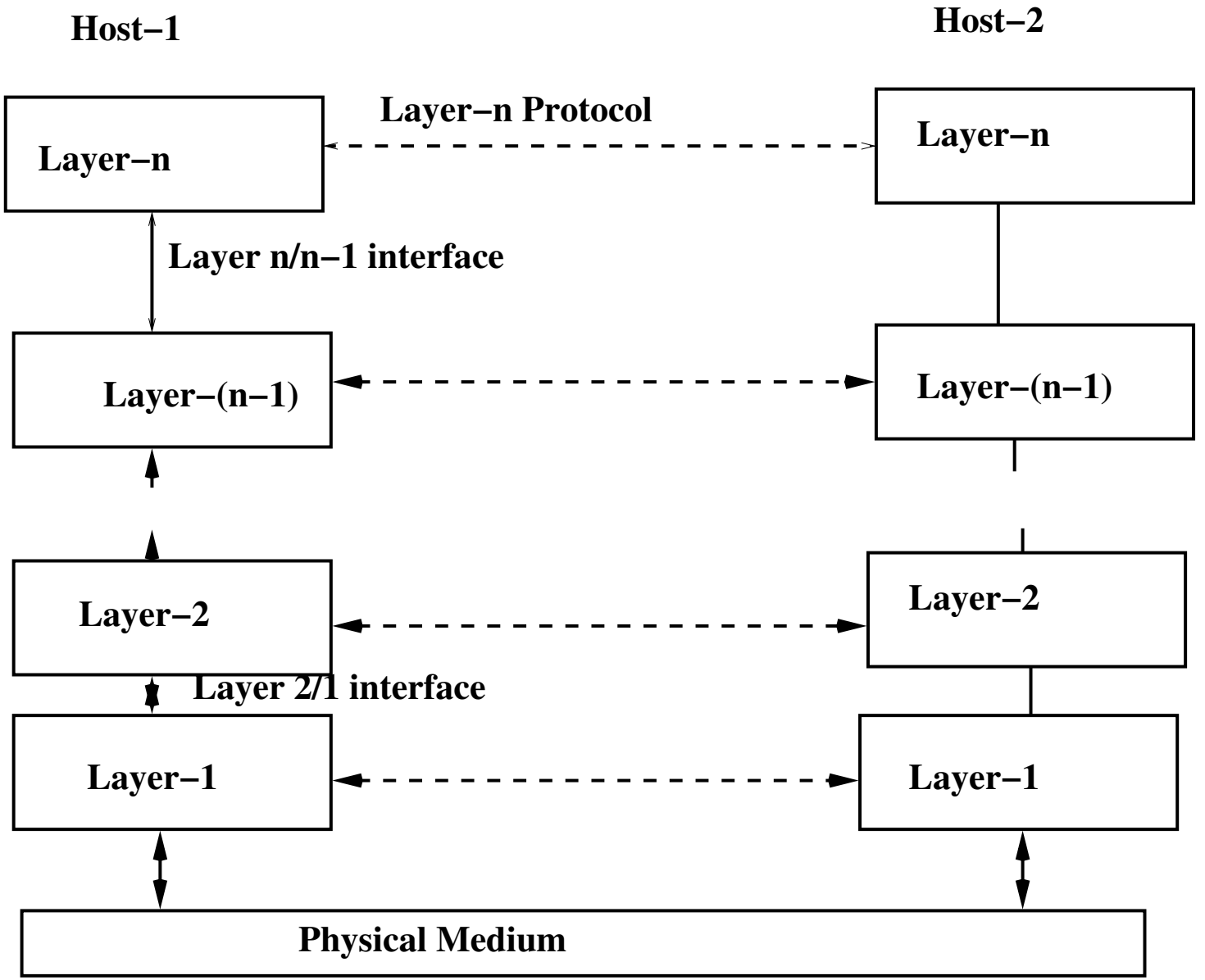
Why layering is needed?

- 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.

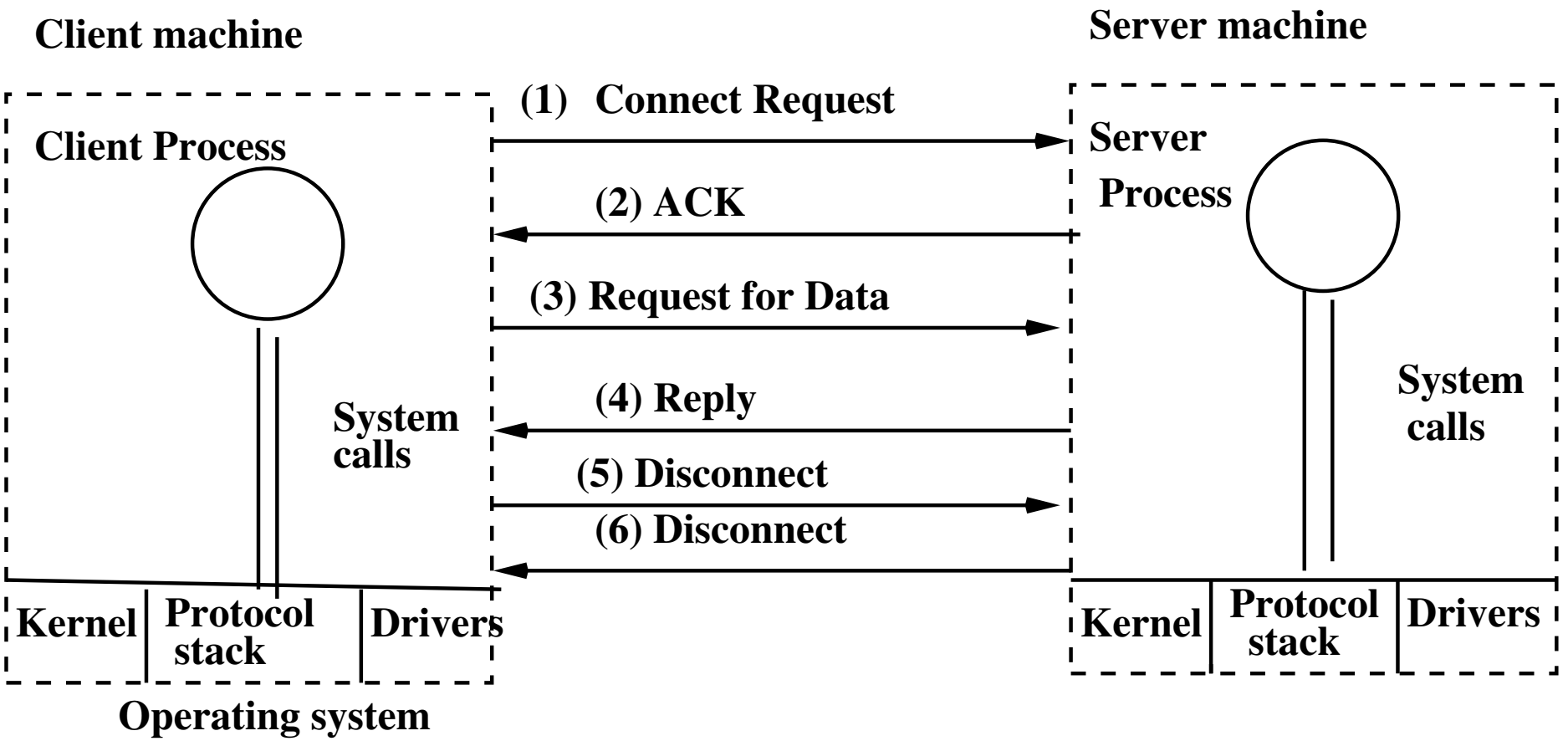
Protocol Hierarchies

- 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



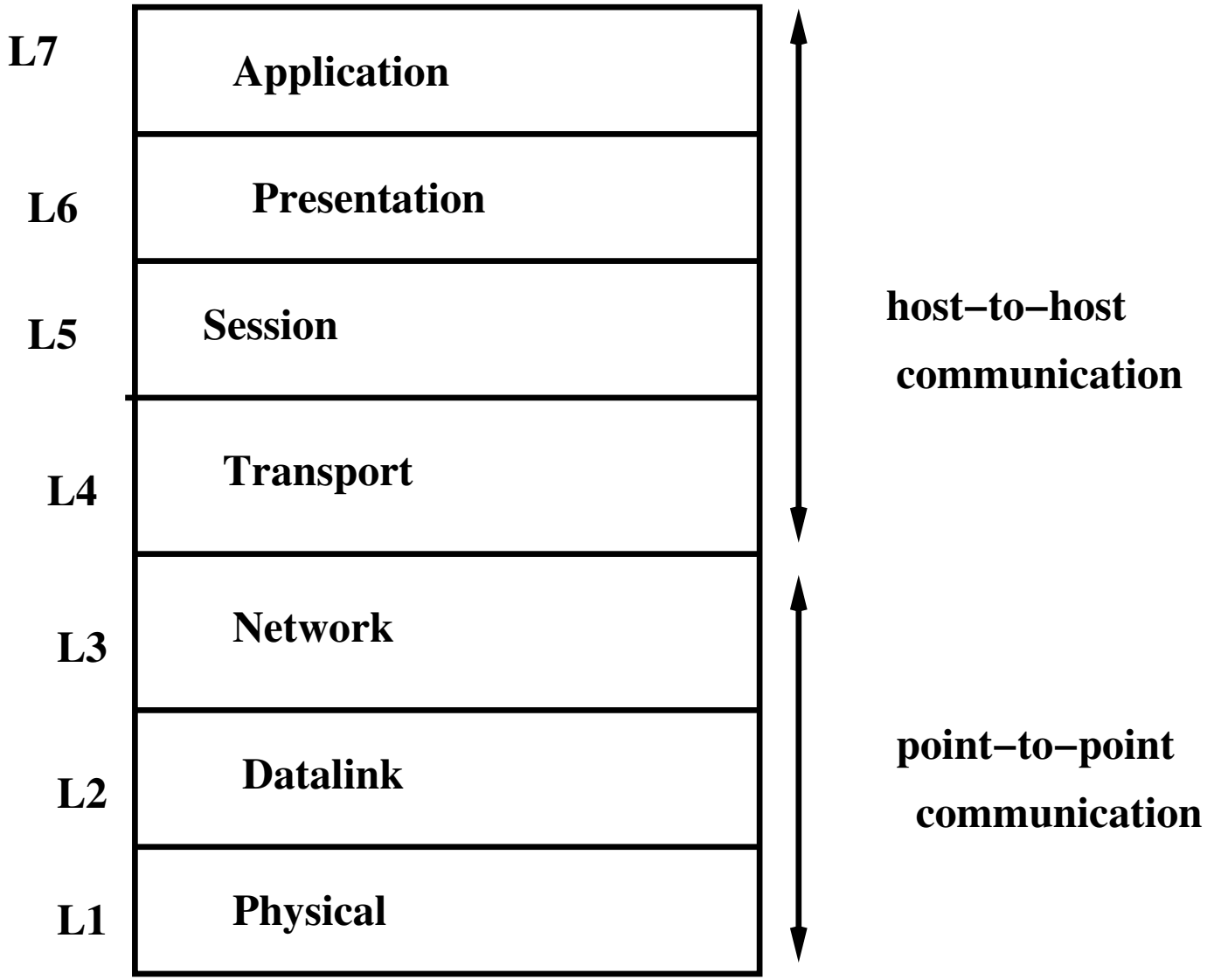
A simple client-server interaction on a connection-oriented network



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

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

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

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.

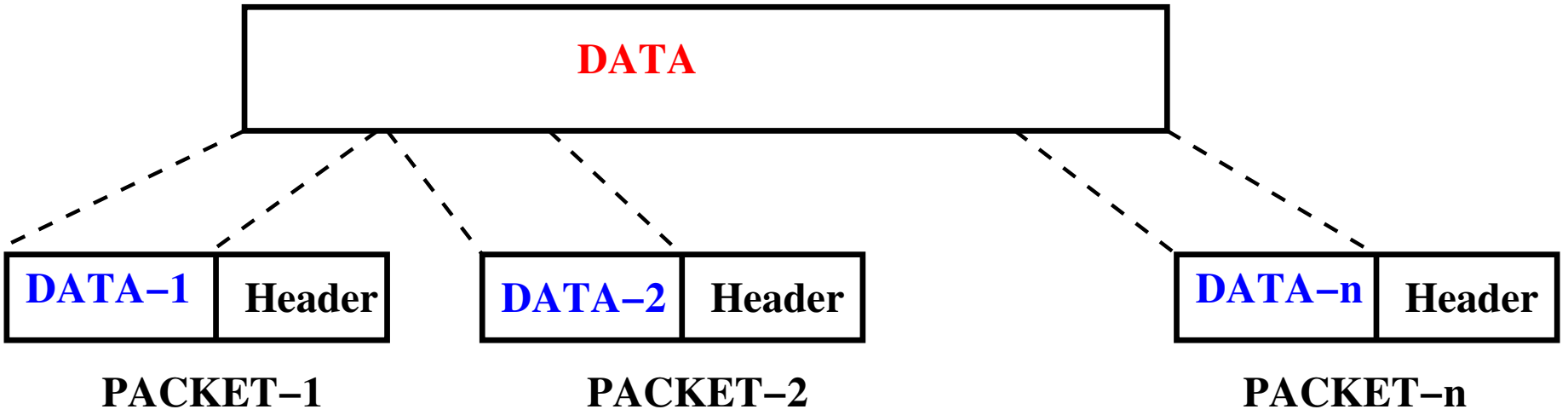
Layer functions

Application Layer:

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

Layer functions

Data handled in a particular layer:

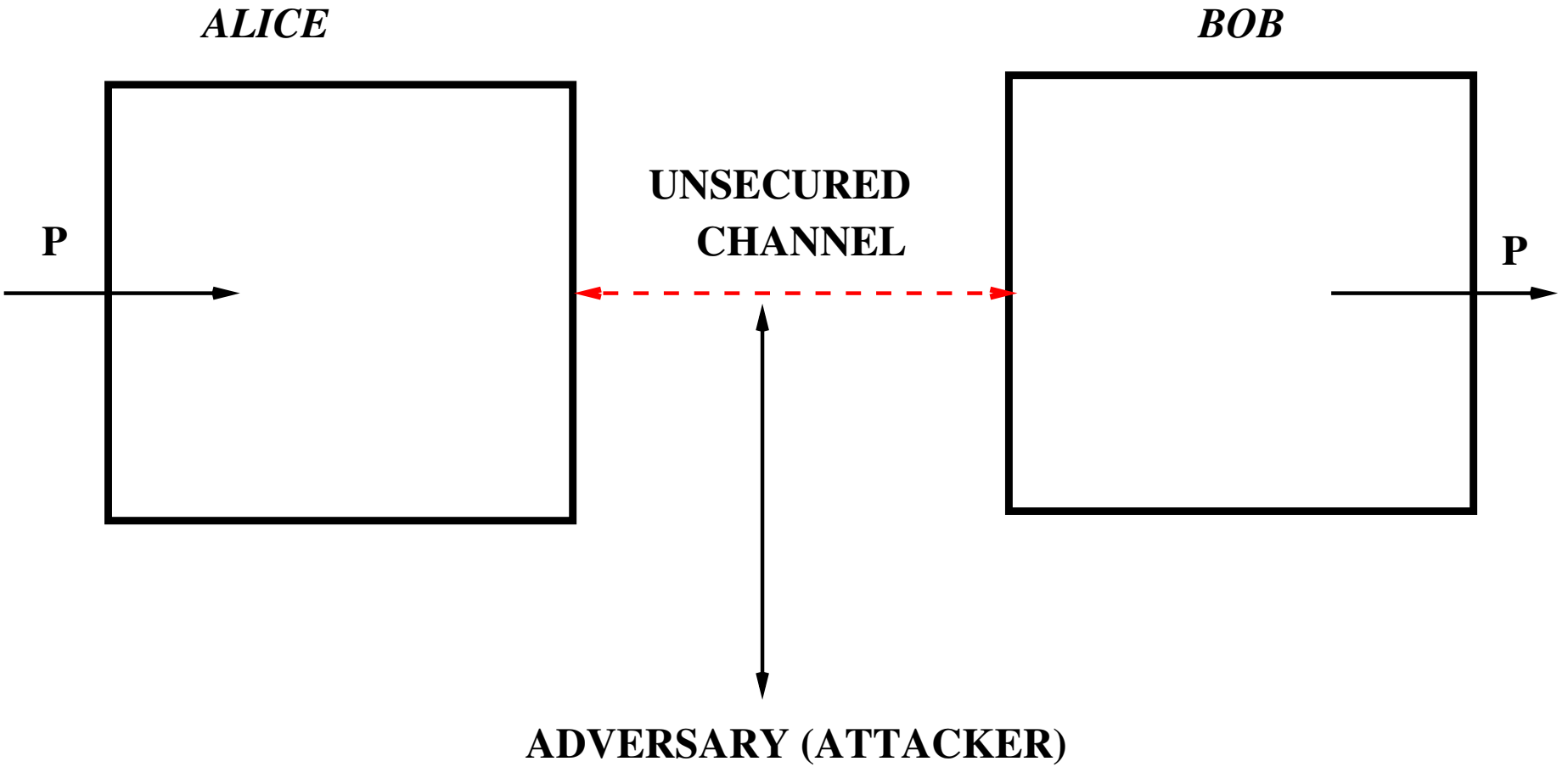


Encrypting Communications Chennels

- This is a classical Alice (user A) and Bob (user B) problem: “Alice wants to send Bob a secure message”.
- Question: What does she do ?
- Answer: She encrypts the message.

Encrypting Communications Chennels

The classical Alice (user A) and Bob (user B) problem:



Encrypting Communications Chennels

- In theory, the encryption can take place at any layer in the OSI communication model.
- In practice, it takes place either at the lowest layers (one and two) or at higher layers.
- If it takes place at the lowest layers, it is called “link-by-link encryption” (LLE).
- In LLE, everything going through a particular data link is encrypted.

Encrypting Communications Chennels

- If the encryption takes place at the higher layers, it is called “end-to-end encryption” (EEE).
- In EEE, the data are encrypted selectively and stay encrypted until they are decrypted by the intended final recipient.
- Each approach has its own benifits and drawbacks.

Thank You!!!

Basics of Symmetric and Public Key Cryptography

Dr. Ashok Kumar Das

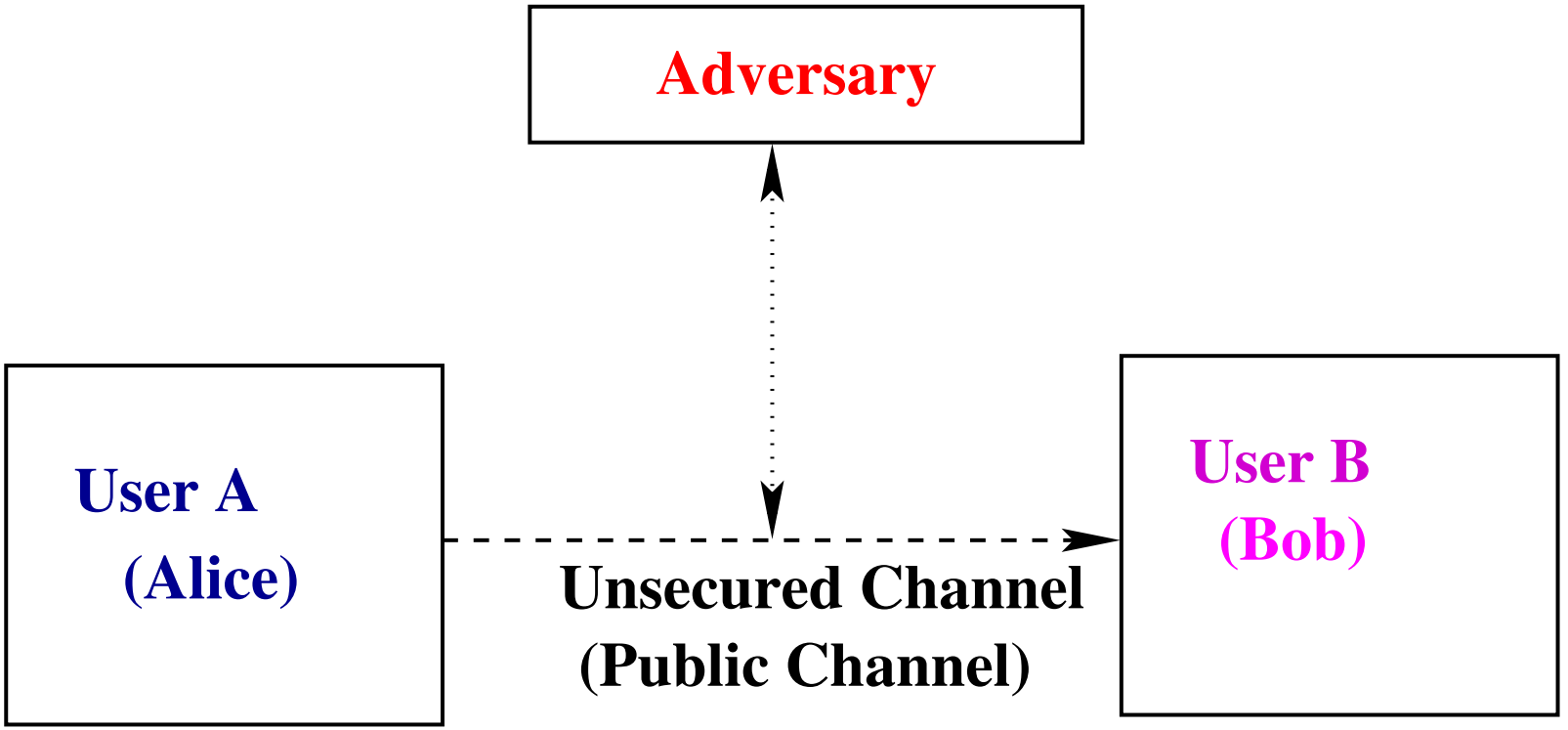
IEEE Senior Member
Professor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

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

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.

Consider the following simple two-party communication model:



Introduction to Cryptography

- 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 (public) 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).

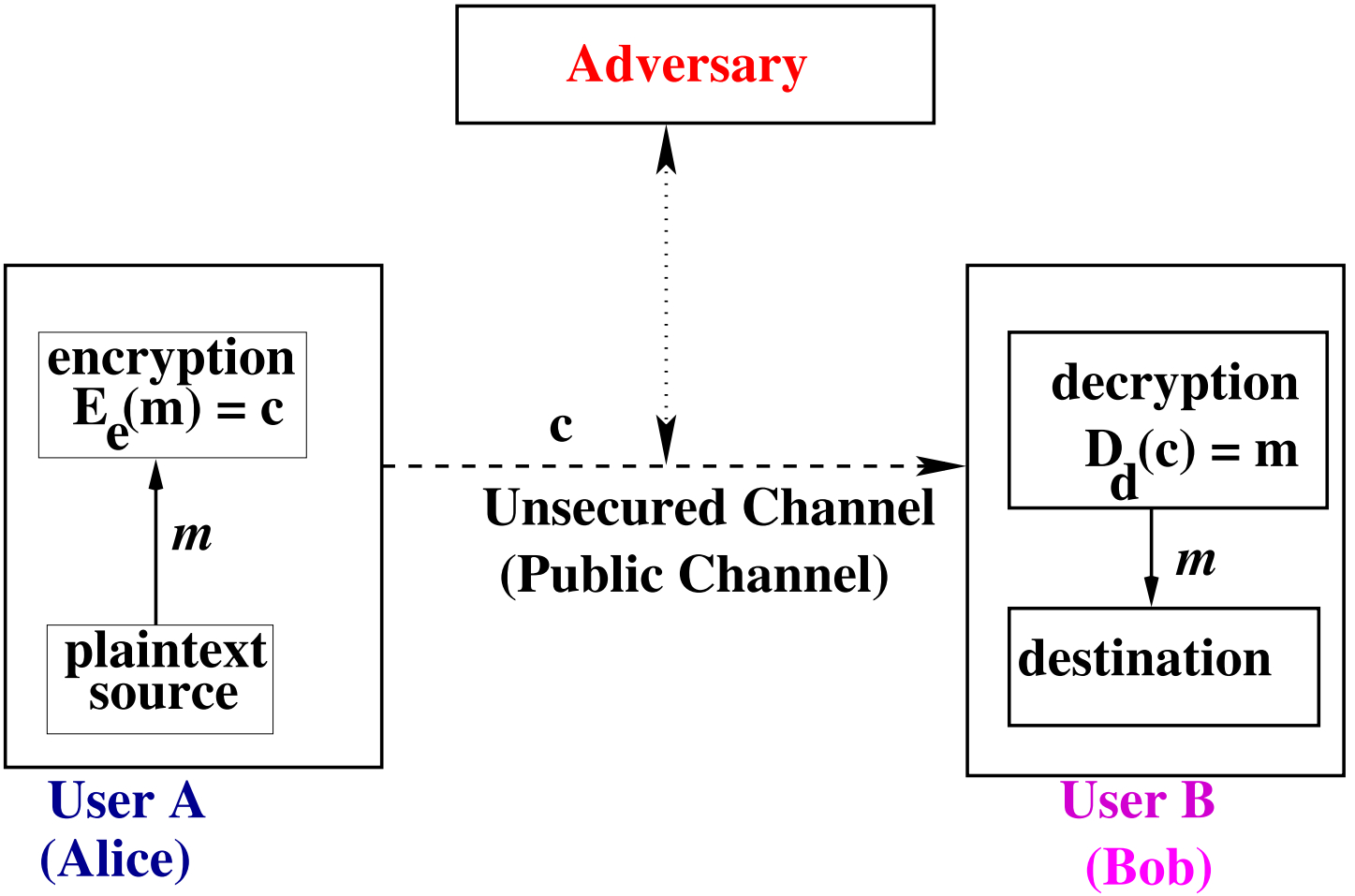
Cryptographic goals (objectives)

- **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.

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.

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



- **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

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), 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 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Symmetric-Key Encryption

Model of conventional (symmetric key) 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

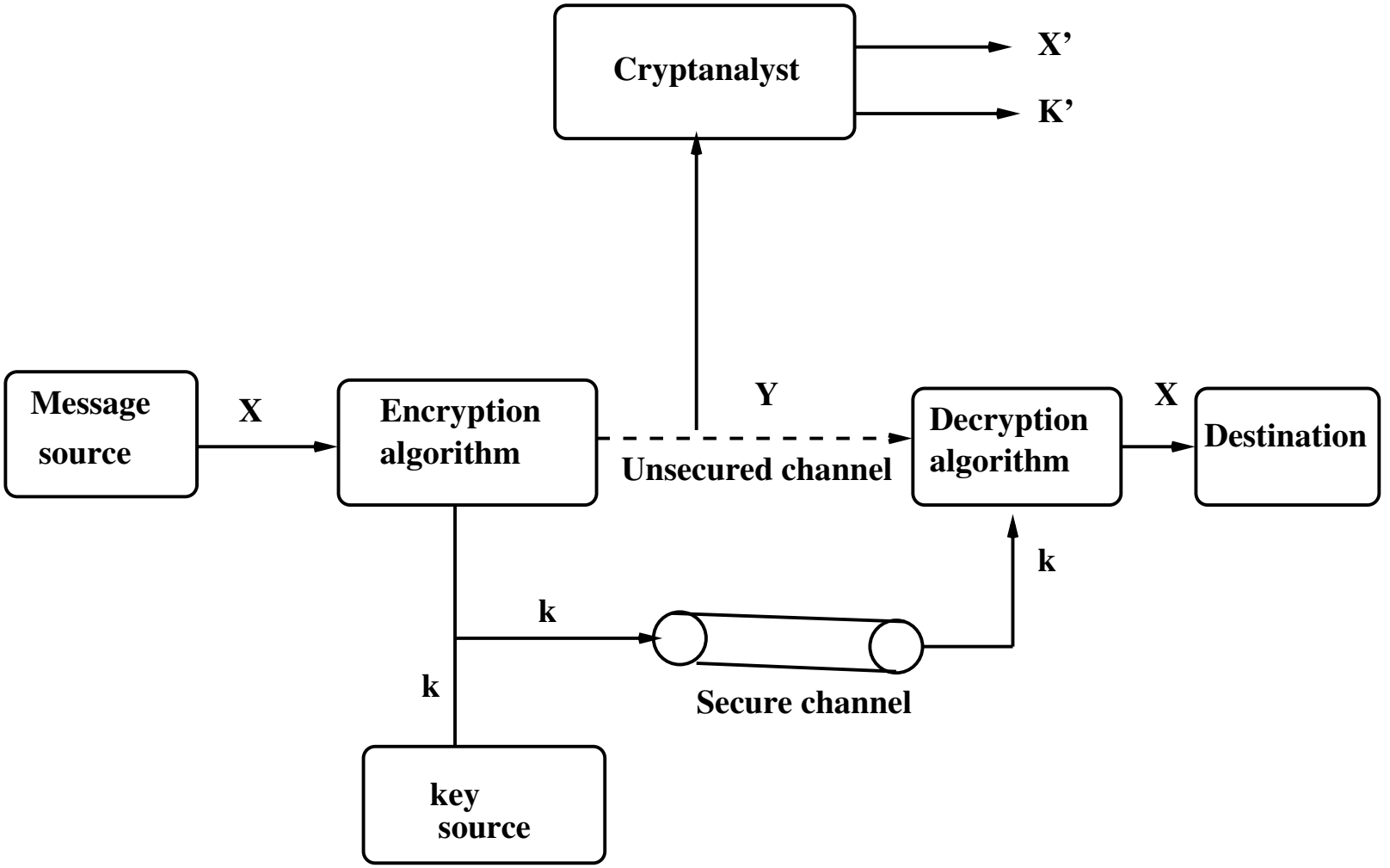


Figure: Model of conventional encryption

Symmetric-Key Encryption

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

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 $k = 3$.

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 $0 \leq k \leq 25$.

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

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: $p_i = c_i \oplus k_i$.

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.

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.

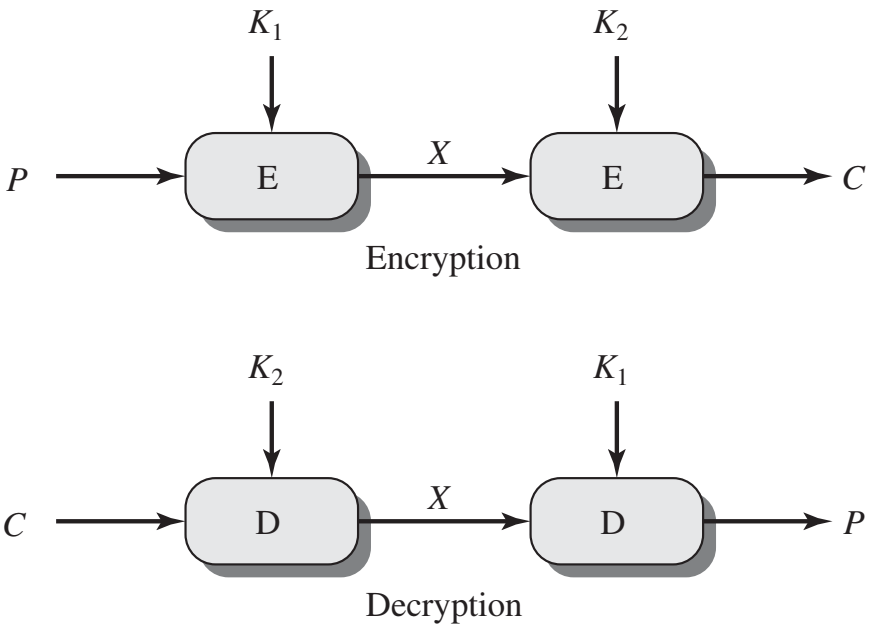
Symmetric-Key Encryption

Data Encryption Standard (DES)

- DES finally and definitely proved insecure in July 1998, when the Electronics Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose “DES cracker” machine that was built for less than 250,000 USD.
- The attack took less than three days.

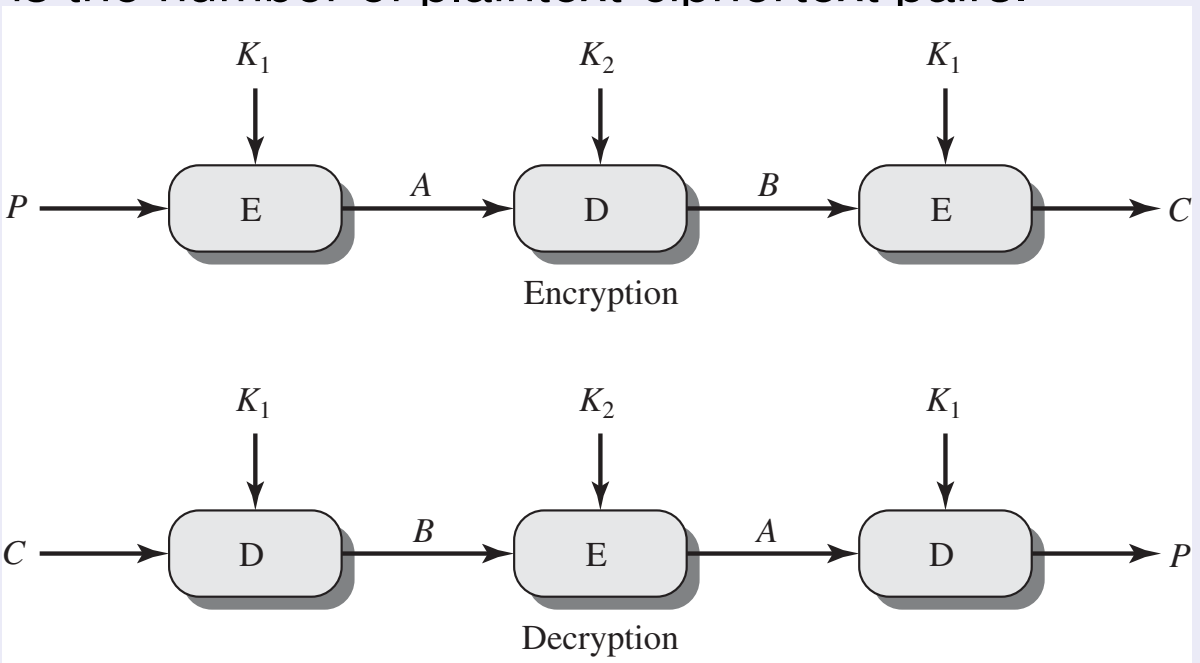
Double DES (2DES)

- It uses two 56-bit keys K_1 and K_2, and 64-bit plaintext block.
- It produces 64-bit ciphertext block.
- Known-plaintext attack (meet-in-the-middle attack) is possible against 2DES to derive two keys K_1 and K_2, which has a key size of 112 bits and with an effort on the order of 2^{56}.

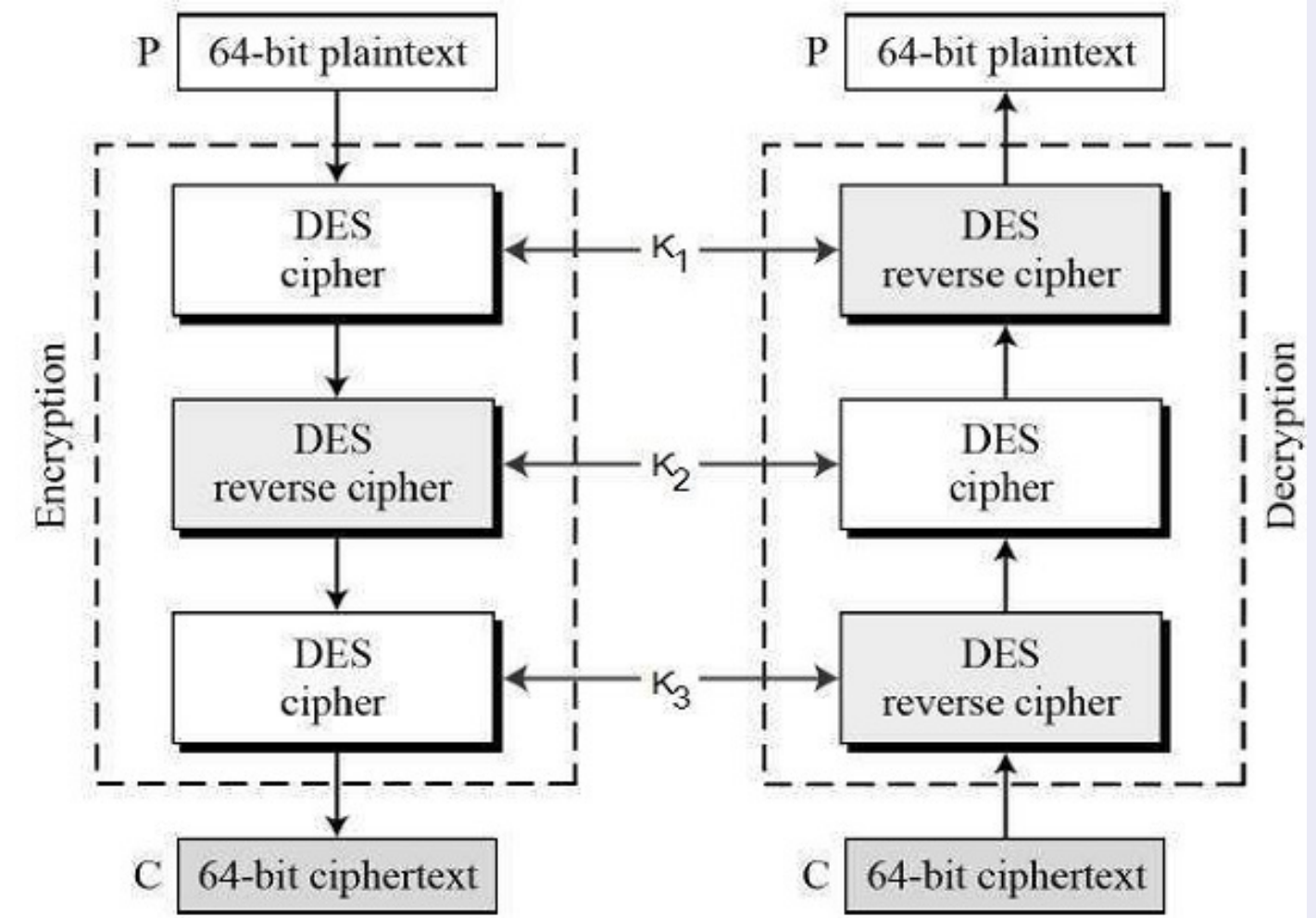


Triple DES with Two Keys (3DES with Two Keys)

- It uses two 56-bit keys K_1 and K_2, and 64-bit plaintext block.
- It produces 64-bit ciphertext block.
- It is also vulnerable to known-plaintext attack (meet-in-the-middle attack) to derive two keys K_1 and K_2.
- The expected running time of this attack is on the order of $2^{120-\log_2 n}$, where n is the number of plaintext-ciphertext pairs.



Triple DES with Three Keys (3DES with Three Keys)



K_1, K_2 and K_3: three 56-bit keys

DES (continued...)

Online Demo on DES Encryption and Decryption

- Generating parameters
- Symmetric key establishment
- Message encoding
- Encryption
- Decryption
- Message decoding

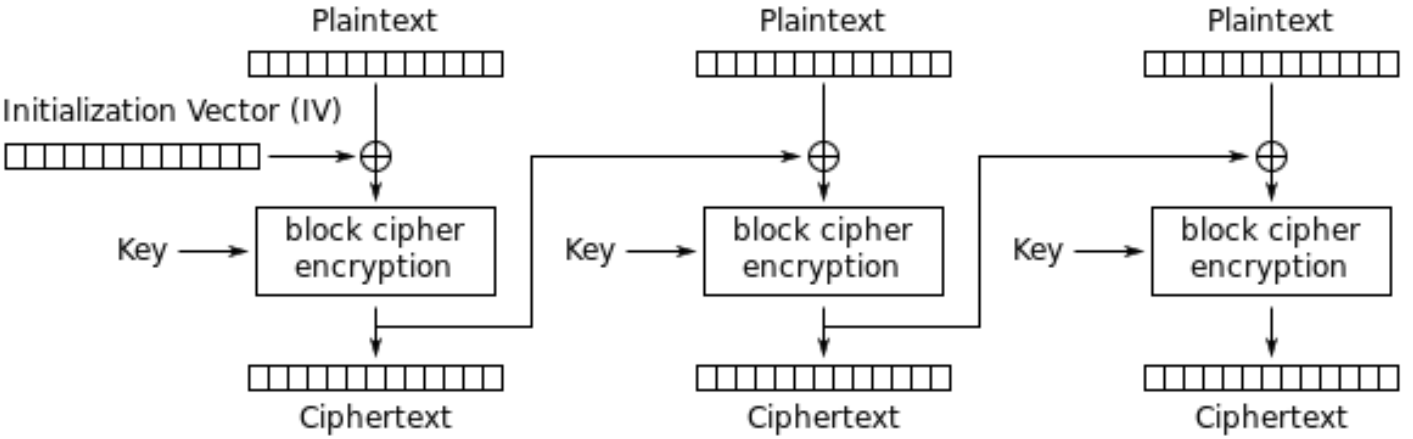
`https://cryptographyacademy.com/des/protocol/`

Symmetric-Key Encryption

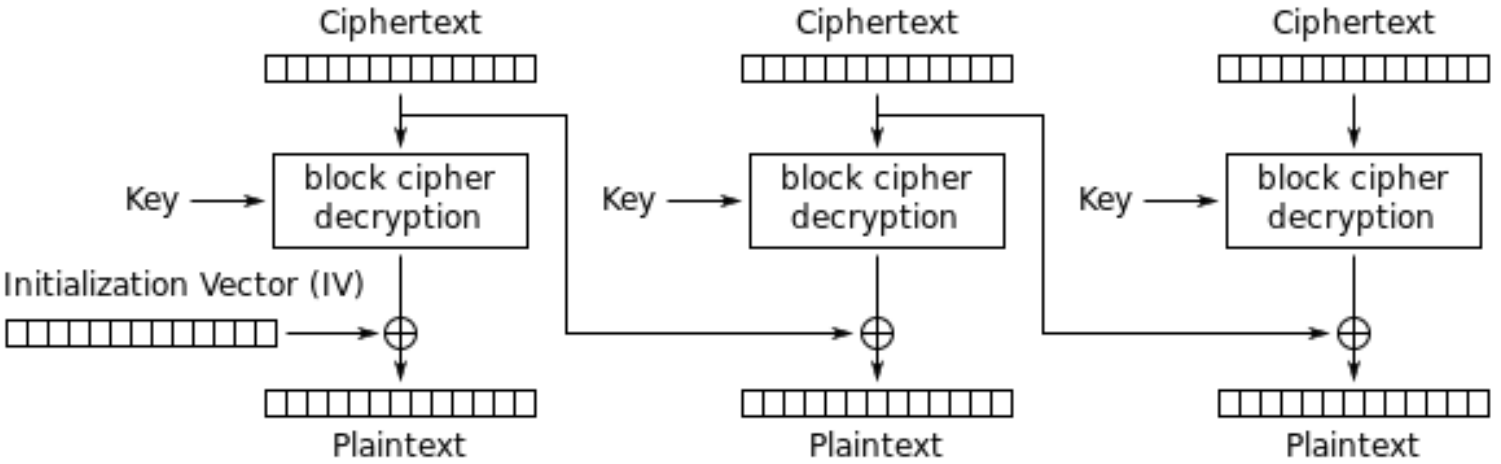
Various modes of operation of Data Encryption Standard (DES)

- Electronic Codebook Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFB)
- Counter Mode (CTR)

Various modes of operation



Cipher Block Chaining (CBC) mode encryption



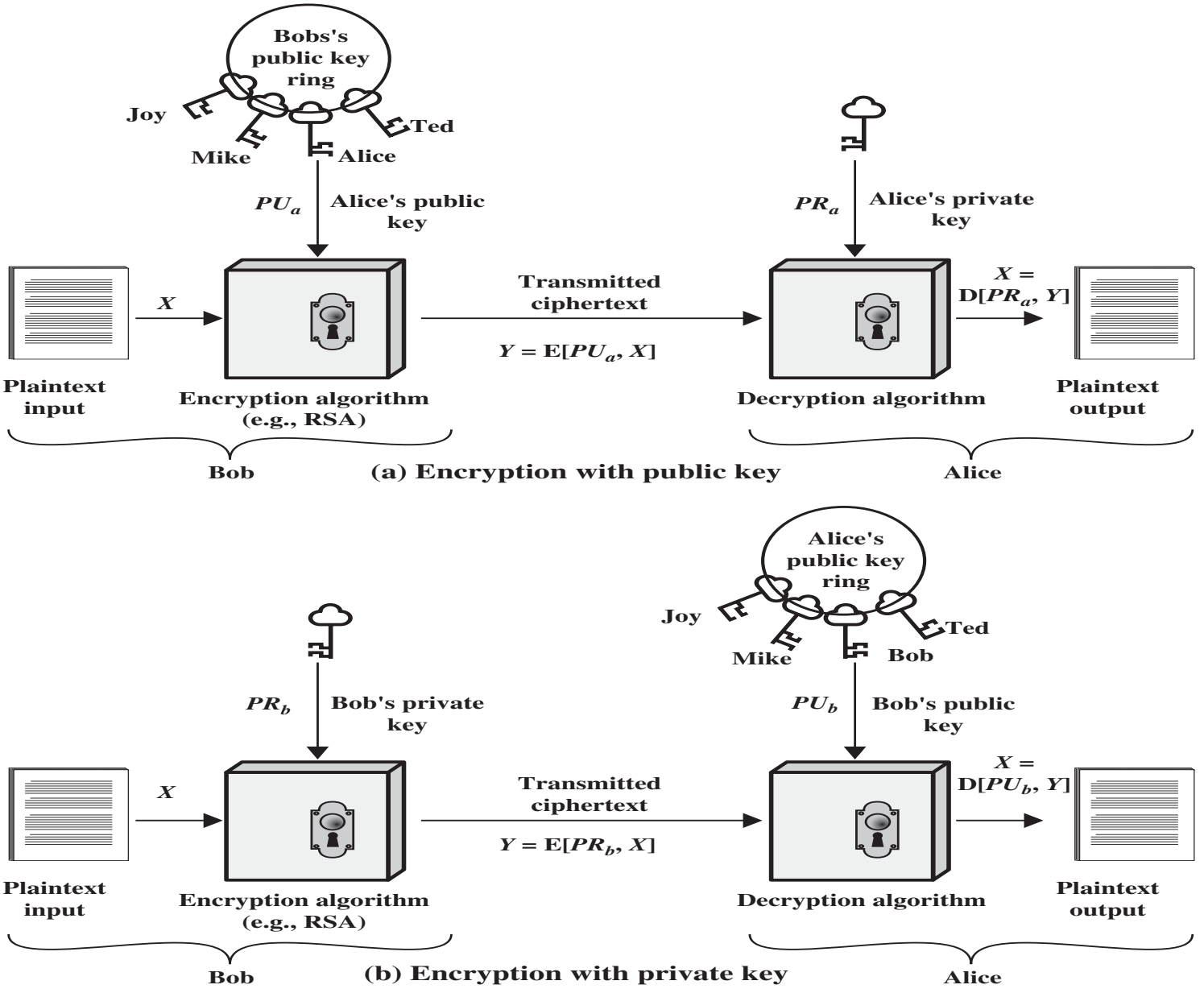
Cipher Block Chaining (CBC) mode decryption

Public-Key Encryption

Model of public key 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 public-key or asymmetric-key, if for each associated encryption/decryption key pair (e, d), called public/private key pair, it is computationally "infeasible" to determine private key d from public key e.

Public-Key Cryptography



The RSA Algorithm

Introduction

- In 1978, Rivest, Shamir and Adleman at MIT, USA discovered a public-key cryptosystem, known as RSA algorithm.
- They received Turing Award (equivalent to Nobel Prize in Computer Science field).
- Their approach is based on elementary number theory concepts.
- The RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and $n - 1$ for some n. A typical size for n is 1024 bits, or 309 decimal digits.

