



Department of CSE(HONORS)

CRYPTANALYSIS & CYBER DEFENSE 21CS304IRA

Topic:

MODES OF OPERATION

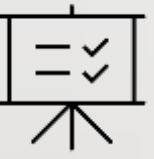
Session -8

AIM OF THE SESSION



To familiarize students with the basic concepts of various Modes of operation in block cipher.


INSTRUCTIONAL OBJECTIVES



The objective of this session is to introduce

1. Basic concepts of various Modes of operation in block cipher.
2. Demonstrate Block Cipher Modes of Operation
3. Describe Block Cipher Modes of Operation List out the Block
4. Describe the Modes of Operation

LEARNING OUTCOMES



At the end of this session, students are expected to know

- Define Block Cipher
- Summarize Block Cipher Modes of Operation

In this module we will discuss basic concepts of different modes of operation in cryptography. This module cover in depth analysis of working principle of these mode of operation with their computational complexity.

This Session provides an overview of modes of operation. Much of the theory of public-key cryptosystems is based on number theory. This session is designed to provide a comprehensive understanding of different modes of operation used in block ciphers. Block ciphers are widely used cryptographic algorithms that encrypt fixed-size blocks of data. The proper selection and understanding of block cipher modes of operation are essential for secure and efficient encryption of data.

SESSION DESCRIPTION

MODES OF OPERATION

- Modes of operation are methods used to transform a block cipher into a stream cipher or a hashing algorithm.
- These modes allow a block cipher to process larger amounts of data than the block size of the cipher, by dividing the data into blocks and applying the block cipher to each block in turn.
- Some commonly used modes of operation include Electronic Codebook (ECB), Cipher Block Chaining (CBC), Counter (CTR), and Galois/Counter Mode (GCM).

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

➤ Electronic Codebook (ECB)

- In the Electronic Codebook (ECB) mode of operation:
- The plaintext is divided into fixed-size blocks, such as "Block 1," "Block 2," and so on.
- Each block is encrypted independently using the encryption algorithm "E" with the same key "K," producing the corresponding ciphertext block.

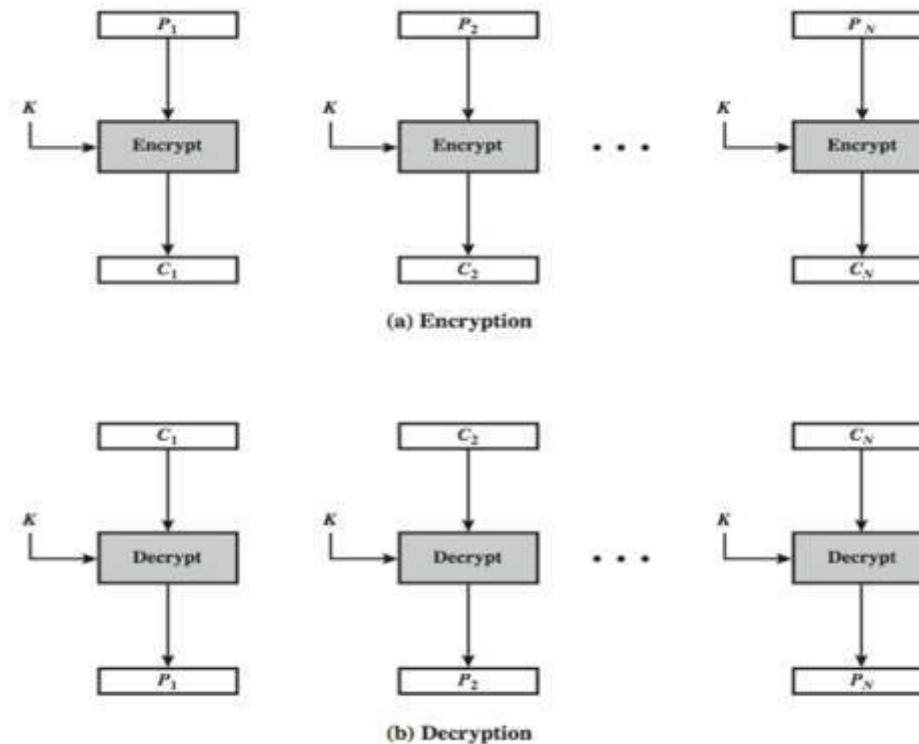
SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