Inventors

Figure: Ronald L. Rivest

Inventors

Figure: Adi Shamir

Inventors

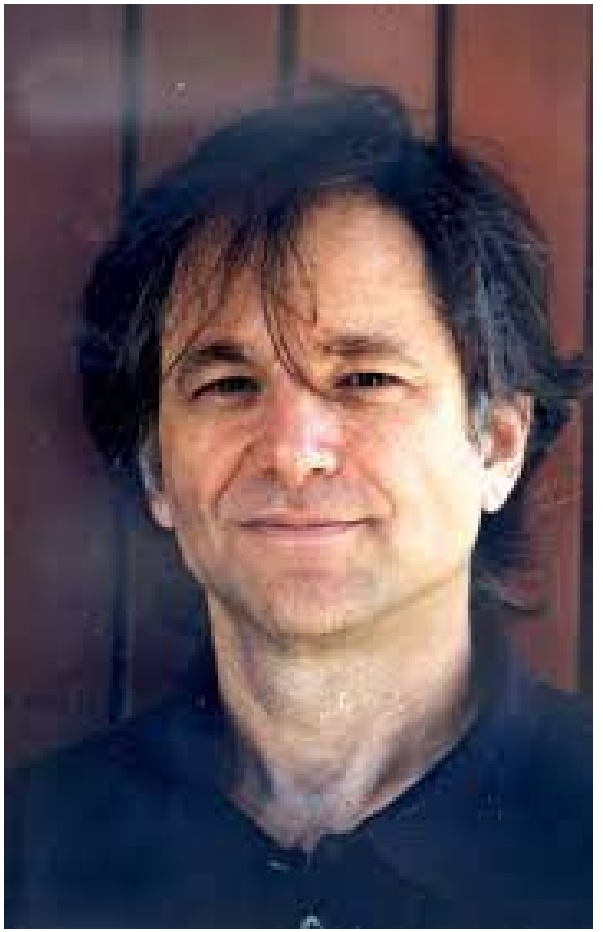


Figure: Leonard M. Adleman

Key Generation

Table: Key generation of the RSA algorithm

Select p, q	p and q both prime, $p \neq q$ (p and q are large)
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$ (Euler phi function)	
Select integer e	$\gcd(e, \phi(n)) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \mod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

The RSA Algorithm

Table: Encryption of the RSA algorithm

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Table: Decryption of the RSA algorithm

Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

The RSA Algorithm

Security of the RSA algorithm

- **Brute force:** This involves trying all possible private keys.
- **Mathematical attacks:** There are several approaches, all equivalent in effect to factoring the the product of two primes (Integer Factorization Problem (IFP)).
  Given a composite integer n of the form $n = p \times q$, to find the prime factors p and q.
  IFP is computationally infeasible (not solvable in polynomial-time factoring algorithm when n is very large, for example, when n is 1024 bits or 2048 bits number.
- **Timing attacks:** These depend on the running time of the decryption algorithm.

The RSA Algorithm

Problem:

The ciphertext message produced by the RSA algorithm with the public key $(e, n) = (223, 1643)$ is:
1451 0103 1263 0560 0127 0897.
Determine the original plaintext message.
Use the standard encoding procedure:
A = 01, B = 02, . . ., Z = 26,
, = 27, . = 28, ? = 29,
0 = 30, 1 = 31, . . ., 9 = 39, ! = 40,
with 00 as the blank space.

The RSA Algorithm

Solution:

- Here $e = 223$, $n = 1643 = 31 \times 53 = p \times q$, say, where p and q are distinct primes.
- $\phi(n) = \phi(1643) = (p-1) \times (q-1) = 30 \times 52 = 1560$.
- Using the Extended Euclid's GCD algorithm, $ed \equiv 1 \pmod{\phi(n)}$, that is, $d = 7$.
- The private key is then $(d, n) = (7, 1643)$.
- The given ciphertext blocks are as follows:
  $C_1 = 1451$,
  $C_2 = 0103$,
  $C_3 = 1263$,
  $C_4 = 0560$,
  $C_5 = 0127$
  $C_6 = 0897$.

The RSA Algorithm

Solution (Continued...):

- The deciphertext (recovered plaintext) of each block C_i is given below (using the repeated square-and-multiply method).
- $M_1 = C_1^d \pmod{n} = 1451^7 \pmod{1643} = 180$
- $M_2 = C_2^d \pmod{n} = 103^7 \pmod{1643} = 516$
- $M_3 = C_3^d \pmod{n} = 1263^7 \pmod{1643} = 122$
- $M_4 = C_4^d \pmod{n} = 560^7 \pmod{1643} = 500$
- $M_5 = C_5^d \pmod{n} = 127^7 \pmod{1643} = 141$
- $M_6 = C_6^d \pmod{n} = 897^7 \pmod{1643} = 523$
- Hence, the original plaintext message using the decoding method given here is as follows:
  $M = M_1 M_2 M_3 M_4 M_5 M_6 = 18\ 05\ 16\ 12\ 25\ 00\ 14\ 15\ 23$
  $= \text{REPLY NOW}$

The RSA Algorithm

Online Demo on RSA Algorithm

- Generating private/public keys pair
- Encrypting a message
- Decrypting a message

`https://8gwifi.org/rsafunctions.jsp`

`https://www.mobilefish.com/services/rsa_key_generation/rsa_key_generation.php`

Diffie-Hellman Key Exchange Protocol

Overview

- Diffie-Hellman key agreement (also called exponential key exchange or Diffie-Hellman key exchange) provided the first practical solution to the secret key distribution problem.
- It is based on public-key cryptography.
- This protocol enables two parties, say A and B, which have never communicated before, to establish a mutual secret key by exchanging messages over a public channel.

Inventors

Figure: Prof. Whitfield Diffie

Inventors

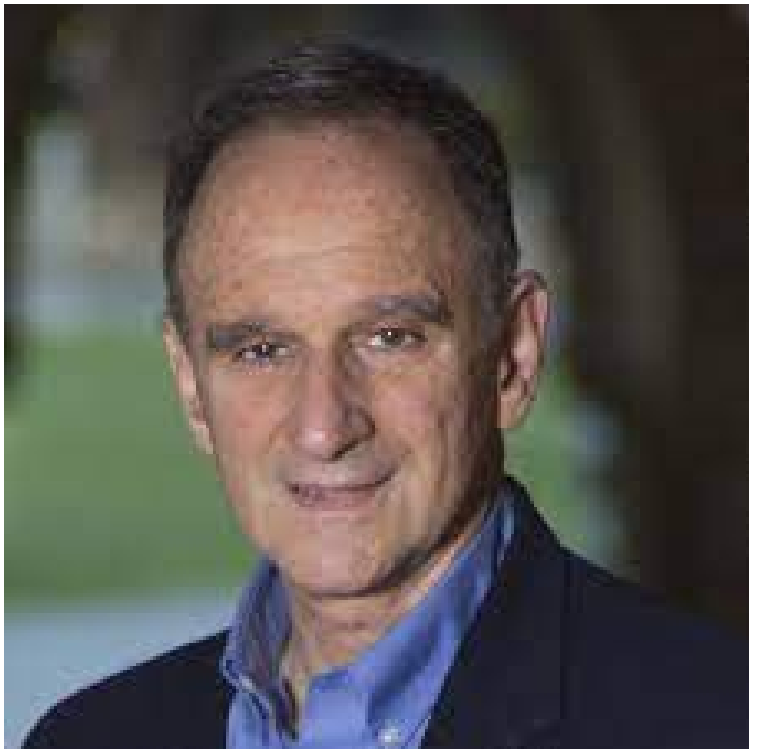


Figure: Prof. Martin Hellman

Me with Prof. Martin Hellman (15 February 2018 at IIIT Hyderabad)

Diffie-Hellman Key Exchange Protocol (continued...)

Global Public Elements

- q : a sufficiently large prime, such that it is intractible to compute the discrete logarithms in $Z_q^* = \{1, 2, \cdots, q-1\}$
(Given α, q and $y = \alpha^x \pmod q$, to find discrete logarithm $x \in Z_q^*$).
- α : $\alpha < q$ and α a primitive root of q.
(Compute $\alpha^1 \pmod q$, $\alpha^2 \pmod q$, $\cdots$, $\alpha^{q-1} \pmod q$.
If all are distinct and $\alpha^{q-1} \pmod q = 1$, α is primitive root of q)

User A Key Generation

- Select private X_A such that $X_A < q$
- Calculate public Y_A such that $Y_A = \alpha^{X_A} \bmod q$

$A \rightarrow B : \{Y_A, q, \alpha\}$
Here $A \rightarrow B : M$ denotes party A sends a message M to party B.

Diffie-Hellman Key Exchange Protocol (continued...)

User B Key Generation

- Select private X_B such that $X_B < q$
- Calculate public Y_B such that $Y_B = \alpha^{X_B} \mod q$

$B \rightarrow A : \{Y_B\}$

Generation of secret key by User A

- $K_{A,B} = (Y_B)^{X_A} \mod q$

Generation of secret key by User B

- $K_{B,A} = (Y_A)^{X_B} \mod q$

Diffie-Hellman Key Exchange Protocol (continued...)

Summary

User A	User B
1. Select private X_A	
2. Calculate public Y_A	
3. $Y_A = \alpha^{X_A} \mod q \xrightarrow{\quad\quad}$	
	1. Select private X_B
	2. Calculate public Y_B
	3. $Y_B = \alpha^{X_B} \mod q \xleftarrow{\quad\quad}$
4. $K_{A,B} = (Y_B)^{X_A} \mod q$	4. $K_{B,A} = (Y_A)^{X_B} \mod q$

Diffie-Hellman Key Exchange Protocol (continued...)

Correctness Proof

$$
\begin{aligned}
K_{A,B} &= (Y_B)^{X_A} \bmod q \text{ [User A]} \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha)^{X_B \cdot X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q \\
&= K_{B,A} \text{ [User B]}
\end{aligned}
$$

Diffie-Hellman Key Exchange Protocol (continued...)

Problem [Diffie-Hellman Key Exchange]

Users A and B use the Diffie-Hellman key exchange technique with a common prime $q = 71$ and a primitive root $\alpha = 7$.
(a) If user A has private key $X_A = 5$, what is the A's public key Y_A?
(b) If user B has private key $X_B = 12$, what is the B's public key Y_B?
(c) What is the secret shared key?
Solution: Here $q = 71$ and $\alpha = 7$.
(a) A's public key Y_A is given by

$$\begin{aligned} Y_A &= \alpha^{X_A} \bmod q \\ &= 7^5 \bmod 71 \\ &= (7^1 \bmod 71) \times (7^4 \bmod 71) \bmod 71 \\ &= 51 \end{aligned}$$

Diffie-Hellman Key Exchange Protocol (continued...)

Problem [Diffie-Hellman Key Exchange] (Continued...)

(b) B's public key Y_B is given by

$$\begin{aligned} Y_B &= \alpha^{X_B} \bmod q \\ &= 7^{12} \bmod 71 \\ &= (7^4 \bmod 71) \times (7^8 \bmod 71) \bmod 71 \\ &= 4 \end{aligned}$$

(c) The secret shared key K is given by

$$\begin{aligned} K_{A,B} &= (Y_B)^{X_A} \bmod q \text{ [User A]} \\ &= 4^5 \bmod 71 \\ &= 30 \end{aligned}$$

Diffie-Hellman Key Exchange Protocol (continued...)

Problem [Diffie-Hellman Key Exchange] (Continued...)

$$\begin{aligned} K_{B,A} &= (Y_A)^{X_B} \bmod q \text{ [User B]} \\ &= 51^{12} \bmod 71 \\ &= 30 \end{aligned}$$

$K = K_{A,B} = K_{B,A} = 30$ is the required secret shared key between A and B.

■

Diffie-Hellman Key Exchange Protocol (continued...)

Online Demo on Diffie-Hellman Key Exchange Protocol

- Generating primitive root of prime
- Computing the shared session key between two parties

`http://www.irongeek.com/diffie-hellman.php?`

Further Readings (Cryptography and Network Security)

- William Stallings, “Cryptography and Network Security: Principles and Practices”, Pearson Education, 2010.
- Behrouz A. Forouzan, “Cryptography and Network Security”, Special Indian Edition.
- Bernard Menezes, “Network Security and Cryptography”, Cengage Learning, 2010.
- A. Menezes, P. Oorschot and S. Vanstone, “Handbook of Applied Cryptography”, CRC Press.
- B. Schneier, “Applied Cryptography”, Reading, MA: Addison-Wesley, 2006.
- D. Stinson, “Cryptography: Theory and Practice”, Chapman & Hall/CRC, 2006.
- Neal Koblitz, “A course in number theory and cryptography”, Springer.

Thank you

Research in Information Security

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
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/`

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.

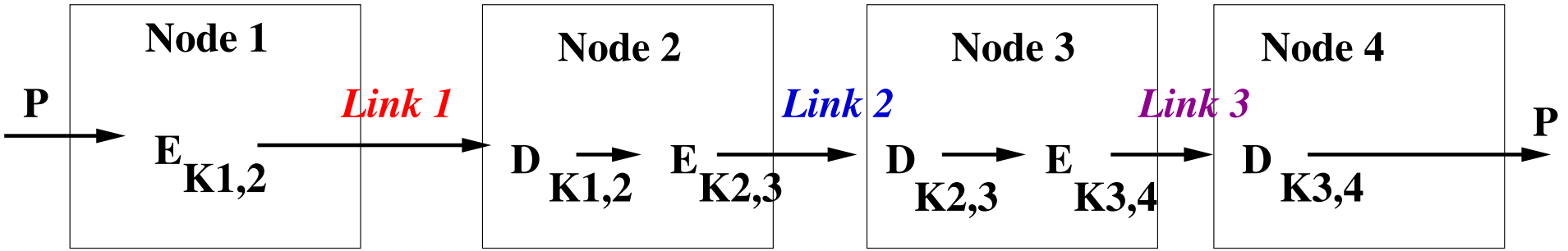
Encrypting Communications Channels

- 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.

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

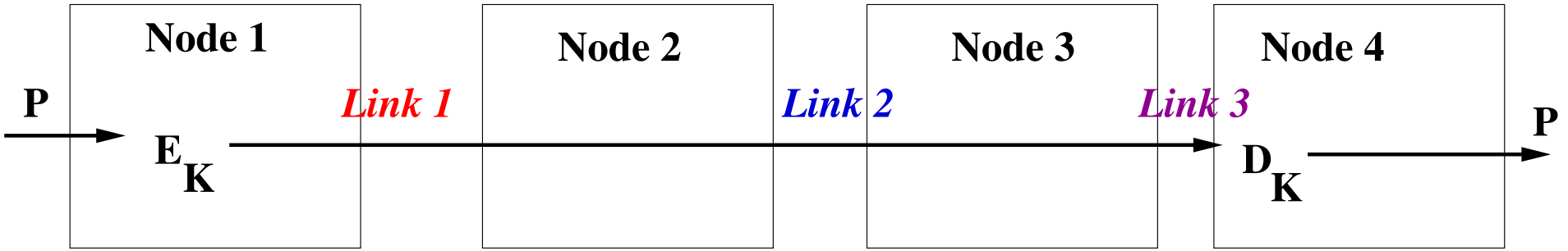
Disadvantages

- 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

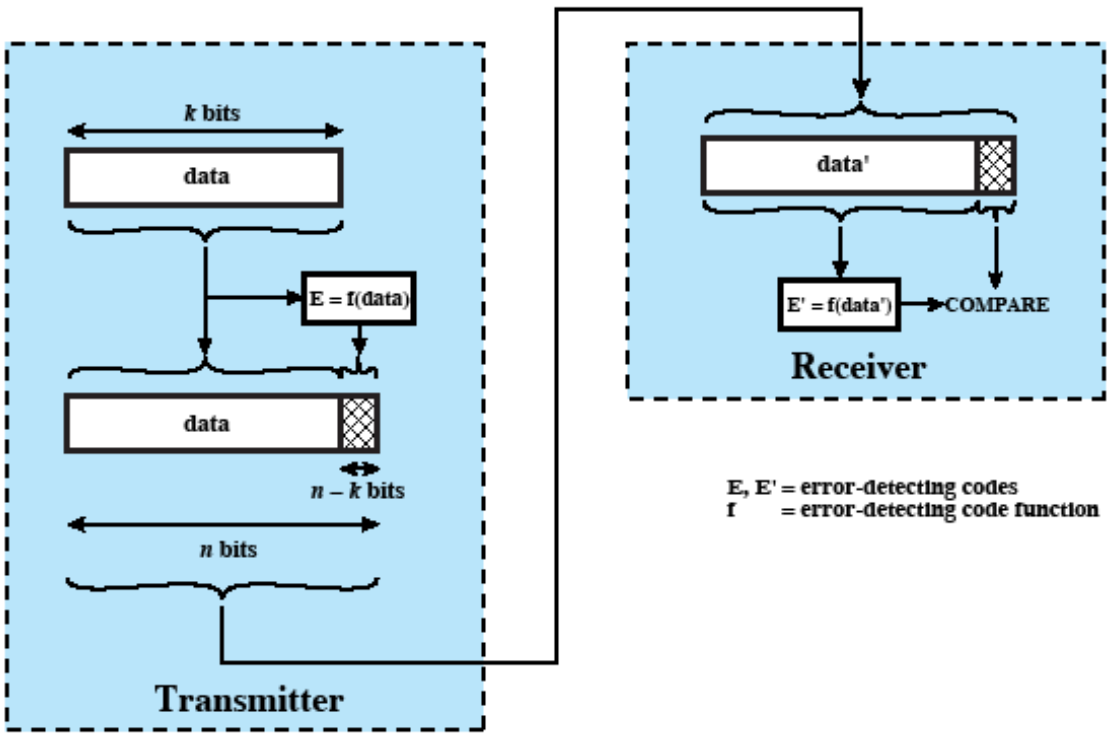
Link-by-link encryption	End-to-end encryption
Role of user	
1. Applied by sending host.	1. Applied by sending process.
2. Invisible to user.	2. User applies encryption.
3. Host maintains encryption.	3. User must find algorithm.
4. One facility for all users.	4. User selects encryption.
5. Can be done 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.

Security at the Datalink Layer

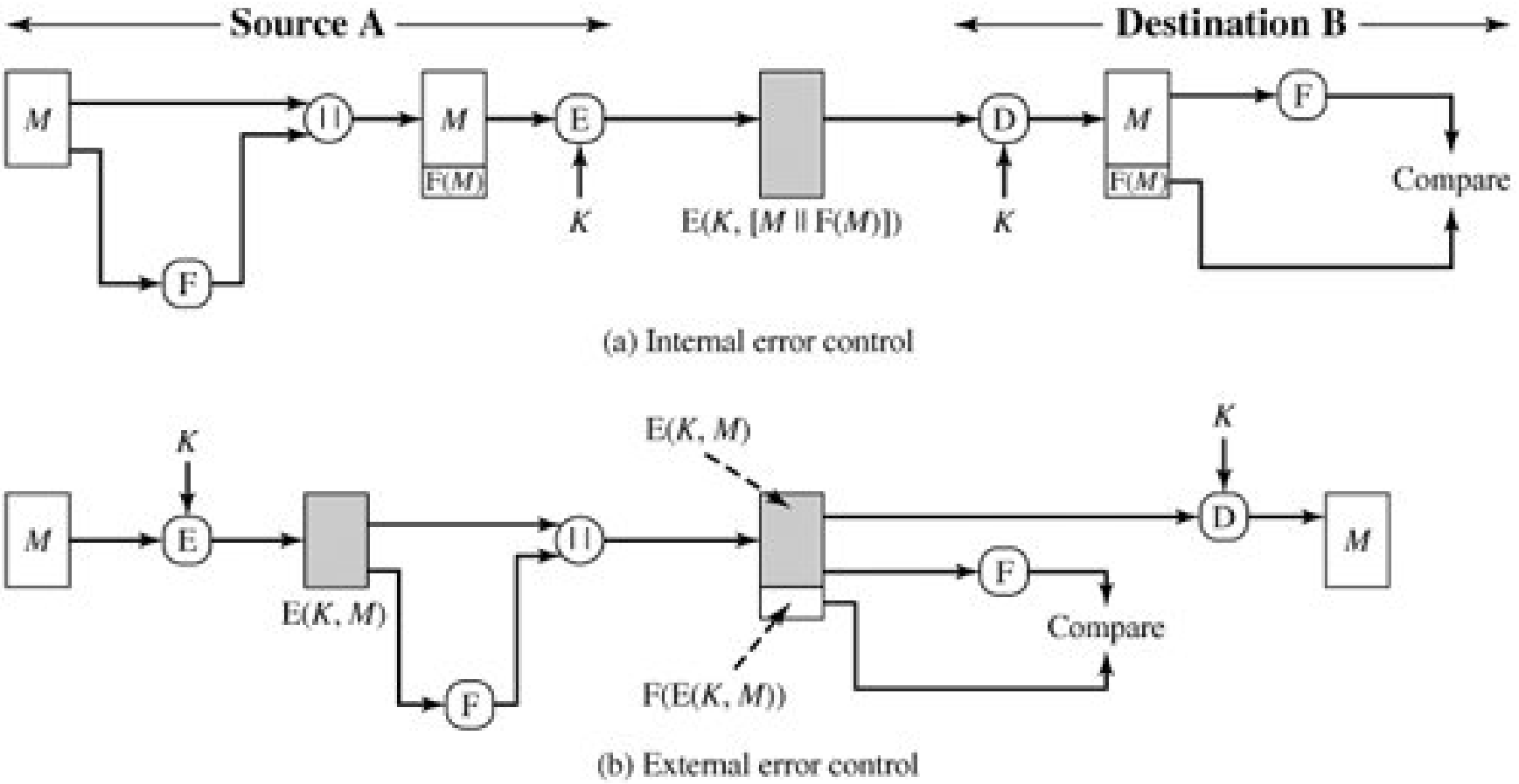
Error Detection Process



Internal and External Error Control

- It may be difficult to determine *automatically* if incoming ciphertext decrypts to intelligible plaintext.
- For example, if the plaintext is a binary object file or digitized X-rays, determination of properly formed and therefore authentic plaintext may be difficult.
- Thus, an opponent could achieve a certain level of disruption simply by issuing messages with random content purporting to come from a legitimate user.
- One solution to this problem is to force the plaintext to have some structure that is easily recognized but that cannot be replicated without recourse to the encryption function.
- We could, for example, append an error-detecting code, also known as a frame check sequence (FCS) or checksum, to each message before encryption.

Internal and External Error Control



M: plaintext message; F: function that produces an FCS (frame check sequence); $||$: concatenation operation; $E(K, M)$: encryption of M using key K; $D(K, M)$: decryption of M using key K; K: shared key between source A and destination B.

Internal and External Error Control

- Note that the order in which the FCS and encryption functions are performed is critical.
- With internal error control, authentication is provided because an opponent would have difficulty generating ciphertext that, when decrypted, would have valid error control bits.
- If instead the FCS is the outer code, an opponent can construct messages with valid error-control codes.
- Although the opponent cannot know what the decrypted plaintext will be, he or she can still hope to create confusion and disrupt operations.

Research in Information Security

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
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/`

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.

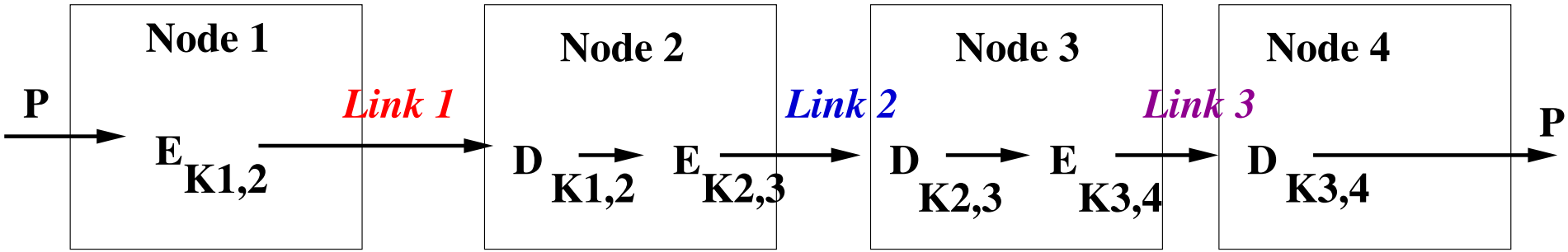
Encrypting Communications Channels

- 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.

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

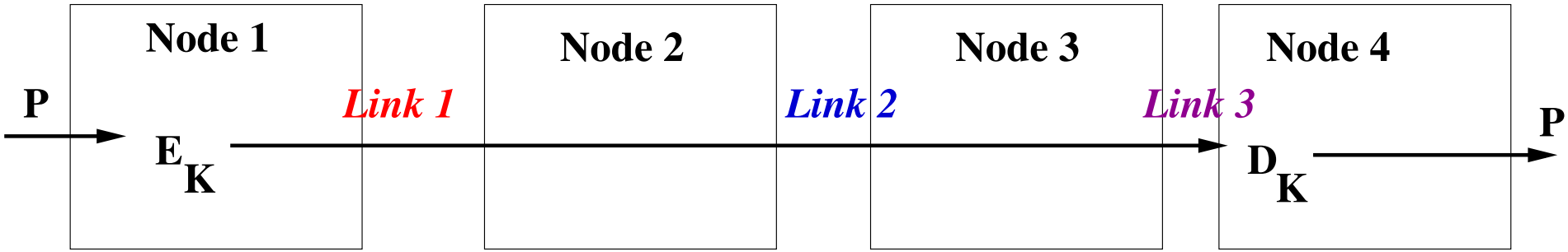
Disadvantages

- 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

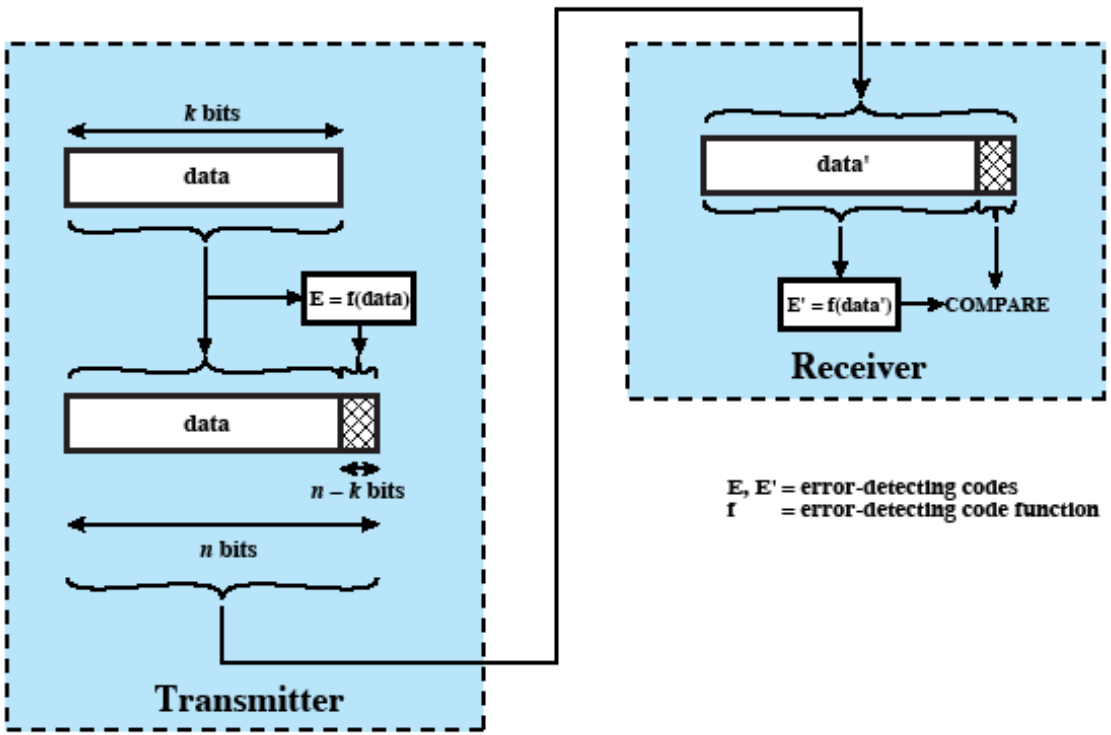
Link-by-link encryption	End-to-end encryption
Role of user	
1. Applied by sending host.	1. Applied by sending process.
2. Invisible to user.	2. User applies encryption.
3. Host maintains encryption.	3. User must find algorithm.
4. One facility for all users.	4. User selects encryption.
5. Can be done 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.

Security at the Datalink Layer

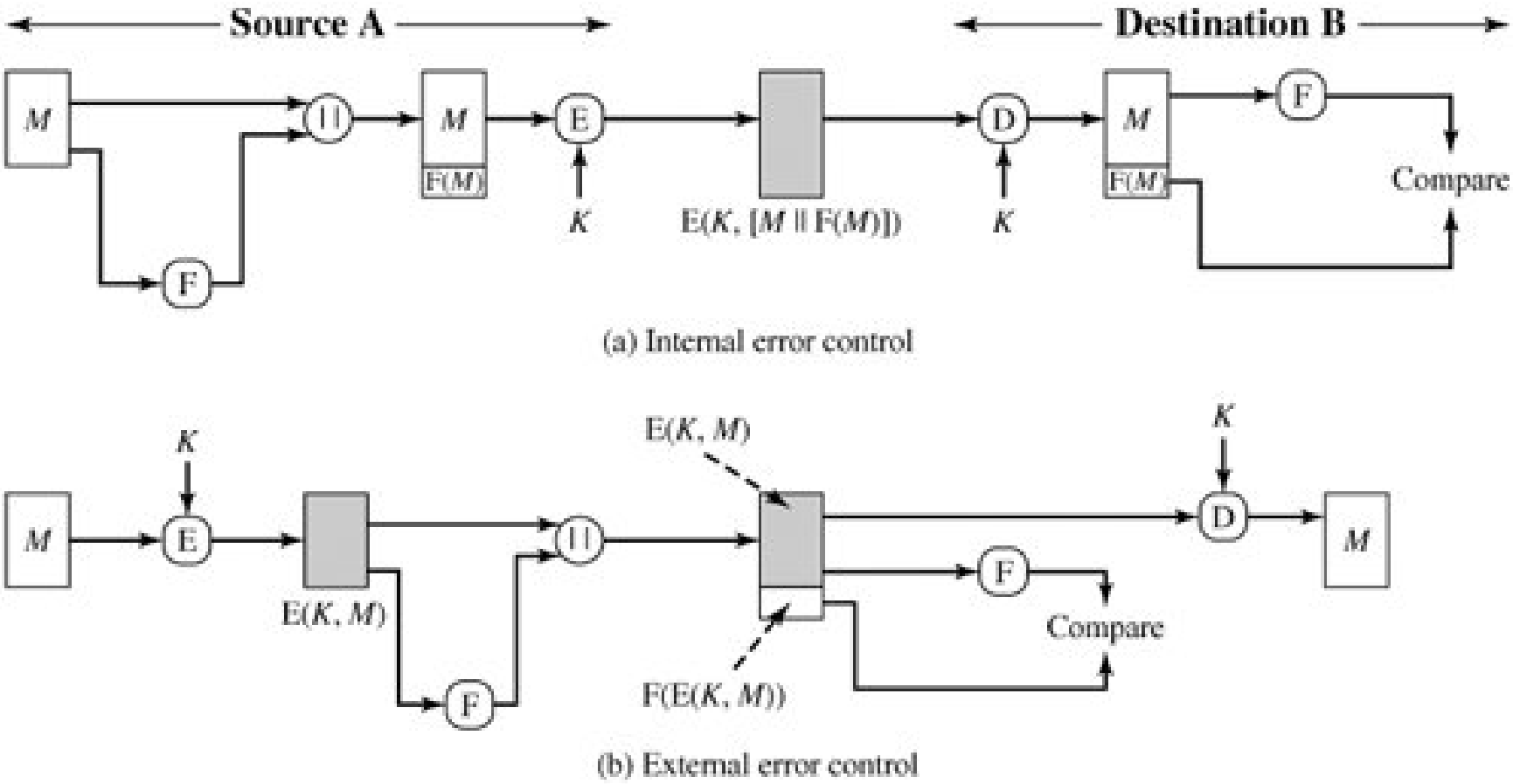
Error Detection Process



Internal and External Error Control

- It may be difficult to determine *automatically* if incoming ciphertext decrypts to intelligible plaintext.
- For example, if the plaintext is a binary object file or digitized X-rays, determination of properly formed and therefore authentic plaintext may be difficult.
- Thus, an opponent could achieve a certain level of disruption simply by issuing messages with random content purporting to come from a legitimate user.
- One solution to this problem is to force the plaintext to have some structure that is easily recognized but that cannot be replicated without recourse to the encryption function.
- We could, for example, append an error-detecting code, also known as a frame check sequence (FCS) or checksum, to each message before encryption.

Internal and External Error Control



M: plaintext message; F: function that produces an FCS (frame check sequence); $||$: concatenation operation; $E(K, M)$: encryption of M using key K; $D(K, M)$: decryption of M using key K; K: shared key between source A and destination B.

Internal and External Error Control

- Note that the order in which the FCS and encryption functions are performed is critical.
- With internal error control, authentication is provided because an opponent would have difficulty generating ciphertext that, when decrypted, would have valid error control bits.
- If instead the FCS is the outer code, an opponent can construct messages with valid error-control codes.
- Although the opponent cannot know what the decrypted plaintext will be, he or she can still hope to create confusion and disrupt operations.

Password Management

Password Management

Password Protection

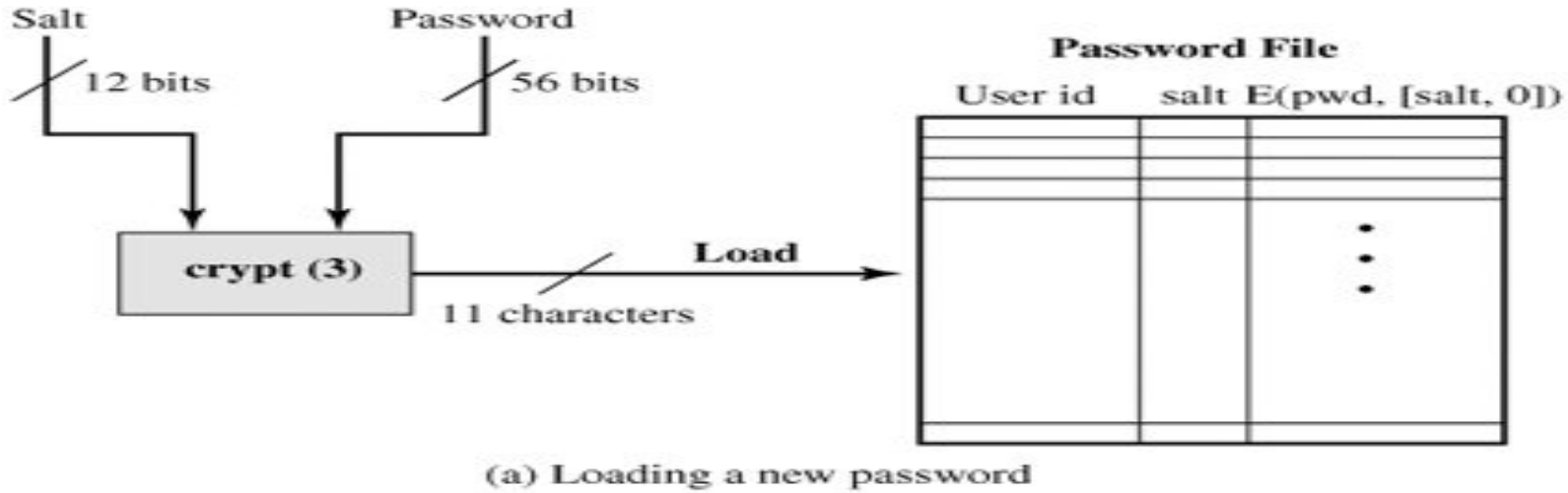
- The front line of defense against intruders is the password system.
- Virtually all multiuser systems require that a user provide not only a name or identifier (ID) but also a password.
- The password serves to authenticate the ID of the individual logging on to the system. In turn, the ID provides security in the following ways:
  - The ID determines whether the user is authorized to gain access to a system. In some systems, only those who already have an ID filed on the system are allowed to gain access.
  - The ID determines the privileges accorded to the user. A few users may have supervisory or “superuser” status that enables them to read files and perform functions that are especially protected by the operating system.
  - The ID is used in what is referred to as discretionary access control. For example, by listing the IDs of the other users, a user may grant permission to them to read files owned by that user.

Password Management

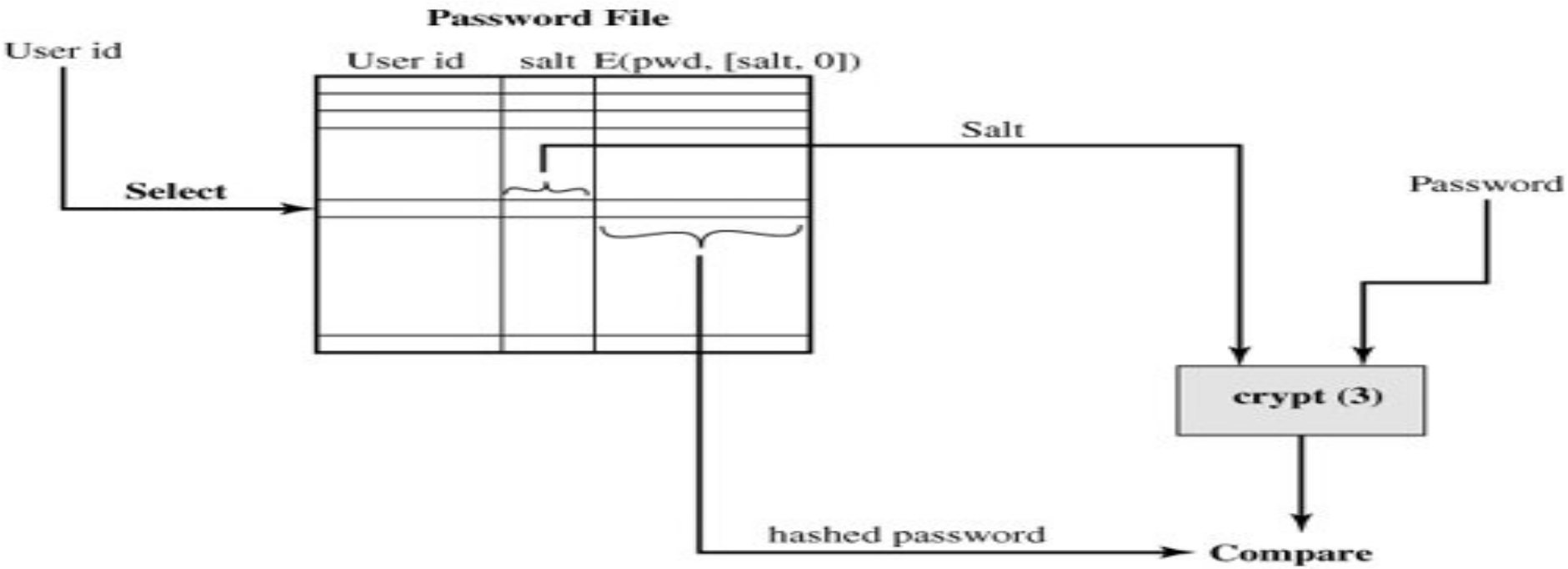
The Vulnerability of Passwords

- To understand the nature of the threat to password-based systems, let us consider a scheme that is widely used on UNIX, in which passwords are never stored in the clear.
- Each user selects a password of up to eight printable characters in length. This is converted into a 56-bit value (using 7-bit ASCII) that serves as the key input to an encryption routine.
- The encryption routine, known as crypt(3), is based on DES. The DES algorithm is modified using a 12-bit "salt" value. Typically, this value is related to the time at which the password is assigned to the user. The modified DES algorithm is exercised with a data input consisting of a 64-bit block of zeros. The output of the algorithm then serves as input for a second encryption. This process is repeated for a total of 25 encryptions. The resulting 64-bit output is then translated into an 11-character sequence.

Password Management in UNIX



(a) Loading a new password



(b) Verifying a password

Password Management

The Vulnerability of Passwords

The salt serves three purposes:

- It prevents duplicate passwords from being visible in the password file. Even if two users choose the same password, those passwords will be assigned at different times. Hence, the "extended" passwords of the two users will differ.
- It effectively increases the length of the password without requiring the user to remember two additional characters.
- It prevents the use of a hardware implementation of DES, which would ease the difficulty of a brute-force guessing attack.

Password Management

The Vulnerability of Passwords

There are two threats to the UNIX password scheme:

- First, a user can gain access on a machine using a guest account or by some other means and then run a password guessing program, called a password cracker, on that machine. The attacker should be able to check hundreds and perhaps thousands of possible passwords with little resource consumption.
- In addition, if an opponent is able to obtain a copy of the password file, then a cracker program can be run on another machine at leisure. This enables the opponent to run through many thousands of possible passwords in a reasonable period.

Password Management

The Vulnerability of Passwords

[Alvare, A. "How Crackers Crack Passwords or What Passwords to Avoid." Proceedings, UNIX Security Workshop II, August 1990.] reports the following techniques for learning passwords:

- Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.
- Exhaustively try all short passwords (those of one to three characters).
- Try words in the system's online dictionary or a list of likely passwords. Examples of the latter are readily available on hacker bulletin boards.
- Collect information about users, such as their full names, the names of their spouse and children, boyfriends and girlfriends, pictures in their office, and books in their office that are related to hobbies.
- Try users' phone numbers, Social Security numbers, and room numbers.
- Try all legitimate license plate numbers for this state.
- Use a Trojan horse to bypass restrictions on access.

Password Management

Password Selection Strategies

Four basic techniques are in use:

- User education
- Computer-generated passwords
- Reactive password checking
- Proactive password checking

Password Management

Password Selection Strategies

User education:

- Users can be told the importance of using hard-to-guess passwords and can be provided with guidelines for selecting strong passwords.
- This user education strategy is unlikely to succeed at most installations, particularly where there is a large user population or a lot of turnover.
- Many users will simply ignore the guidelines.

Password Management

Password Selection Strategies

Computer-generated passwords:

- If the passwords are quite random in nature, users will not be able to remember them.
- Even if the password is pronounceable, the user may have difficulty remembering it and so be tempted to write it down.
- In general, computer-generated password schemes have a history of poor acceptance by users.

Password Management

Password Selection Strategies

Reactive password checking:

- This strategy is one in which the system periodically runs its own password cracker to find guessable passwords.
- The system cancels any passwords that are guessed and notifies the user.
- This tactic has a number of drawbacks: It is resource intensive if the job is done right.

Password Management

Password Selection Strategies

Proactive password checking:

- This is the most promising approach to improve password security.
- A user is allowed to select his or her own password. However, at the time of selection, the system checks to see if the password is allowable and, if not, rejects it.
- Such checkers are based on the philosophy that, with sufficient guidance from the system, users can select memorable passwords from a fairly large password space that are not likely to be guessed in a dictionary attack.
- For example, the following rules could be enforced:
  - All passwords must be at least eight characters long.
  - In the first eight characters, the passwords must include at least one each of uppercase, lowercase, numeric digits, and punctuation marks.

Password Management

Zipf's Law in Passwords

- D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's Law in Passwords," **IEEE Transactions on Information Forensics and Security**, vol. 12, no. 11, pp. 2776–2791, Nov 2017.

Password Management

Biometrics and Fuzzy Extractor

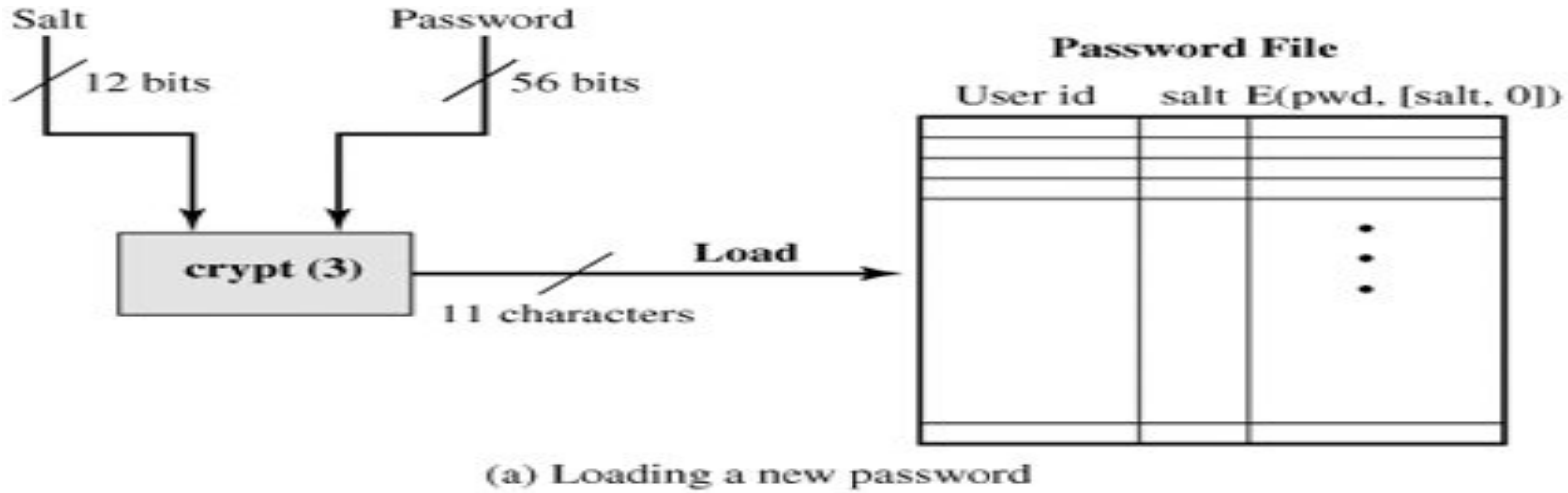
- Let $\mathcal{M} = \{0,1\}^v$ denote a finite v-dimensional metric space of biometric data points, $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{Z}^+$ a distance function, which can be used to calculate the distance between two points based on the metric chosen, l the number of bits of the output string, and t the error tolerance, where $\mathbb{Z}^+$ represents the set of all positive integers.
- The fuzzy extractor is a tuple $(\mathcal{M}, l, t)$, which is composed of the following two algorithms, called *Gen* and *Rep*:
  - **Gen:** It is a probabilistic algorithm, which takes a biometric information $B_i \in \mathcal{M}$ as input, and then outputs a secret key data $\sigma_i \in \{0,1\}^l$ and a public reproduction parameter τ_i, where $Gen(B_i) = \{\sigma_i, \tau_i\}$.
  - **Rep:** This is a deterministic algorithm, which takes a noisy biometric information $B_i' \in \mathcal{M}$ and a public parameter τ_i and t related to B_i, and then it reproduces (recovers) the biometric key data σ_i. In other words, we have $Rep(B_i', \tau_i) = \sigma_i$ provided that the condition $d(B_i, B_i') \leq t$ is met.

Password Management

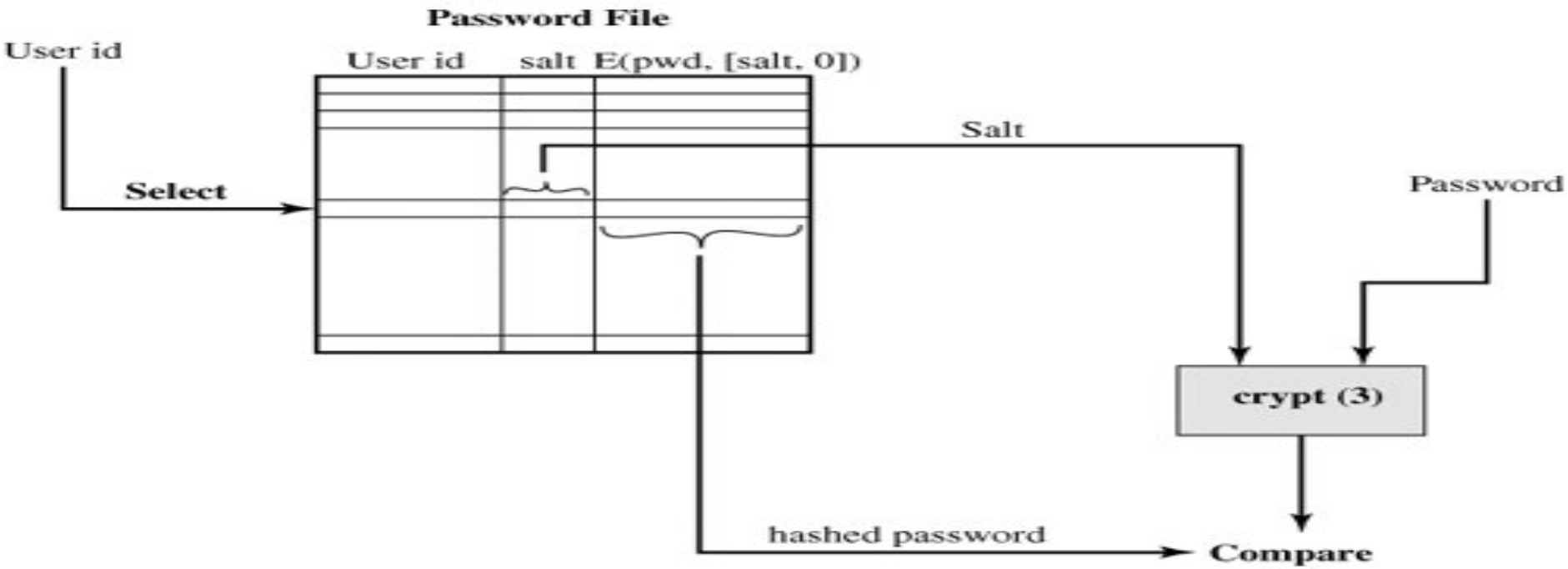
Biometrics and Fuzzy Extractor

- Vanga Odelu, **Ashok Kumar Das**, and Adrijit Goswami. ”A Secure Biometrics-Based Multi-Server Authentication Protocol using Smart Cards,” in ***IEEE Transactions on Information Forensics and Security***, Vol. 10, No. 9, pp. 1953 - 1966, 2015. (2015 SCI Impact Factor: 2.441) [This article is one of the top 50 most frequently downloaded documents for Popular Articles (June 2015 - June 2016)]

Password Management in UNIX



(a) Loading a new password



(b) Verifying a password

Principles of 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
Personal Home Page: http://sites.google.com/site/iitkgpakdas/

Digital Signatures

Signature Schemes

- A *signature scheme* is a five-tuple $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, where the following conditions are satisfied:
- 1. $\mathcal{P}$ is a finite set of possible messages;
- 2. $\mathcal{A}$ is a finite set of possible signatures;
- 3. $\mathcal{K}$, the keyspace, is a finite set of possible keys;
- 4. For each $k \in \mathcal{K}$, there is a signing algorithm $sig_k \in \mathcal{S}$ and a corresponding verification algorithm $ver_k \in \mathcal{V}$. Each $sig_k : \mathcal{P} \rightarrow \mathcal{A}$ and $ver_k : \mathcal{P} \times \mathcal{A} \rightarrow \{true, false\}$ are functions such that the following equation is satisfied for every message $x \in \mathcal{P}$ and for every signature $y \in \mathcal{A}$:
  $ver_k(x, y) =$ true, if $y = sig_k(x)$,
  $ver_k(x, y) =$ false, if $y \neq sig_k(x)$.
- The pair (x, y) with $x \in \mathcal{P}$ and $y \in \mathcal{A}$ is called a *signed message*.

The Digital Signature Algorithm (DSA)

- The DSA is based on the difficulty of computing logarithms and is based on schemes originally presented by ElGamal and Schnorr.

Table: Global Public-Key Components

p	prime number where $2^{L-1} < p < 2^L$ for $512 \leq L \leq 1024$ and L is a multiple of 64.
q	prime divisor of $(p-1)$, where $2^{159} < q < 2^{160}$; i.e., bit length of 160 bits.
g	$= h^{(p-1)/q} \mod p$, where h is any integer with $1 < h < p-1$ such that $h^{(p-1)/q} \mod p > 1$.

The Digital Signature Algorithm (DSA) (Continued...)

Table: User's Private Key

x	random or psuedo-random integer with $0 < x < q$.

Table: User's Public Key

y	$= g^x \mod p$.

Table: User's Per-Message Secret Number

k	random or psuedo-random integer with $0 < k < q$.

The Digital Signature Algorithm (DSA) (Continued...)

Table: Signing Phase

$r = (g^k \mod p) \mod q$
$s = [k^{-1}(H(M) + x.r)] \mod q$
Signature = (r, s)
Send $(M, (r, s))$ to reviever.

M : message to be signed

Digital Signatures

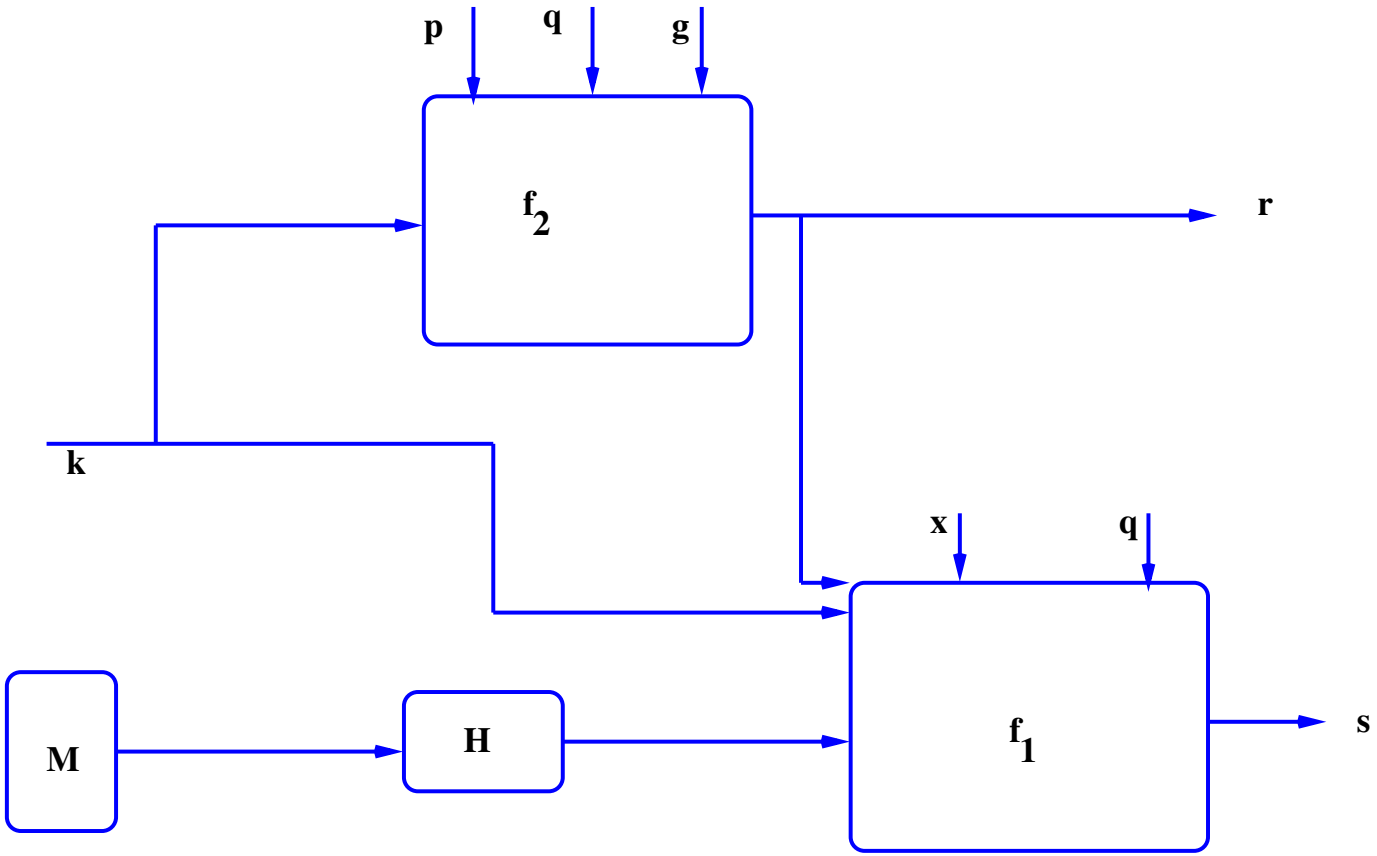


Figure: (a) Signing

$$s = f_1(H(M), k, x, r, q) = [k^{-1}(H(M) + x.r)] \mod q$$
$$r = f_2(k, p, q, g) = (g^k \mod p) \mod q$$

The Digital Signature Algorithm (DSA) (Continued...)

Table: Verification Phase

$w = (s')^{-1} \mod q$
$u1 = [H(M').w] \mod q$
$u2 = (r').w \mod q$
$v = (g^{u1}.y^{u2} \mod p) \mod q$
TEST: $v = r'$. If so accept; otherwise reject.

M', r', s' : received versions of M, r, s

Digital Signatures

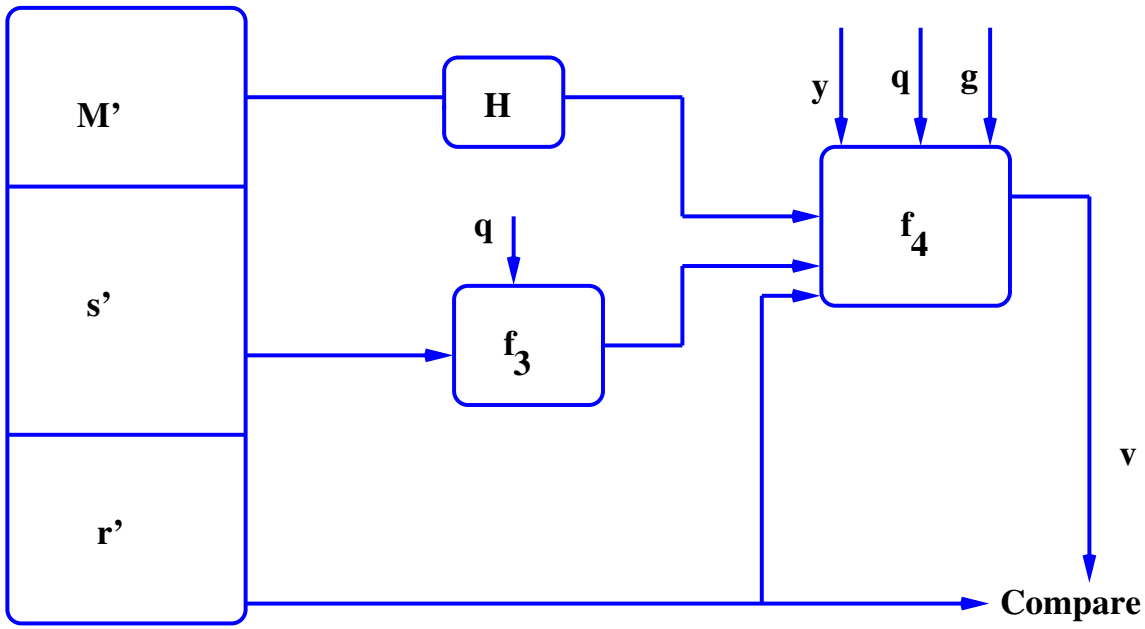


Figure: (b) Verifying

$$w = f_3(s', q) = (s')^{-1} \bmod q$$
$$v = f_4(y, q, g, H(M'), w, r')$$
$$= ((g^{(H(M').w) \bmod p}.y^{(r'.w) \bmod q}) \bmod p) \bmod q$$

The RSA Digital Signature

- The The RSA Digital Signature algorithm is based on the difficulty of factoring a composite number into two prime factors.

Table: Key generation of the RSA digital signature algorithm

Select p, q	p and q both prime, $p \neq q$ (p and q are large)
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(e, \phi(n)) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \mod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

The RSA Digital Signature (Continued...)

Setup Phase

- $n = p \times q$
- $\mathcal{P} = \mathcal{A} = \mathcal{Z}_n$
- Define the key space as
  $\mathcal{K} = \{(n, p, q, e, d) : n = pq; p, q \text{ primes}, \ ed \equiv 1 \pmod{\phi(n)}, \gcd(e, \phi(n)) = 1\}$
- Let two users A and B be Alice and Bob, respectively.
- Alice publishes $\{e, n\}$ and keeps $\{d, n\}$ as private.
- $x \in \mathcal{P}$ be a message to be signed by the Alice (User A).

The RSA Digital Signature (Continued...)

Signature Generation Phase

- A computes $y = sig_K(x) = x^d \pmod{n}$.
- The signature on the message x is y.
- The signed message on x is (x, y).
- $Alice \rightarrow Bob : (x, y)$.

The RSA Digital Signature (Continued...)

Signature Verification Phase

- B computes $x' = y^e \pmod{n}$.
- Bob accepts signature as valid if $x' = x$.
- Bob otherwise rejects the signature.

The RSA Digital Signature (Continued...)

Correctness Proof of Signature Verification Phase

$$
\begin{aligned}
x' &= y^e \pmod{n} \\
&= (x^d \pmod{n})^e \pmod{n} \\
&= x^{ed} \pmod{n}, \text{ since } ed \equiv 1 \pmod{\phi(n)} \\
&= x^{k\phi(n)+1} \pmod{n} \\
&= x.
\end{aligned}
$$

The ElGamal Digital Signature Scheme

Setup Phase

- p is selected as a large prime so that the discrete logarithm problem (DLP) in Z_p is intractable.
- $\alpha \in Z_p^*$ is a primitive root of p.
- $\mathcal{P} = Z_p^*$.
- $\mathcal{A} = Z_p^* \times Z_{p-1}$.
- Key space $\mathcal{K} = \{(p, \alpha, a, \beta) : 1 < a < p, \beta = \alpha^a \pmod p\}$.
- The public key is (p, α, β).
- The private key is a.

The ElGamal Digital Signature Scheme (Continued...)

Signature Generation Phase

- Let $x \in Z_p^*$ be a message to be signed by the signer.
- Signer A selects a random number $k \in Z_{p-1}^*$.
- Signer A computes $\gamma = \alpha^k \pmod{p}$.
- Signer A computes $\delta = (x - a\gamma)k^{-1} \pmod{p-1}$.
- Signature on message x is $sig_K(x) = (\gamma, \delta)$.
- Signer sends the signed message $(x, (\gamma, \delta))$ to the verifier V.

The ElGamal Digital Signature Scheme (Continued...)

Signature Verification Phase

- Verifier V has x, $\gamma \in Z_p^*$, $\delta \in Z_{p-1}$, and public key (p, α, β).
- Verfier V computes $\beta^{\gamma} \pmod{p}$.
- Verfier V computes $\gamma^{\delta} \pmod{p}$.
- Verifier V checks the condition: $\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} \pmod{p}$.
- If above holds, V accepts A's signature; otherwise, rejects it.

The ElGamal Digital Signature Scheme (Continued...)

Correctness Proof of Signature Verification Phase

- We have,

$$\begin{aligned} \delta &= (x - a\gamma)k^{-1} \pmod{p-1} \\ k\delta &= x - a\gamma \pmod{p-1} \\ a\gamma + k\delta &= x \pmod{p-1} \end{aligned}$$

- Now,

$$\begin{aligned} \beta^{\gamma}\gamma^{\delta} &= (\alpha^{a} \pmod{p})^{\gamma}(\alpha^{k} \pmod{p})^{\delta} \pmod{p} \\ &= \alpha^{a\gamma + k\delta} \pmod{p} \\ &= \alpha^{x} \pmod{p} \end{aligned}$$

The ElGamal Digital Signature Scheme (Continued...)

Example [ElGamal Digital Signature]

- Take a prime $p = 467$.
- Take a primitive root $\alpha = 2$.
- Choose randomly $a = 127$.
- Compute by repeated square-and-multiply algorithm,

$$\begin{aligned} \beta &= \alpha^a \pmod{p} \\ &= 2^{127} \pmod{467} \\ &= 132. \end{aligned}$$

- The public key is $(p, \alpha, \beta) = (467, 2, 132)$.
- The private key is $a = 127$.

The ElGamal Digital Signature Scheme (Continued...)

Example [ElGamal Digital Signature] (Continued...)

- Suppose Alice wants to sign the message $x = 100$.
- Alice chooses the random number $k = 213 \in Z_p^*$. Note that $\gcd(k, p-1) = \gcd(213, 466) = 1$.
- Alice computes $k^{-1} \pmod{p-1} = 213^{-1} \pmod{466} = 431$, using the extended Euclid's GCD algorithm.
- Alice computes by repeated square-and-multiply algorithm,

$$\begin{aligned} \gamma &= \alpha^k \pmod{p} \\ &= 2^{213} \pmod{467} \\ &= 29, \\ \delta &= (x - a\gamma)k^{-1} \pmod{p-1} \\ &= (100 - 127 \times 29) \times 431 \pmod{466} \\ &= 51. \end{aligned}$$

The ElGamal Digital Signature Scheme (Continued...)

Example [ElGamal Digital Signature] (Continued...)

- Alice sends the signed message $(m, (\gamma, \delta)) = (100, (29, 51))$ to verifier, Bob.
- Bob verifies this signature by checking that $\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} \pmod{p}$.
- Bob computes by repeated square-and-multiply algorithm,

$$\begin{aligned} \beta^{\gamma}\gamma^{\delta} &= 132^{29} \times 29^{51} \pmod{467} \\ &= 189, \\ \alpha^{x} &= 2^{100} \pmod{467} \\ &= 189. \end{aligned}$$

- Hence, the signature is verified.

Thank You

Research in Information Security

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
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/`

Algorithms and Approaches of Proxy Signatures

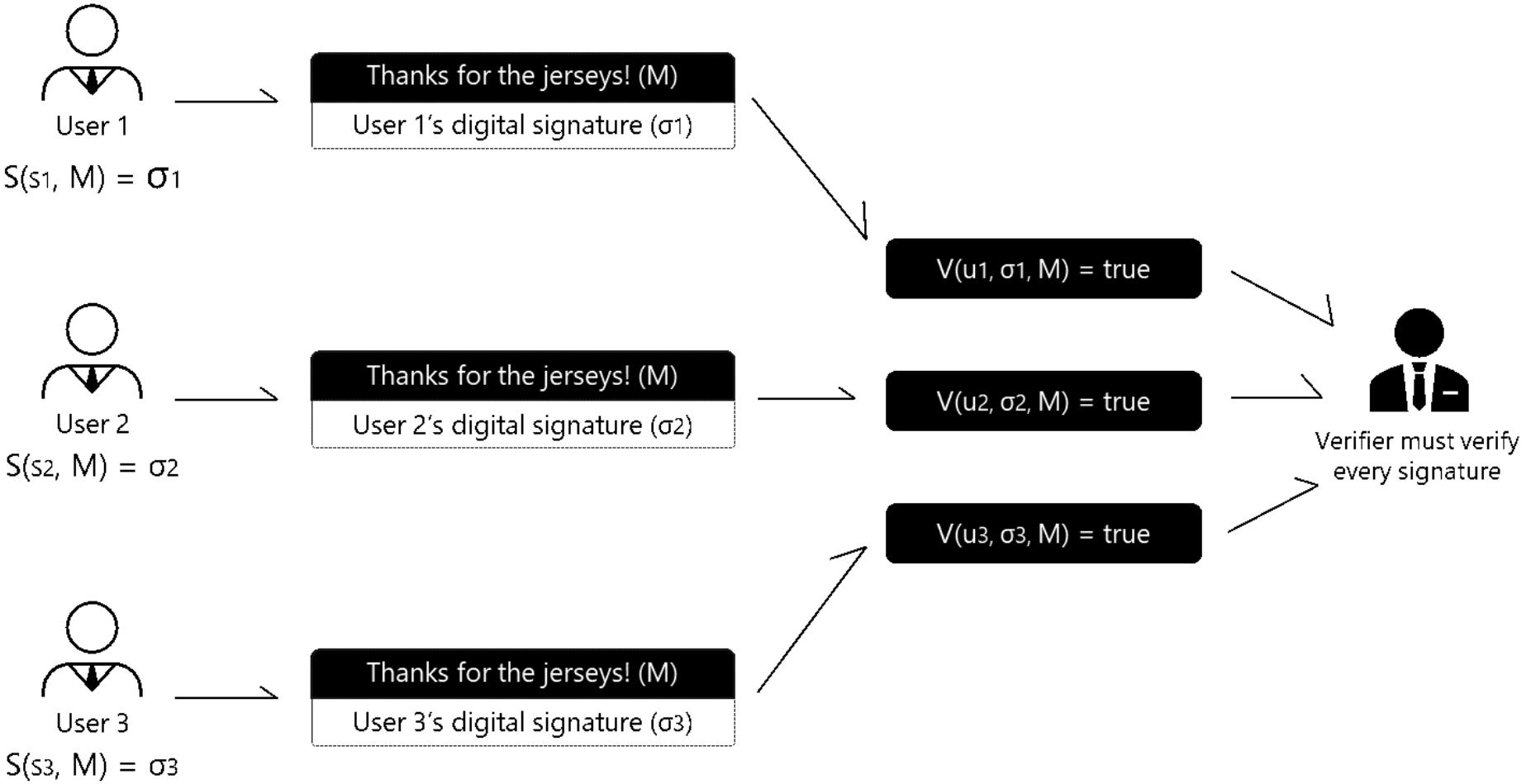
Proxy Signatures

Digital Signature

- Digital signature is a cryptographic means through which the authenticity, data integrity and signer's non-repudiation can be verified.
- Typically, digital signature of a document is a piece of information encrypted by the signer's private key.
- Numerous researches have shown significant contributions to this field using various cryptographic primitives.
- Nevertheless, there are many practical environments where digital signatures do not possess specific requirements, and thereby digital signatures appear in several other forms, namely proxy signatures, multi signatures, blind signatures, ring signatures etc.

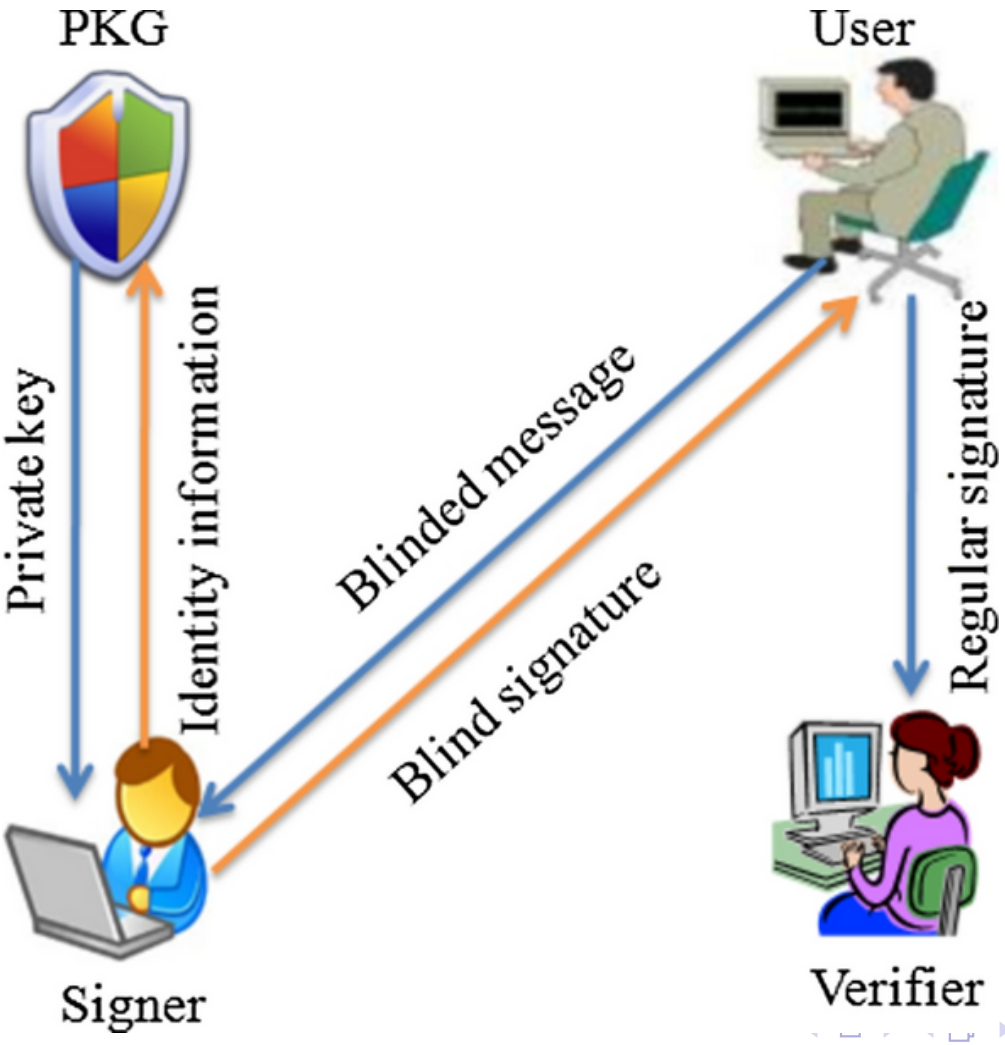
Multi-signature

- A multi-signature scheme enables a group of signers to produce a compact, joint signature on a common document, and has many potential uses.



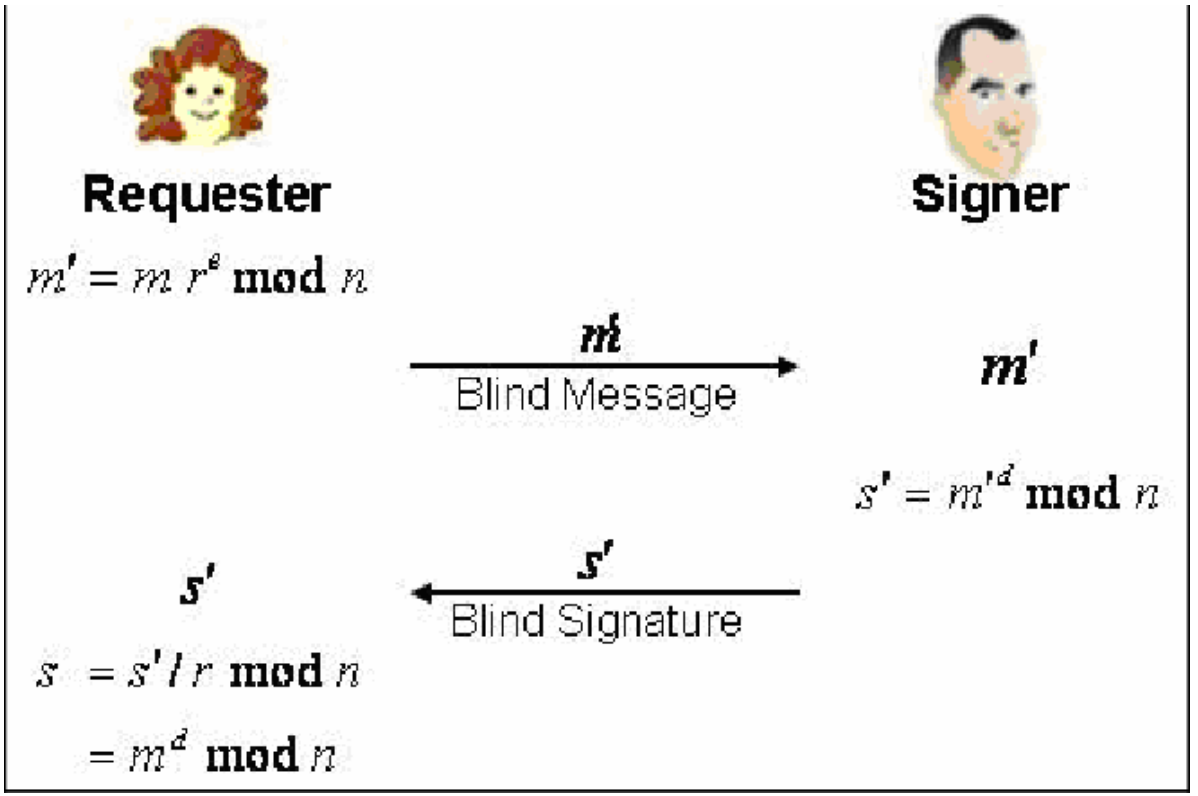
Blind signature

- In many applications involving anonymity, it is desirable to allow a participant to sign a message without knowing what the message is. This is called a blind signature.



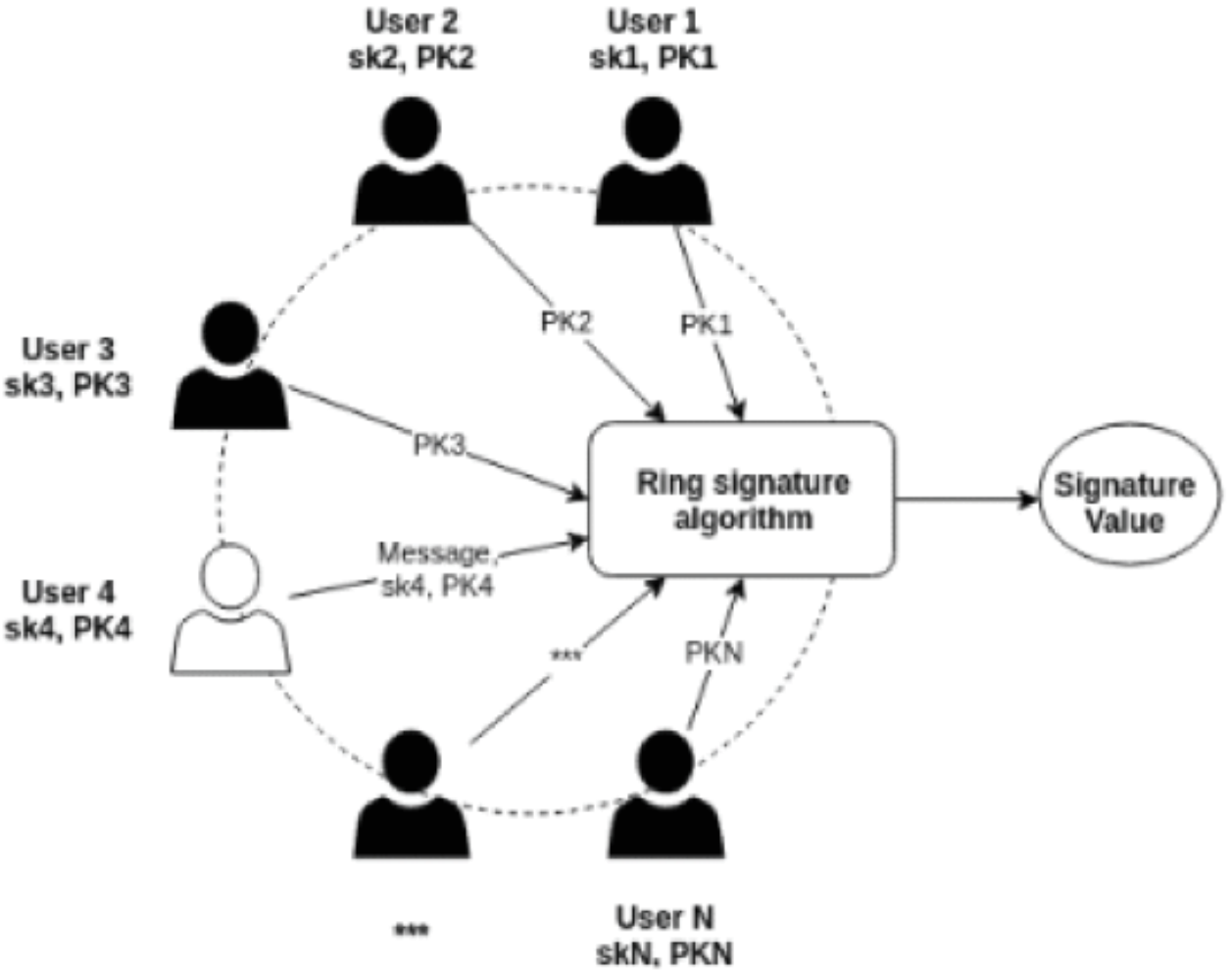
Blind signature

- **Bleumer, G. (2011). Chaum Blind Signature Scheme. In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA.**



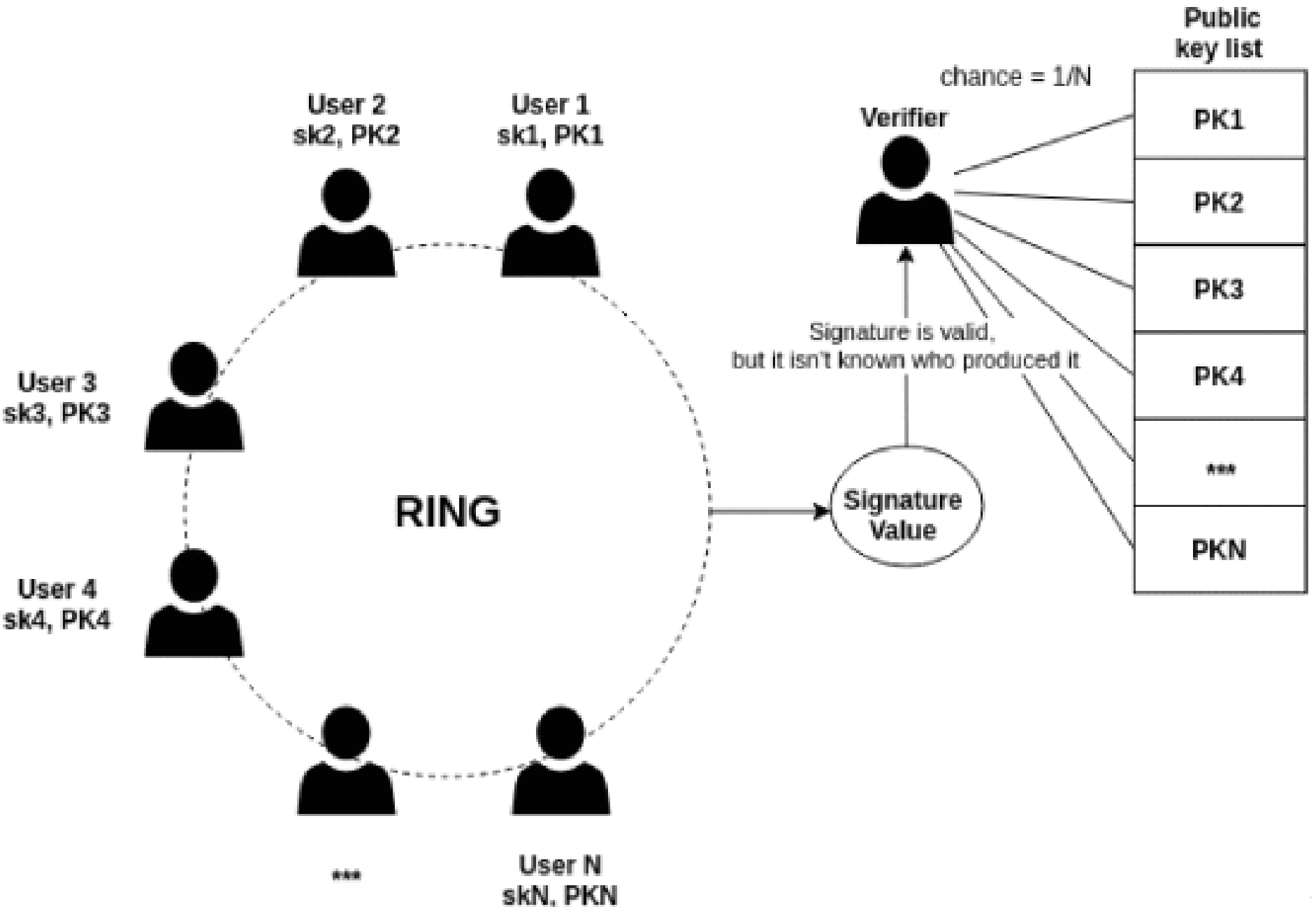
Ring signature

- Ring signatures, first introduced by Rivest, Shamir, and Tauman, enable a user to sign a message so that a ring of possible signers (of which the user is a member) is identified, without revealing exactly which member of that ring actually generated the signature.



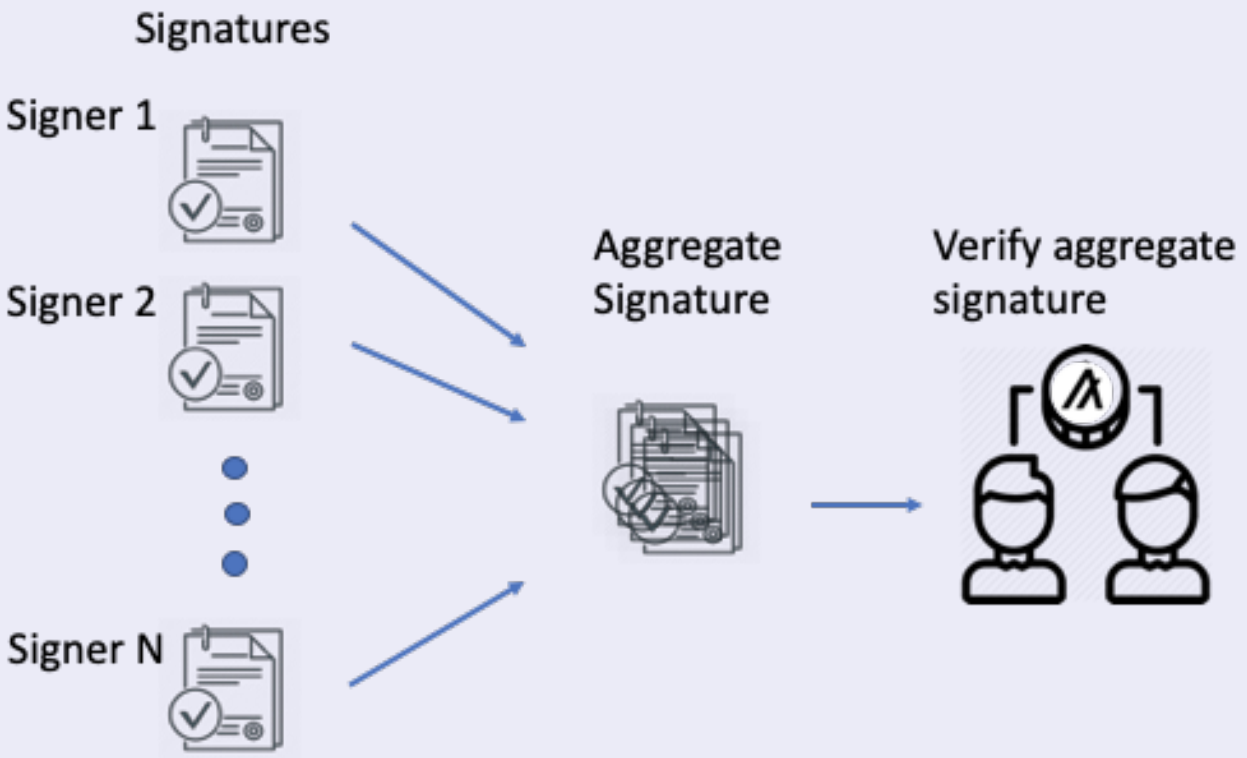
Signature calculation process

Ring signature verification process



Aggregate Signature

- In a general aggregate signature scheme, signatures are generated by individual users. They can then be combined into an aggregate signature by some aggregating party.
- An aggregate signature is the same length as an ordinary signature in the underlying scheme.

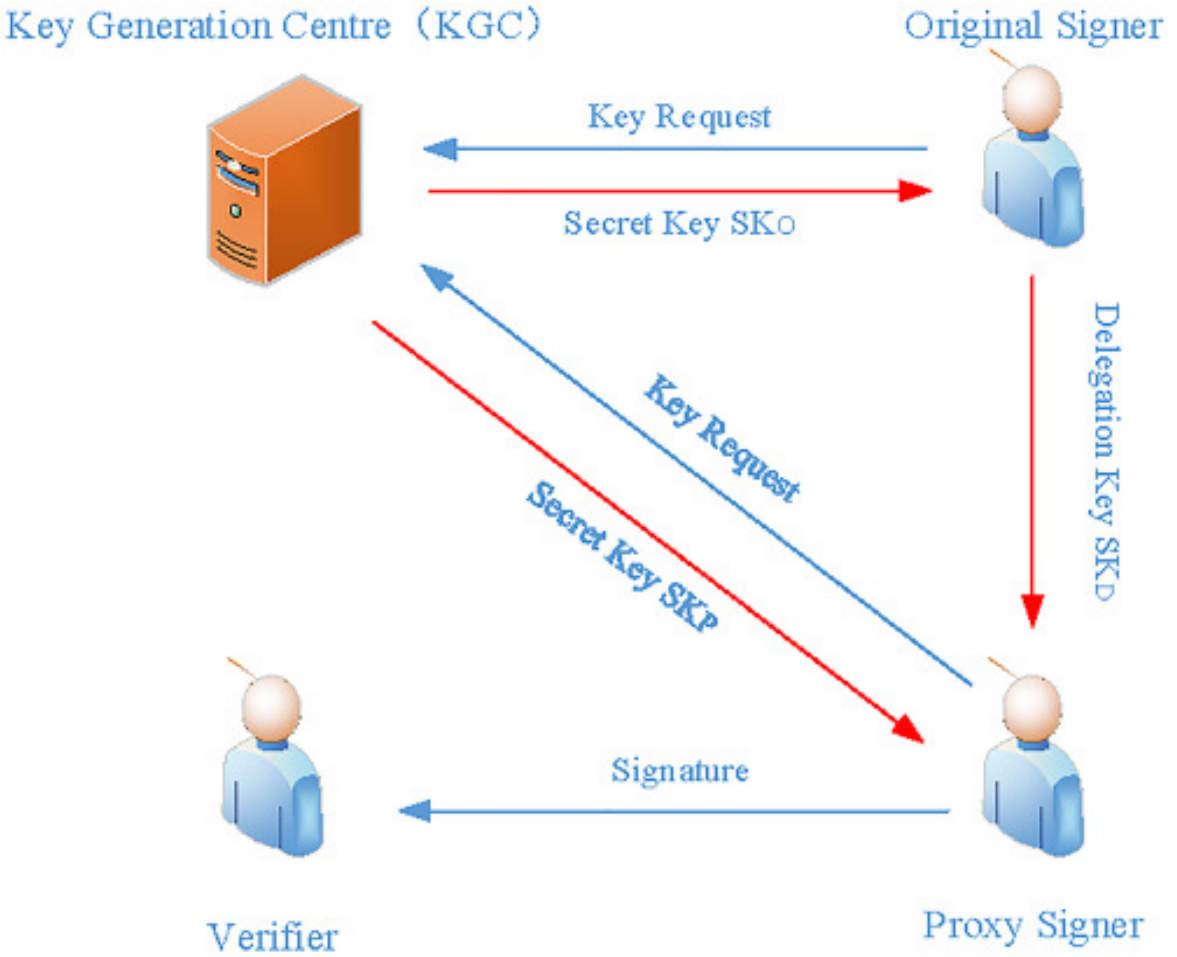


Proxy Signatures

Proxy signatures

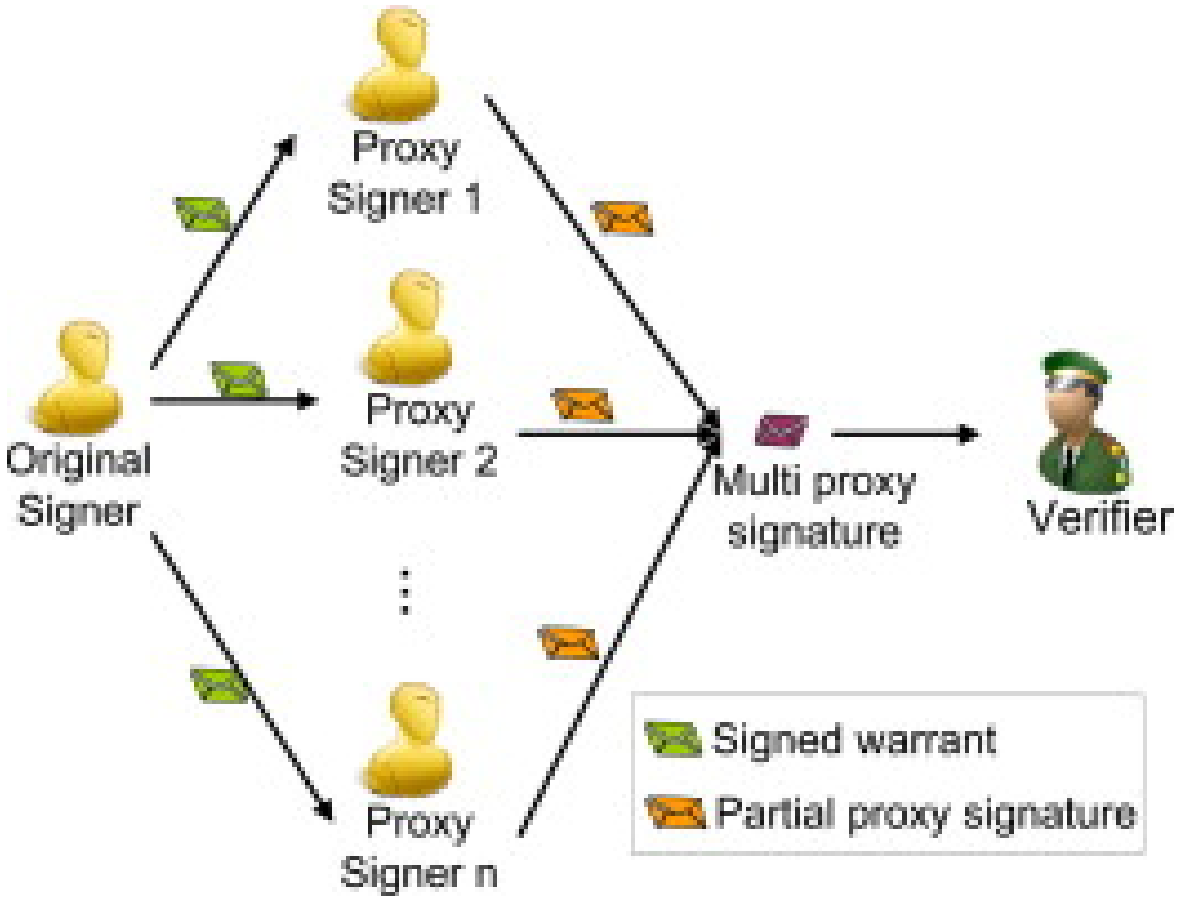
- Proxy signature is a digital signature where an original signer delegates her signing capability to a proxy signer, and then the proxy signer performs message signing on behalf of the original signer.
- **Example:** A manager of a company wants to go for a long trip. She would need a proxy agent, to whom she would delegate her signing capability, and thereafter the proxy agent would sign the documents on behalf of the manager.

Proxy Signature



Wu, F, Yao, W, Zhang, X, Wang, W, Zheng, Z. "Identity-based proxy signature over NTRU lattice," in ***Int J Commun Syst.***, 2019; 32:e3867. https://doi.org/10.1002/dac.3867

Multi-proxy Signature



Hu Xiong, Jianbin Hu, Zhong Chen, Fagen Li. "On the security of an identity based multi-proxy signature scheme," in ***Computers & Electrical Engineering***, vol. 37, no. 2, pp. 129-135, 2011.

Proxy Signatures

Proxy signatures

- The notion of proxy signature has been evolved over a long time (over 25 years now).
- However, the cryptographic treatment on proxy signature was introduced by Mambo *et al.* in 1996.
  M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signatures: Delegation of the Power to Sign Messages," IEICE Transactions Fundamentals, vol. E79-A, no. 9, pp. 1338-1353, 1996.
- Mambo *et al.* classified the proxy signature on the basis of delegation, namely ***full delegation***, ***partial delegation*** and ***delegation by warrant***.

Proxy Signatures

Full delegation

- In full delegation, an original signer gives her private key to a proxy signer and the proxy signer signs document using original signer's private key.
- The drawback of proxy signature with full delegation is that the absence of a distinguishability between original signer and proxy signer.

Proxy Signatures

Partial delegation

- In partial delegation, the original signer derives a proxy key from her private key and hands it over to the proxy signer as a delegation capability.
- In this case, the proxy signer can misuse the delegation capability, because partial delegation cannot restrict the proxy signer's signing capability.

Proxy Signatures

Delegation by warrant

- The weaknesses of full delegation and partial delegation are eliminated by partial delegation with warrant.
- A warrant explicitly states the signers' identity, delegation period and the qualification of messages on which the proxy signer can sign, etc.

Discrete logarithm problem and its applications

- The discrete logarithm is the inverse of discrete exponentiation in a finite cyclic group.
- **Instance:** A multiplicative group $(G, \cdot)$, an element $g \in G$ having order n and $y = g^x \mod n$.
  Question: Find x.
  This problem is computationally infeasible when n is large.

Formal definition of discrete logarithm problem

Let G be a cyclic group of order n, g a generator of G, and A an algorithm that returns an integer in Z_n, where $Z_n = \{0, 1, \ldots, n-1\}$. Let $a \in_R S$ denote that a is chosen randomly from the set S. Consider the following experiment, $EXP^{DLP}_{G,g}(A)$ in Algorithm 1.

Algorithm 1: $EXP^{DLP}_{G,g}(A)$

1: $x \in_R Z_n$
2: $X \leftarrow g^x \mod n$
3: $x' \leftarrow A(X)$
4: **if** $g^{x'} = X \mod n$ **then**
5: **return** 1 (Success)
6: **else**
7: **return** 0 (Failure)
8: **end if**

Formal definition of discrete logarithm problem

- The DLP-advantage of A is defined by $Adv_{G,g}^{DLP}(A) = Pr[Exp_{G,g}^{DLP}(A) = 1]$, where $Pr[E]$ denotes the probability of an event E.
- The discrete logarithm problem (DLP) is said to be a hard problem in G if the DLP-advantage of any adversary of reasonable resources is small, where resources are measured in terms of the time complexity of the adversary including its code size as usual.
- In other words, DLP is called a hard problem, if $Adv_{G,g}^{DLP}(A) \leq \epsilon$, for any sufficiently small $\epsilon > 0$.
- **Reference: Ashok Kumar Das. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. Networking Science (Springer), 2(1-2):12-27, 2013.**

The Schnorr signature scheme

The scheme is based on hardness of solving DLP. It consists of the following phases.

- **Setup** ($\mathcal{SP}_{dlp}$)**:** Takes input 1^k and outputs **params-dlp**.
  The **params-dlp** consists of primes q and l such that $2^{k-1} \leq q < 2^k$, an element $g \in Z_q^*$ of order l that divides $q-1$, and a hash function $h : \{0,1\}^* \rightarrow Z_l$.
- **KeyGen** ($\mathcal{KG}_{dlp}$)**:** The users agree on a group G (multiplicative group of integers modulo q) for some prime q with generator g of prime order l in which the DLP is hard problem.
  The user chooses a private key $x \in Z_l$ and then computes the publlic key as $y = g^x \pmod{q}$.
  In other words,
  user public key $\leftarrow \mathcal{KG}_{dlp}$(**params-dlp**, user private key), that is,
  $y \leftarrow \mathcal{KG}_{dlp}$(**params-dlp**, x).