- This encryption process is repeated for each block of the plaintext, resulting in a series of ciphertext blocks.
- During decryption, each ciphertext block is decrypted independently using the decryption algorithm "D" with the same key "K," yielding the corresponding plaintext block.
- The decryption process is performed for each ciphertext block, allowing for the reconstruction of the original plaintext blocks.

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION



ECB Mode of Operation.

Note: Copy Rights of this figure are reserved for the original author

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

- **Advantages and Limitations of ECB**
 - The ECB method is ideal for a short amount of data, such as an encryption key.
 - The most significant characteristic of ECB is that if the same b-bit block of plaintext appears more than once in the message, it always produces the same ciphertext.
 - Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

- Simple way of block cipher.
- For lengthy messages, the ECB mode may not be secure. If the message is highly structured, it may be possible for a cryptanalyst to exploit these regularities.
- Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

- Cipher Block Chaining Mode (CBC)
 - The plaintext is divided into fixed-size blocks, such as "Block 1," "Block 2," and so on.
 - An Initialization Vector (IV) is generated, serving as the initial input for the encryption process.
 - For each plaintext block, the XOR operation is performed between the plaintext block and the previous ciphertext block or the IV.

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

- The result of the XOR operation is then encrypted using the encryption algorithm "E" with the same key "K," producing the ciphertext block.
- The ciphertext block becomes the input for the XOR operation in the next iteration.
- This process is repeated for each block of the plaintext, resulting in a series of ciphertext blocks.

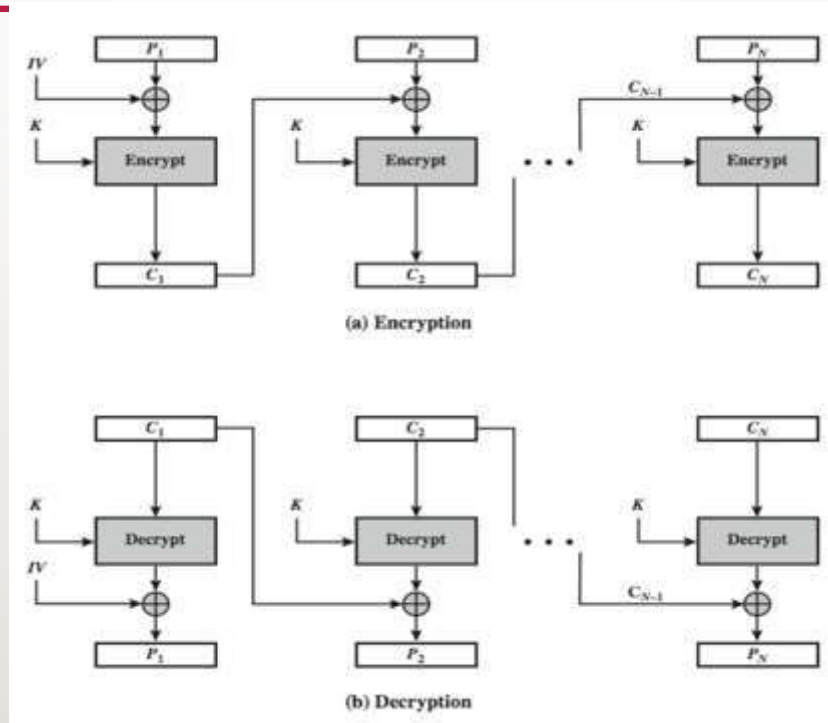
SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

- During decryption, the ciphertext blocks are decrypted using the decryption algorithm "D" with the same key "K."
- The decrypted ciphertext blocks are then XORed with the previous ciphertext block or the IV to obtain the corresponding plaintext blocks.
- The decryption process is performed for each ciphertext block, allowing for the reconstruction of the original plaintext blocks.

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION



CBC Mode of Operation.

Note: Copy Rights of this figure are reserved for the original author

SESSION DESCRIPTION

BLOCK CIPHER MODES OF OPERATION

➤ Advantages and Limitations of CBC

- CBC works well for input greater than b bits.
- CBC is a good authentication mechanism.
- Better resistive nature towards cryptanalysis than ECB.
- Parallel encryption is not possible since every encryption requires previous cipher.

SELF-ASSESSMENT QUESTIONS

1. In ECB mode, how is the plaintext divided for encryption?

- a) Into fixed-size blocks
- b) Into variable-sized blocks
- c) Into a single block
- d) It is not divided

2. What is the primary disadvantage of ECB mode?

- a) Lack of parallelism
- b) Vulnerability to plaintext attacks
- c) Slow encryption process
- d) None of the above

SELF-ASSESSMENT QUESTIONS

3. In ECB mode, how are the blocks encrypted?

- a) Each block is encrypted independently using the same key
- b) Each block is encrypted using a different key
- c) Blocks are not encrypted individually
- d) Encryption depends on the size of the block

4. In ECB mode, what happens if two plaintext blocks are identical?

- a) Both blocks are encrypted differently
- b) Both blocks are encrypted with the same ciphertext
- c) Only the first block is encrypted
- d) Encryption fails

SUMMARY

Throughout the session, students will gain hands-on experience with implementing and analyzing various block cipher modes of operation through practical exercises and programming assignments. By the end of the course, students will have a solid understanding of different block cipher modes and be able to make informed decisions regarding their selection and use in real-world cryptographic applications.

TERMINAL QUESTIONS

1. Demonstrate the basic concept of the ECB mode of operation.
2. Summarize how the plaintext is divided and encrypted in the ECB mode.
3. List the advantages of using ECB mode for encryption?
4. Illustrate the basic concept of the CBC (Cipher Block Chaining) mode of operation in cryptography.
5. List the advantages of using CBC mode for encryption?

REFERENCES FOR FURTHER LEARNING OF THE SESSION

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 5th edition.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier, Second Edition, John Wiley & Sons, Inc., 2015.
3. Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, by Hamid R. Nemati and Li Yang, IGI Global, 2011
4. Forouzon B, "Cryptography and Network Security," Indian Edition, TMH (2010).

THANK YOU



Team – CACD



Department of CSE(HONORS)

CRYPTANALYSIS & CYBER DEFENSE 21CS304IRA

Topic:

MODES OF OPERATION

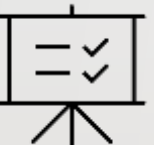
Session -9

AIM OF THE SESSION




To make the students well versed with the basic concepts of various Modes of operation in block cipher.

INSTRUCTIONAL OBJECTIVES



The objective of this session is to introduce basic concepts of various Modes of operation in block cipher and demonstrate Block Cipher Modes of Operation.

LEARNING OUTCOMES



At the end of this session, students are expected to know

- Define Block Cipher
- Demonstrate CFB, OFB & Counter Block Cipher Modes of Operation

This module demonstrates block cipher modes of operation. It also helps to identify which ciphers operate as block ciphers and stream ciphers. Principles of Pseudorandom Number Generators are listed and types of Pseudorandom Number Generators are discussed. Finally, Stream Ciphers and its example RC4 algorithm are demonstrated.

This Session provides an overview of modes of operation. Much of the theory of public-key cryptosystems is based on number theory. This session is designed to provide a comprehensive understanding of different modes of operation used in block ciphers. Block ciphers are widely used cryptographic algorithms that encrypt fixed-size blocks of data. The proper selection and understanding of block cipher modes of operation are essential for secure and efficient encryption of data.