The Schnorr signature scheme (Continued...)

- **Sign** ($\mathcal{S}_{dlp}$)**:** To sign a message, say m, the signer has to choose a random number $t \in Z_l$ and calculate $r = g^t \pmod{q}$.
  Then the signer computes $c = h(m||r)$ and $\sigma = (t - xc) \pmod{l}$.
  The signature on message m is then (σ, c).
  In other words, $\sigma \leftarrow \mathcal{S}_{dlp}($ **params-dlp**, $(t, r), x, m)$.
- **Verify** ($\mathcal{V}_{dlp}$)**:** The verifier calculates $r' = g^{\sigma} y^{c} \pmod{q}$ and $c' = h(m||r')$.
  If the condition $c' = c$ is satisfied, the signature is treated as valid; otherwise, the signature is invalid.
  In other words, **result** $\leftarrow \mathcal{V}_{dlp}($ **params-dlp**, $y, \sigma, m)$,
  where **result** $\in \{valid, invalid\}$.

Remark: The Schnorr signature scheme is proven to be secure under the assumption that the DLP is intractable (NP-hard).

Proxy Signatures

Security properties of proxy signature

- **Strong unforgeability:** A designated proxy signer can create a valid proxy signature on behalf of the original signer. But the original signer and other third parties cannot create a valid proxy signature.
- **Strong identifiability:** Anyone can determine the identity of corresponding proxy signer from the proxy signature.
- **Strong undeniability:** Once a proxy signer creates a valid proxy signature on behalf of the original signer, he cannot deny the signature creation.
- **Verifiability:** The verifier can be convinced of the signers' agreement from the proxy signature.
- **Distinguishability:** Proxy signatures are distinguishable from the normal signatures by everyone.
- **Secrecy:** The original signer's private key cannot be derived from any information, such as the shares of the proxy key, proxy signatures, etc.

Generalized DLP-based Proxy Signature Model

Generalized DLP-based Proxy Signature Model

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Participants

- An **original signer**, who delegates her signing capability to a **proxy signer**.
- A proxy signer, who signs the message (document) on behalf of the original signer.
- A **verifier**, who verifies the proxy signature and decides to accept or reject the signature.
- A **trusted authority**, who certifies the public key.

Generalized DLP-based Proxy Signature Model

Mechanism

- An original signer selects a private key x_o and computes her public key y_o as

$$y_o \leftarrow \mathcal{KG}_{dlp}(\textbf{params-dlp}, x_o)$$

- A proxy signer selects a private key x_p and computes her public key y_p as

$$y_p \leftarrow \mathcal{KG}_{dlp}(\textbf{params-dlp}, x_p)$$

- **Delegation capability generation**:
  It takes **params-dlp**, original signer's chosen parameters (k_o, r_o), original signer's private key x_o, a warrant w as input; and outputs signature σ_o on w as

$$\sigma_o \leftarrow \mathcal{S}_{dlp}(\textbf{params-dlp}, (k_o, r_o), x_o, w)$$

Generalized DLP-based Proxy Signature Model

Mechanism (Continued...)

- **Delegation capability verification**: It takes **params-dlp**, y_o, w, σ_o as input; produces the output **Result** where **Result** $\in \{Valid, Invalid\}$, that is,

$$\textbf{Result} \leftarrow \mathcal{V}_{dlp}(\textbf{params-dlp}, y_o, \sigma_o, w)$$

- **Proxy key generation (PKeyGen$_{dlp}$)**: It takes **params-dlp**, σ_o, x_p and random number as input; and outputs proxy key ρ_p. Typically, the proxy signer applies simple arithmetic operation to form a proxy key as

$$\rho_p \leftarrow y_o\sigma_o + x_p y_p \pmod{q}$$

Procedurally,

$$\rho_p \leftarrow PKeyGen_{dlp}(\textbf{params-dlp}, \sigma_o, x_p, \text{pub-parameters})$$

Generalized DLP-based Proxy Signature Model

Mechanism (Continued...)

- **Proxy signature generation**:
  It takes **params-dlp**, proxy key σ_p and message m as input; outputs proxy signature σ_p on m, that is,

$$\sigma_p \leftarrow \mathcal{S}_{dlp}(\textbf{params-dlp}, \rho_p, m)$$

- **Proxy signature verification**:
  It takes **params-dlp**, y_o, y_p, m and σ_p as input; outputs **Result**, that is,

$$\textbf{Result} \leftarrow \mathcal{V}_{dlp}(\textbf{params-dlp}, (y_o, y_p), \sigma_p, m)$$

Mambo *et al.*'s scheme: A case study of generalized DLP-based proxy signature model

M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signatures: Delegation of the Power to Sign Messages," IEICE Transactions Fundamentals, vol. E79-A, no. 9, pp. 1338-1353, 1996.

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Assumptions

- This scheme is based on the discrete logarithm problem.
- $v \equiv g^s \pmod{p}$, where $s \in_R Z_{p-1} \backslash \{0\}$ is a secret of the original signer and p is a prime number whose length is taken greater than 512 bits. g is a generator for $Z_p^* = \{1, 2, 3, \ldots, p-1\}$.
- To determine the discrete logarithm s.

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Proxy Generation

- An original signer generates a random number $k \in_R Z_{p-1} \setminus \{0\}$ and computes $K = g^k \pmod{p}$. After that, he calculates $\sigma = s + kK \pmod{p-1}$.

Proxy Delivery

- The original signer gives (σ, K) to a proxy signer in a secure way.

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Proxy Verification

- The proxy signer checks a congruence such that

$$g^{\sigma} \equiv vK^{K} \pmod{p}$$

- If (σ, K) passes this congruence, she accepts it as a valid proxy. Otherwise, she rejects it and requests him a valid one, or she stops this protocol.

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Signing by the proxy signer

- When the proxy signer signs a document m_p for the sake of the original signer, she uses the σ as an alternative to s, and executes the ordinary signing operation. Then, (m_p, (signature of the original signer), K) serves as a created proxy signature.

Verification of the proxy signature

- The verification of the proxy signature is carried out by the same checking operation as in the original signature scheme except for the extra computation
  $vK^K \pmod p$. The computed value $v'(\equiv vK^K \pmod p)$ is dealt with as a new public value.

Mambo *et al.*'s Proxy Signature Scheme: An Examples of Proxy Signature

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Assumptions

- A public key v of an original signer is computed by $v = g^s \pmod{p}$. p satisfies $|p| \leq 512$ for preserving scheme's security.
- s_p is a secret of a proxy signer, and v_p is a proxy signer's public key satisfying $v_p \equiv g^{s_p} \pmod{p}$.
- After proxy submission process, (σ_p, K) satisfying the following congruences is obtained.
- $\sigma_p \equiv s + kK \pmod{p-1}$, where $K \equiv g^k \pmod{p}$ and $k \in_R Z_{p-1} \setminus \{0\}$

Mambo *et al.*'s Proxy Signature Scheme: An Examples of Proxy Signature

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Main Phases

1. **Signing**. A proxy signer generates a random number $r \in_R Z^*_{p-1}$, and computes $x = g^r \pmod{p}$. For a message m_p to be signed the proxy signer computes $y = r^{-1}(m_p - x\sigma_p) \pmod{p-1}$. After that, $(m_p, (x, y, K))$ is sent to a verifier.
2. **Verification.** A verifier checks $(m_p, (x, y, K))$ by the congruence $g^{m_p} \equiv (vK^K)^x x^y \pmod{p}$.
   If the check has succeeded, $(m_p, (x, y, K))$ is properly signed. Otherwise, signature is not valid, or the message might be manipulated.

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Research in Information Security

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
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/`

Intrusion Prevention and Detection

Intrusion Prevention and Detection

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- An intrusion is the act of gaining unauthorized access to a system so as to cause loss or harm.
- **Examples:**
  - Unauthorized login to a system by illegally acquiring a password (through, for example, a password guessing attack).
  - Worm injections that use the system as a launch pad to spread and infect other machines.
  - Injection of spyware that passively monitors the activities of the user and replay this information back to the attacker (over the Internet, for example).
  - Flooding the host with spurious connection requests that attempt to exhaust the target's resources - processing power, memory, or communication bandwidth.

Intrusion Prevention and Detection

- Two ways of handling attempted intrusions are the following:
  - Intrusion prevention
  - Intrusion detection
- A real-life example is as follows:
  - Modern day ailments like diabetes and high blood pressure are known to be linked to lifestyle choices.
  - Preventive "treatment" for these ailments includes regular exercises, a stress-free life, and a diet rich in fruits and vegetables.
  - However, an individual who strictly follows these three practices cannot be guaranteed freedom from those medical problems.
  - We know the regular health checkups (especially for those beyond 50 years of age) can help prolong life.
  - In the current context, periodic monitoring of blood sugar levels and blood pressure are strongly recommended so that timely action can be taken if there is deviation from the norm.
  - The regular health check-ups are analogous to "**intrusion detection**", while positive lifestyle choices are analogous to "**intrusion prevention**".

Classification of Intruders

- **Masquerader:** An individual who is not authorized to use the computer and who penetrates a system access controls to exploit a legitimate user's account.
- **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his/her privileges.
- **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.
- The masquerade is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider.

Intrusion Techniques

- The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system.
- Generally, this requires the intruder to acquire information that should have been protected.
- In most cases, this information is in the form of a user password.
- The password file can be protected in one of the two ways:
  - One-way encryption [password management, which is already discussed in the class]
  - Access control: Access to the password file is limited to one or a very few accounts.
- If one or both these countermeasures are in place, some effort is needed for a potential intruder to learn passwords.

Prevention versus Detection

Intrusion Detection

- A system second line of defense is intrusion detection, and this has been the focus of much research in recent years.
- This interest is motivated by a number of considerations, including the following:
  - If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.
    Even if the detection is not sufficiently timely to preempt the intruder, the sooner that the intrusion is detected, the less amount of damage and the more quickly that recovery can be achieved.
  - An effective intrusion detection system (IDS) can serve as a deterrent, so acting to prevent intrusions.
  - Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Prevention versus Detection

Intrusion Detection (Continued...)

- Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in way that can be quantified.
- Of course, we can not expect that there will be a crisp, exact distinction between an attack by an intruder and the normal use of resources by an authorized user. Rather, we must expect that there will be some overlap.
- Although the typical behavior of an intruder differs from the typical behavior of an authorized user, there is an overlap in these behaviors.

Prevention versus Detection

Intrusion Detection (Continued...)

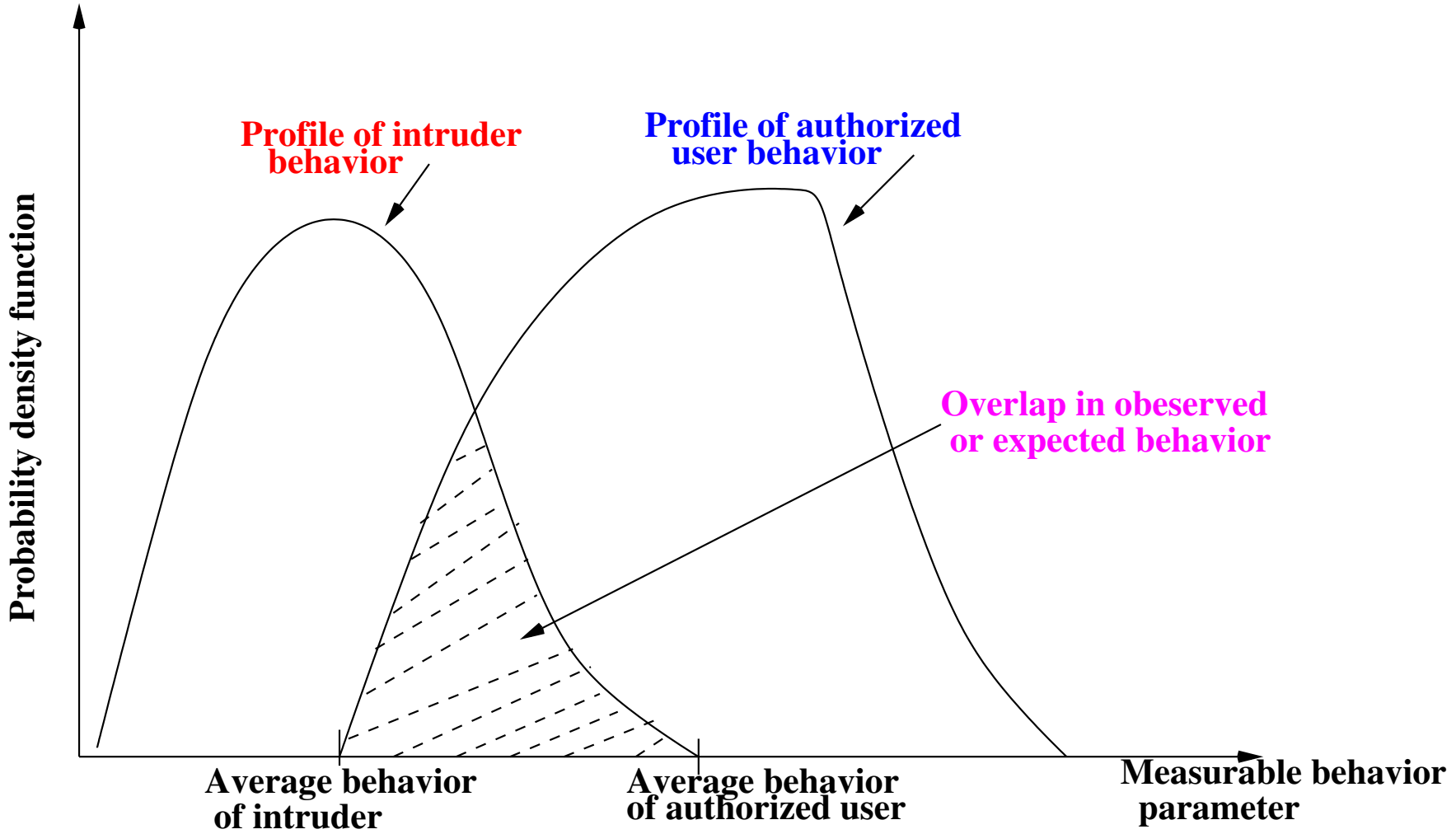


Figure: Profiles of behavior of intruders and authorized users

Prevention versus Detection

Intrusion Detection (Continued...)

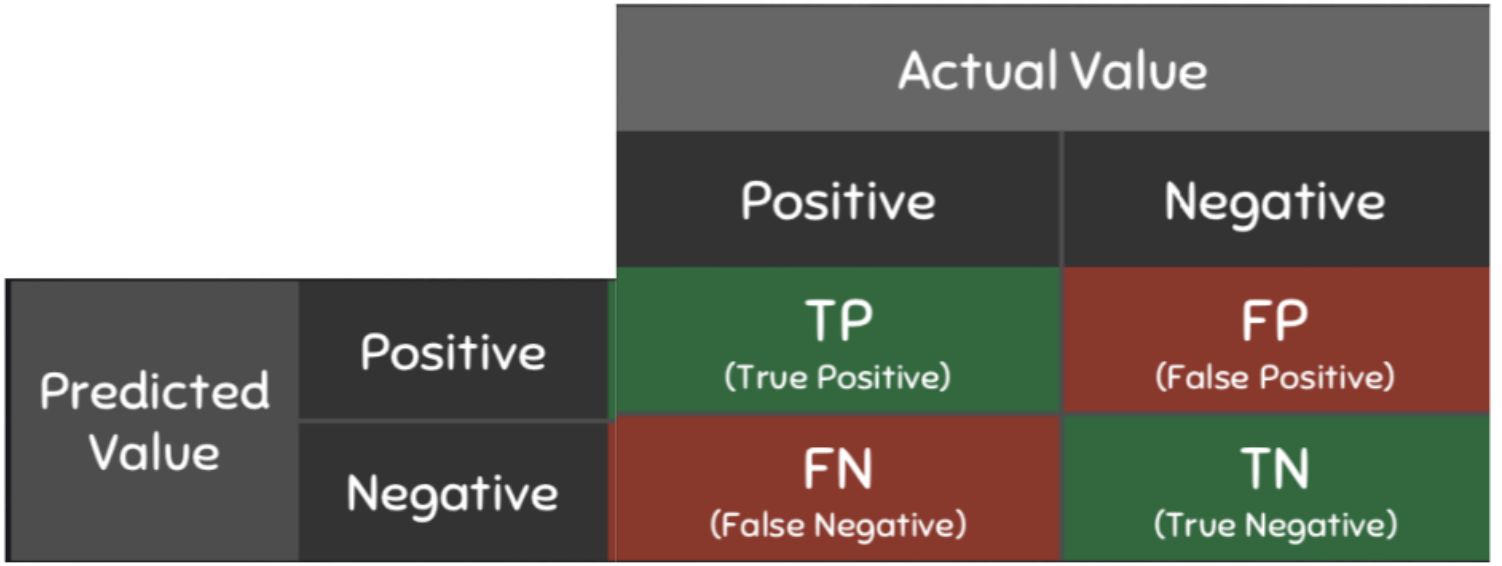
- **False positive:** A loose interpretation of intruder behavior, which will catch more intruders, will also lead to a number of "false positives" or authorized users are identified as intruders.
  The probability of a false accept is called the ***false accept rate (FAR)*** or ***fraud rate***.
- **False negative:** An attempt to limit false positives by a tight interpretation of intruder behavior will lead to an increase in "false negatives" or intruders are not identified as intruders.
  The probability of a false reject is called the ***false reject rate (FRR)*** or ***insult rate***.

Prevention versus Detection

Intrusion Detection (Continued...)

- False negatives are riskier than false positives.
- If there was not an attack and the IDS picked it (false positive), it is not much of a harm.
- But if there was an attack and the IDS does not detect it (false negative), it can be disastrous.

Metrics Used



- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Figure: Structure of a confusion matrix

Metrics Used

- *Accuracy*:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Recall*:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- *Precision*:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- *F1 score*:

$$\text{F1 score} = \frac{2 * TP}{2 * TP + FP + FN}$$

Prevention versus Detection

Intrusion Detection (Continued...)

- An IDS performs the following three tasks:
  - First, it monitors “event of interest” occurring in the target system or in the network.
    An event of interest may be
    - ★ A system call (a call made to the OS) to, for example, open a file containing sensitive data.
    - ★ Another event of interest may be the attempted establishment of a TCP connection from a specific IP address to a certain port.
  - An IDS generates a large amount data which it then analyzes and converts into valuable information to be used by system administrators.
    The information gleaned may lead to conclusions such as “the source addresses in 90% of the packets received in the past 5 minutes are never-seen-before IP addresses”.
  - The IDS creates a database of “interesting events”. It raises an alert each time it observes any such event.

Prevention versus Detection

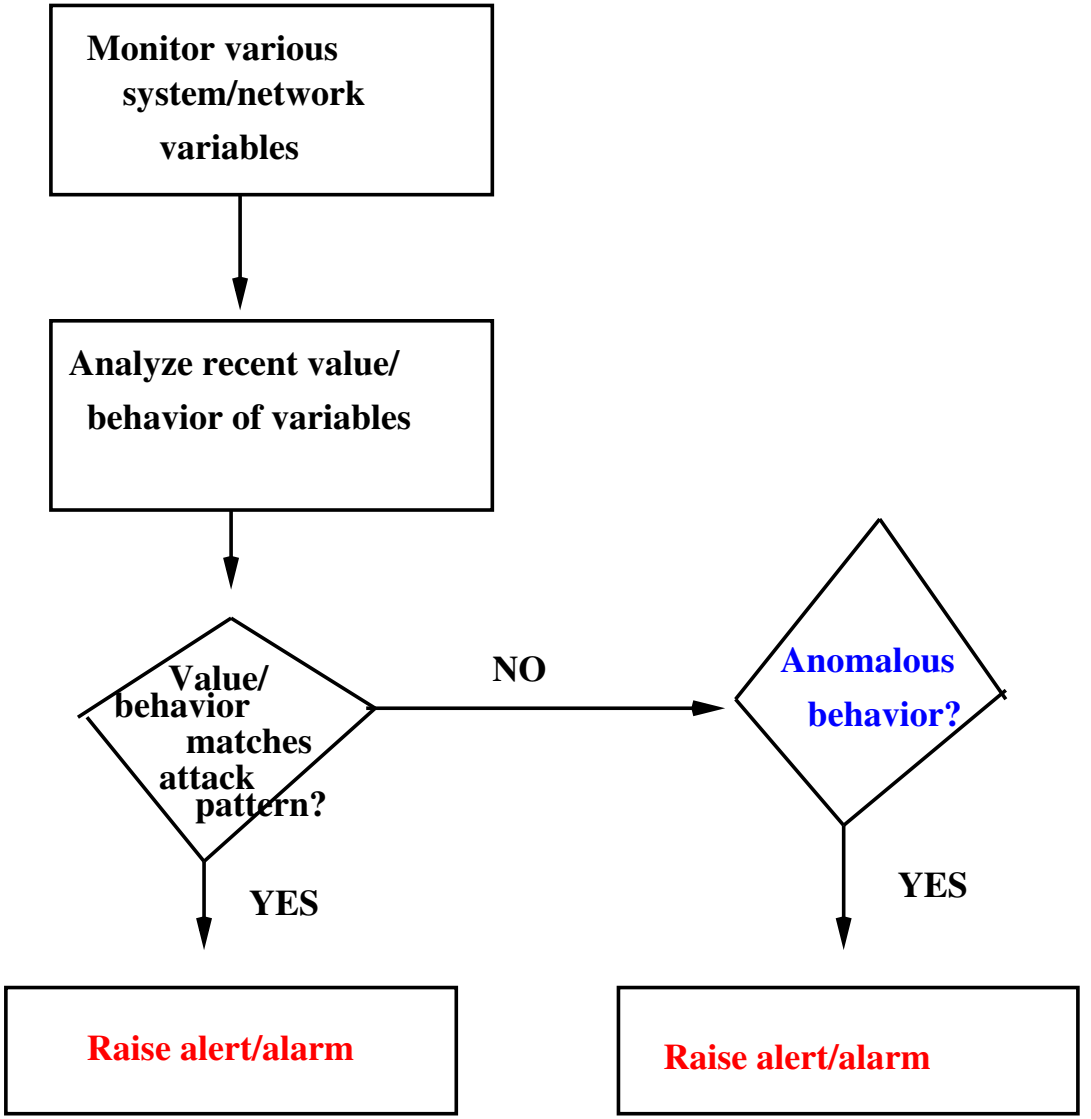


Figure: Tasks performed by an IDS.

Prevention versus Detection

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Table: Events of interest to an IDS

Variable monitored	**Event of interest**	**Possible attack**
Number of accesses of specific file	Tenfold increase over norm	DoS attack
Login frequency to a particular account	Unusually high	Attempted break-in
Number of distinct source IP addresses of arriving packets	Very high	Worm attack
Ratio of ARP request packets to ARP response packets	$>> 1$	Network scan to identify local active hosts
Ratio of TCP SYN packets to TCP FIN packets	$>> 1$	Possible DoS attack

Prevention versus Detection

Table: Events of interest to an IDS (Continued...)

Variable monitored	**Event of interest**	**Possible attack**
Percentage of half-open TCP connections	Sudden surge	Possible DoS/DDoS attack
TCP header flags	Invalid combination	Port scan, OS fingerprinting
TCP connection establishment	Unused destination port	Attempt to find which services are open
Payload of incoming packets	Specific byte sequence present	Specific worm open
OS calls	Particular sequence of calls	Specific virus attack

Approaches to Intrusion Detection

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Statistical anomaly detection

- Involves the collection of data related to the behavior of legitimate users over a period of time.
- The statistical tests are then applied to observed behavior to determine with a high level of confidence whether the behavior is not legitimate user behavior.
  - **Chi Square test**: Compares observed frequencies to expected frequencies. It is for a population variance
    The chi-square distribution results when v independent variables with standard normal distributions are squared and summed. The formula for the probability density function of the chi-square distribution is

$$f(x) = \frac{e^{-\frac{x}{2}} x^{\frac{v}{2}-1}}{2^{\frac{v}{2}}\Gamma(\frac{v}{2})}, for\ x \geq 0$$

    where v is the shape parameter and Γ is the gamma function. The formula for the gamma function is

$$\Gamma(n) = \int_0^\infty t^{n-1} e^{-t}\, dt$$

Approaches to Intrusion Detection

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Statistical anomaly detection

- **t-Test**: Looks at differences between two groups on some variable of interest. It is for a population mean (variance unknown)
  If we take a sample of n observations from a normal distribution, then the t-distribution with $v = n - 1$ degrees of freedom can be defined by the following probability density function:

$$f(x) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\Gamma(\frac{v}{2})}\left(1 + \frac{x^2}{v}\right)^{-\frac{v+1}{2}} \quad \text{for } x \in (-\infty, +\infty)$$

Statistical anomaly detection

- **F-Test**: It is for two population variances
  The F-distribution is the ratio of two chi-square distributions with degrees of freedom v_1 and v_2, respectively, where each chi-square has first been divided by its degrees of freedom. The formula for the probability density function of the F-distribution is

$$f(x) = \frac{\Gamma(\frac{v_1+v_2}{2})(\frac{v_1}{v_2})^{\frac{v_1}{2}} x^{\frac{v_1}{2}-1}}{\Gamma(\frac{v_1}{2})\Gamma(\frac{v_2}{2})(1+\frac{v_1 x}{v_2})^{\frac{v_1+v_2}{2}}} \text{ for } x \geq 0$$

  where v_1 and v_2 are the shape parameters and Γ is the gamma function.

Approaches to Intrusion Detection

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Statistical anomaly detection (Continued...)

- (a) **Threshold detection**: This approach involves defining thresholds, independent of users, for the frequency of occurrence of various events.
- (b) **Profile based**: A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

Approaches to Intrusion Detection

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Rule-Based Detection

- Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.
  - (a) Anomaly detection: Rules are developed to detect deviation from previous usage patterns.
  - (b) Penetration identification: An expert system approach that searches for suspicious behavior.

Approaches to Intrusion Detection

Statistical anomaly detection (Continued...)

- The foundation of the statistical anomaly detection approach is an analysis of audit records.
- Examples of metrics that are useful for profile-based intrusion detection are the following:
  - **Counter:** A non-negative integer that may be incremented but not decremented until it is reset by management action.
    Typically, a count of certain event types is kept over a particular period of time.
    * **Examples:** The number of logins by a single user during an hour, the number of times a given command is executed during a single user session, the number of password failures during a minute.

Approaches to Intrusion Detection

Statistical anomaly detection (Continued...)

- **Gauge:** A non-negative integer that may be incremented or decremented.
  Typically, a gauge is used to measure the current value of some entity.
  * **Examples:** The number of logical connections assigned to a user application,
  the number of outgoing messages queued for a user process.
- **Interval timer:** The length of the time between two events.
  * **Example:** The length of time between successive logins to an account.
- **Resource utilization:** Quantity of resources consumed during a specified period.
  * **Examples:** The number of pages printed during a user session, the total time consumed by a program execution.

Approaches to Intrusion Detection

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Statistical anomaly detection (Continued...)

- Given the above general metrics, various tests can be performed to determine whether current activity fits within acceptable limits. The following approaches that can be taken:
  - **Mean and standard deviation:** This test gives a reflection of the average behavior and its variability.
    The use of mean and standard deviation is applicable to a wide variety of counters, timers, and resource measures.
  - **Multivariate:** This model is based on correlations between two or more variables.
    Intruder behavior may be characterized with greater confidence by considering such correlations (for example, processor time and resource usage, or login frequency and session elapsed time).

Approaches to Intrusion Detection

Statistical anomaly detection (Continued...)

- **Markov process:** This kind of model is used to establish transition probabilities among various states.
  Example: This model might be used at transitions between certain commands.
- **Time series:** This model focuses on time intervals, looking for sequences of events that happen too rapidly or too slowly.
  A variety of statistical tests can be applied to characterize abnormal timing.

Distributed Denial-of-Service (DDoS) Detection

- In a SYN flood attack, the victim sees a dis-appropriate number of SYN packets compared to FIN packets.
  - By a SYN packet, we mean any incoming packet with the SYN flag set.
    Recall that such a packet is a TCP connection request packet.
  - Likewise, FIN and RST packets are, respectively, those with the FIN and RST flags set.
  - A FIN packet is sent by the side that wishes to terminate the TCP connection. If other party agrees to termination, it responds with its own FIN packet. Thus, SYN and FIN packets usually occur in pair.

DDoS Detection (Continued...)

- TCP connections that terminate normally involve one SYN packet (from the client) and a corresponding FIN packet to initiate or confirm termination of a connection.
  The total number of incoming SYN packets should equal the number of incoming FIN packets.
- In the event of a connection being abort, an RST packet takes the place of the FIN.
  In most cases, the number of RST packets is a small fraction of the number of FIN packets, so we ignore them in our analysis.

DDoS Detection (Continued...)

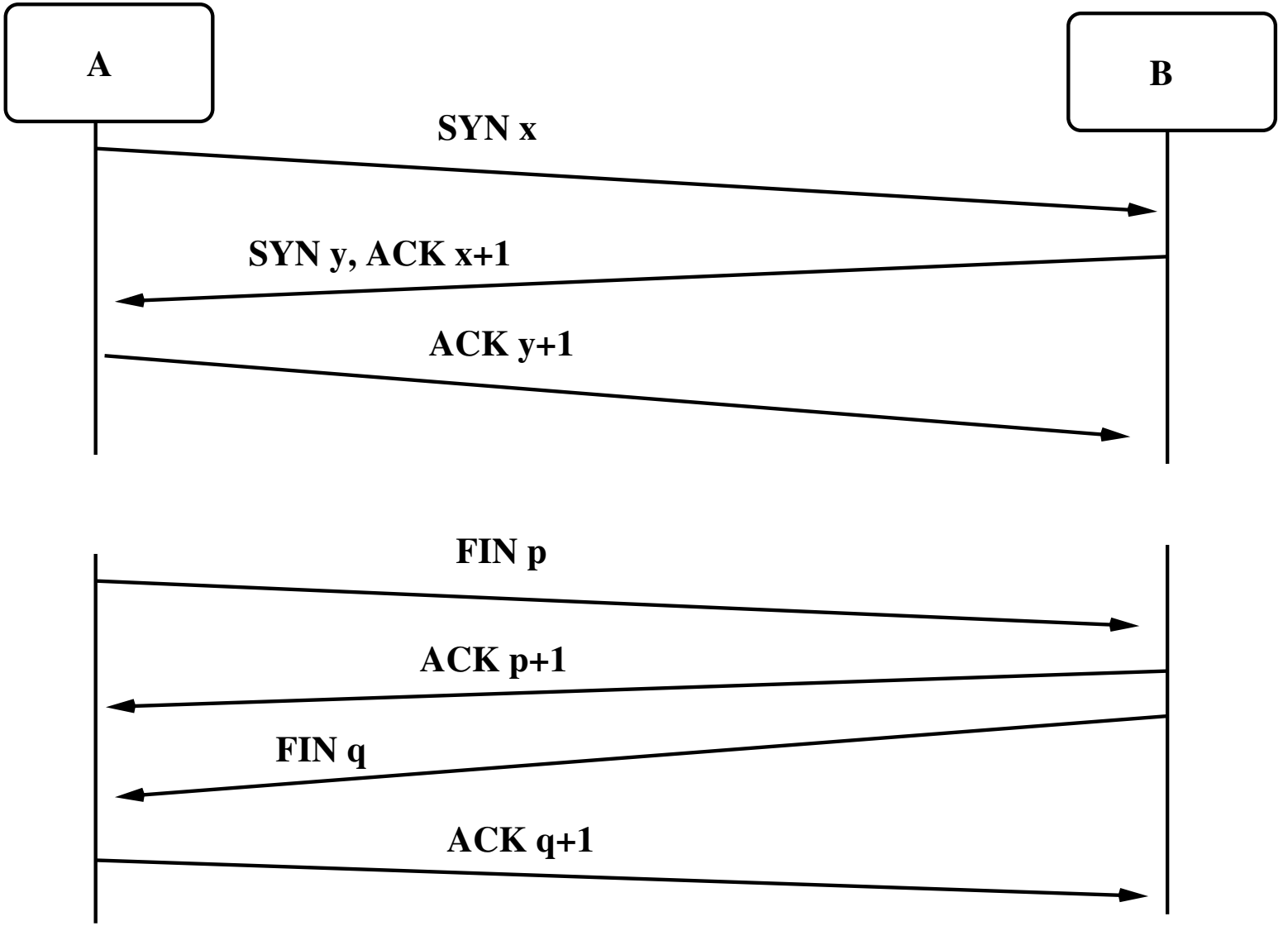


Figure: TCP connection establishment and termination

An anomaly detection system

- **Haining Wang, Danlu Zhang, Kang G. Shin: Change-Point Monitoring for the Detection of DoS Attacks. *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 4, pp.193–208, 2004.**

An anomaly detection system (Continued...)

- Define the following variables:
  - $S_i = \#$ of SYN packets arrivals in the i^{th} observation interval, T_i
  - $F_i = \#$ of FIN packets arrivals in the i^{th} observation interval, T_i'
  - $D_i =$ Normalized difference between $\#$ of SYN and FIN packets in the i^{th} observation interval, that is,

$$D_i = \frac{S_i - F_i}{F_i}.$$

  - $\tau_1, \tau_2, \tau_3 =$ thresholds for detection
- Consider the time series, $D_1, D_2, D_3, \ldots$

An anomaly detection system (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Detection Algorithm 1

- Raise an alert, if the most recently computed detection variable D_i exceeds the threshold τ_1, that is,

$$D_i > \tau_1.$$

- The IDS may raise many alarms since it bases its decision on point values.

An anomaly detection system (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Detection Algorithm 2

- Raise an alert, if the "smoothed average" of the previous values of D exceeds the threshold τ_2.
- This approach uses the well-known technique of exponential smoothing.
- The decision variable at the end of the i^{th} observation interval is the smoothed average, S_i computed using

$$\begin{aligned} S_i &= \alpha D_i + (1-\alpha) S_{i-1}, 0 < \alpha < 1, \\ S_0 &= 0, \end{aligned}$$

where α is known as "smoothing constant".

An anomaly detection system (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Detection Algorithm 2 (Continued...)

- Thus,

$$\begin{aligned} S_i &= \alpha D_i + (1-\alpha) S_{i-1} \\ &= \alpha D_i + (1-\alpha)[\alpha D_{i-1} + (1-\alpha) S_{i-2}] \\ &= \alpha D_i + \alpha(1-\alpha) D_{i-1} + (1-\alpha)^2 S_{i-2} \\ &\vdots \\ &= \alpha D_i + \alpha(1-\alpha) D_{i-1} + \alpha(1-\alpha)^2 D_{i-2} + \ldots \end{aligned}$$

- For example, for $\alpha = 0.40$, we get,

$$S_i = 0.40 D_i + 0.24 D_{i-1} + 0.14 D_{i-2} + \ldots$$

An anomaly detection system (Continued...)

Detection Algorithm 2 (Continued...)

- The decision variable S_i is thus a "weighted sum" of D_i, D_{i-1}, D_{i-2}, ... which degrades weights assigned to earlier values of D.
- Thus, "earlier" values of D count less.
- An alarm will be raised if $S_i > \tau_2$.

An anomaly detection system (Continued...)

Detection Algorithm 2 (Continued...)
Observations

- The value of τ_2 is set based on empirical data.
  - If it is too low, it will result in many "false positives".
  - If it too high, it will result in "false negatives".
- Another design parameter is the "smoothing" constant α.
  - It a value close to 1 is selected, it will give dis-appropriate importance to the most recent value of D_i.
  - If the limiting case of $\alpha = 1$, this algorithm reduces to Detection Algorithm 1.
  - If the close of zero α, the more even are the weights assigned to all values of D_i.

An anomaly detection system (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Detection Algorithm 3

- Define a "modified cumulative sum" of previous values of D.
- Raise an alert, if this value exceeds a threshold τ_3.
- This method makes use of a technique, called the sequential change point detection
  [M. Basseville and I. V. Nikiforov, "Detection of Abrupt Changes: Theory and Application, Pentice Hall, 1993.]

An anomaly detection system (Continued...)

Detection Algorithm 3 (Continued...)

- During normal operation, the number of FIN packets will balance out the number of SYN packets, and hence $D_i = \frac{S_i - F_i}{F_i}$ will be close to 0.
- Let u be an upper bound on the mean of D_i during the normal opetations.
- Let D_i' be a shifted version of D_i, that is,

$$D_i' = D_i - u.$$

- The decision variable M_i used here is defined by

$$\begin{aligned} M_i &= (M_{i-1} + D_i')^+, \\ M_0 &= 0. \end{aligned}$$

- The notation x^+ is defined as follows:

$$x^+ = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases}$$

An anomaly detection system (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Detection Algorithm 3

- The IDS sounds an alert at the end of j^{th} interval, if

$$M_j > \tau_3,$$

  where τ_3 is a threshold determined empirically.
- With Detection Algorithm 3 (cumulative sum method), the false positives and false negatives encountered with Detection Algoritm 1 are both avoided.

Hierarchical Access Control

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
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

Hierarchical Access Control

Hierarchical Access Control

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Overview of Hierarchical Access Control

- Hierarchical access control is a fundamental problem in computer and network systems.
- In a hierarchical access control, a user of higher security level class has the ability to access information items (such as message, data, files, etc.) of other users of lower security classes.
- A user hierarchy consists of a number n of disjoint security classes, say, SC_1, SC_2, ..., SC_n. Let this set be $SC = \{SC_1, SC_2, \ldots, SC_n\}$.
- A binary partially ordered relation $\geq$ is defined in SC as $SC_i \geq SC_j$, which means that the security class SC_i has a security clearance higher than or equal to the security class SC_j.

Hierarchical Access Control

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Overview of Hierarchical Access Control

- In addition the relation $\geq$ satisfies the following properties:
  - **[Reflexive property]** $SC_i \geq SC_i, \forall SC_i \in SC$.
  - **[Anti-symmetric property]** If $SC_i, SC_j \in SC$ such that $SC_i \geq SC_j$ and $SC_j \geq SC_i$, then $SC_i = SC_j$.
  - **[Transitive property]** If $SC_i, SC_j, SC_k \in SC$ such that $SC_i \geq SC_j$ and $SC_j \geq SC_k$, then $SC_i \geq SC_k$.
- If $SC_i \geq SC_j$, we call SC_i as the predecessor of SC_j and SC_j as the successor of SC_i. If $SC_i \geq SC_k \geq SC_j$, then SC_k is an intermediate security class. In this case SC_k is the predecessor of SC_j and SC_i is the predecessor of SC_k.
- In a user hierarchy, the encrypted message by a successor security class is only decrypted by that successor class as well as its all predecessor security classes in that hierarchy.

Overview of Hierarchical Access Control

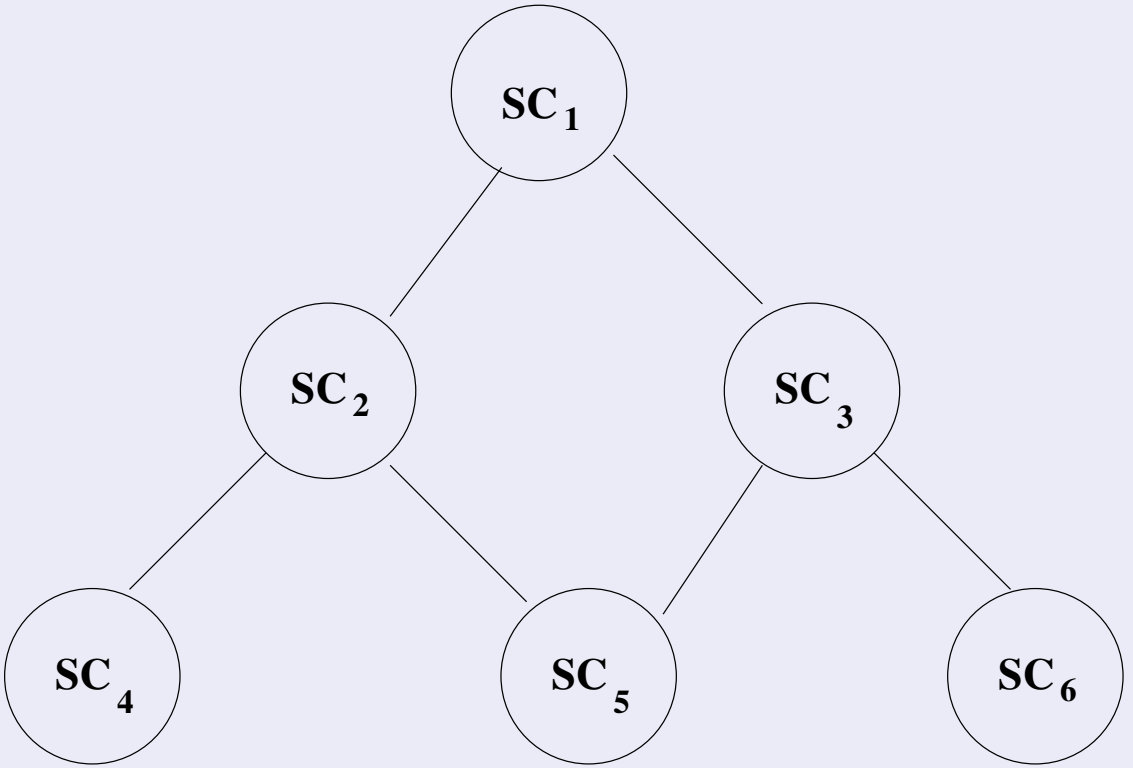


Figure: A small sample of poset in a user hierarchy.

Hierarchical Access Control

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Applications of Hierarchical Access Control

- Military
- Government schools and colleges
- Private corporations
- Computer network systems
- Operating systems
- Database management systems

Chung et al.'s User Hierarchical Access Control Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Reference

- Y. F. Chung, H. H. Lee, F. Lai and T. S. Chen, "Access control in user hierarchy based on elliptic curve cryptosystem", Information Sciences (Elsevier), vol. 178, no. 1, pp. 230-243, 2008 (2018 SCI Impact Factor: 5.524).

Chung et al.'s User Hierarchical Access Control Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Relationship Building Phase

- CA (central authority) builds a hierarchical structure for controlling access according to the relationships among the nodes in the hierarchy.
- Let $U = \{SC_1, SC_2, \ldots, SC_n\}$ be a set of n security classes in the hierarchy. Assume that SC_i is a security class with higher clearance and SC_j a security class with lower clearance, that is, $SC_i \geq SC_j$.
- A legitimate relationship $(SC_i, SC_j) \in R_{i,j}$ between two security classes SC_i and SC_j exists in the hierarchy if SC_i can access SC_j.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Generation Phase

CA performs the following steps:

- **Step 1:** Randomly selects a large prime p.
- **Step 2:** Selects an elliptic curve $E_p(a, b)$ defined over Z_p such that the order of $E_p(a, b)$ lies in the interval $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$.
- **Step 3:** Selects a one-way function $h(\cdot)$ to transform a point into a number and a base point G_j from $E_p(a, b)$ for each security class SC_j $1 \leq j \leq n$.
- **Step 4:** For each security class SC_j $(1 \leq j \leq n)$, selects a secret key sk_j and a sub-secret key s_j.
- **Step 5:** For all $\{SC_i | (SC_i, SC_j)\} \in R_{i,j}$, computes the followings:
  $s_i G_j = (x_{j,i}, y_{j,i})$,
  $h(x_{j,i} || y_{j,i})$, where $||$ is a bit concatenation operator.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Key Generation Phase (Continued...)

- **Step 6:** Finally, computes the public polynomial $f_j(x)$ using the values of $h(x_{j,i}||y_{j,i})$ as

$$f_j(x) = \prod_{SC_i \geq SC_j} (x - h(x_{j,i}||y_{j,i})) + sk_j \pmod{p}$$

- **Step 7:** Sends sk_j and s_j to the security class SC_j via a secret channel.
- **Step 8:** Announces $p, h(\cdot), G_j, f_j(x)$ as public.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Derivation Phase

In order to compute the secret keys sk_j of all successors, SC_j, the predecessor SC_i, for which the relationships $(SC_i, SC_j) \in R_{i,j}$ between SC_i and SC_j hold, proceeds as follows:

- Step 1: For $\{SC_i|(SC_i, SC_j)\} \in R_{i,j}$, computes the followings:
  $s_i G_j = (x_{j,i}, y_{j,i})$,
  $h(x_{j,i}||y_{j,i})$.
- Step 2: Computes the secret key sk_j using $h(x_{j,i}||y_{j,i})$ as follows:

$$f_j(x) = \prod_{SC_i \geq SC_j} (x - h(x_{j,i}||y_{j,i})) + sk_j \pmod{p},$$

$$f_j(h(x_{j,i}||y_{j,i})) = sk_j \pmod{p}.$$

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Derivation Phase (Continued...)

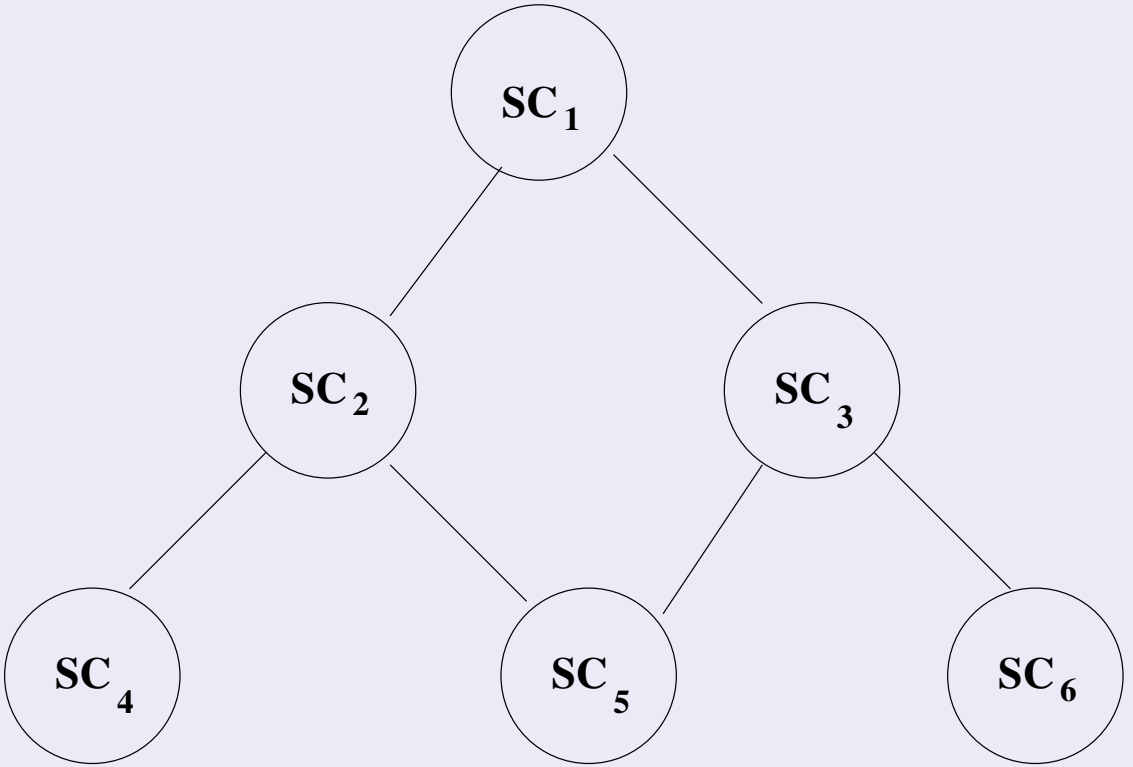


Figure: A small sample of poset in a user hierarchy.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Derivation Phase (Continued...)

$$f_j(x) = \prod_{SC_i \geq SC_j} [x - h(x_{j,i}||y_{j,i})] + sk_j \pmod p,$$

$SC_1 : f_1(x) = [x - h(x_{1,0}||y_{1,0})] + sk_1 \pmod p$, where s_0 is given by CA

$SC_2 : f_2(x) = [x - h(x_{2,1}||y_{2,1})] + sk_2 \pmod p,$

$SC_3 : f_3(x) = [x - h(x_{3,1}||y_{3,1})] + sk_3 \pmod p,$

$SC_4 : f_4(x) = [x - h(x_{4,1}||y_{4,1})][x - h(x_{4,2}||y_{4,2})] + sk_4 \pmod p,$

$SC_5 : f_5(x) = [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})][x - h(x_{5,3}||y_{5,3})] + sk_5 \pmod p,$

$SC_6 : f_6(x) = [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] + sk_6 \pmod p$

Chung et al.'s Scheme (Continued...)

Key Derivation Phase (Continued...)

To derive the secret key sk_5 of SC_5 by its predecessor class SC_2, SC_2 needs to do following:

- Computes $s_2 G_5 = (x_{5,2}, y_{5,2})$ and then $h(x_{5,2}||y_{5,2})$.
- Determines sk_5 using $h(x_{5,2}||y_{5,2})$ from the public polynomial $f_5(x) = [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})][x - h(x_{5,3}||y_{5,3})] + sk_5 \pmod p$ as $sk_5 = f_5(h(x_{5,2}||y_{5,2})) \pmod p$.

Chung et al.'s Scheme (Continued...)