SESSION DESCRIPTION

MODES OF OPERATION

Cipher Feedback Mode

- In CFB mode, the previous ciphertext block is used to encrypt the next plaintext block, creating a feedback loop that enables the encryption and decryption processes. CFB mode operates on individual bits or bytes of data and converts a block cipher into a stream cipher.
- To begin, an Initialization Vector (IV) is generated, which acts as the initial input for the encryption process. The IV should be unique for each encryption session and must be kept secret.
- In the feedback mechanism, the IV (or the previous ciphertext block) is encrypted using the block cipher algorithm. The resulting ciphertext is then XORed with the corresponding plaintext bits (or bytes) to produce the encrypted output.

SESSION DESCRIPTION

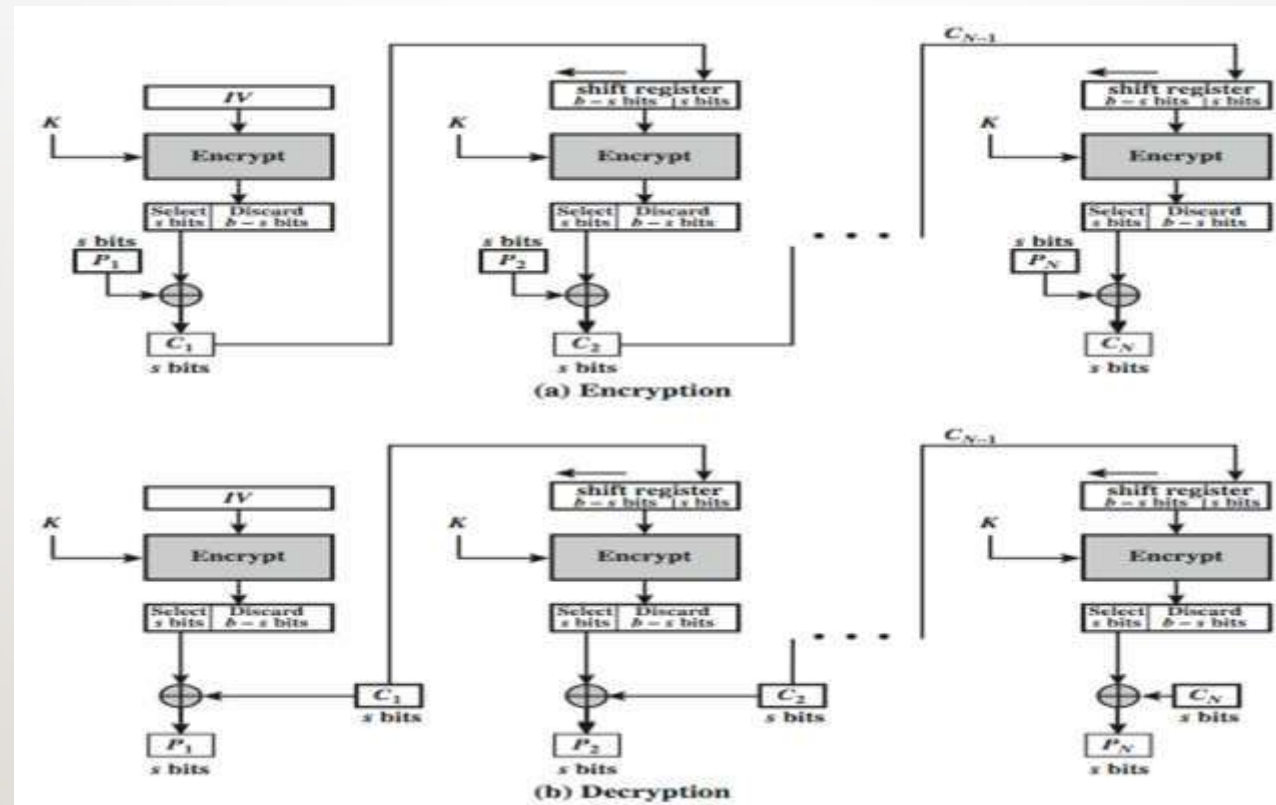
MODES OF OPERATION

Cipher Feedback Mode

- The encryption process starts with the IV, and the block cipher algorithm takes the IV as input, generating a block of ciphertext. This ciphertext is XORed with the first plaintext bits (or bytes), producing the encrypted output.
- In subsequent iterations, the previous ciphertext block becomes the input for the block cipher algorithm. The block cipher encrypts the previous ciphertext block, and the resulting ciphertext is XORed with the next plaintext bits (or bytes) to produce the next encrypted output. This feedback loop ensures that each block of plaintext is encrypted using the previous ciphertext block.
- During decryption, the same process is followed, using the IV (or the previous ciphertext block) as the input for the block cipher algorithm. The ciphertext is XORed with the corresponding encrypted bits (or bytes) to produce the decrypted output.