Inserting New Security Classes Phase

If a new security class SC_k is inserted into the hierarchy such that $SC_i \geq SC_k \geq SC_j$, then the relationships $(SC_i, SC_k) \in R_{i,k}$ for $SC_i \geq SC_k$ and $(SC_k, SC_j) \in R_{k,j}$ for $SC_k \geq SC_j$ need to be updated into the hierarchy. CA needs the following steps to manage the accessing priority of SC_k in the hierarchy.

- Step 1: Updates the partial relationships R that follows when the security class SC_k joins the hierarchy.
- Step 2: Randomly selects the secret key sk_k, the sub-secret key s_k and the base point G_k for the class SC_k.
- Step 3: For all $\{SC_i|(SC_i, SC_k)\} \in R_{i,k}$ that satisfies $SC_i \geq SC_k$ when the new class SC_k is inserted in the hierarchy, computes $s_i G_k = (x_{k,i}, y_{k,i})$, and $h(x_{k,i}||y_{k,i})$.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Inserting New Security Classes Phase (continued...)

- Step 4: Computes the public polynomial $f_k(x)$ as follows:

$$f_k(x) = \prod_{SC_i \geq SC_k} (x - h(x_{k,i}||y_{k,i})) + sk_k \pmod{p}$$

- Step 5: For all $\{SC_i|(SC_i, SC_k)\} \in R_{i,k}$ and $\{SC_k|(SC_k, SC_j)\} \in R_{k,j}$ that satisfy $SC_i \geq SC_k \geq SC_j$ when the new class SC_k is inserted in the hierarchy, computes
  $s_k G_j = (x_{j,k}, y_{j,k})$,
  $s_i G_j = (x_{j,i}, y_{j,i})$,
  $h(x_{j,k}||y_{j,k})$ and $h(x_{j,i}||y_{j,i})$.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Inserting New Security Classes Phase (continued...)

- Step 6: Computes the public polynomial $f_j'(x)$ as follows:

$$f_j'(x) = \prod_{SC_i \geq SC_k \geq SC_j} (x - h(x_{j,i}||y_{j,i}))(x - h(x_{j,k}||y_{j,k})) + sk_j \pmod{p}$$

- Step 7: Replaces $f_j(x)$ with $f_j'(x)$, and sends sk_k and s_k to SC_k via a secure channel, and announces publicly G_k, $f_k(x)$ and $f_j'(x)$.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Inserting New Security Classes Phase (continued...)

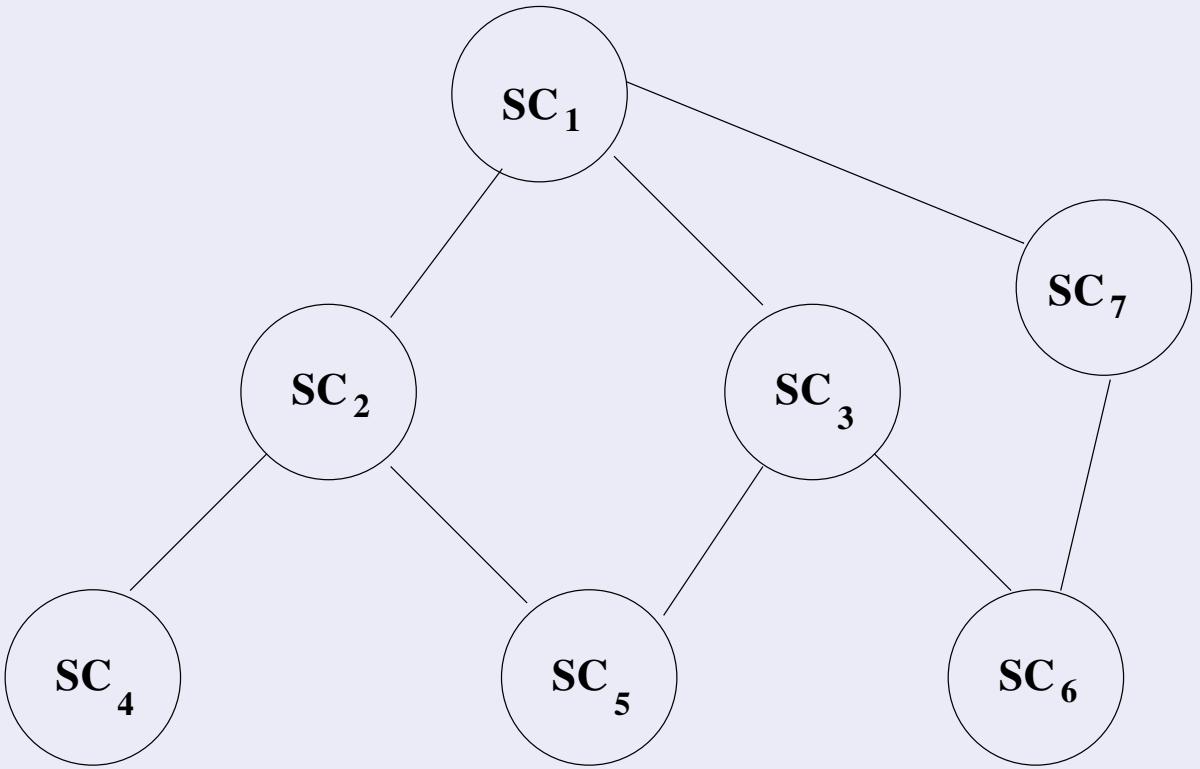


Figure: A small sample of poset in a user hierarchy: when a new security class SC_7 is added into the hierarchy.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Inserting New Security Classes Phase (continued...)

$$
\begin{aligned}
SC_1 : f_1(x) &= [x - h(x_{1,0}||y_{1,0})] + sk_1 \pmod p, \text{ where } s_0 \text{ is given by CA} \\
SC_2 : f_2(x) &= [x - h(x_{2,1}||y_{2,1})] + sk_2 \pmod p, \\
SC_3 : f_3(x) &= [x - h(x_{3,1}||y_{3,1})] + sk_3 \pmod p, \\
SC_4 : f_4(x) &= [x - h(x_{4,1}||y_{4,1})][x - h(x_{4,2}||y_{4,2})] + sk_4 \pmod p, \\
SC_5 : f_5(x) &= [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})][x - h(x_{5,3}||y_{5,3})] \\
&\quad + sk_5 \pmod p, \\
SC_6 : f'_6(x) &= [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})][x - h(x_{6,7}||y_{6,7})] \\
&\quad + sk_6 \pmod p \\
SC_7 : f_7(x) &= [x - h(x_{7,1}||y_{7,1})] + sk_7 \pmod p
\end{aligned}
$$

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Removing Existing Security Classes Phase

If an existing member SC_k, such that the relationship $SC_i \geq SC_k \geq SC_j$ breaks up, wants to leave from a user hierarchy, then CA not only directly revokes information related to SC_k, but also alters the accessing relationship between the involved ex-predecessor SC_i and ex-successor SC_j of SC_k. In this phase, CA executes the following steps.

- Step 1: Updates the partial relationship R that follows when SC_k is removed.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Removing Existing Security Classes Phase (Continued...)

- Step 2: For all $\{SC_k|(SC_k, SC_j)\} \in R_{k,j}$ does the followings:
  - Step 2.1: Renews the secret key sk_j as sk_j' and the base point G_j as G_j' of SC_j.
  - Step 2.2: For all $\{SC_i|(SC_i, SC_j)\} \in R_{i,j}$ does the followings:
    - Step 2.2.1: Renews $\{SC_i|(SC_i, SC_j)\} \in R_{i,j}$ after removing SC_k.
    - Step 2.2.2: Computes $s_i G_j' = (x_{j,i}, y_{j,i})$.
    - Step 2.2.3: Computes $h(x_{j,i}, y_{j,i})$.
    - Step 2.2.4: Computes the public polynomial $f_j'(x)$ as

$$f_j'(x) = \prod_{SC_i \geq SC_j} (x - h(x_{j,i}||y_{j,i})) + sk_j' \pmod{p}$$

    - Step 2.2.5: Replaces $f_j(x)$ with $f_j'(x)$.
- Step 3: Sends sk_j' to SC_j via a secret channel and announces G_j' and $f_j'(x)$ as public.

Chung et al.'s Scheme (Continued...)

Removing Existing Security Classes Phase (continued...)

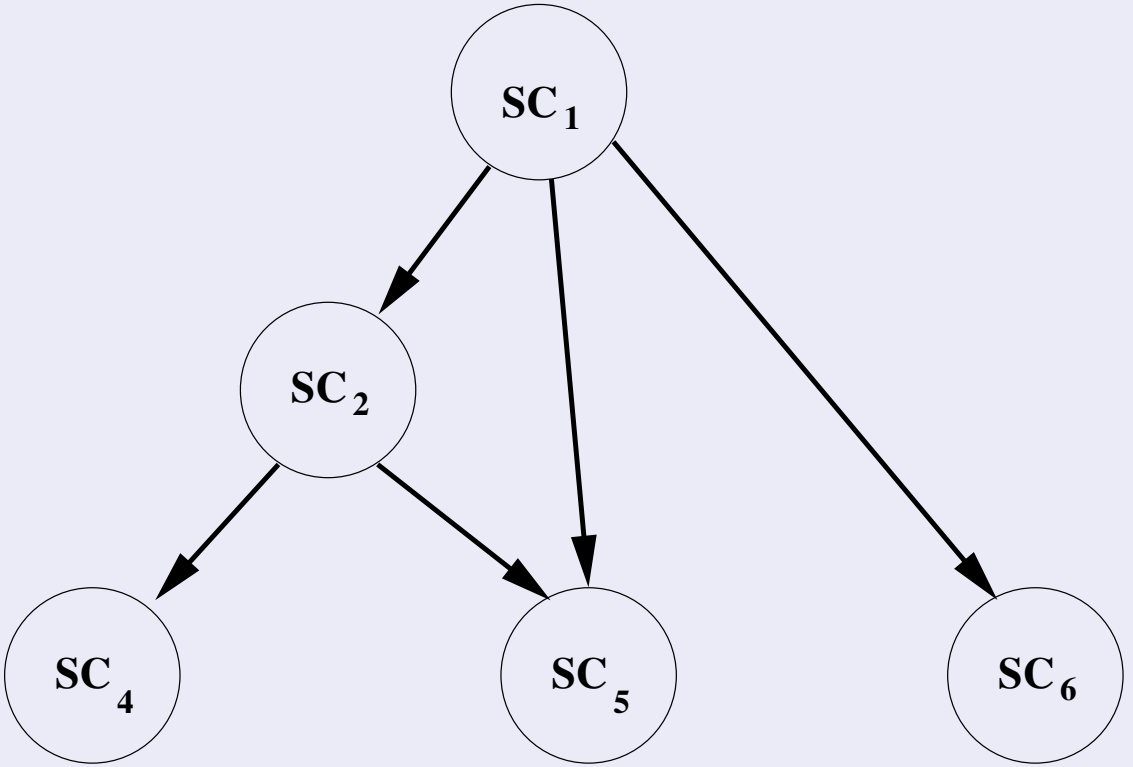


Figure: A small sample of poset in a user hierarchy: when an existing security class SC_3 is removed into the hierarchy.

Chung et al.'s Scheme (Continued...)

Removing Existing Security Classes Phase (continued...)

- Before deleting SC_3, $f_5(x)$ and $f_6(x)$ are formed as
  $f_5(x) = [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})][x - h(x_{5,3}||y_{5,3})] + sk_5 \pmod{p}$
  $f_6(x) = [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] + sk_6 \pmod{p}$.
- After deleting SC_3, $f'_5(x)$ and $f'_6(x)$ are formed as
  $f'_5(x) = [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})] + sk'_5 \pmod{p}$
  $f'_6(x) = [x - h(x_{6,1}||y_{6,1})] + sk'_6 \pmod{p}$.

Chung et al.'s Scheme (Continued...)

Creating New Relationships

- Suppose we want to create a new relationship between SC_5 and SC_6 in the hierarchy (Figure 1) such that $SC_2 \geq SC_5 \geq SC_6$.

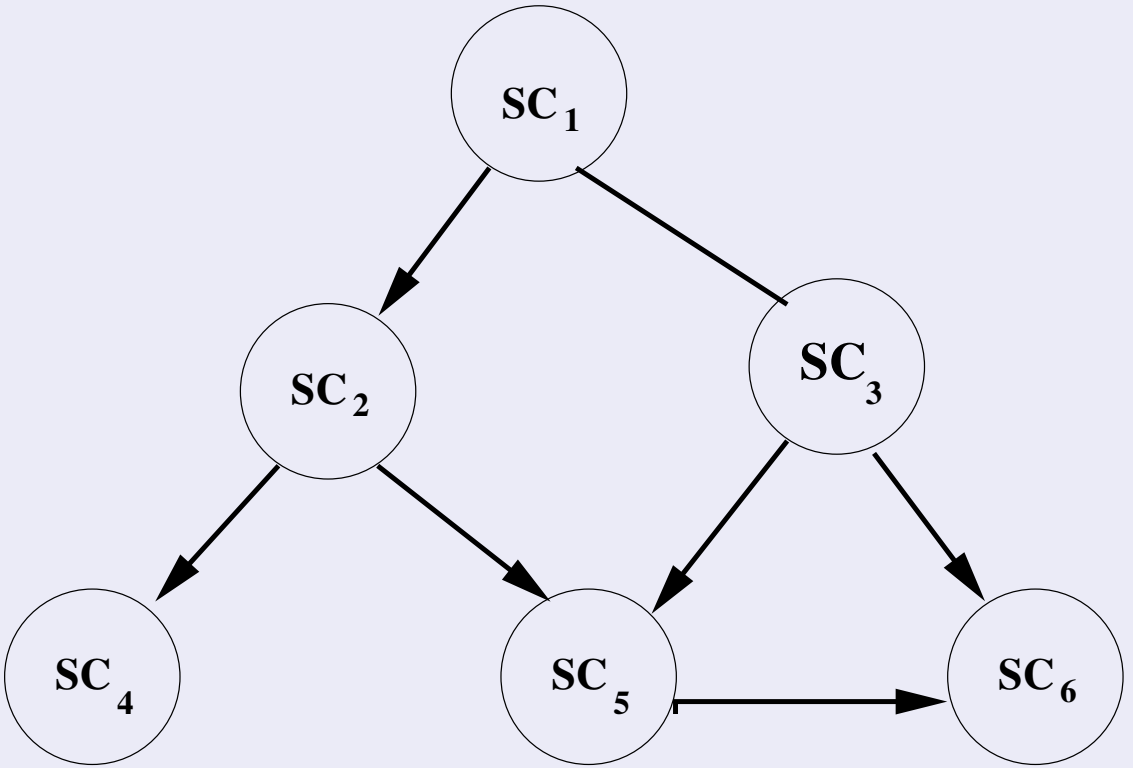


Figure: The consequent poset after creating $SC_5 \geq SC_6$ in Figure 1.

Chung et al.'s Scheme (Continued...)

Creating New Relationships

- Before creating the relationship $SC_2 \geq SC_5 \geq SC_6$, $f_6(x)$ is formed as follows:

$$f_6(x) = [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] + sk_6 \pmod{p}.$$

- After creating the relationship $SC_2 \geq SC_5 \geq SC_6$, updated public polynomial $f_6'(x)$ is formed as follows:

$$\begin{aligned} f_6'(x) = {} & [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] \\ & [x - h(x_{6,2}||y_{6,2})][x - h(x_{6,5}||y_{6,5})] + sk_6 \pmod{p}. \end{aligned}$$

Chung et al.'s Scheme (Continued...)

Revoking Existing Relationships

- Suppose we want to revoke the existing relationship $\{SC_2|(SC_2, SC_5) \in R_{2,5}\}$ in the following figure such that $\{SC_2|(SC_2, SC_5) \notin R_{2,5}\}$

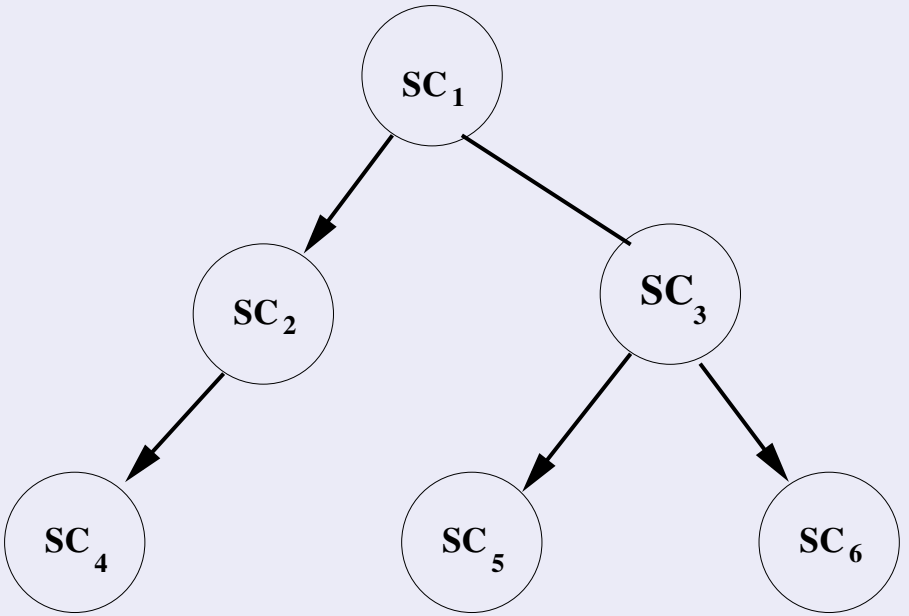


Figure: The consequent poset after revoking $SC_2 \geq SC_5$ in Figure 1.

Chung et al.'s Scheme (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Revoking Existing Relationships

- Before revoking $\{SC_2|(SC_2, SC_5) \in R_{2,5}\}$, $f_5(x)$ is formed as follows:

$$\begin{aligned} f_5(x) &= [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})] \\ &\quad [x - h(x_{5,3}||y_{5,3})] + sk_5 \pmod{p}. \end{aligned}$$

- After revoking $\{SC_2|(SC_2, SC_5) \in R_{2,5}\}$, $f_5(x)$ is replaced with the updated $f_5'(x)$ as follows:

$$f_5'(x) = [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,3}||y_{5,3})] + sk_5' \pmod{p}.$$

after renewing the secret key sk_5' in place of sk_5.

Chung et al.'s Scheme (Continued...)

Changing Secret Keys

- A secret key must be changeable to maximize security.
- To change a secret key sk_j to sk_j', CA must replace the base point G_j with G_j' and the public polynomial $f_j(x)$ with $f_j'(x)$ as follows.
  - Step 1: Replace the secret key sk_j with sk_j' and the base point G_j with G_j'.
  - Step 2: For all $\{SC_i|(SC_i, SC_j)\} \in R_{i,j}$:
    - Step 2.1: Determine $s_i G_j' = (x_{j,i}, y_{j,i})$
    - Step 2.2: Determine $h(x_{j,i}||y_{j,i})$, where $||$ is a bit concatenation operator

Chung et al.'s Scheme (Continued...)

Changing Secret Keys (Continued...)

- Step 3: Determine the public polynomial $f_j'(x)$ as follows

$$f_j'(x) = \prod_{SC_i \geq SC_j} [x - h(x_{j,i} || y_{j,i})] + sk_j' \pmod{p}$$

- Step 4: Replace $f_j(x)$ with $f_j'(x)$
- Step 5: Send sk_j' to SC_j via a secret channel, and announce G_j' and $f_j'(x)$

Cryptanalysis and Improvement of Chung et al.'s Scheme

- **Ashok Kumar Das, Nayan Ranjan Paul, and Laxminath Tripathy. "Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem," in *Information Sciences (Elsevier)*, Vol. 209, No. C, pp. 80 - 92, 2012, doi: http://dx .doi.org/10.1016/ j.ins.2012.04.036. (2018 SCI Impact Factor: 5.524)**

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Root finding of a polynomial in finite field $GF(q)$

- Suppose we are given a polynomial $f(x) \in GF(q)[x]$ of degree n ($deg.f = n$), in $GF(q)$ and we want to find all the roots $a \in GF(q)$ of $f(x) = 0$. Let q be odd.
- M. Ben-or showed that the expected number of operations used by his proposed algorithm for finding all roots of $f(x) = 0$ is $O(n.\log n.L(n).\log q)$, where $L(n) = \log n.\log\log n$. In other words, the expected running time of the algorithm is polynomial in n.

Reference: M. Ben-Or, Probabilistic algorithms in finite fields, in: Proceedings of 22nd Annual Symposium on Foundations of Computer Science (IEEE FOCS'81), 1981, pp. 394-398.

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Root finding of a polynomial in finite field $GF(q)$

Algorithm: Finding roots of $f_1(x) \in GF(q)[x] : ROOTS(f_1(x))$

if $(deg.f_1(x) = 1$, i.e., $f_1(x) = (x - a))$ **then**
  return a;
end if
repeat
  Choose $\delta \in GF(q)$ randomly;
  Compute $f_2(x) = \gcd[f_1(x), (x + \delta)^{(q-1)/2} - 1]$;
until $(0 < deg.f_2 < deg.f_1)$
return $ROOTS(f_2(x)) \cup ROOTS(f_1(x)/f_2(x))$;

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Factorization of a polynomial in finite field $GF(q)$

- Let we want to factor a polynomial $f(x) \in GF(q)[x]$ of degree n into its irreducible factors. Assume that $f(x)$ has no repeated factors and we want to factor $f(x)$ as $f(x) = g_1(x).g_2(x)\ldots g_n(x)$, where $g_d(x)$ is the product of irreducible factors of $f(x)$ of degree d.
- M. Ben-or showed that the expected number of operations to factor an n-degree polynomial is $O(n^2.L(n).\log q)$. That is, the expected running time of the algorithm is also polynomial in n.

Reference: M. Ben-Or, Probabilistic algorithms in finite fields, in: Proceedings of 22nd Annual Symposium on Foundations of Computer Science (IEEE FOCS'81), 1981, pp. 394-398.

Cryptanalysis and Improvement of Chung et al.'s Scheme

Factorization of a polynomial in finite field $GF(q)$

Algorithm: Factorization

$r_0(x) = x$;
$f_0(x) = f(x)$;
for $d = 1 \rightarrow n$ **do**
  $r_d(x) = (r_{d-1}(x))^q \pmod{f_{d-1}(x)}$, $deg.r_d < deg.f_{d-1}$;
  $g_d(x) = \gcd(f_{d-1}(x), r_d(x) - x)$;
  $f_d(x) = f_{d-1}(x)/g_d(x)$;
end for
return $g_1(x), g_2(x), \ldots, g_n(x)$;

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

The exterior root finding attack on Chung et al.'s scheme

- Consider the case when a security class SC_k is inserted in the user hierarchy with the relationship $SC_i \geq SC_k \geq SC_j$.
- Thus, when a new security class SC_k is added as a predecessor of SC_j, CA updates the public polynomial of SC_j by replacing $f_j(x)$ by $f_j'(x)$.
- However, for those predecessors, which remain as predecessors of SC_j in $f_j'(x)$, their secrets are still at the same positions of $f_j'(x)$.
- Now, knowing the public polynomial $f_j(x)$ of SC_j before adding the security class SC_k and the public polynomial $f_j'(x)$ of SC_j after adding the security class SC_k until the secret key sk_j of SC_j has been changed by CA, an attacker can generate a polynomial by taking the difference of $f_j(x)$ and $f_j'(x)$.
- Let this difference polynomial be denoted by $\phi(x) = f_j(x) - f_j'(x)$.

Cryptanalysis and Improvement of Chung et al.'s Scheme

Exterior root finding attack on Chung et al.'s scheme

- It is noted that

$$
\begin{aligned}
\phi(x) &= f_j(x) - f'_j(x) \\
&= \Big(\prod_{SC_i > SC_j} [x - h(x_{j,i}||y_{j,i})] + sk_j \pmod p\Big) - \\
&\quad \Big(\prod_{SC_i > SC_k > SC_j} [x - h(x_{j,i}||y_{j,i})][x - h(x_{j,k}||y_{j,k})] \\
&\quad + sk_j \pmod p\Big) \\
&= \prod_{SC_i > SC_j} [x - h(x_{j,i}||y_{j,i})] - \\
&\quad \prod_{SC_i > SC_k > SC_j} [x - h(x_{j,i}||y_{j,i})][x - h(x_{j,k}||y_{j,k})] \pmod p
\end{aligned}
$$

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Exterior root finding attack on Chung et al.'s scheme

- Further, we observe that the constructed polynomial $\phi(x)$ has common factors $(x - h(x_{j,i}||y_{j,i}))$. Then the attacker finds the roots of the equation $\phi(x) = f_j(x) - f'_j(x) = 0$ in a polynomial time using M. Ben-or's method.
- With the knowledge of the roots, the attacker can easily derive the secret key sk_j of the security class SC_j.
- The attacker, who is not a user in hierarchy of security classes, first obtains the roots $h(x_{j,i}||y_{j,i})$ and then computes the secret key sk_j of SC_j as $sk_j = f_j(h(x_{j,i}||y_{j,i})) = f'_j(h(x_{j,i}||y_{j,i})) \pmod{p}$.

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Example [Exterior root finding attack on Chung et al.'s scheme]

- In Figure 1, the user hierarchy consists of six security classes, denoted by $U = \{SC_1, SC_2, SC_3, SC_4, SC_5, SC_6\}$.

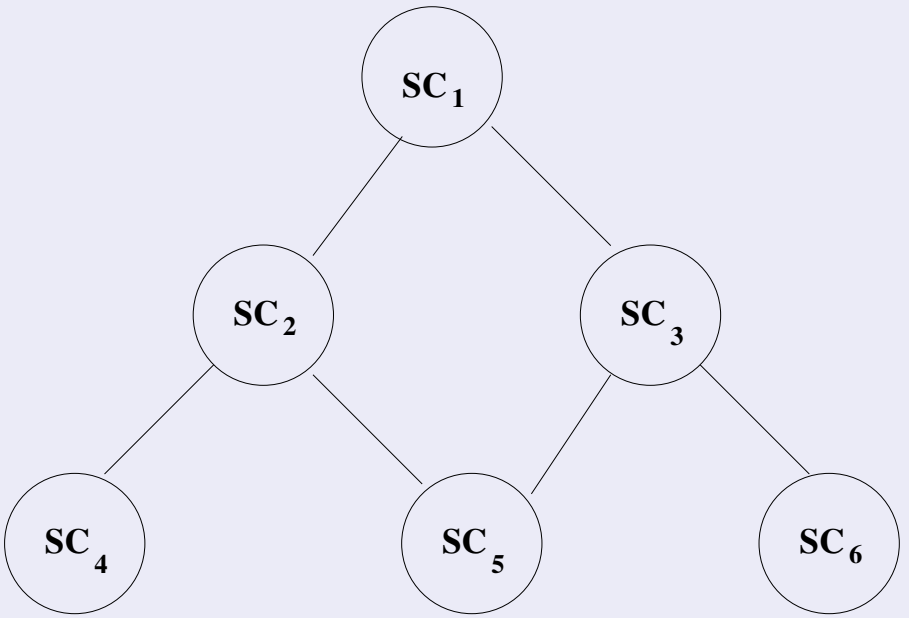


Figure: A small sample of poset in a user hierarchy.

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Example [Exterior root finding attack on Chung et al.'s scheme]

- CA computes the public elliptic curve polynomial $f_j(x)$ for each SC_j. Each SC_i derives the secret keys of its successors SC_j:

$$
\begin{aligned}
SC_1 : f_1(x) &= [x - h(x_{1,0}||y_{1,0})] + sk_1 \pmod p, \text{ where } s_0 \text{ is given by CA} \\
SC_2 : f_2(x) &= [x - h(x_{2,1}||y_{2,1})] + sk_2 \pmod p, \\
SC_3 : f_3(x) &= [x - h(x_{3,1}||y_{3,1})] + sk_3 \pmod p, \\
SC_4 : f_4(x) &= [x - h(x_{4,1}||y_{4,1})][x - h(x_{4,2}||y_{4,2})] + sk_4 \pmod p, \\
SC_5 : f_5(x) &= [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})][x - h(x_{5,3}||y_{5,3})] \\
&\quad + sk_5 \pmod p, \\
SC_6 : f_6(x) &= [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] + sk_6 \pmod p
\end{aligned}
$$

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Example [Exterior root finding attack on Chung et al.'s scheme]

- Suppose a new security class SC_7 is inserted into the existing user hierarchy shown in Figure 1, with the relationship $SC_1 \geq SC_7 \geq SC_6$. The resulting user hierarchy is shown below.

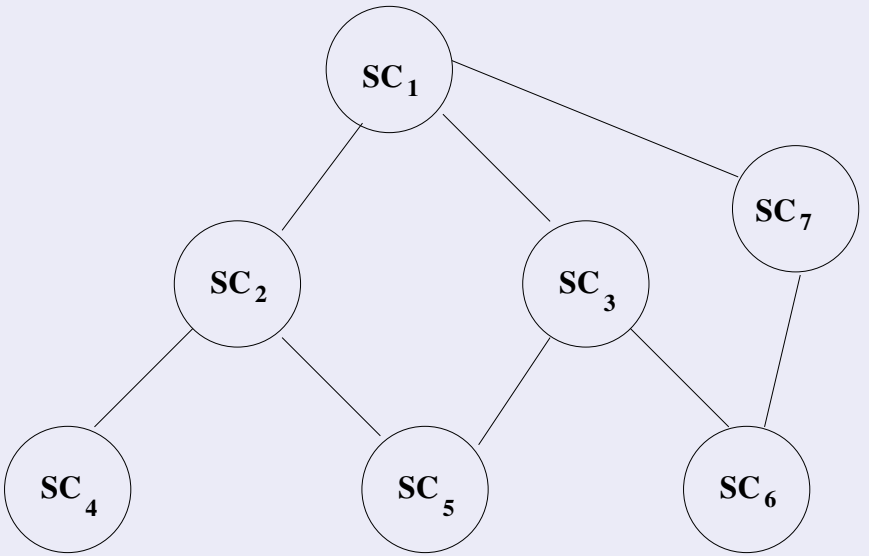


Figure: A small sample of poset in a user hierarchy: when a new security class SC_7 is added into the hierarchy.

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Example [Exterior root finding attack on Chung et al.'s scheme]

- Due to insertion of SC_7, CA needs to select randomly sk_7, s_7 and G_7. Since SC_7 is a successor of SC_1 and a predecessor of SC_6, CA constructs the public polynomial $f_7(x)$ and replaces the public polynomial $f_6(x)$ with $f_6'(x)$.
- Before joining the security class SC_7 into the hierarchy, the public elliptic curve polynomial for security class SC_6 was

$$f_6(x) = [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] + sk_6 \pmod p \quad (1)$$

- After joining the security class SC_7, the public polynomial $f_6'(x)$ for SC_6 and $f_7(x)$ for SC_7 are formed as follows:

$$\begin{aligned} f_6'(x) &= [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] \\ &\quad [x - h(x_{6,7}||y_{6,7})] + sk_6 \pmod p \quad (2) \\ f_7(x) &= [x - h(x_{7,1}||y_{7,1})] + sk_7 \pmod p \quad (3) \end{aligned}$$

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Example [Exterior root finding attack on Chung et al.'s scheme]

- Knowing the public polynomials $f_6(x)$ and $f_6'(x)$ in Equations (4) and (5), an attacker finds the roots of the equation

$$\begin{aligned} \phi(x) &= f_6(x) - f_6'(x) = 0 \\ \Rightarrow & [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] \\ & [1 - (x - h(x_{6,7}||y_{6,7}))] = 0 \,(\text{mod}\, p) \end{aligned} \quad (4)$$

- The attacker obtains the roots as $x = h(x_{6,1}||y_{6,1})$, $h(x_{6,3}||y_{6,3})$ and $1 + h(x_{6,7}||y_{6,7})$. Out of these roots, $h(x_{6,1}||y_{6,1})$ and $h(x_{6,3}||y_{6,3})$ satisfy both Equations (4) and (5).

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Example [Exterior root finding attack on Chung et al.'s scheme]

- Thus, knowing these values, the attacker easily computes the secret key sk_6 of security class SC_6 as

$$\begin{aligned} sk_6 &= f_6(h(x_{6,1}||y_{6,1}))\,(\text{mod } p) \\ &= f'_6(h(x_{6,1}||y_{6,1}))\,(\text{mod } p) \\ &= f_6(h(x_{6,3}||y_{6,3}))\,(\text{mod } p) \\ &= f'_6(h(x_{6,3}||y_{6,3}))\,(\text{mod } p). \end{aligned}$$

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Improvement of Chung et al.'s Scheme

- Improvement over Chung et al.'s scheme is provided in the inserting new security classes phase in order to withstand the exterior root finding attack on Chung et al.'s scheme.
- For more details, see the paper:
  Ashok Kumar Das, Nayan Ranjan Paul, and Laxminath Tripathy. "Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem," in *Information Sciences (Elsevier)*, Vol. 209, No. C, pp. 80 - 92, 2012, doi: http://dx .doi.org/10.1016/ j.ins.2012.04.036. (2018 SCI Impact Factor: 5.524)

Cryptanalysis and Improvement of Chung et al.'s Scheme

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Improvement of Chung et al.'s Scheme

Inserting new security classes: If a new security class SC_k is inserted into the hierarchy such that $SC_i \geq SC_k \geq SC_j$, then the relationships $(SC_i, SC_k) \in R_{i,k}$ for $SC_i \geq SC_k$ and $(SC_k, SC_j) \in R_{k,j}$ for $SC_k \geq SC_j$ need to be updated into the hierarchy. In this phase, CA renews the secret keys sk_j of successors SC_j of the newly added security class SC_k. Moreover, CA must change the public base points G_j by G'_j and the public elliptic curve polynomials $f_j(x)$ with $f'_j(x)$ of SC_j. CA needs the following steps:

- Step 1: Updates the partial relationships R that follow when the security class SC_k joins the hierarchy.
- Step 2: Randomly selects the secret key sk_k, the sub-secret key s_k and the base point G_k for the class SC_k.
- Step 3: For all $\{SC_i | (SC_i, SC_k) \in R_{i,k}\}$ that satisfies $SC_i \geq SC_k$ when the new class SC_k is inserted in the hierarchy, computes $s_i G_k = (x_{k,i}, y_{k,i})$, and $h(x_{k,i} || y_{k,i})$.

Cryptanalysis and Improvement of Chung et al.'s Scheme

Improvement of Chung et al.'s Scheme

- Step 4: Computes the public polynomial $f_k(x)$ as follows:

$$f_k(x) = \prod_{SC_i > SC_k} (x - h(x_{k,i}||y_{k,i})) + sk_k \pmod{p}$$

- Step 5: For all $\{SC_i|(SC_i, SC_k) \in R_{i,k}\}$ and $\{SC_j|(SC_k, SC_j) \in R_{k,j}\}$ that satisfy $SC_i \geq SC_k \geq SC_j$ when the new class SC_k is inserted in the hierarchy:
  Replaces the secret key sk_j with sk'_j and the base point G_j with G'_j of the successor security class SC_j of SC_k.
  Computes $s_k G'_j = (x'_{j,k}, y'_{j,k})$.
  Computes $s_i G'_j = (x'_{j,i}, y'_{j,i})$.
  Computes $h(x'_{j,k}||y'_{j,k})$ and $h(x'_{j,i}||y'_{j,i})$ using the one-way function $h(\cdot)$.

Cryptanalysis and Improvement of Chung et al.'s Scheme

Improvement of Chung et al.'s Scheme

- Step 6: Computes the public polynomial $f_j'(x)$ as follows:

$$f_j'(x) = \prod_{SC_i > SC_k > SC_j} (x - h(x'_{j,i} || y'_{j,i}))(x - h(x'_{j,k} || y'_{j,k})) + sk'_j \pmod{p}$$

- Step 7: Replaces $f_j(x)$ with $f_j'(x)$, and sends sk'_j to SC_j via a secure channel, and announces publicly G'_j and $f_j'(x)$.
- Step 8: Sends sk_k and s_k to SC_k via a secure channel, and announces publicly G_k and $f_k(x)$.

Cryptanalysis and Improvement of Chung et al.'s Scheme

Key derivation: For the relationship $(SC_i, SC_j) \in R_{i,j}$ between two security classes SC_i and SC_j, if the predecessor SC_i wants to compute the updated secret key sk_j' of its successor SC_j, then SC_i needs to proceed the following steps:

- Step 1: For $\{SC_i | (SC_i, SC_j) \in R_{i,j}\}$, computes $s_i G_j' = (x_{j,i}', y_{j,i}')$ and $h(x_{j,i}' || y_{j,i}')$.
- Step 2: Computes the secret key sk_j' using the computed hash value $h(x_{j,i}' || y_{j,i}')$ as follows:
  SC_i has the updated public elliptic curve polynomial for SC_j as

$$f_j'(x) = \prod_{SC_i > SC_k > SC_j} (x - h(x_{j,i}' || y_{j,i}'))(x - h(x_{j,k}' || y_{j,k}')) + sk_j' \pmod p.$$

Cryptanalysis and Improvement of Chung et al.'s Scheme

Cryptanalysis and Improvement of Chung et al.'s Scheme

- Step 3: Computes

$$sk'_j = f'_j(h(x'_{j,i}||y'_{j,i}))\,(\text{mod } p).$$

Important References

- N. Koblitz, “Elliptic Curve Cryptosystems”, Mathematics of Computation, vol. 48, pp. 203-209, 1987.
- V. Miller, “Uses of elliptic curves in cryptography”, in Proceedings of Advances in Cryptology (CRYPTO’85), LNCS 218, pages 417-426, 1986, Springer-Verlag.
- Y. F. Chung, H. H. Lee, F. Lai and T. S. Chen, “Access control in user hierarchy based on elliptic curve cryptosystem”, Information Sciences (Elsevier), vol. 178, no. 1, pp. 230-243, 2008. (2018 SCI Impact Factor: 5.524)
- Ashok Kumar Das, Nayan Ranjan Paul, and L. Tripathy. “Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem,” in Information Sciences (Elsevier), Vol. 209, No. C, pp. 80 - 92, 2012. (2018 SCI Impact Factor: 5.524)
- Vanga Odelu, Ashok Kumar Das, and A. Goswami. “A secure effective key management scheme for dynamic access control in a large leaf class hierarchy,” in Information Sciences (Elsevier), Vol. 269, No. C, pp. 270-285, 2014. (2018 SCI Impact Factor: 5.524)

Important References

- S.G. Akl, P.D. Taylor, Cryptographic solution to a problem of access control in a hierarchy, ACM Transactions on Computer Systems 1 (3) (1983) 239-248.
- S. Mackinnon, P. Taylor, H. Meijer, S. Akl, An optimal algorithm for assigning cryptographic keys to control access in a hierarchy, IEEE Transactions on Computers 34 (9) (1985) 797-802.
- Y.-L. Lin, C.-L. Hsu, Secure key management scheme for dynamic hierarchical access control based on ECC, Journal of Systems and Software 84 (4) (2011) 679-685.
- J.-W. Lo, M.-S. Hwang, C.-H. Liu, An efficient key assignment scheme for access control in a large leaf class hierarchy, Information Sciences 181 (2011) 917-925.
- M. Nikooghadam, A. Zakerolhosseini, Secure Communication of Medical Information sing Mobile Agents, Journal of Medical Systems 36 (6) (2012) 3839-3850.

Important References

- M. Nikooghadam, A. Zakerolhosseini, M.E. Moghaddam, Efficient utilization of elliptic curve cryptosystem for hierarchical access control, Journal of Systems and Software 83 (10) (2010) 1917-1929.
- R.S. Sandhu, Cryptographic implementation of a tree hierarchy for access control, Information Processing Letters 27 (2) (1988) 95-98.
- Vanga Odelu, Ashok Kumar Das, and Adrijit Goswami. “An Effective and Secure Key-Management Scheme for Hierarchical Access Control in E-Medicine System,” in Journal of Medical Systems (Springer), Vol. 37, No. 2, pp. 1 - 18, 2013. (2016 SCI Impact Factor: 2.456)

Thank you!

Wireless Sensor Network Security

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/

Overview of Wireless Sensor Networks

Overview of Wireless Sensor Networks

- In a sensor network, many tiny computing nodes called sensors are scattered in an area for the purpose of sensing some data and transmitting data to nearby *base stations* for further processing.
- A sensor node, also known as a *mote*, is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. The transmission between the sensors is done by short range radio communication.
- The base station is assumed to be computationally well-equipped whereas the sensor nodes are resource-starved.
- The sensor nodes are usually scattered either randomly or manually in a *sensor field* (i.e., deployment area or target field).
- Data are routed back to the base station by a multi-hop infrastructure-less architecture through sensor nodes.

Overview of Wireless Sensor Networks

Ad hoc Networks

- An ad hoc network is a group of mobile, wireless hosts which co-operatively and spontaneously form a network independently of any fixed infrastructure or centralized administration.
- In particular, an ad hoc network has no base stations: a host, also called node, communicates directly with nodes within its wireless range and indirectly with all other destinations using a multi-hop route through other nodes in the network.

Overview of Wireless Sensor Networks

Differences between sensor networks and ad hoc networks

- The number of nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad hoc network.
- Sensor nodes are densely deployed.
- Sensor nodes are prone to failures.
- The topology of a sensor network may change frequently.
- Sensor nodes are limited in power, computation capacities as well as memory.
- Sensor nodes mainly use broadcast communication paradigm whereas most ad hoc networks are based on point-to-point communications.

Overview of Wireless Sensor Networks

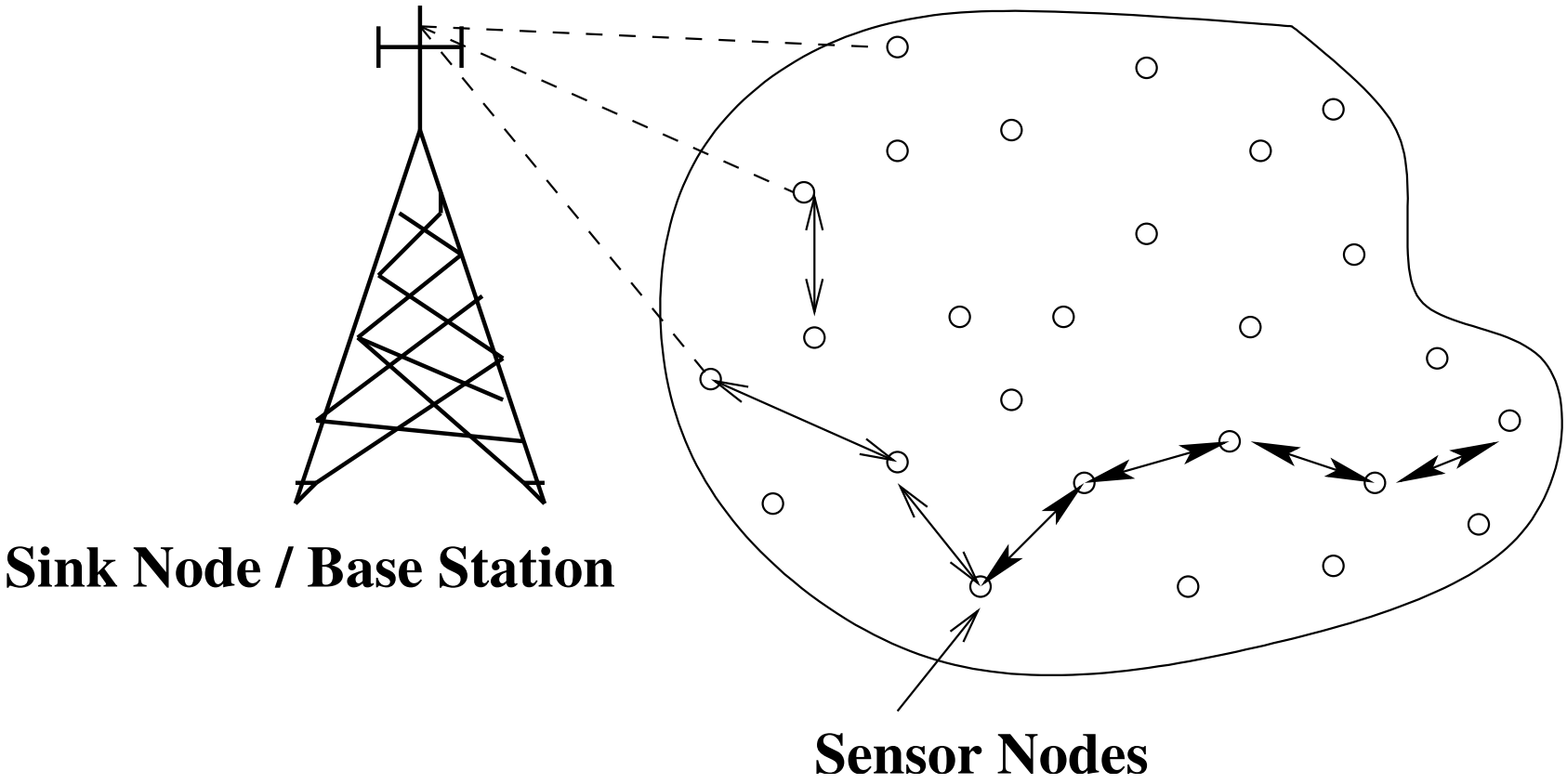


Figure: A distributed wireless sensor network (DWSN)/homogeneous architecture.

Overview of Wireless Sensor Networks

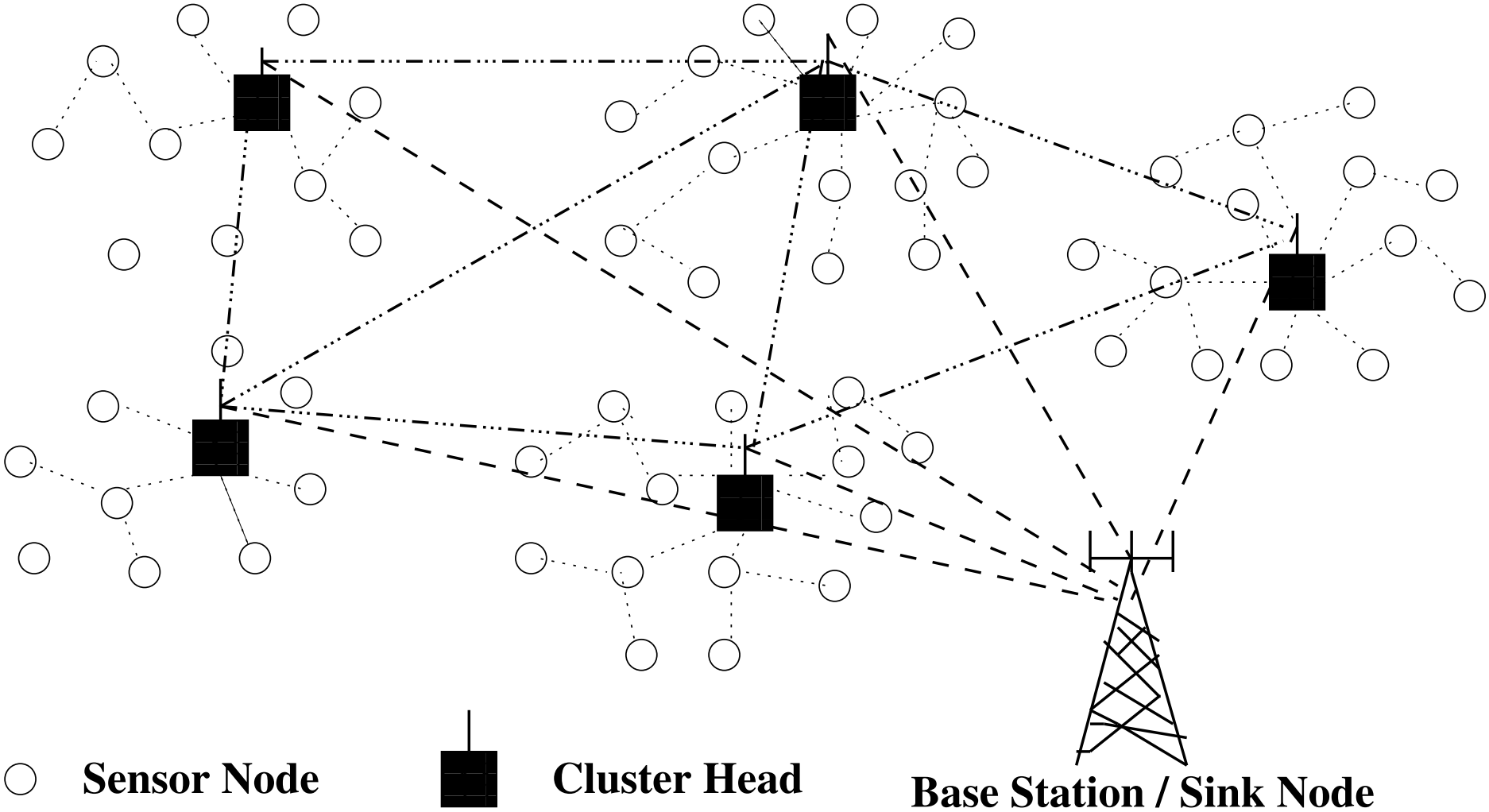


Figure: A hierarchical wireless sensor network (HWSN)/heterogenous architecture.

Overview of Wireless Sensor Networks

Hardware constraints

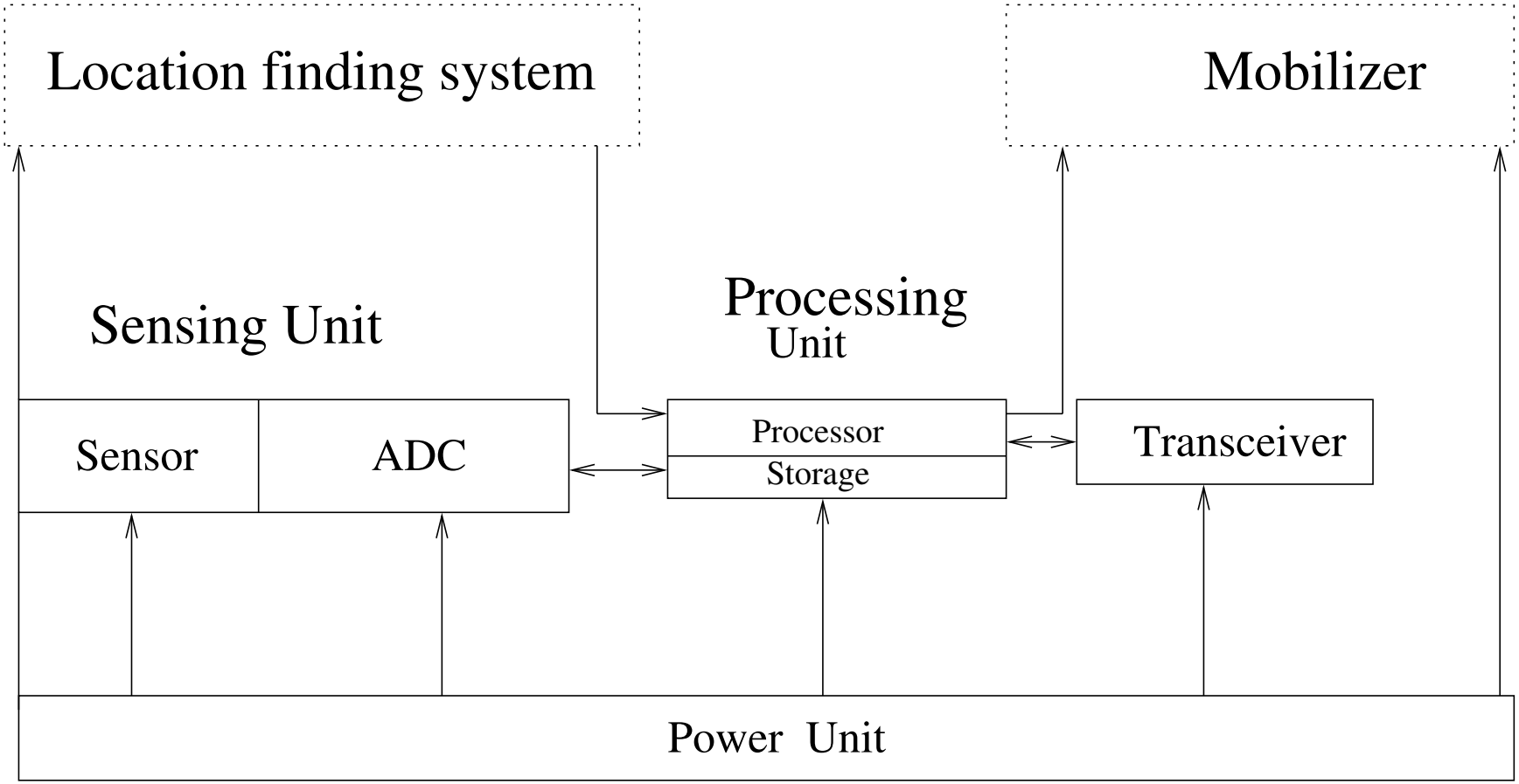


Figure: The components of a sensor node.

Overview of Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Table: Basic characteristics of typical MICA2 and MICA2-DOT motes

	MICA2	MICA2-DOT
Processor	8-bit 7.7 MHz Atmega 128	8-bit 4 MHz Atmega 128
RAM	4K bytes	4K bytes
ROM	128K bytes	128K bytes
EEPROM	512K bytes	512K bytes
Data rate	38.4K baud	38.4K baud
Default packet size (under TinyOS)	29 bytes	29 bytes
Power supply	2 AA batteries	1 coin cell battery

Overview of Wireless Sensor Networks

Sensor network topology

- ***Pre-deployment and deployment phase:*** Sensor nodes can be deployed in mass or placed one by one in the sensor field (target field).
- ***Post-deployment phase:*** The topology of a sensor network can change after deployment due to sensor nodes' available energy, mobility of nodes, etc.
- ***Redeployment of additional nodes phase:*** Additional sensor nodes can be redeployed at any time to replace the faulty or compromised sensor nodes.

Overview of Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Applications of sensor networks

- Military applications
- Environmental monitoring
- Classroom/home
- Health monitoring
- Habitat monitoring
- Detecting and monitoring car thefts
- Vehicle tracking and detection
- Practical application of WSNs (Vehicular Ad Hoc Networks, VANETs)

Overview of Wireless Sensor Networks

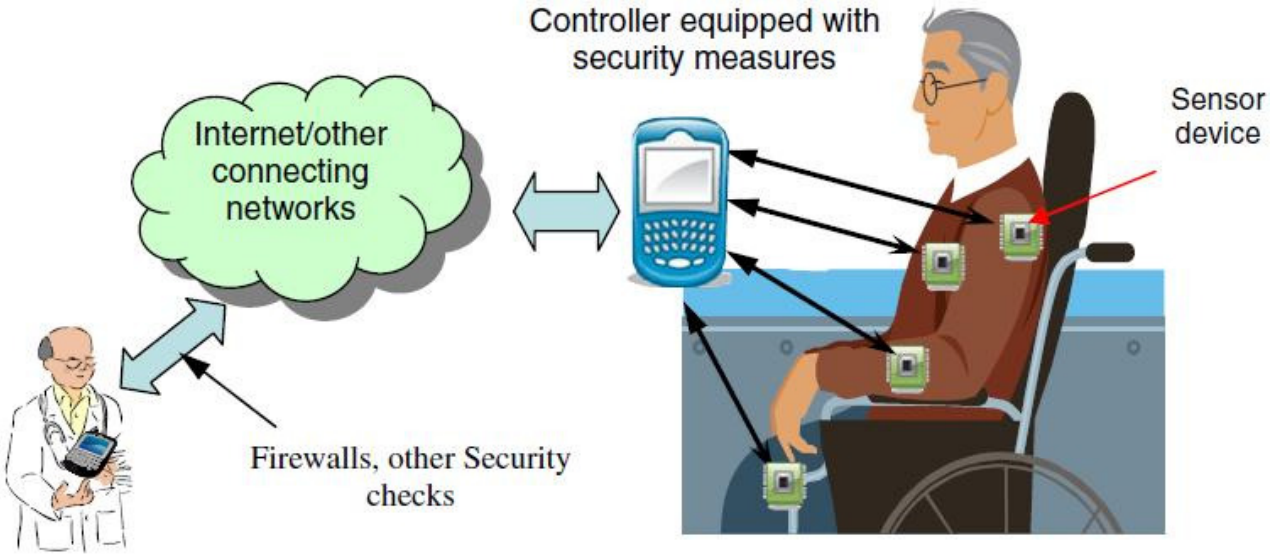
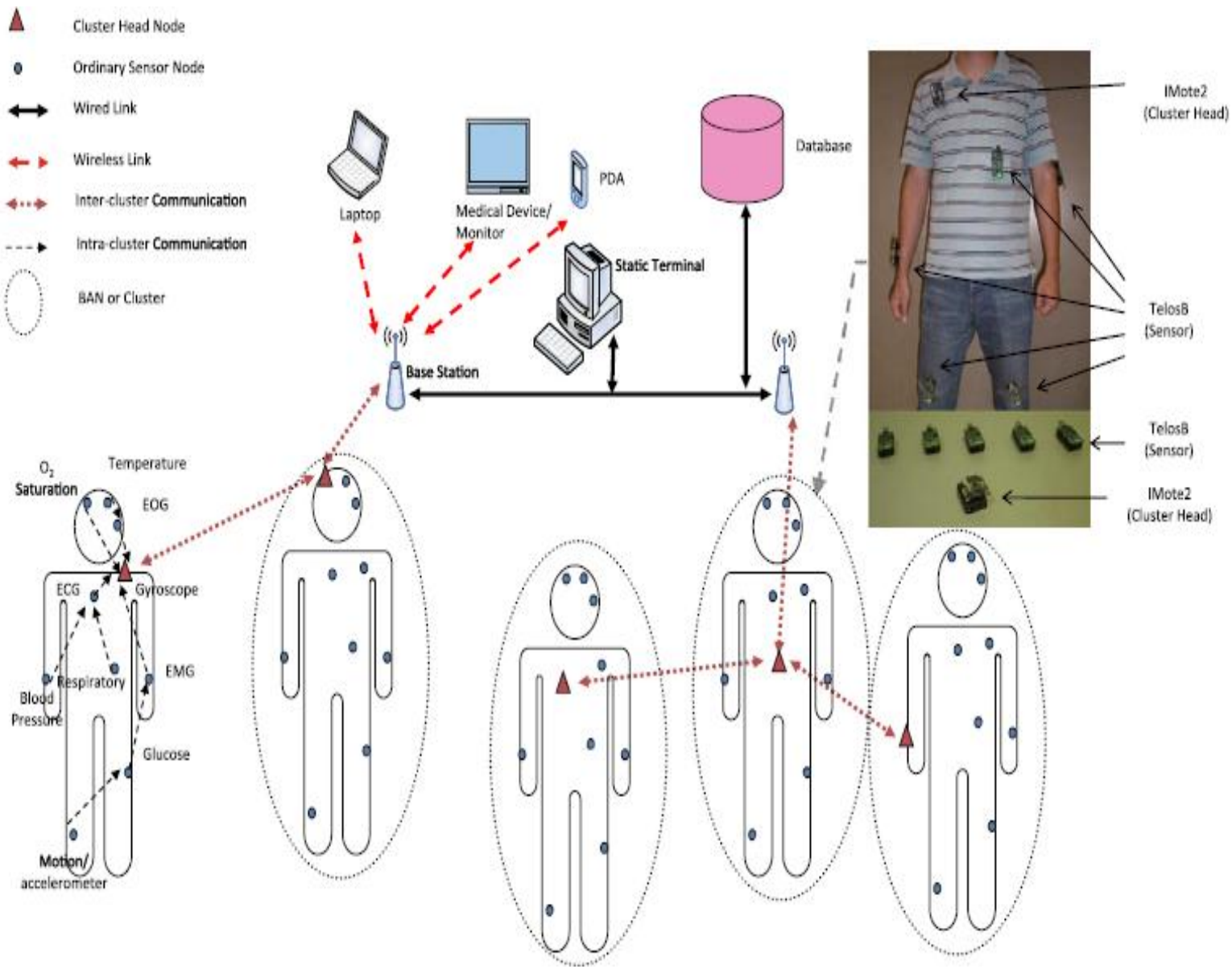


Figure: Application in wireless body area networks.

A hierarchical WBAN with the clustering heads



Battlefield surveillance

- A large number of sensor nodes are rapidly deployed in a battlefield via airplanes or trucks.
- Each individual sensor node monitors conditions and activities in its surrounding after deployment in the battlefield and then reports these sensing observations to the nearby base stations via wireless communications with its neighbor sensor nodes.
- The base station then conducts a more accurate detection on the activities (for example, possible attacks) of the opposing force after collecting a large number of sensing observations from the sensor nodes.
- Thus, the appropriate decisions as well as responses can be made quickly in the battlefield.

Overview of Wireless Sensor Networks

References on Wireless Sensor Networks Surveys

- I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A Survey. Computer Networks, Vol. 38, No. 4, pp. 393-422, 2002.
- H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. Computer Networks, Vol. 54, No. 15, pp. 2688-2710, 2010.

Overview of Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

General security requirements

- ***Authentication:*** authenticating other sensor nodes, cluster heads, and base stations before granting a limited resource, or revealing information.
- ***Integrity:*** ensuring that message or the entity under consideration is not altered.
- ***Confidentiality:*** providing privacy of the wireless communication channels to prevent eavesdropping.
- ***Availability:*** ensures that the desired network services are available even in the presence of denial of service attacks.
- ***Non-repudiation:*** preventing malicious nodes to hide their activities.
- ***Authorization:*** ensures that only the sensor nodes those who are authorized can be involved in providing information to network services.
- ***Freshness:*** ensures that the data is recent and no adversary can replay old messages.

General security requirements (Continued...)

We also need to consider the forward and backward secrecy as new sensors are deployed in the network and old sensors fail due to energy problems.

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

Sensor network limitations

- *Limited resources in sensor nodes:* Each sensor node has a primitive processor featuring very low computing power and only small amount of programmable memory.
- *Limited life-time of sensor nodes:* Each sensor node is battery-powered. So, after several weeks or months of operation, some nodes in the network may exhaust their power and as a result, the security protocols used must be energy efficient.
- *Limited communication abilities of sensor nodes:* Sensor nodes have the ability to communicate with each other and the base stations using short range wireless radio transmission at low bandwidth.

Sensor network limitations

- *Lack of knowledge about deployment configuration:* In most applications, the post-deployment network configuration is not possible to decide a prior. As a result, it may not be always possible to use security algorithms that have strong dependence on locations of sensor nodes in a sensor network.
- *Issue of node capture:* Wireless sensor networks often operate in an unattended environment. An adversary may physically capture some sensors to compromise their stored sensitive secret data and codes from their memory as they are not generally equipped with tamper-resistant hardware.

Securing Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- Key management
- User authentication
- Access control
- User access control
- Intrusion detection

Key management

- In this method, the practical approach is to preload a set of keying information before the deployment of sensor nodes in the target field.
- After deployment, they discover their neighbor nodes and then establish the secret keys between them using the preloaded keying information.
- Neighbor sensor nodes then use the established secret keys for their future secure communications.

User authentication

- In a user authentication in WSN, a legitimate user is allowed to query and collect the real-time data at any time from a sensor node or cluster head of the network as and when he/she demands for it.
- As most of the applications in wireless sensor network (WSN) are real-time based, so users are generally interested in accessing real-time information.
- This is possible if the users (called the external parties) are allowed to access the real-time data directly from the nodes inside WSN and not from the base station *BS*.
- Usually, the information from nodes are gathered periodically in the *BS* and so, the gathered information may not be real-time.
- In order to get the real-time information from the nodes, the user needs to be first authorized to the nodes as well as the *BS* so that illegal access to nodes do not happen.

Access control

- An access control scheme consists of two tasks: *node authentication* and *key establishment*.
- In *node authentication*, a deployed node needs to prove its identity to its neighbor nodes and also to prove that it has the right to access the existing sensor network.
- On the other hand, in *key establishment*, the secret shared keys need to be established between a deployed node and its neighbor nodes to protect secure communications among them.

User access control

- User access control mechanism provides the access rights for the correct information and resources for different services in wireless sensor network.
- Using user access control, an authorized user can access only those information for which he/she is permitted to access.
- In WSNs, specially in case of WBAN, for accessing of medical data, there exist different groups of users.
- In medical applications, different types of information belonging to various security levels can be generated by all kinds of sensors.
- With the proper access privilege, selected types of the authorized users should access proper data. This means that accessibility of a particular type of data to users is based solely on necessity.

Intrusion detection

- **External versus internal attacks:**
  - In the normal flow of the network, the nodes are honest and cooperative entities, whereas attacker nodes are precluded from the network and have no access to the network.
  - The external attacks can be launched only from the outside of the scope of the network. So, the impact of these attacks is limited especially in case of WSN.
  - The attacker can physically capture a sensor node and extract useful information such as its identity, secret key, etc. from that node, and can deploy some fake sensor nodes by using that extracted information. In this way an internal attack can be performed.
  - Internal attacks such as blackhole, misdirection, wormhole, sinkhole, etc. are very harmful in nature as they cause severe damage to the performance of the network.

Attacks on the different layers of WSN stack

Table: Layer-wise attacks on WSN stack

Layer	Attacks
Physical Layer	Tampering, Sybil attack, Jamming, Interception
Data Link Layer	Sybil attack, Collision, Exhaustion, Replay attack, Spoofing and altering routing attack, Traffic analysis and monitoring
Network Layer	Selective forwarding attack, Blackhole attack, Sybil attack, Hello flood attack, Spoofing attack, Internet smurf attack, Wormhole attack, Misdirection attack
Transport Layer	Desynchronization, Flooding attack
Application Layer	False data injection, Spoofing and altering routing attack

Summary of some network layer attacks
Blackhole attack

- A blackhole attack occurs when an intruder captures and re-programs a set of nodes in the network to block the packets that they receive instead of forwarding them towards the base station.
- As a result, any information that enters in the blackhole region is compromised by an attacker.
- This attack is easy to constitute and it is capable of undermining network effectiveness by partitioning the network such that the important event information do not reach the base station.
- The network performance parameters such as throughput and end-to-end delay are affected in the presence of the blackhole nodes.

Blackhole attack scenario

- A blackhole attack scenario is shown in Figure. In this case, there are three sources S_1, S_2 and S_3, which send data to the destination D. However, in the presence of blackhole attacker node X, the packets do not reach to the destination D because they are captured by X.

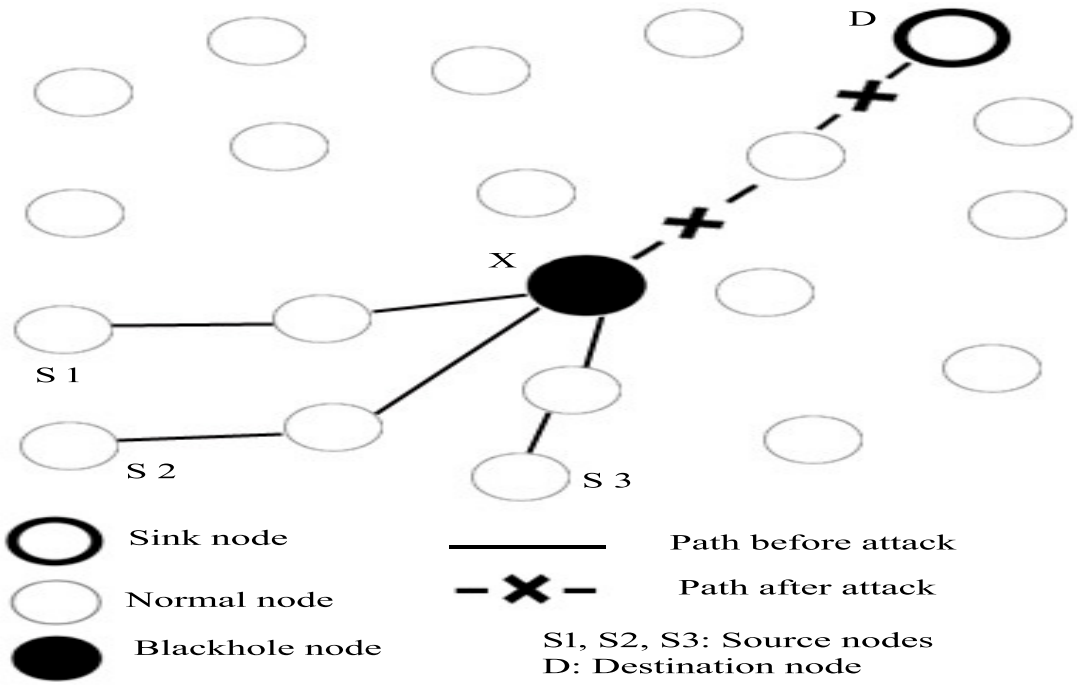


Figure: An example of a blackhole attack in WSN

Misdirection attack

- In a misdirection attack, an attacker routes the packets from its neighbors to other distant nodes, but not necessarily to its legitimate destination nodes.
- This produces a long delay in packet delivery and decreases throughput of the network. Under this attack, packets reach to the destination, but from a different route which further produces long delay and thus, also decreases throughput of the network.

Misdirection attack scenario

- There are two scenarios: one for the normal flow and other for the misdirection attack.
- Nodes S and R communicate via intermediate nodes A and D. Let node A be a misdirection attacker node. A then forwards the messages to a node, which is far away from the destination.
- Thus, the messages reach to the node R, but from a different path $\langle S, A, B, C, D \rangle$ that further increases the delay.

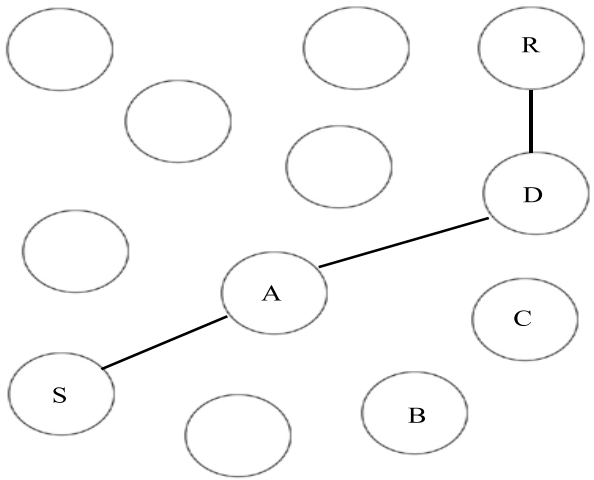


Figure: Normal flow

Misdirection attack scenario cont..

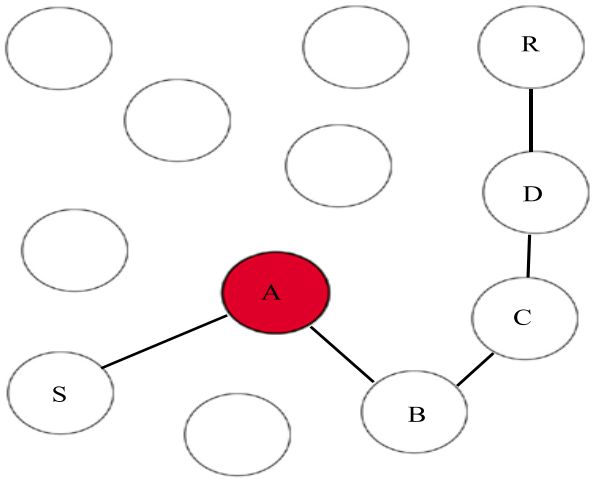


Figure: Under misdirection attack

Wormhole attack

- In a wormhole attack, an attacker can tunnel the packets between two distant locations in the network through an in-band channel or out-of-band channel. In this case, a wormhole tunnel is formed by a pair of attackers.
- The wormhole tunnel gives two distant nodes a misapprehension that they are close to each other.
- The existing wormhole can attract and bypass a large amount of network traffic, and thus the wormhole node can easily get the network traffic and perform the manipulation.
- The attacker is able to launch a variety of attacks including the sniffing, modification and dropping.

Wormhole attack scenario

- Figure depicts a scenario for the wormhole attack. Source node *A* sends the packets to a sink node *E* via intermediate nodes *B*, *C*, and *D*.
- At the same time, a wormhole node, say *WH*1 advertises a path with less hop distance so that node *A* attracts towards that path and it starts sending the packets via *WH*1 and *WH*2, where *WH*2 is the colluded node of the wormhole node *WH*1.

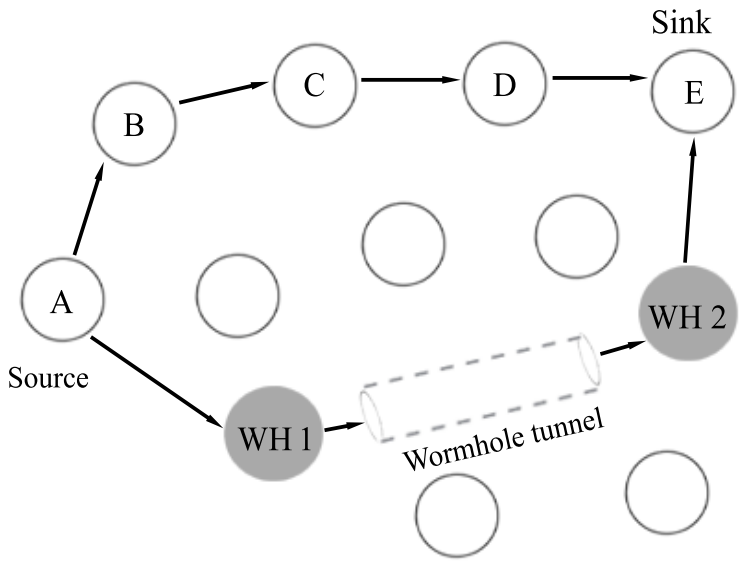


Figure: An example of wormhole attack

Sinkhole attack

- In a sinkhole attack, a malicious (sinkhole) node advertises a best possible route to the base station which misguides its neighbors in order to use that route more frequently.
- The malicious node thus gets an opportunity to tamper with the data, damage the regular network operations.
- In a sinkhole attack, the attacker node utilizes a compromised node to launch the attack in which a route with the less hope distance is advertised to misguide its neighbors.
- This assures the neighbors to forward all the traffic through such an advertised route.
- The route not only captivates the neighbors of the sinkhole, but also it captivates other nodes that are closer to the sinkhole than to the base station.

Sinkhole attack scenario

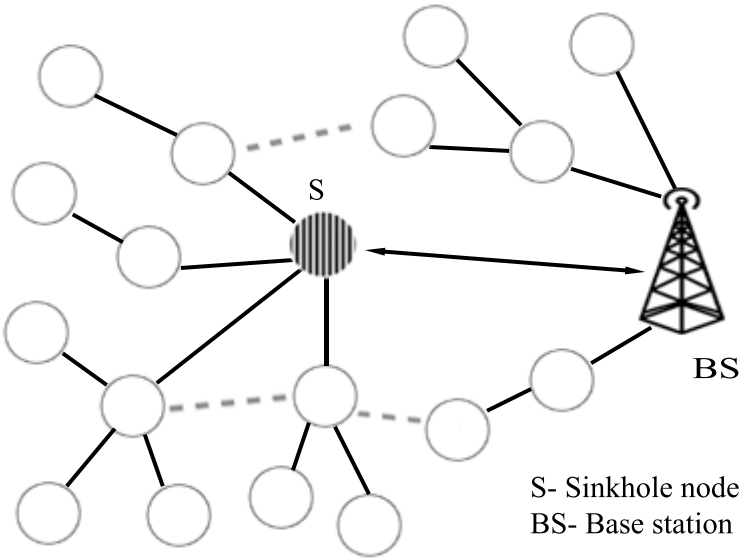


Figure: An example of sinkhole attack

Sinkhole attack scenario cont..

- The sinkhole attack can also be conducted using the wormhole attack, where a malicious node first captures the packets from its neighbors and utilizes a secret tunnel (wormhole tunnel) to send the packets to another colluded node.
- The colluded node eventually delivers the packets to the base station.
- The two ends of the wormhole tunnel can be at a longer distance as compared with other routes, but still it can prevent the source from discovering other routes greater than two hops away from the base station.

Sinkhole attack scenario cont..

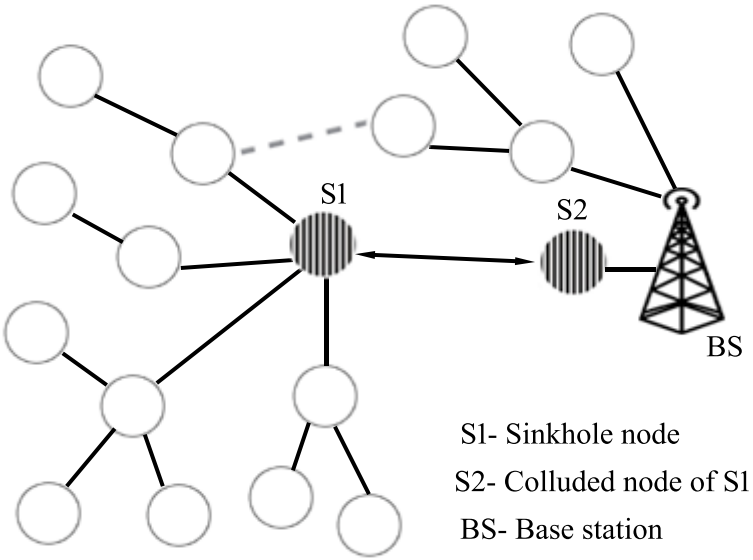
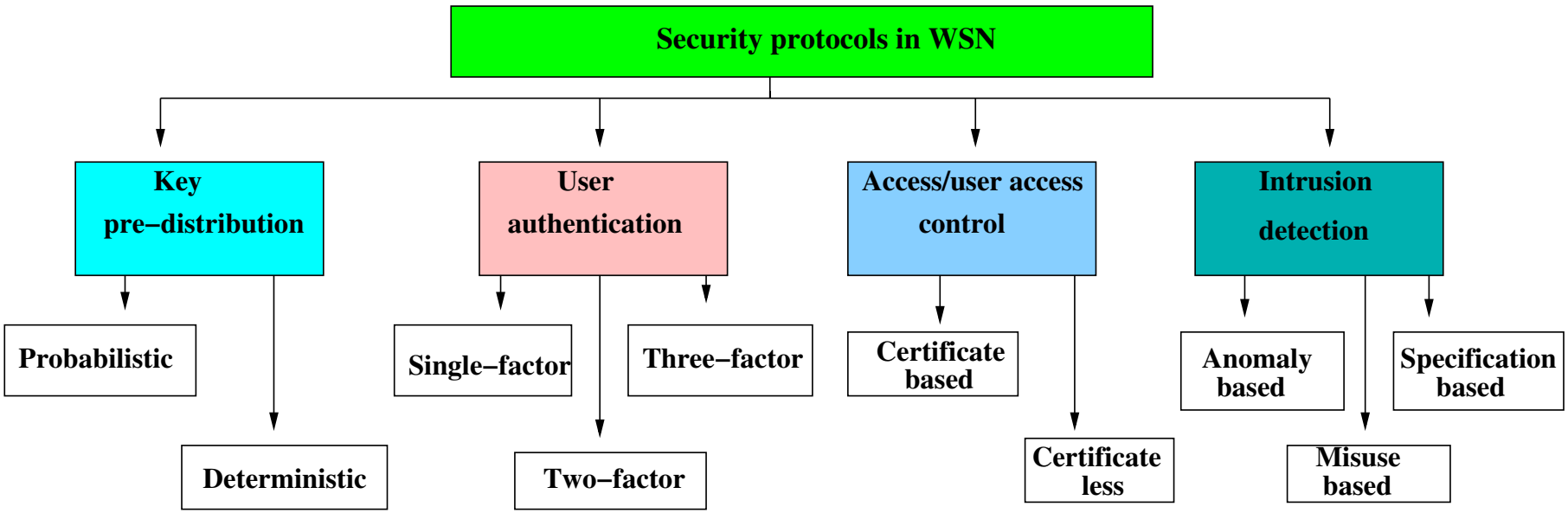


Figure: An example of sinkhole attack using wormhole tunnel

Hybrid anomaly

- In WSN, multiple attacks can be launched at a time.
- Hybrid anomaly is a type of anomaly that contains different types of attacker nodes, such as blackhole nodes, misdirection nodes, etc.
- Hybrid anomaly has the ability to degrade the network performance rapidly, and it can also trouble the attack specific detection mechanism.

Security protocols in WSNs: a taxonomy



Thank you

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Key Management in Wireless Sensor Networks

Dr. Ashok Kumar Das

IEEE Senior Member
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/

Part I

Key Management in Wireless Sensor Networks

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

The bootstrapping protocol

- Establishes cryptographically secure communication links among the communicating sensor nodes.
- Must not only enable a newly deployed sensor node to initiate a secure infrastructure, but it must also allow nodes deployed at a later time to join the network securely.
- A challenging area of sensor networks due to resource limitations of sensor nodes as well as vulnerable to physical capture of nodes by an adversary in a sensor network.
- Public key routines (such as RSA, Diffie-Hellman, ElGamal) are not so much viable options. However, recent research demonstrates that ECC is viable.
- Symmetric key ciphers (such as DES, AES, RC5) are viable options

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Different phases of the bootstrapping protocol

- **Key pre-distribution phase:** Done in offline by the key setup server (usually, the base station).
- **Direct key establishment phase:** Performed by each sensor node after its deployment in the network.
- **Path key establishment phase:** Required if nodes do not establish direct keys during the direct key establishment phase.

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Requirements of the bootstrapping protocol

- *R1:* Deployed sensor nodes must be able to establish secure node-to-node communication.
- *R2:* Illegal sensor nodes should not be able to gain entry/access into the network, either through packet injection or masquerading as a legitimate sensor node.
- *R3:* One can always add new sensor nodes dynamically at any time after the initial deployment and these additional deployed nodes can form secure connections with the already deployed nodes in the sensor network. Thus, the bootstrapping information must always be present and can not be simply erased after deployment to prevent compromise in the event of capture.

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Evaluation metrics of the bootstrapping protocol

- ***Scalability:*** It should support a large-scale sensor network.
- ***Storage overhead:*** The amount of memory required to store security credentials must be minimum.
- ***Communication overhead:*** The number of messages exchanged during a key establishment procedure must be less.
- ***Computational overhead:*** The amount of processor cycles required to establish a secret key between two communicating sensor nodes should be minimum due to resource limitations of sensor nodes.
- ***Network connectivity:*** This is the probability that two sensor nodes can establish a secret key.

Evaluation metrics of the bootstrapping protocol (Continued...)

- ***Resilience against node capture:*** For any two non-compromised sensor nodes u and v, we have to find out what is the probability that the adversary can decrypt the secret communications between u and v when c sensor nodes are already compromised ?
  Let $P_e(c)$ denote the fraction of total secure communications compromised after capturing c sensor nodes by an attacker in a sensor network.
  If $P_e(c) = 0$, we call a key establishment scheme as *unconditionally secure against node capture* or *perfectly resilience against node capture*.

Key distribution using a single network-wide key

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Protocol

- The simplest solution is the use of a single network-wide master key for the entire network.
- Each node is given the same mission key before deployment in the network by the key setup server.
- After deployment, any two neighbor nodes can communicate securely with each other using this key.

Key distribution using a single network-wide Key

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Properties

- Only a single network-wide cryptographic key is needed to be stored in each sensor node's memory.
- Works well without needing to perform the direct key establishment phase.
- Provides 100% network connectivity.
- NO computational overhead.
- NO communication overhead.
- Scalable.

Key distribution using a single network-wide Key

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Drawbacks

- The compromise of even a single node in a network would reveal the secret key and thus allow decryption of all network traffic.
- Another solution is to use a single shared network-wide key to establish session keys between any two neighbor nodes, and then erase the network-wide key.
- The main difficulty of this variant of the key establishment procedure is that it does not allow addition of new nodes after initial deployment.

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Key Management in Wireless Sensor Networks

Dr. Ashok Kumar Das

IEEE Senior Member
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/

Part I

Key Management in Wireless Sensor Networks

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

The bootstrapping protocol

- Establishes cryptographically secure communication links among the communicating sensor nodes.
- Must not only enable a newly deployed sensor node to initiate a secure infrastructure, but it must also allow nodes deployed at a later time to join the network securely.
- A challenging area of sensor networks due to resource limitations of sensor nodes as well as vulnerable to physical capture of nodes by an adversary in a sensor network.
- Public key routines (such as RSA, Diffie-Hellman, ElGamal) are not so much viable options. However, recent research demonstrates that ECC is viable.
- Symmetric key ciphers (such as DES, AES, RC5) are viable options

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Different phases of the bootstrapping protocol

- **Key pre-distribution phase:** Done in offline by the key setup server (usually, the base station).
- **Direct key establishment phase:** Performed by each sensor node after its deployment in the network.
- **Path key establishment phase:** Required if nodes do not establish direct keys during the direct key establishment phase.

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Requirements of the bootstrapping protocol

- *R1:* Deployed sensor nodes must be able to establish secure node-to-node communication.
- *R2:* Illegal sensor nodes should not be able to gain entry/access into the network, either through packet injection or masquerading as a legitimate sensor node.
- *R3:* One can always add new sensor nodes dynamically at any time after the initial deployment and these additional deployed nodes can form secure connections with the already deployed nodes in the sensor network. Thus, the bootstrapping information must always be present and can not be simply erased after deployment to prevent compromise in the event of capture.

Key Distribution in Wireless Sensor Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Evaluation metrics of the bootstrapping protocol

- ***Scalability:*** It should support a large-scale sensor network.
- ***Storage overhead:*** The amount of memory required to store security credentials must be minimum.
- ***Communication overhead:*** The number of messages exchanged during a key establishment procedure must be less.
- ***Computational overhead:*** The amount of processor cycles required to establish a secret key between two communicating sensor nodes should be minimum due to resource limitations of sensor nodes.
- ***Network connectivity:*** This is the probability that two sensor nodes can establish a secret key.

Evaluation metrics of the bootstrapping protocol (Continued...)

- ***Resilience against node capture:*** For any two non-compromised sensor nodes u and v, we have to find out what is the probability that the adversary can decrypt the secret communications between u and v when c sensor nodes are already compromised ?
  Let $P_e(c)$ denote the fraction of total secure communications compromised after capturing c sensor nodes by an attacker in a sensor network.
  If $P_e(c) = 0$, we call a key establishment scheme as *unconditionally secure against node capture* or *perfectly resilience against node capture*.

Key distribution using a single network-wide key

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Protocol

- The simplest solution is the use of a single network-wide master key for the entire network.
- Each node is given the same mission key before deployment in the network by the key setup server.
- After deployment, any two neighbor nodes can communicate securely with each other using this key.

Key distribution using a single network-wide Key

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Properties

- Only a single network-wide cryptographic key is needed to be stored in each sensor node's memory.
- Works well without needing to perform the direct key establishment phase.
- Provides 100% network connectivity.
- NO computational overhead.
- NO communication overhead.
- Scalable.

Key distribution using a single network-wide Key

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Drawbacks

- The compromise of even a single node in a network would reveal the secret key and thus allow decryption of all network traffic.
- Another solution is to use a single shared network-wide key to establish session keys between any two neighbor nodes, and then erase the network-wide key.
- The main difficulty of this variant of the key establishment procedure is that it does not allow addition of new nodes after initial deployment.

Random key distribution

(L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks", in *9th ACM CCS*, pp. 41-47, Nov. 2002.)

Key pre-distribution phase

- Done in offline by the key setup server (base station).
- Each node u is assigned a unique node identifier id_u.

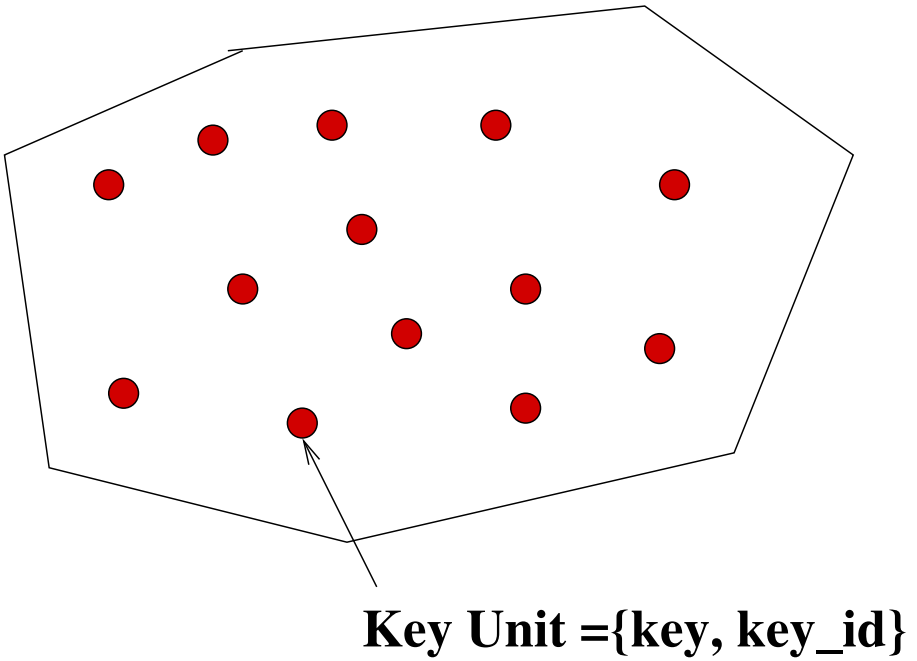


Figure: Key pool $\mathcal{M}$ of size M.

Random key distribution

Key pre-distribution phase (Continued...)

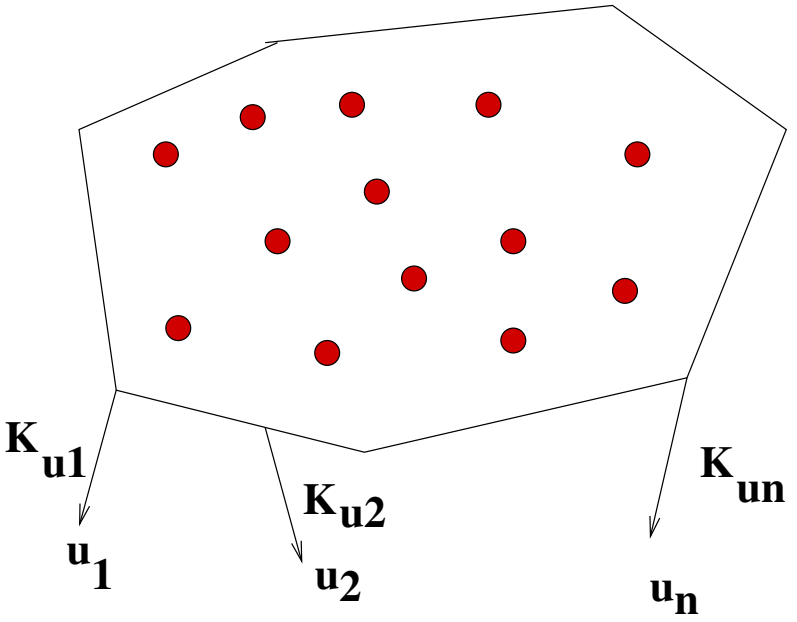


Figure: Key ring K_{u_i} selection of a sensor node u_i.

- For each node u_i, a small subset K_{u_i} of size m is selected randomly without replacement from the key pool $\mathcal{K}$.
- Each node u_i is pre-loaded with (i) id_{u_i}, and (ii) K_{u_i}.

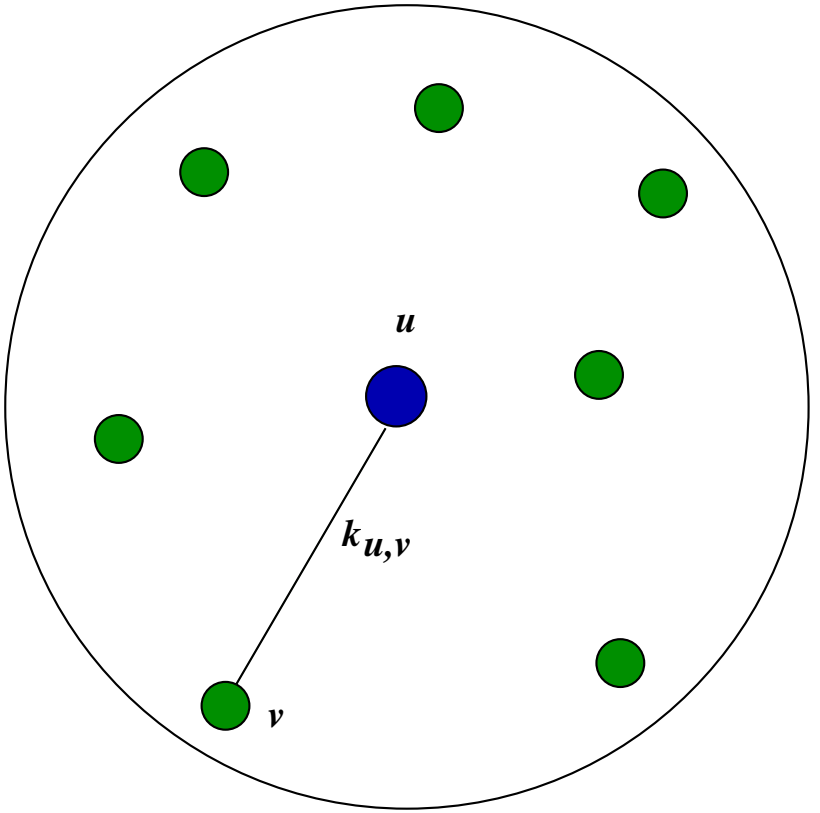
Random key distribution

Direct key establishment phase

- Executed by each sensor node after deployment in the network.
- Each node broadcasts a HELLO message (containing its own identifier).
  $u \rightarrow * : HELLO$
- Each node prepares a list of physical neighbors in its communication range.
  $NL_u = \{v_1, v_2, \ldots, v_d\}$ is the list of d neighbors of a node u.
- Key neighbors
- Direct neighbors

Random key distribution

Direct key establishment phase (Continued...)



- $u \rightarrow v : id_u || \{$ list of key ids $\}$
- $v \rightarrow u : id_v || \{$ list of key ids $\}$

Random key distribution

Direct key establishment phase (Continued...)

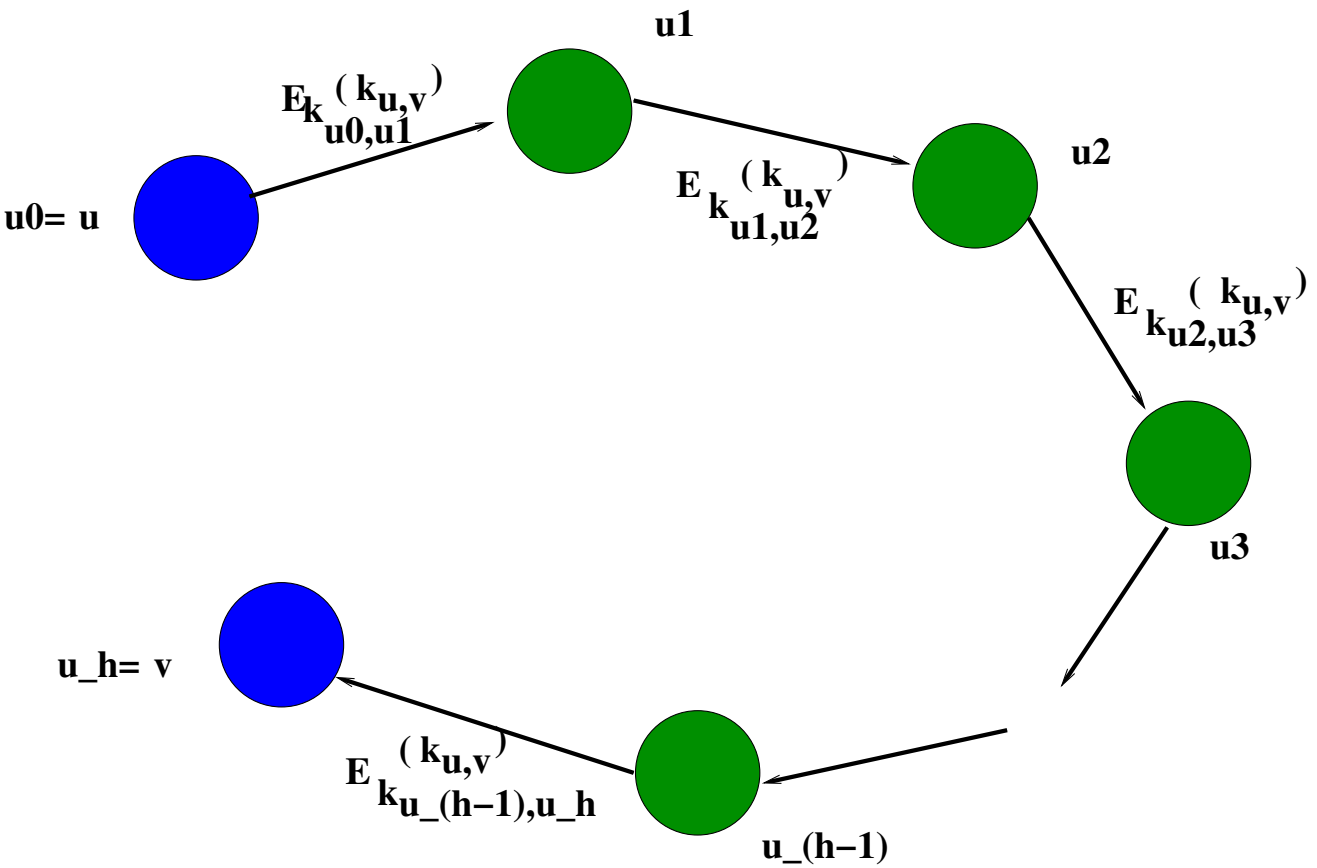
Another procedure of key discovery which is more secure, but slower, could utilize client puzzles such as a Merkle puzzle (R. Merkle, "Secure communication over insecure channels", *Communications of the ACM*, 21(4):294-299, 1978).