SESSION DESCRIPTION

MODES OF OPERATION



CFB Mode of Operation

Copy rights of this diagram belongs to original author

SESSION DESCRIPTION

MODES OF OPERATION

Advantages and Limitations of CFB

- With 8-bit CFB, if a byte is lost, one byte of plaintext will be lost and the next 8 bytes will be garbled. After that, the plaintext will decrypt properly.
- If a byte is added to the ciphertext, a byte of garbage will be added, and the following 8 bytes will be garbled, the rest will be ok.
- Random stream can no longer be computed in advance
- Since, there is some data loss due to use of shift register, thus it is difficult for applying cryptanalysis.

SESSION DESCRIPTION

MODES OF OPERATION

Output feedback Mode of Operation

- Initialization: An IV (Initialization Vector) is generated, which serves as the initial input for the encryption process. The IV should be unique for each encryption session and must be kept secret. A secret key is also chosen, which remains constant throughout the encryption and decryption process.

SESSION DESCRIPTION

MODES OF OPERATION

Keystream Generation:

The IV is encrypted using the block cipher algorithm, producing the initial keystream block. The keystream block is stored. To generate subsequent keystream blocks, the previous keystream block is encrypted using the block cipher algorithm, creating a feedback loop. The keystream blocks are generated independently of the plaintext, allowing for parallel encryption and decryption.

SESSION DESCRIPTION

MODES OF OPERATION

Encryption Process:

- The keystream blocks are XORed with the corresponding plaintext blocks to produce the ciphertext. Each bit or byte of the plaintext is combined with the corresponding bit or byte of the keystream using the XOR operation.
- The resulting ciphertext is stored. Decryption Process: The same keystream generation process is followed as in encryption, using the IV and the secret key. The keystream blocks are XORed with the corresponding ciphertext blocks to produce the plaintext. Each bit or byte of the ciphertext is combined with the corresponding bit or byte of the keystream using the XOR operation.

SESSION DESCRIPTION

MODES OF OPERATION

Random access encryption and decryption:

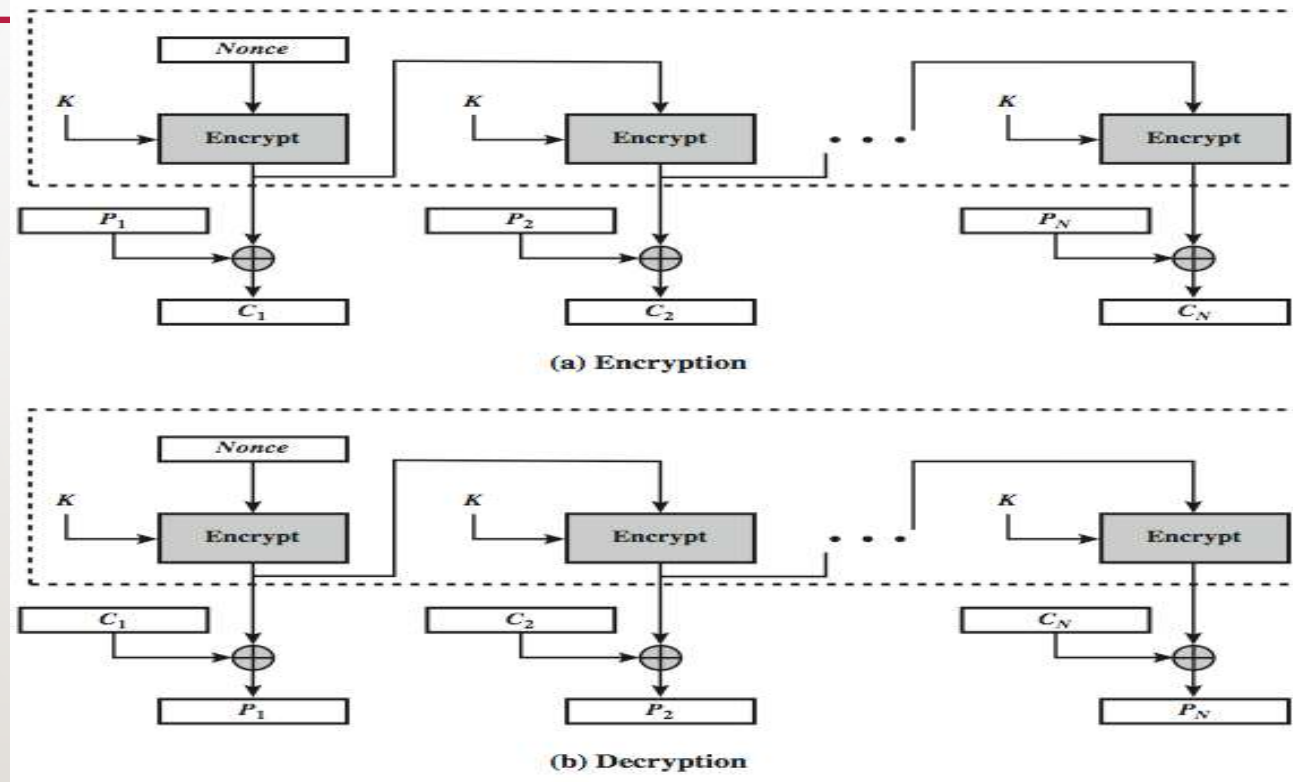
Any ciphertext block can be decrypted without the need for preceding blocks.

Efficient parallel processing:

- The keystream blocks can be generated independently, allowing for efficient parallel encryption and decryption.

SESSION DESCRIPTION

MODES OF OPERATION



OFB Mode of Operation

Copy rights of this diagram belongs to original author

SESSION DESCRIPTION

MODES OF OPERATION

Advantages and Limitations of OFB

- One advantage of the OFB method is that bit errors in transmission do not propagate. For example, if a bit error occurs in CI, only the recovered value of PI is affected; subsequent plaintext units are not corrupted.
- With CFB, CI also serves as input to the shift register and therefore causes additional corruption downstream.
- The disadvantage of OFB is that it is more vulnerable to a message stream modification attack than is CFB.

SESSION DESCRIPTION

MODES OF OPERATION

Counter Mode of Operation

- Counter (CTR) mode of operation is a cryptographic technique used to transform a block cipher into a stream cipher.
- It operates by encrypting a unique counter value for each plaintext block, generating a stream of pseudorandom bits or bytes that are then XORed with the plaintext to produce the ciphertext.
- The key idea behind CTR mode is to create a unique counter value for each block, ensuring that the encryption process remains deterministic and allowing for parallel encryption and decryption.

SESSION DESCRIPTION

MODES OF OPERATION

Counter Mode of Operation

Here's how CTR mode works:

- Initialization: An Initialization Vector (IV) is generated, which serves as the initial value for the counter. The IV should be unique for each encryption session and must be kept secret. A secret key is chosen, which remains constant during the encryption and decryption process.
- Counter Generation: A counter value is created for each plaintext block. The counter value can be a simple incrementing value or a more complex scheme, depending on the specific implementation. The counter value is usually combined with the IV and the block index to ensure uniqueness.