- Assume u and v be two neighbor nodes.
- u generates m client puzzles, say, $P_1, \ldots, P_m$, one for each of the m keys in its key ring.
- $u \rightarrow v : \{E_{k_i}(P_i), MAC_{k_i}(P_i)\}$, $i = 1, 2, \ldots, m$.
- v decrypts an encrypted puzzle, say, $E_{k_i}(P_i)$ with one of the keys residing in its key ring and computes the corresponding MAC. If the computed MAC and the received MAC are equal, then u and v use this key for future communication.
- Though this method is secure one, but it introduces a lot of communication and computational overheads to establish pairwise keys among neighbor nodes.

Random key distribution

Path key establishment phase

- Executed after direct key establishment phase by a sensor node in the network, if required.



Random key distribution

Dynamic node addition phase

- Assume a node u needs to be deployed in the existing sensor network.
- The key setup server assigns a unique identifier id_u and selects a key ring K_u of size m from the key pool $\mathcal{K}$. These information are loaded in its memory before deployment.
- After deployment, u establishes keys with its neighbor nodes.
- Path key establishment could be executed by the node u, if necessary.

Random key distribution

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Analysis

- **Storage overhead:** m keys.
- **Communication overhead:** list of m key ids (clear-text broadcasting) or list of m challenge messages (private shared-key discovery).
- **Computational overhead:** $\frac{2m+p_{EG}-p_{EG}\cdot m}{2}$ $\log m$ comparisons, where p_{EG} is the probability that two neighbor nodes can establish a secret key during the direct key establishment phase (network connectivity probability)[clear text broadcasting]. Required additional cost due to $2m$ encryptions and decryptions and $2m$ MACs for MAC verifying the puzzles [private shared-key discovery].

Random key distribution

Analysis (Continued...)

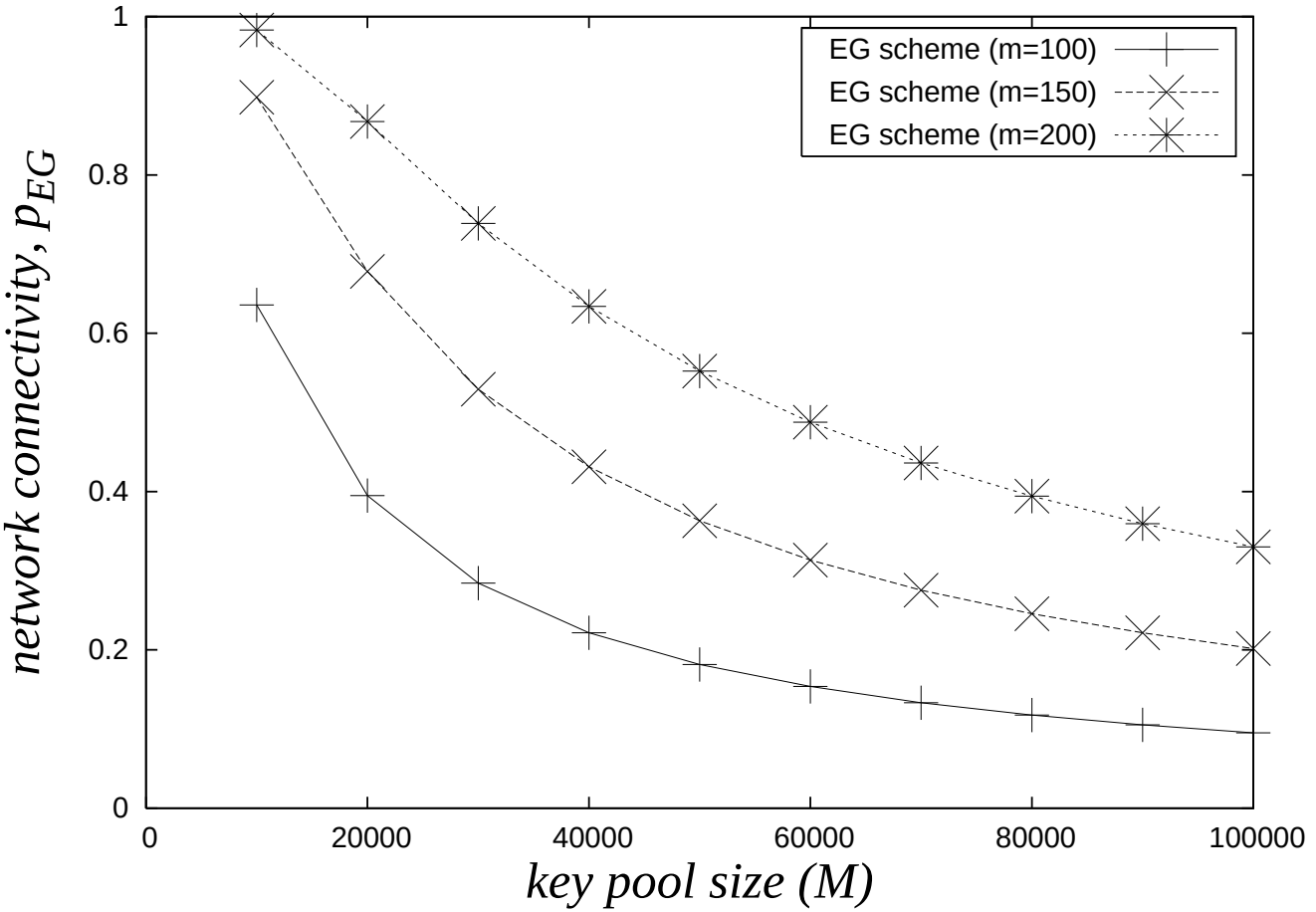
- **Network connectivity for direct key establishment phase:** The probability of establishing a direct pairwise key between two sensor nodes u and v is

$$\begin{aligned} p_{EG} &= 1 - \frac{\binom{M-m}{m}}{\binom{M}{m}} \\ &= 1 - \prod_{i=0}^{m-1} \frac{M-m-i}{M-i}, \end{aligned} \tag{1}$$

  where M is the key pool size and m the key ring size of a sensor node.

Random key distribution

Network connectivity of the EG scheme for different combinations of M and m.



Random key distribution

Network connectivity for path key establishment phase:

- Let d be the average number of neighbor nodes that each sensor node can contact.
- The probability of two sensor nodes establishing a pairwise key (directly or indirectly) is

$$p_1 = 1 - (1 - p)(1 - p^2)^d. \tag{2}$$

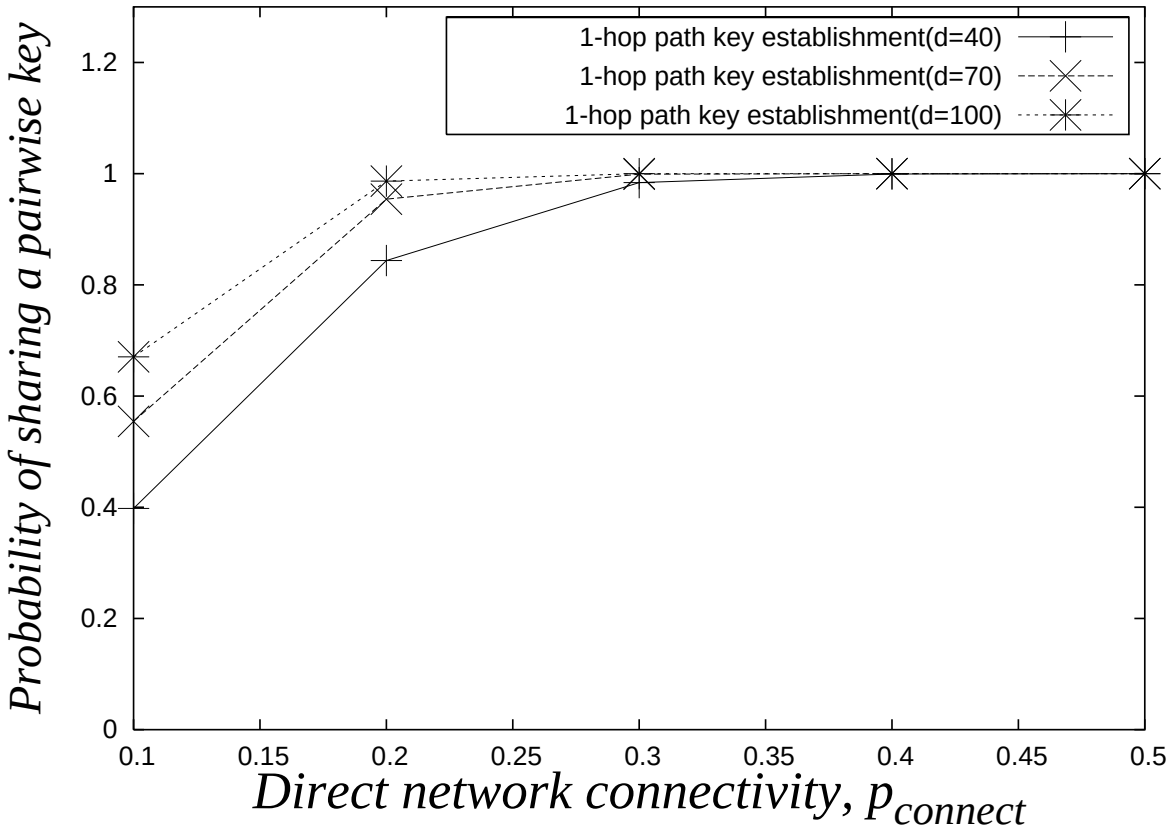
- If p_h is the probability that two neighbor sensor nodes can establish a key using a h-hop path key establishment phase, it is easy to deduce that

$$p_h = 1 - (1 - p_{h-1})(1 - p.p_{h-1})^d \text{ for all } h \geq 1, \tag{3}$$

where $p_0 = p$.

Random key distribution

The probability p_1 of establishing a pairwise key v.s. the probability p that two sensor nodes establish a direct pairwise key, with $d = 40, 70, 100$.



Random key distribution

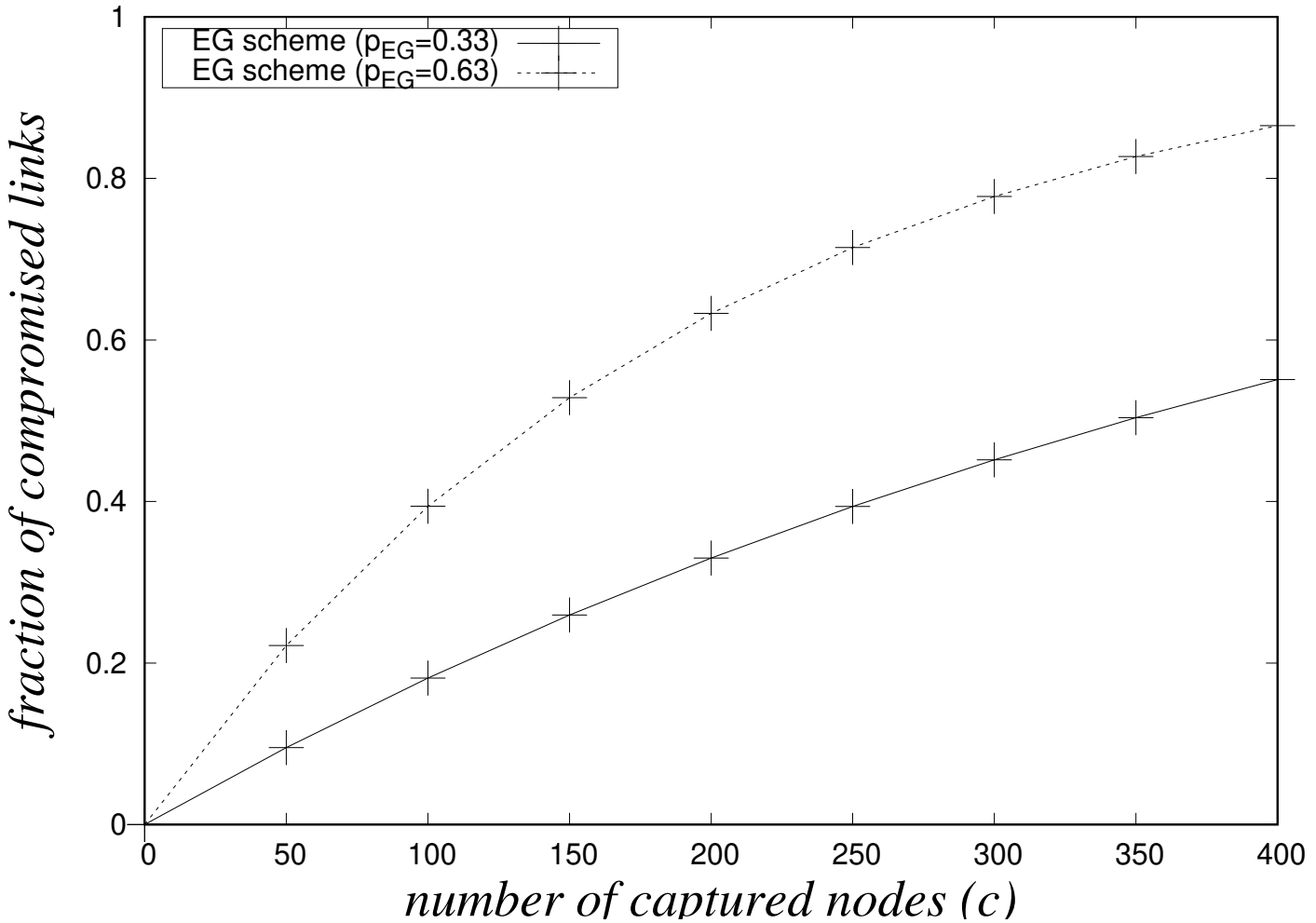
Resilience against node capture attacks during direct key establishment phase

- Node capture model: Random / Selective.
- For any two non-compromised sensor nodes u and v, we have to find out what is the probability that the adversary can decrypt the secret communications between u and v when c sensor nodes are already compromised ?
- When c nodes are already captured, the resilience against node capture is given by

$$P_e(c) = 1 - \left(1 - \frac{m}{M}\right)^c. \tag{4}$$

Random key distribution

Resilience against node capture attacks during direct key establishment phase (Continued...)



Random key distribution

Resilience against node capture attacks during path key establishment phase

- Consider a secure h-hop path $\langle u = u_0, u_1, u_2, \ldots, u_{h-1}, u_h = v \rangle$ between two neighbor nodes u and v through which u and v can establish a pairwise direct secret key between them.
- The secure link (u, v) is compromised by an attacker if either of its end points u and v are compromised, or any one of the intermediate nodes $u_1, u_2, \ldots, u_{h-1}$ is compromised.
- If a fraction f of sensor nodes are captured by an attacker in the network during the path key establishment phase, the probability that the secure link (u, v) is compromised is 1− (probability that the link (u, v) is not compromised) = $1 - (1 - f)^{h+1}$.
- p and p_h be the probabilities that two neighbors can establish a secure link during the direct key establishment phase and h-hop path key establishment phase, respectively.

Random key distribution

Resilience against node capture attacks during path key establishment phase (Continued...)

- Let there be n sensor nodes deployed in the network and each node have in average d physical neighbors in its communication range.
- The total number of secure links in the network is $\frac{nd}{2} \times p + \frac{nd}{2} \times (1-p) \times p_h$.
- The resilience against node capture during the h-hop path key establishment phase due to capture of a fraction $f(= \frac{c}{n})$ of sensor nodes in the network can be estimated as

$$
\begin{aligned}
P_e(c)_{pathkey} &= \frac{\frac{nd}{2} \times P_e(c) + \frac{nd}{2} \times (1-p) \times p_h \times (1-(1-f)^{h+1})}{\frac{nd}{2} \times p + \frac{nd}{2} \times (1-p) \times p_h} \\
&= \frac{P_e(c)}{p + (1-p) \times p_h} + (1 - \frac{p}{p + (1-p) \times p_h}) \\
&\quad \times (1-(1-f)^{h+1}). \qquad (5)
\end{aligned}
$$

Random key distribution

Important observations

- Network connectivity p depends on size M of key pool with a fixed key ring size m. Smaller M leads to higher network connectivity.
- Resilience against node capture $P_e(c)$ depends on key pool size M and number of captured nodes c. Smaller key pool size leads to have low resilience against node capture. Further, as c increases, resilience also decreases.
- Needs to make a better trade-off between network connectivity and resilience against node capture.

Random key distribution

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Important References

- L. Eschenauer and V. D. Gligor. A Key Management Scheme for Distributed Sensor Net- works. In 9th ACM Conference on Computer and Communication Security, pages 41-47, November 2002.
- H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In IEEE Symposium on Security and Privacy, pages 197-213, Berkeley, California, 2003.
- W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In 23rd Conference of the IEEE Communications Society (Infocom'04), volume 1, pages 586-597, Hong Kong, China, March 21-25 2004.
- W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In ACM Conference on Computer and Communications Security (CCS'03), pages 42-51, Washington DC, USA, October 27-31 2003.

Random key distribution

Important References

- D. Liu and P. Ning. “Establishing Pairwise Keys in Distributed Sensor Networks,” in Proceedings of 10th ACM Conference on Computer and Communications Security (CCS), pages 52-61, Washington DC, Oct 27-31 2003.
- Ashok Kumar Das. “A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks,” in International Journal of Information Security (Springer), Vol. 11, No. 3, pp. 189-211, 2012, doi: 10.1007/s10207-012 -0162-9.
- Ashok Kumar Das. “Improving Identity-Based Random Key Establishment Scheme for Large-Scale Hierarchical Wireless Sensor Networks,” in International Journal of Network Security, Vol. 14, No. 1, pp. 1 - 21, 2012.

Random key distribution

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Important References

- Ashok Kumar Das. “An Efficient Random Key Distribution Scheme for Large-Scale Distributed Sensor Networks,” in Security and Communication Networks (Wiley), Vol. 4, No. 2, pp. 162 - 180, 2011.
- Ashok Kumar Das. “A Survey on Analytic Studies of Key Distribution Mechanisms in Wireless Sensor Networks,” in Journal of Information Assurance and Security, Vol. 5, No. 5, pp. 526-553, 2010, Dynamic Publishers Inc., USA.
- Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, “A survey of key management schemes in wireless sensor networks,” *Computer Communications*, vol. 30, no. 11-12, pp. 2314-2341, 2007.
- Y. Wang, G. Attebuty, and B. Ramamurthy, “A Survey of Security Issues in Wireless Sensor Networks”, *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 2-23, 2006.

Thank you

Introduction to Blockchain

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad
(Formerly **Indian Institute of Information Technology, Hyderabad**)

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

September 9, 2024

What is Blockchain?

- A blockchain is considered as a chain of blocks that are created from several blocks and it potentially consists of information.
- By the words “block” and “chain”, we actually specify in the context of digital information (“block”) which is stored in a public domain say database (“chain”).
- Since the digital information is stored in the form of “block” and it is linked in a “chain” form, the linked blocks constitute a chain, and hence, the name “blockchain”.
- The blockchain’s first block is known as the **Genesis block**.

What is Blockchain?

The reasons why the blockchain have gained so much admiration are that:

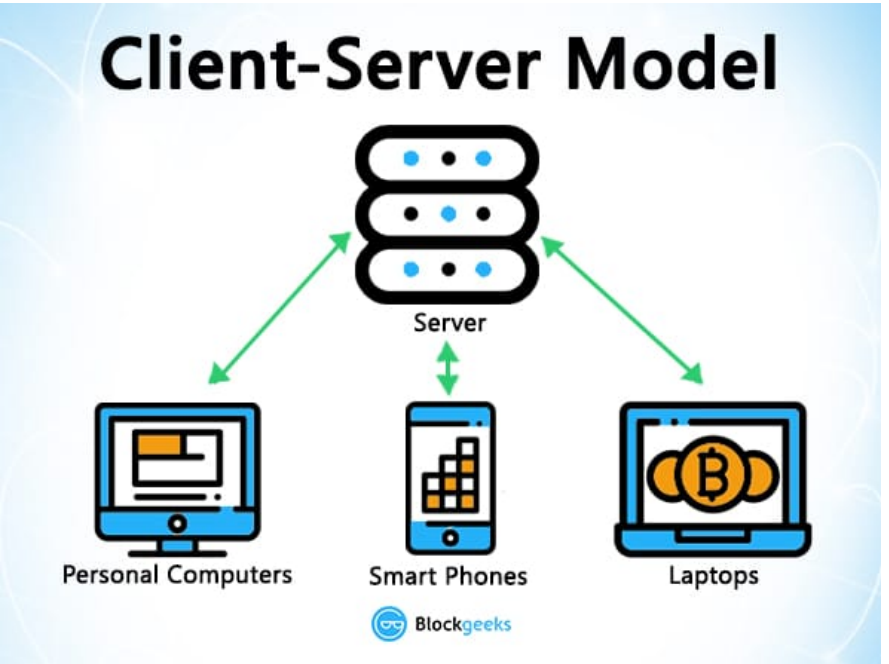
- It is not owned by a single entity, hence it is **decentralized**
- The data is **cryptographically** stored inside
- The blockchain is **immutable**, so no one can tamper with the data that is inside the blockchain
- The blockchain is **transparent** so one can track the data if they want to

Three Pillars of Blockchain Technology

- Decentralization
- Transparency
- Immutability

Pillar #1 : Decentralization

- Example of a centralized system is banks. They store all your money, and the only way that you can pay someone is by going through the bank.
- The traditional client-server model is a perfect example of this.
- When you google search for something, you send a query to the server who then gets back at you with the relevant information. That is simple client-server.



Pillar #1 : Decentralization

The centralized systems have treated us well for many years, however, they have several vulnerabilities.

- Firstly, because they are centralized, all the data is stored in one spot. This makes them easy target spots for potential hackers.
- If the centralized system was to go through a software upgrade, it would halt the entire system
- What if the centralized entity somehow shut down for whatever reason? That way nobody will be able to access the information that it possesses
- Worst case scenario, what if this entity gets corrupted and malicious? If that happens then all the data that is inside the blockchain will be compromised.

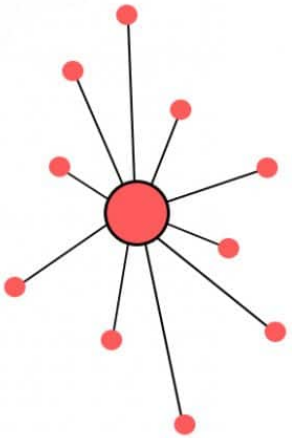
Pillar #1 : Decentralization

- In a decentralized system, the information is not stored by one single entity. In fact, everyone in the network owns the information.
- In a decentralized network, if you wanted to interact with your friend then you can do so directly without going through a third party.
- That was the main ideology behind Bitcoins. You and only you alone are in charge of your money. You can send your money to anyone you want without having to go through a bank.

Pillar #1 : Decentralization

The New Networks

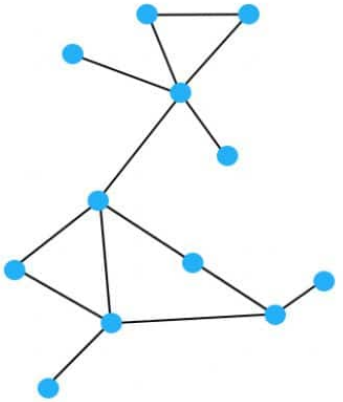
Centralized



Centralized systems have a core authority that dictates the truth to the other participants in the network.

Only priveleged users or institutions can access the history of transactions or confirm new transactions.

Decentralized



Decentralized systems have no core authority to dictate the truth to other participants in the network.

Every participant in the network can access the history of transactions or confirm new transactions.

Blockgeeks

Pillar #2 : Transparency

- While the person's real identity is secure, you will still see all the transactions that were done by their public address.
- This level of transparency has never existed before within a financial system.
- It adds that extra, and much needed, level of accountability which is required by some of these biggest institutions.

Pillar #3 : Immutability

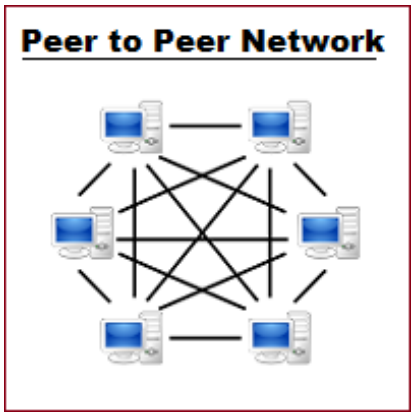
- Immutability, in the context of the blockchain, means that once something has been entered into the blockchain, it cannot be tampered with.
- The reason why the blockchain gets this property is that of cryptographic hash function.

INPUT	HASH
Hi	3639EFCD08ABB273B1619E82E78C29A7DF02C1051B1820E99FC395DCAA3326B8
Welcome to blockgeeks. Glad to have you here.	53A53FC9E2A03F9B6E66D84BA701574CD9CF5F01FB498C41731881BCDC68A7C8

INPUT	HASH
This is a test	C7BE1ED902FB8DD4D48997C6452F5D7E509FBCDBE2808B16BCF4EDCE4C07D14E
this is a test	2E99758548972A8E8822AD47FA1017FF72F06F3FF6A016851F45C398732BC50C

Maintaining the Blockchain – Network and Nodes

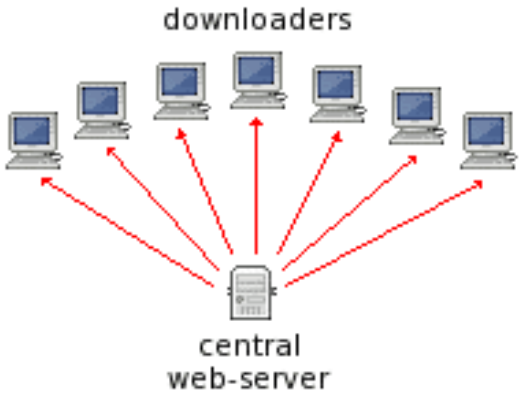
- The blockchain is maintained by a peer-to-peer (P2P) network.
- The network is a collection of nodes which are interconnected to one another.
- Nodes are individual computers which take in input and performs a function on them and gives an output.
- The blockchain uses a special kind of network called *peer-to-peer network* which partitions its entire workload between participants, who are all equally privileged, called *peers*.
- There is no longer one central server, now there are several distributed and decentralized peers.



Maintaining the Blockchain – Network and Nodes

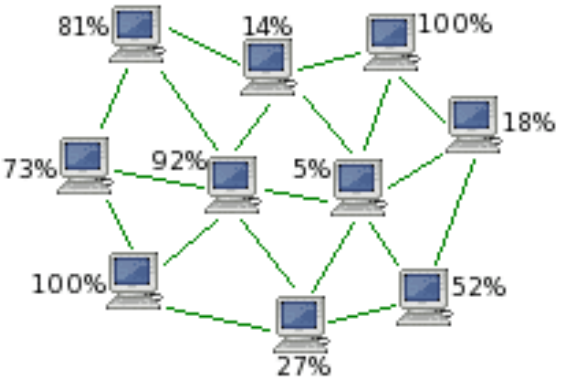
- One of the main uses of the peer-to-peer network is file sharing, also called torrenting.

Traditional Centralized Downloading



- Slow
- Single point of failure
- High bandwidth usage for server

Decentralized Peer-to-Peer Downloading

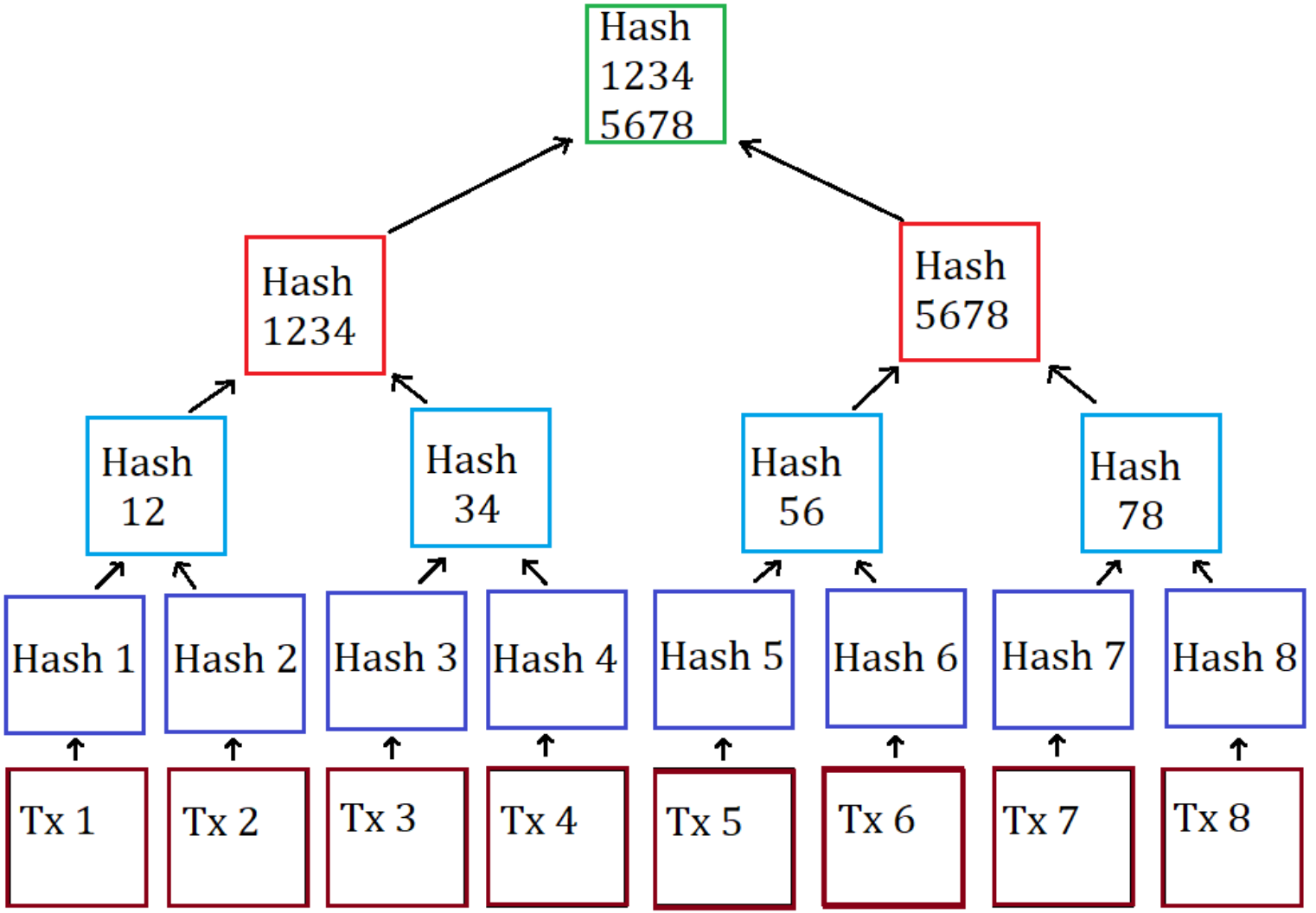


- Fast
- No single point of failure
- All downloaders are also uploaders

Types of blockchain

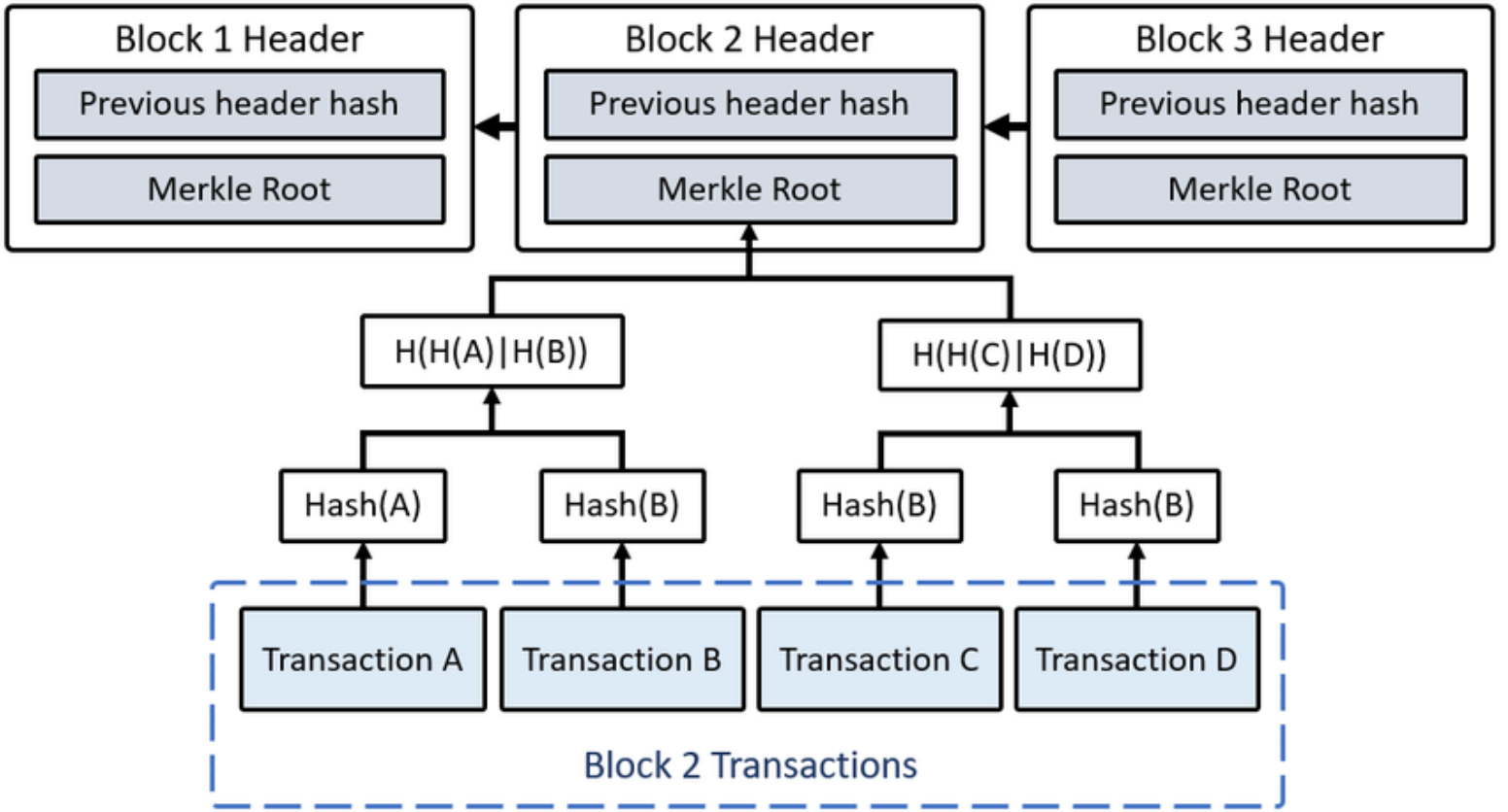
- **Public (Permissionless) Blockchain**: Everyone has the right to join, access, send, verify and receive transactions of the blocks in the blockchain to create a consensus. One widely successful permissionless blockchain is *bitcoin*.
- **Private (Permissioned) Blockchain**: The owner of the network decides which node to assign the right to access, send, receive, join and verify the block for creating an agreement between the nodes. Example: Healthcare Applications
- **Consortium or Hybrid Blockchain**: Internet of Vehicles (IoV) application

How to Make a Merkle Tree?

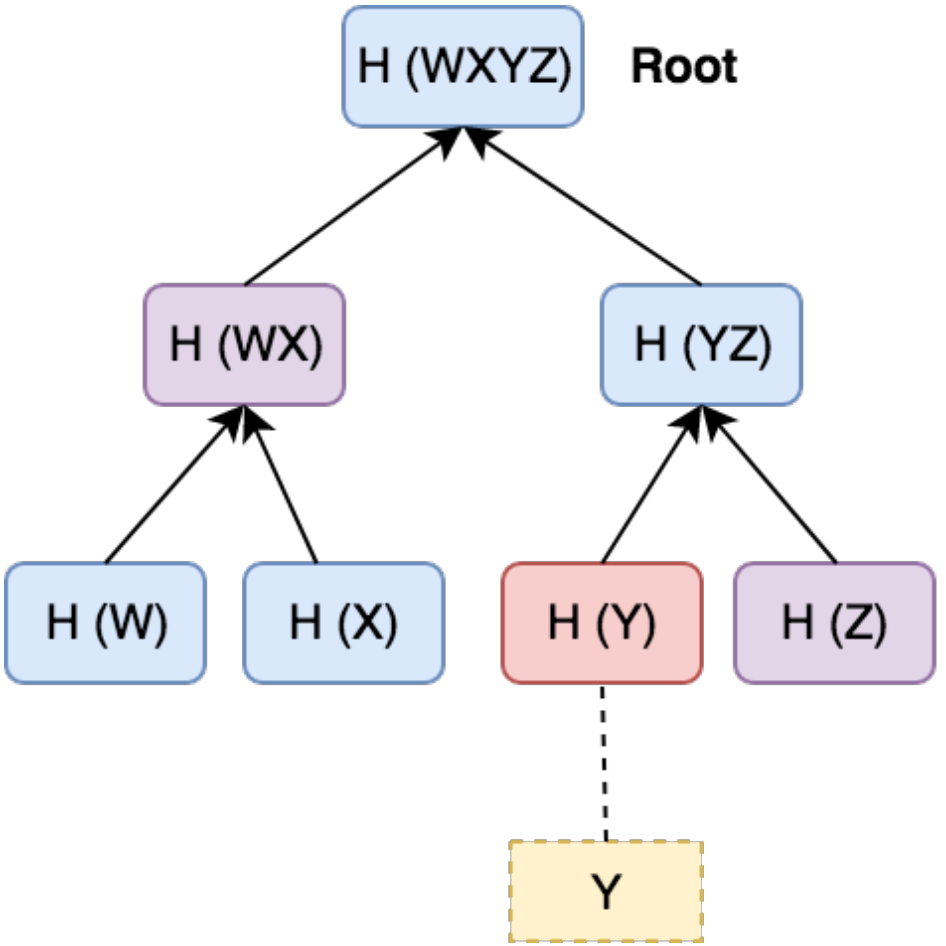


Tx_i: i^{th} transaction; H_i: Hash of i^{th} transaction ($H_{Tx_i} = H(Tx_i)$);
$H_{Tx_1 Tx_2} = H_{Tx_1} \oplus H_{Tx_2}$; $H_{12345678}$: Merkle root

Use of Merkle Tree in Blockchain



Verifying Transactions Using the Merkle Root



To confirm transaction Y, one only needs to know: $H(WX)$, $H(Y)$, $H(Z)$ and $H(WXYZ)$; $H(\cdot)$: hash function

Types of blockchain

Block Header	
Block Version	BV
Previous Block Hash	PBHash
Merkle Tree Root	MTR
Block Type	Public
Timestamp	TS
Owner of Block	ES_l
Public key of signer (ES_l)	Pub_{ES_l}
Block Payload (Transactions)	
Transaction #1	Tx_1
Transaction #2	Tx_2
⋮	⋮
Transaction #n_t	Tx_{n_t}
Current Block Hash	CBHash
Signature on block using ECDSA	BSign

a) Formation of a block on public blockchain

Block Header	
Block Version	BV
Previous Block Hash	PBHash
Merkle Tree Root	MTR
Block Type	Private
Timestamp	TS
Owner of Block	ES_l
Public key of signer (ES_l)	Pub_{ES_l}
Block Payload (Encrypted Transactions)	
Encrypted Transaction #1	$E_{Pub_{ES_l}}(Tx_1)$
Encrypted Transaction #2	$E_{Pub_{ES_l}}(Tx_2)$
⋮	⋮
Encrypted Transaction #n_t	$E_{Pub_{ES_l}}(Tx_{n_t})$
Current Block Hash	CBHash
Signature on block using ECDSA	BSign

b) Formation of a block on private blockchain

Block Header	
Block Version	BV
Previous Block Hash	PBHash
Merkle Tree Root	MTR
Block Type	Hybrid
Timestamp	TS
Owner of Block	ES_l
Public key of signer (ES_l)	Pub_{ES_l}
Block Payload	
Encrypted Transaction #1	$E_{Pub_{ES_l}}(Tx_1)$
Transaction #2	Tx_2
⋮	⋮
Encrypted Transaction #n_t	$E_{Pub_{ES_l}}(Tx_{n_t})$
Current Block Hash	CBHash
Signature on block using ECDSA	BSign

c) Formation of a block on consortium blockchain

Consensus mechanisms

Consensus mechanisms are used to verify transactional data between the nodes in a P2P network.

- **Byzantine Fault Tolerance (BFT)**: An agreement protocol which helps to tolerate the Byzantine failures in a network. BFT maintains the reliable record of transactions in a transparent and tamper-proof way, as long as the number of traitors does not exceed one-third of the general network nodes.
- **Practical Byzantine Fault Tolerance (PBFT)**: This consensus mechanism is used when BFT fails to tolerate the faults in a network system. The algorithm for PBFT works in asynchronous systems and is optimized to achieve high performance along with an impressive overhead runtime.

Practical Byzantine Fault Tolerance (PBFT)

For adding a block in the blockchain, the following procedures in PBFT are required:

- A leader acting as a miner will select by the leader selection algorithm for adding a block.
- Leader receives a block with block adding request from any nodes (or client) into the blockchain.
- Leader sends this block to every node in the network for verifying the transactions.
- After successful verification of the transaction in the block, each received node sends a valid reply for adding that block.
- Leader counts the received reply and checks the number of counts (say, RCount) if it is greater that the twice number of the faulty nodes, i.e., $RCount > 2n_f + 1$ or the two-third nodes give the same reply. The $2n_f + 1$ non-faulty or valid replies provide the liveness of the system, that is, the message delay need to be bounded in due course. If this condition is satisfied, the leader will add the block into the blockchain and broadcasts a commit for adding the block into their respective blockchain for backup purposes.

Consensus mechanisms

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

- **Proof-of-Work (PoW):** This is the original consensus mechanism used to verify the transactions and produce new blocks in the blockchain. Mining is a complex process, and miners need to demonstrate that they can validate the transaction block. Here, the miners are financially rewarded if they perform verification. Thus, as the complexity increases in the mining process over time, the power consumption also increases. In other words, PoW is a costly process as the miners compete with each other to solve a mathematical problem.
- **Proof-of-Stake (PoS)**: In 2017, Ethereum began the process of switching from a PoW mechanism to a PoS system. The latter was designed to mitigate the limitations of PoW, in terms of energy, cost, and processing time. Specifically, it adopts a forging process rather than the mining process to validate transaction blocks.

Consensus mechanisms

- **Delegated Proof-of-Stake (DPoS):** This is a fast, efficient, flexible and most decentralized consensus mechanism. DPoS holds the power of stakeholder for the approval of voting and resolving the consensus issues in an honest and representative way. The deterministic selection of witnesses allows the transactions to be confirmed on an average of just 1 second. This consensus mechanism is designed to protect all the participants in a free, fair, and transparent environment.
- **Proof-of-Burn (PoB)**: An alternative consensus protocol for PoS and PoW. In PoB mechanism, the miners prove that they burn one cryptocurrency to create another currency, i.e., they are sent to a bitcoin address which is unsupendable. Its significance depends on the burning tokens in an unrecoverable manner. As comparative to PoW/PoS, it is easily verifiable and hard to undo.

Ripple Protocol Consensus Algorithm (RPCA)

- It is a voting based consensus algorithm proposed in the literature in order to achieve high accuracy of a correct agreement for adding a block into a blockchain over unreliable distributed network.
- RPCA achieves an agreement in the voting process.
- Every participant node in the voting process maintains a unique node list (UNL), where each node in the list is considered as trusted one.
- The protocol executes with the help of the following steps [Wang et al. [2019]]:
  - In voting process, every participant node constantly receives the transaction. If the transaction is valid, the node integrates the transactions into a set or list, called a “candidate set”.
  - Every participant node dispatches its own candidate set to other participant node as a proposal.

Ripple Protocol Consensus Algorithm (RPCA) (Continued...)

- The remaining steps are as follows [Wang et al. [2019]]:
  - The participant node receives the proposal from other nodes. The node will then check whether the sender node belongs to its UNL list or not. If it is there, the node will verify the transactions with its own local candidate set. If all are valid, the transactions will gain a vote. Only when the transaction gets more than 50% of vote, the transaction will enter into the next round.
  - Next, the participant node sends the transaction that it gains more than 50% of the votes than the others and if it increases to 60% of vote, it needs to wait until it reaches to the threshold of 80% of the votes.
  - Finally, the participant node records the transaction confirmed by the 80% UNL nodes to be added into its ledger data.

The transaction is accepted only if 80% of the votes in the UNL of a participant node agrees with it. Thus, 80% of the UNL is honest, that is, the percentage of the faulty nodes in the UNL is less that for the 20% of the UNL. When the UNL contains d number of nodes in the network, and the RPCA will maintain its correctness as long as $n_f \leq \frac{d-1}{5}$, i.e., $d \geq 5n_f + 1$ where n_f is the Byzantine failure persisted nodes in the network.

Table: Blockchain consensus mechanisms and their applications

Consensus mechanism	**Concept**	**Resource**	**Applications**
Ripple	Voting in multiple rounds	No resource	XRP ledger xrp [2014]
PBFT	Voting	No resource	Tendermint Kwon [2014]
PoW	Hashing	Computations	Bitcoin Nakamoto [2009] Ethereum Wood et al. [2014]
PoS	Digital signatures	Currency	PeerCoin King and Nadal [2012] SnowWhite Bentov et al. [2016] Ouroboros Kiayias et al. [2017]
DPoS	Voting	Currency	BitShared Ark EOS
PoB	Address suspension	Currency	SlimCoin sli [2014]

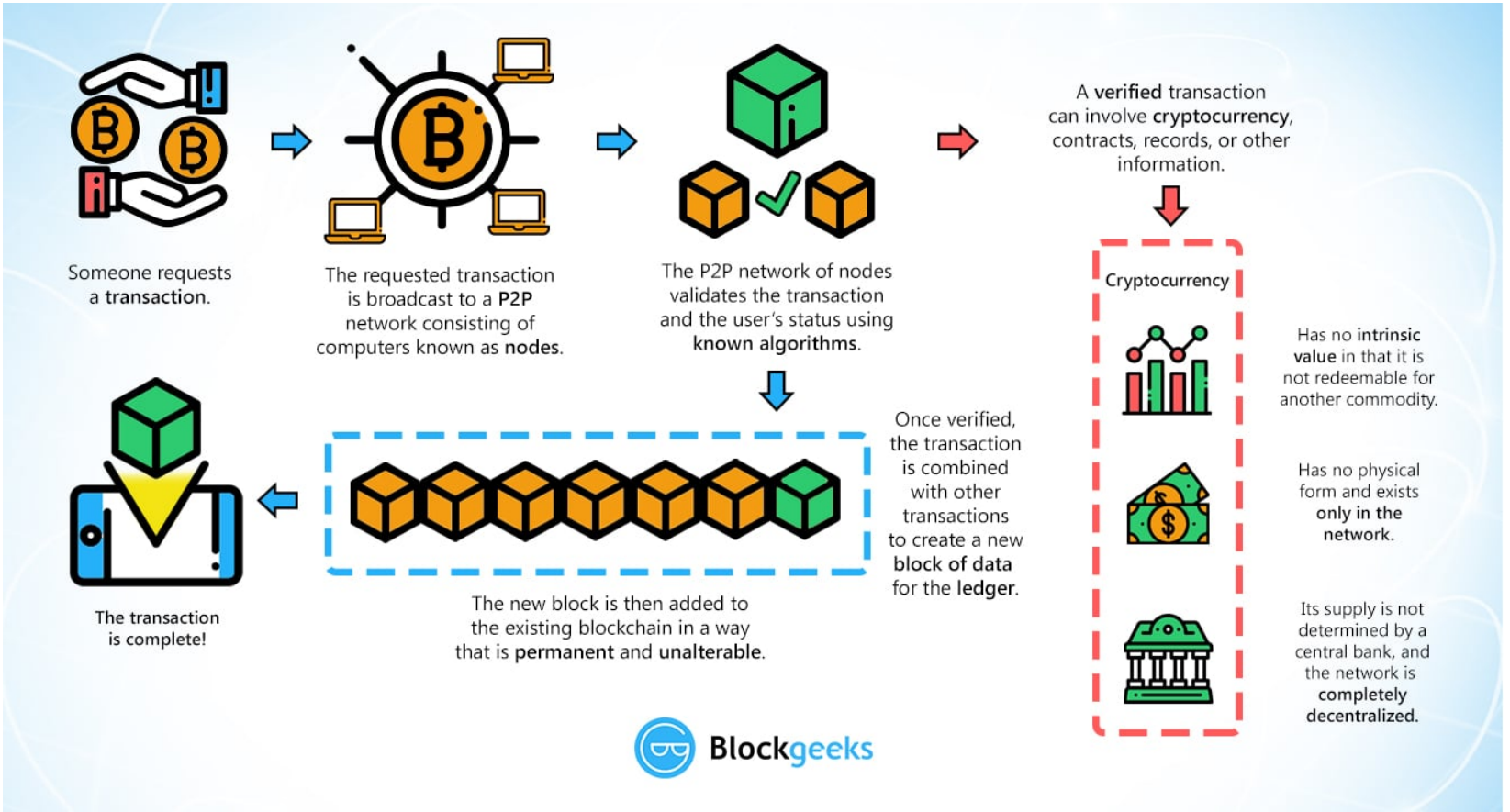
Ref. Anusha Vangala, Ashok Kumar Das, Neeraj Kumar, and Mamoun Alazab. "Smart Secure Sensing for IoT-Based Agriculture: Blockchain Perspective," in *IEEE Sensors Journal*, 2020, DOI: 10.1109/JSEN.2020.3009382.

Table: Attacks on consensus mechanisms

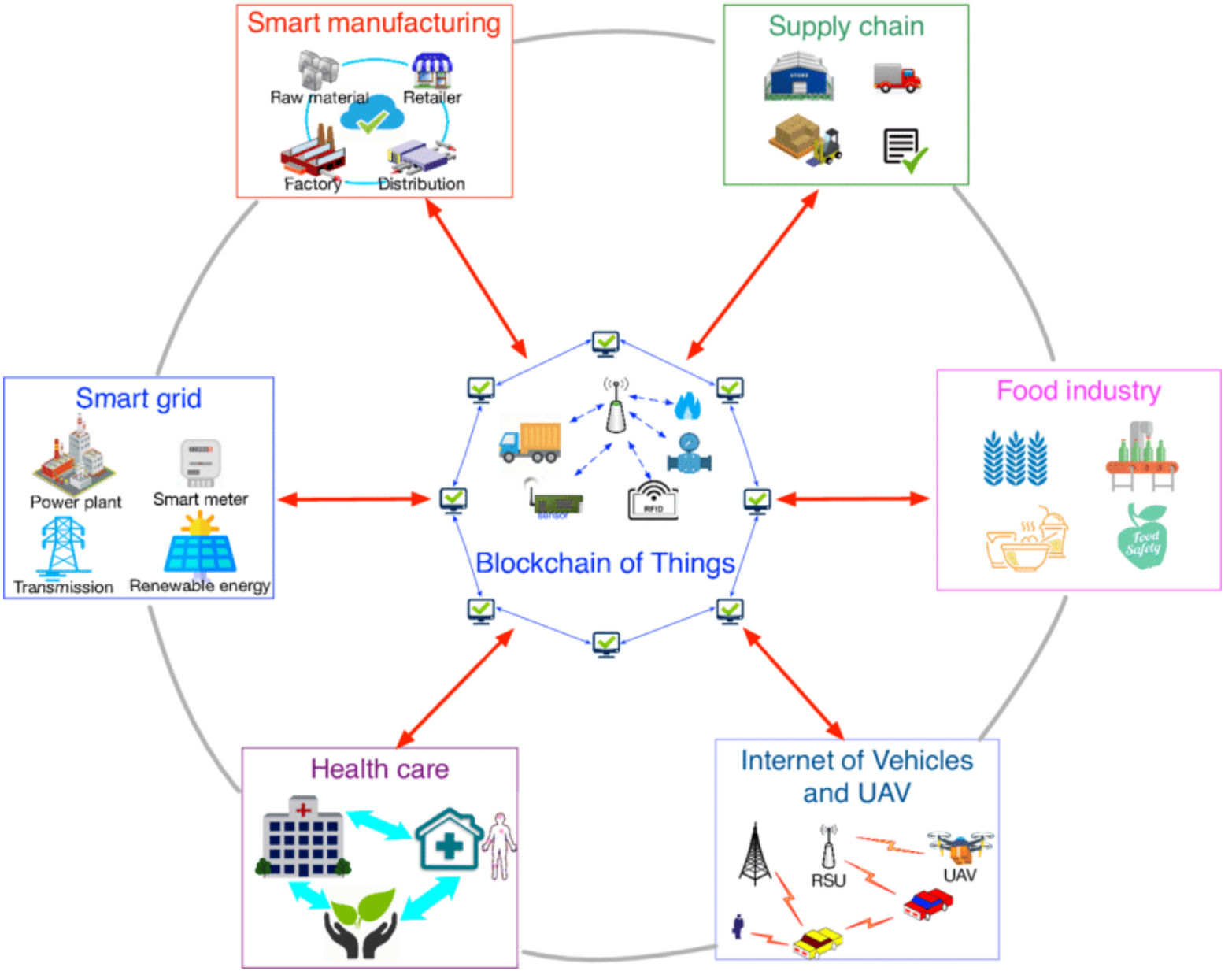
Attack	Affected consensus protocols	Description
Double spending	Most protocols	Repeated usage of token
Selfish mining	PoW	Gain profits by generating blocks privately in a mining pool
Nothing at stake	PoS	Blocks added to all branches in a fork
Bribe attack	PoS	Honest nodes are given incentive to add blocks on private fork
Stake bleeding attack	PoS	Broadcast transactions copied from main chain onto private fork to earn extra fees and increase stake
Fake stake attack	PoS	Increase the smaller valued stakes to higher valued stakes

Ref. S. Zhang and J. Lee, "A Group Signature and Authentication Scheme for Blockchain-Based Mobile-Edge Computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4557-4565, 2020. (2019 SCI Impact Factor: 9.936).

Block creation and addition using consensus mechanism in a blockchain



Applications of Blockchain Technology



References

Slimcoin : A Peer-to-Peer Crypto-Currency with Proof-of-Burn - Mining without Powerful Hardware, 2014. URL https://github.com/slimcoin-project/slimcoin-project.github.io/raw/master/whitepaperSLM.pdf.

The XRP Ledger, 2014. URL https://xrpl.org/consensus-principles-and-rules.html.

Iddo Bentov, Rafael Pass, and Elaine Shi. Snow White: Provably Secure Proofs of Stake. *IACR Cryptology ePrint Archive*, 2016(919), 2016.

Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Advances in Cryptology (CRYPTO'17)*, pages 357–388, Santa Barbara, CA, USA, 2017.

Sunny King and Scott Nadal. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake, 2012.

Jae Kwon. Tendermint: Consensus without mining. *Self-Published Paper (Draft v.0.6)*, 1(11), 2014. URL https://tendermint.com/static/docs/tendermint.pdf.

Satoshi Nakamoto. Bitcoin open source implementation of p2p currency. *P2P foundation*, 18, 2009.

X. Wang, P. Zeng, N. Patterson, F. Jiang, and R. Doss. An Improved Authentication Scheme for Internet of Vehicles Based on Blockchain Technology. *IEEE access*, 7:45061–45072, 2019.

Gavin Wood et al. Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Impact on Blockchain-based AI/ML-enabled Big data analytics for Cognitive Internet of Things environment

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
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/`

AI-enabled Blockchain-based Big data analytics in Cognitive Internet of Things (CIoT)

Reference: Ankush Mitra, Basudeb Bera, Ashok Kumar Das, Sajjad Shaukat Jamal, and Ilsun You. "Impact on Blockchain-based AI/ML-enabled Big data analytics for Cognitive Internet of Things environment," in ***Computer Communications (Elsevier)***, Vol. 197, pp. 173-185, January 2023, DOI: 10.1016/j.comcom.2022.10.010. (2022 SCI Impact Factor: 6)

Cognitive Internet of Things (CIoT)

- Internet of Things (IoT) has become an emerging technology due to a huge enhancement of Information and Communications Technology (ICT). IoT compromises a large number of smart devices, called IoT devices, which can be either physical or virtual objects.
- Cognitive Internet of Things (CIoT) has now become a new network model which is enhancement of IoT. Likewise IoT network, physical and virtual objects are part of CIoT, which work with minimum human intervention and they also communicate with each other based on a “context-aware perception-action cycle”.
- CIoT is thus considered as a field of science where IoT and cognitive computing are applied to make the IoT systems smarter. It provides some kind of thinking ability to the IoT systems.

Cognitive Internet of Things (CIoT)

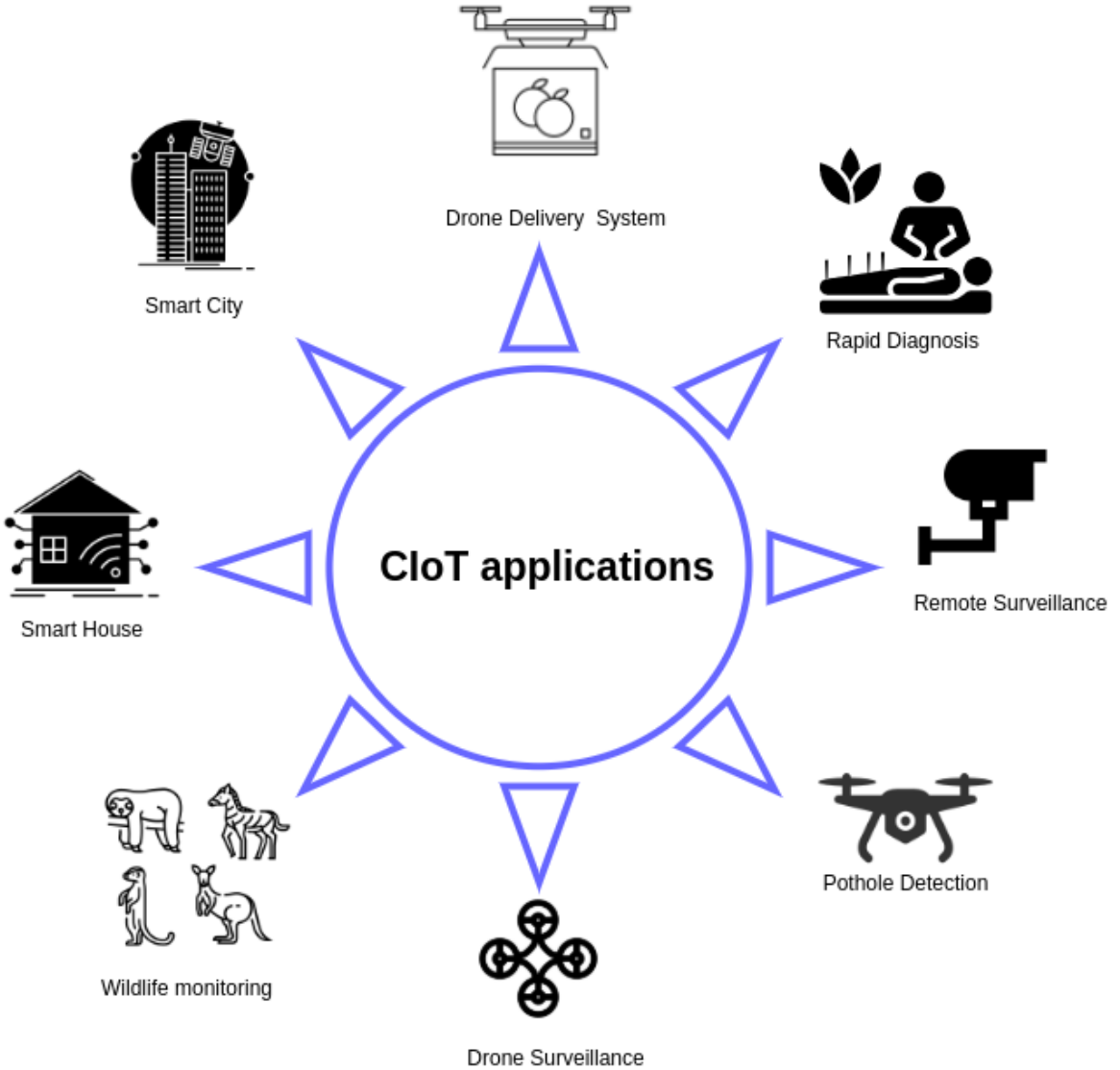


Figure: Various CIoT-enabled applications

Attacks related to AI/ML

AI/ML security becomes an emerging topic in the computer science field in order to make correct and accurate predictions on non-poisonous (corrupted) data. For instance, typically a huge volume of data generated by the IoT smart devices in CIoT can be stored in cloud server(s). As the cloud servers are semi-trusted, there is a possibility by the insider attackers of the cloud servers to perform several attacks.

- *Adversarial input attacks*: These are specially crafted on inputs that have been developed with the aim of being reliably mis-classified in order to evade detection.
- *Data poisoning attacks*: The attacker can then insert false data, alter the label of the data, and also remove the data or insert random noise to the data to poison the training data.
- *Model attacks*: In such type of attack, an attacker may pollute the model's hyper-parameters (that is, the parameters that are learned using AI/ML).
- *Model stealing attacks*: Such kinds of attack scenarios are used to steal or duplicate models or recover training data membership.

Convolution Neural Networks (CNN)

Convolution Neural Networks (CNN) is a deep neural network that is generally used in computer vision.
CNN mainly contains three types of layers:

- *Convolution layer*: It is a convolution tool that splits various features of the input images for analysis.
- *Pooling layer*: The main goal of this layer is that it decreases the size of the convoluted feature map in order to reduce computation costs.
- *Fully connected layer*: This layer applies the output of the convolution layer for making prediction about the best description for the inputs (images).

Convolution Neural Networks (CNN)

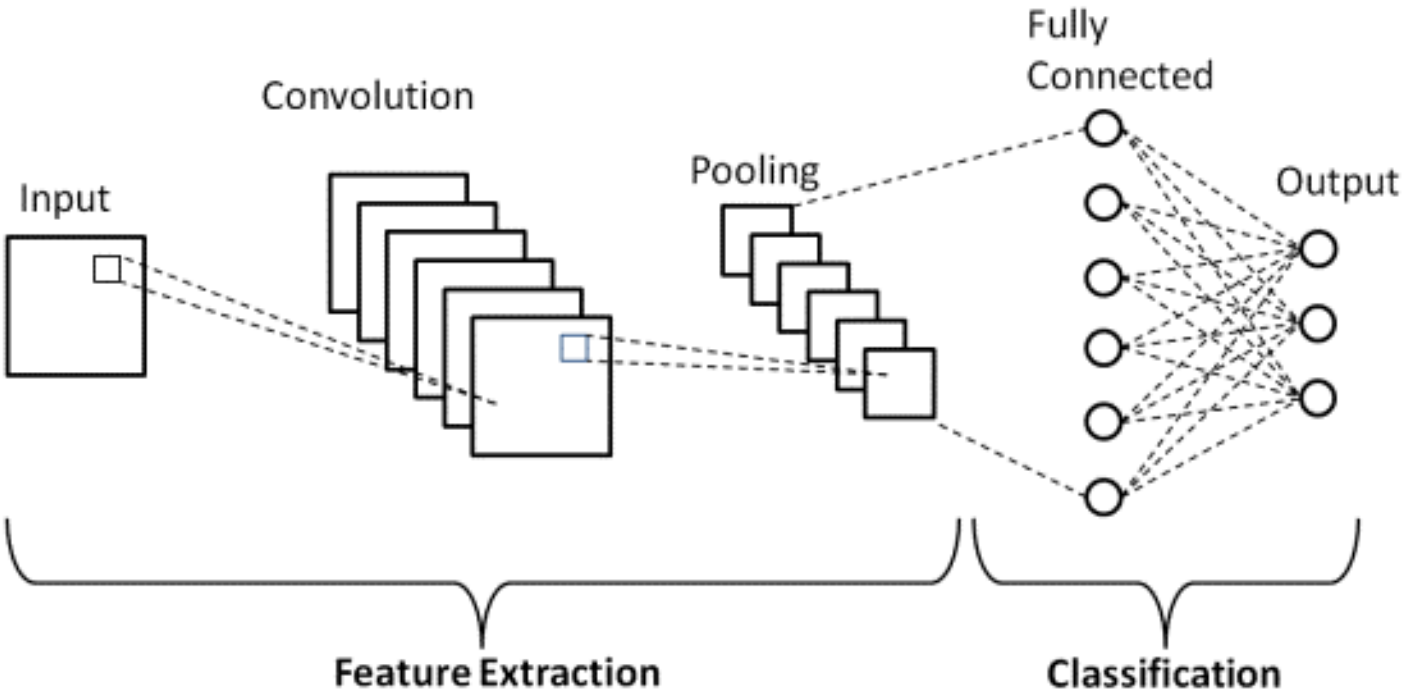
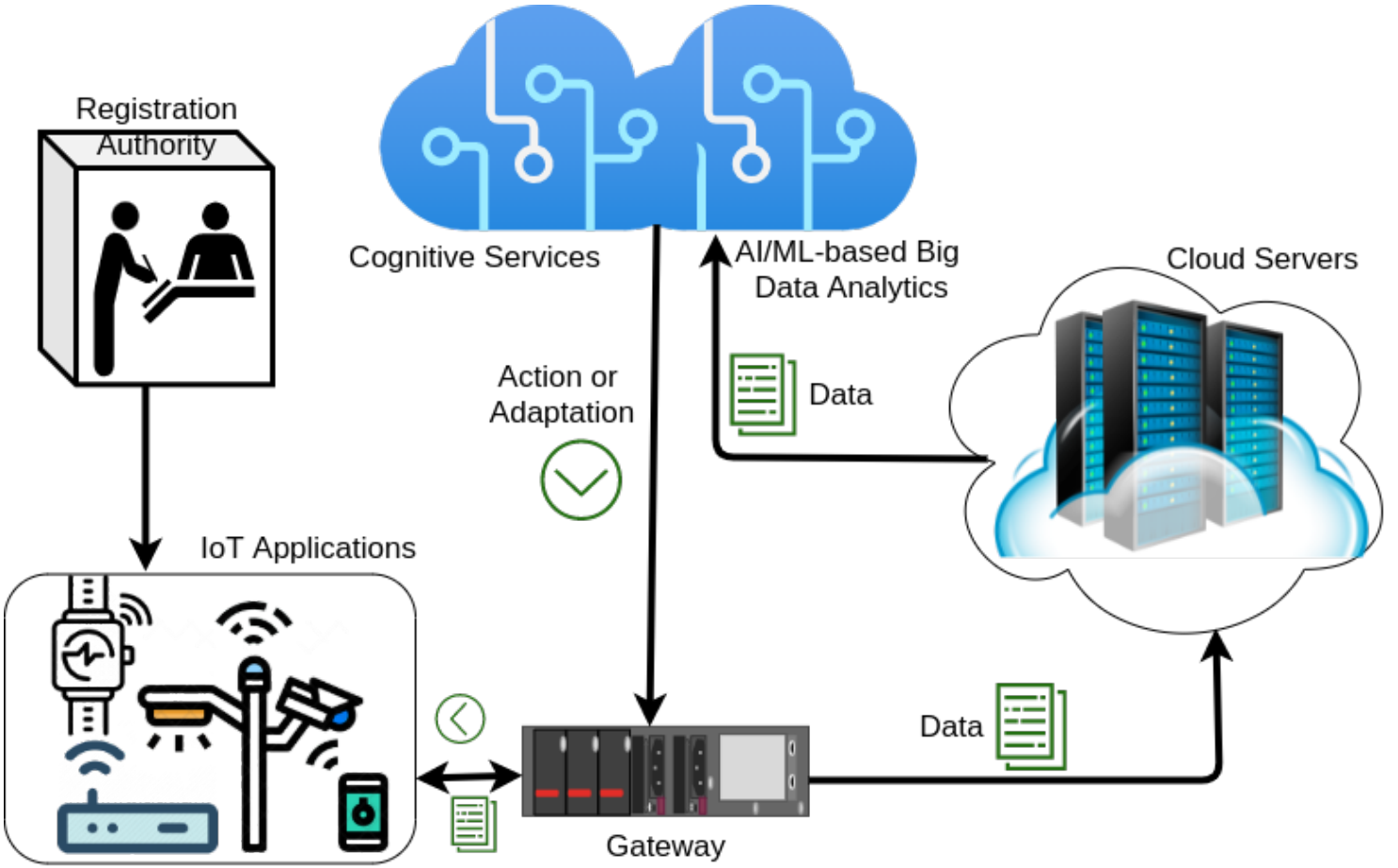
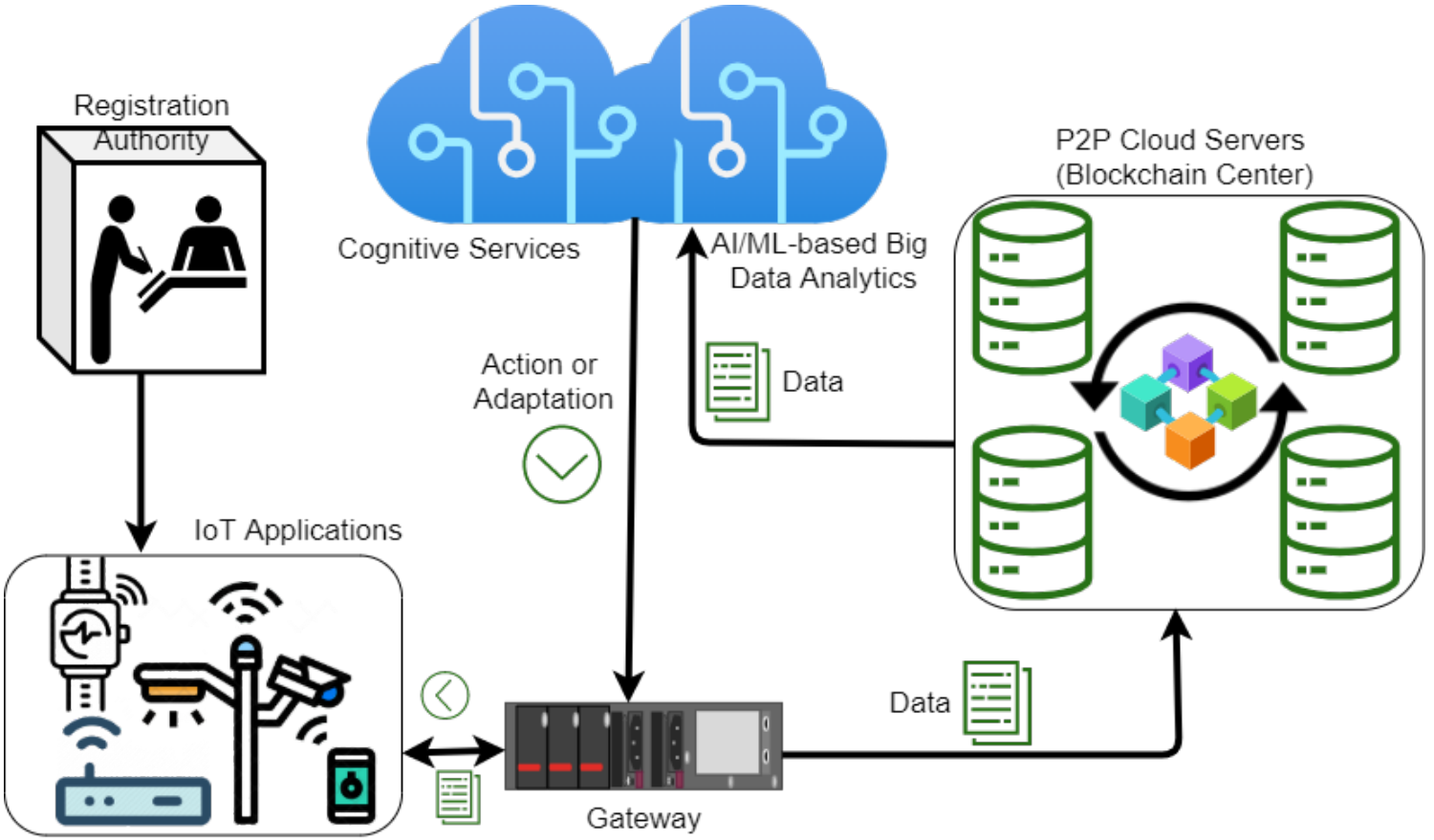


Figure: CNN architecture

CIoT network model without blockchain



Blockchain-envisioned CIoT network model

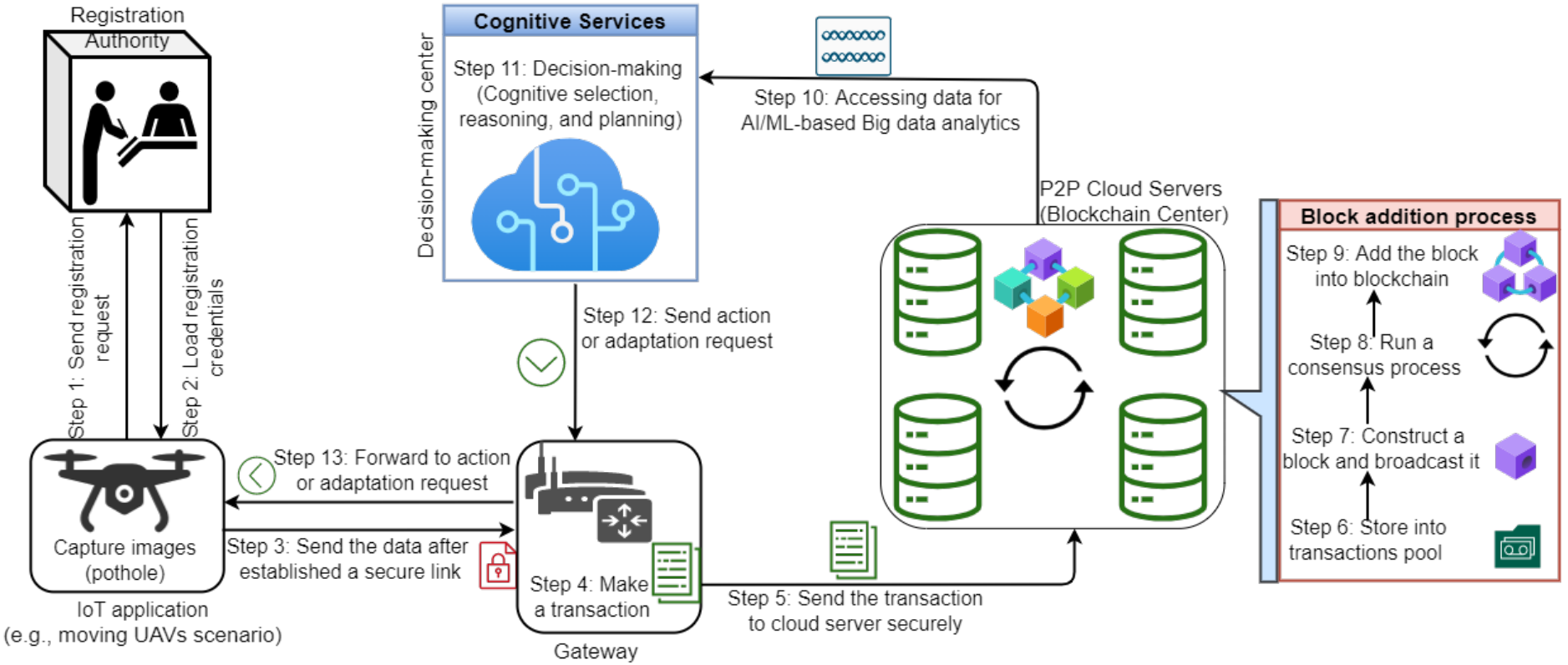


AI-enabled Blockchain-based Big data analytics in CIoT

Block Header
Block Version
Previous Block Hash (PBH)
Merkle Tree Root (MTR)
Timestamp
Signer's Public Key
Block Payload (Transactions)
(Transactions (Tx), ECDSA.Sig(Tx))
Current Block Hash (CBH)

Figure: A block structure in public blockchain

Overall process in blockchain-based AI/ML-enabled Big data analytics



Experimental Results

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Data Sets:

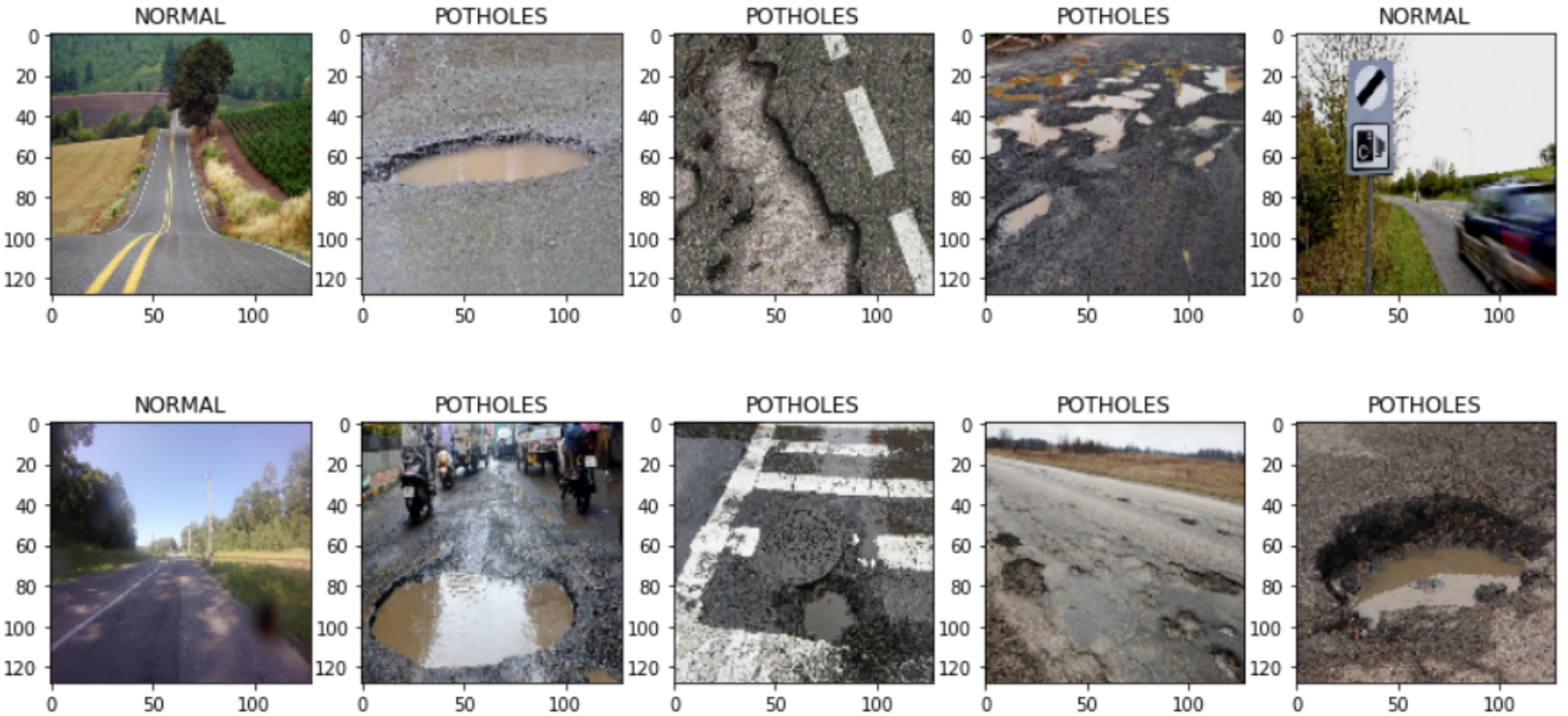
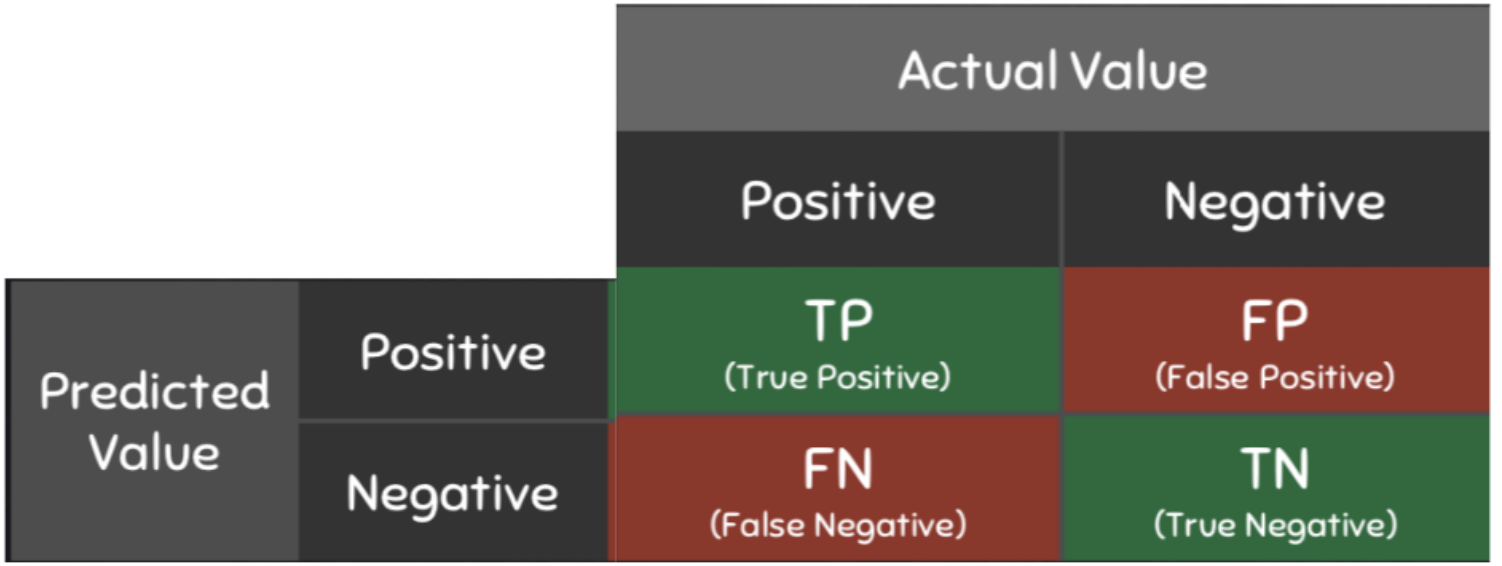


Figure: Pothole data sets (Source: `https://www.kaggle.com/atulyakumar98/pothole-detection-dataset`)

Metrics Used in Experiment



- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Figure: Structure of a confusion matrix

Metrics Used in Experiment

- *Accuracy*:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Recall*:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- *Precision*:

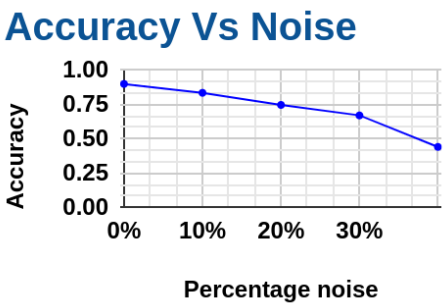
$$\text{Precision} = \frac{TP}{TP + FP}$$

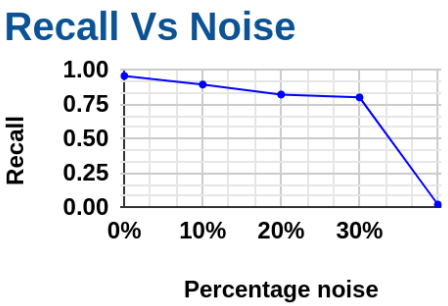
- *F1 score*:

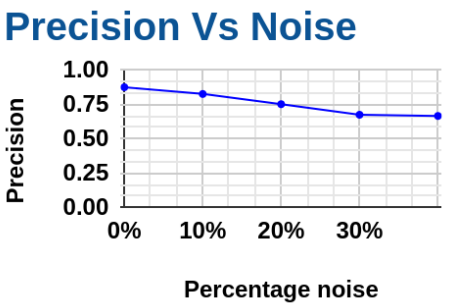
$$\text{F1 score} = \frac{2 * TP}{2 * TP + FP + FN}$$

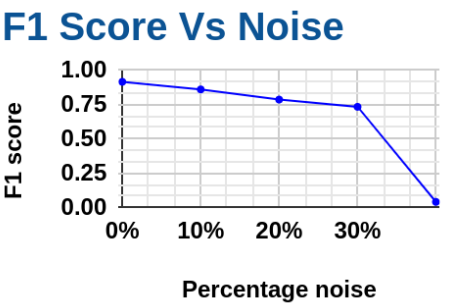
Experimental results under salt noise insertion attacks

Salt noise is a very common type noise seen in the images. It is also known as impulsive noise. In this part of our experiment, we consider that if an adversary tries to inject the salt noise on the data, it effects the machine learning model significantly.



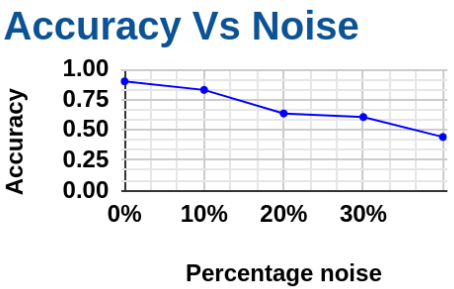


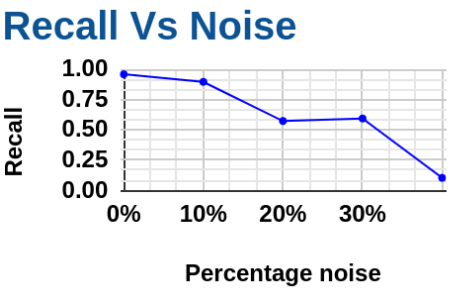


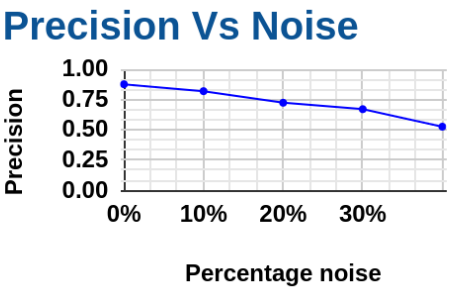


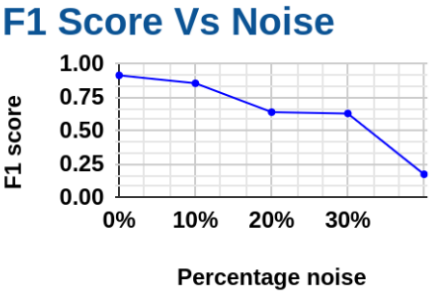
Experimental results under Gaussian noise insertion attacks

Gaussian noise is another standard noise that has used in our experiments. It is referred as a statistical noise having a probability density function equal to that of the Gaussian distribution.



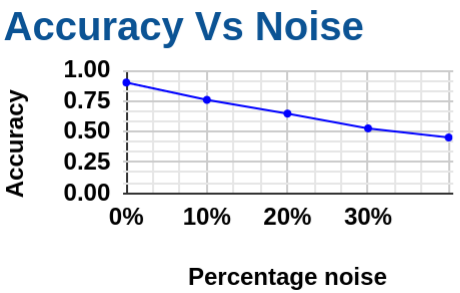


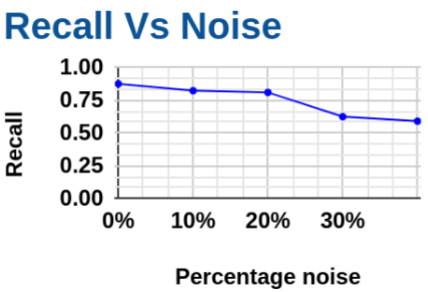


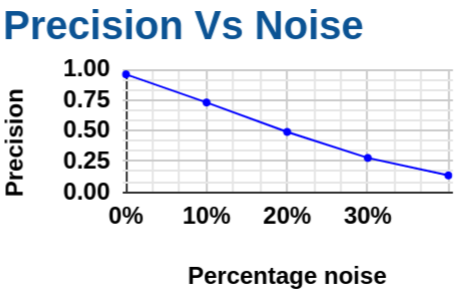


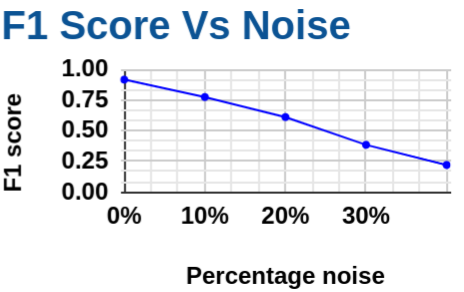
Experimental results under Poisson noise insertion attacks

Poisson noise is a standard noise that is used in the experimental results to check the effect of accuracy, recall, precision and F1 score under the ML model.



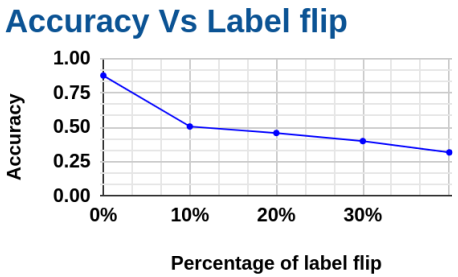


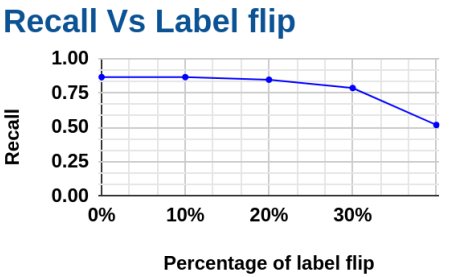


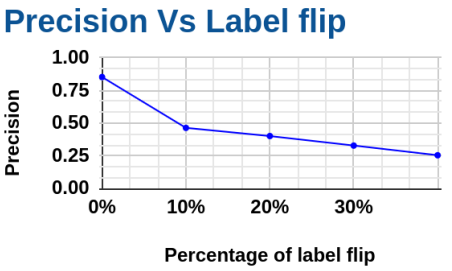


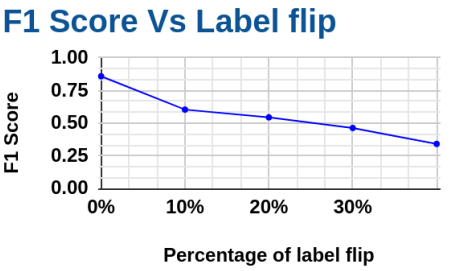
Experimental results under label flipping attacks

In a label filliping attack, if the adversary tries to alter the labels of the original data, how it can effect on the overall performance of the ML model.









Performance of ML model without data poisoning attacks

- The effect of no data poising attacks under the ML model.
- 0% noise insertion means that no attacks on the data.

Table: Experimental results without any data poisoning attacks

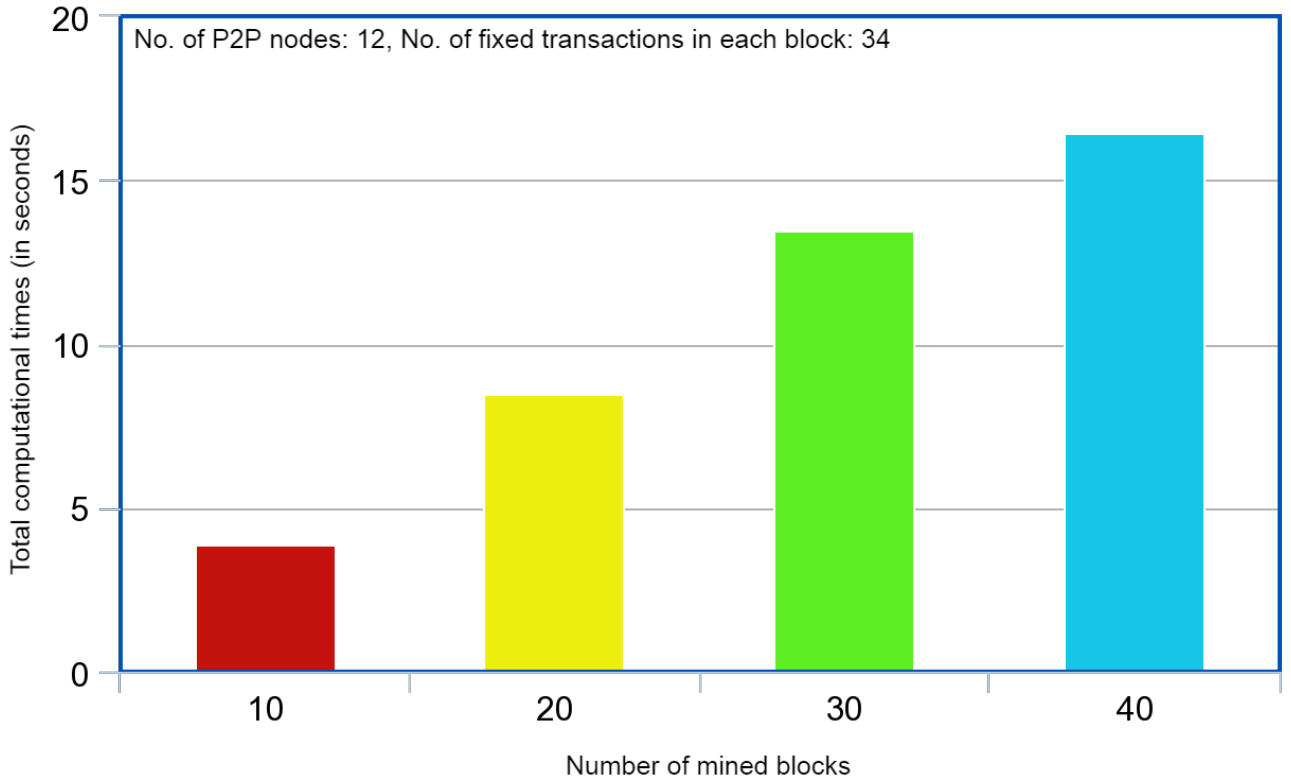
Accuracy	Recall	Precision	F1 Score
0.8764	0.8648	0.8533	0.859

Blockchain Implementation

- We created a blockchain system with the help of node.js script. We also created the virtual distributed blockchain network with twelve P2P (cloud) servers.
- All the servers work on the localhost, but they use different ports for communication among them.
- To simulate the blockchain system, we have used a host computer having the configuration as: “OS: Ubuntu 18.04 LTS, Processor: Intel i5-8400 (2.80GHz), Memory: 7.6 GiB, OS Type: 64 bit, Disk Type: HDD, Disk Size: 152.6 GB”.
- We have considered two different cases.

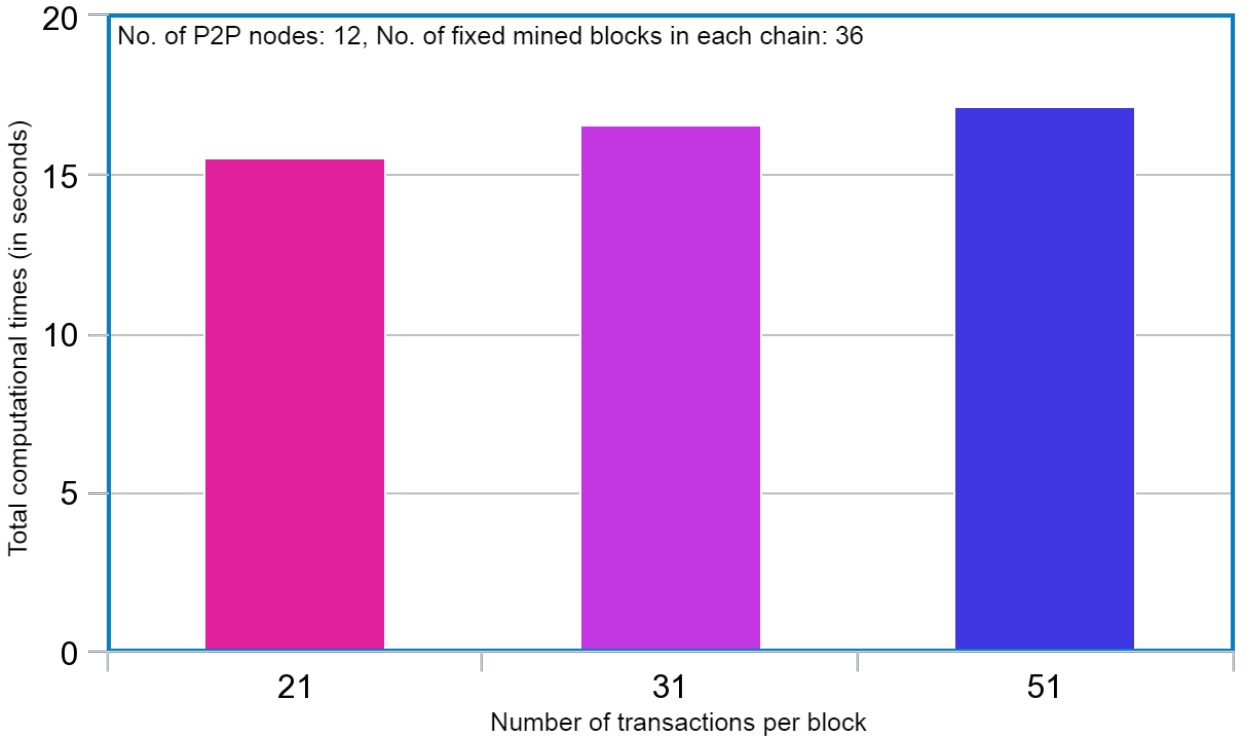
Blockchain Implementation

Case 1. In this case, we fixed the number of transactions to a fixed value as 34 (i.e., the size of each block will be the same) and recorded the time of the blocks addition when we varied the number of blocks in the blockchain. The simulation results show that when the size of each block is fixed, the total computation time linearly varies with the number of blocks mined in the blockchain.



Blockchain Implementation

Case 2. In this case, we fixed the total number of blocks in the blockchain to a fixed value as 36, and recorded the time of the blocks addition when we varied the size of the blocks (i.e., we varied the number of transactions in a block). The simulation results illustrate that when the number of blocks into the blockchain is fixed, the total computation time varies linearly with the number of transactions in each block in the blockchain.



Thank You
For Your Attention