SESSION DESCRIPTION

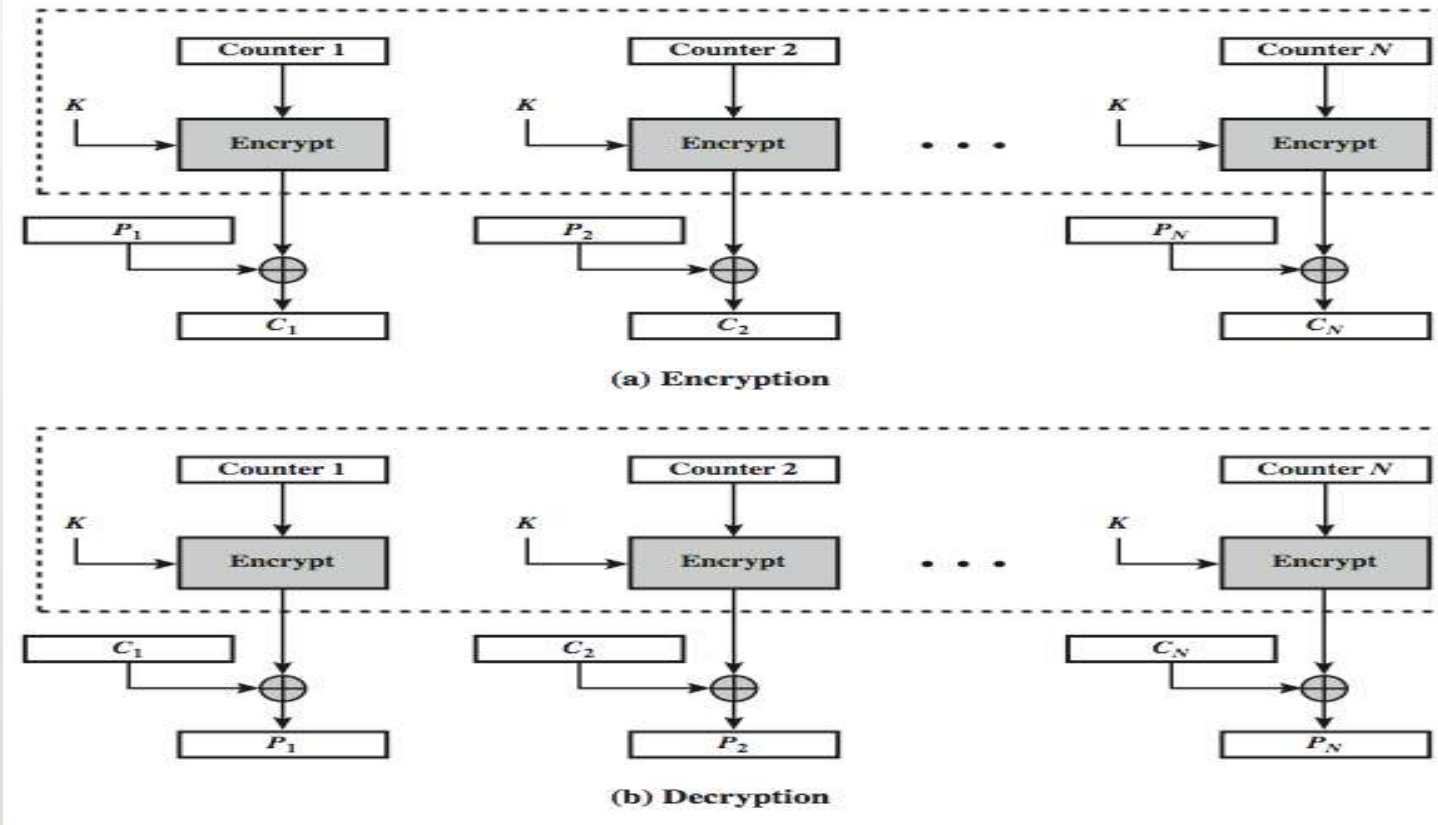
MODES OF OPERATION

Counter Mode of Operation

- **Encryption Process:** The counter value is encrypted using the block cipher algorithm, generating a pseudorandom keystream block. The keystream block is XORed with the corresponding plaintext block, producing the ciphertext. Each bit or byte of the plaintext is combined with the corresponding bit or byte of the keystream using the XOR operation. The resulting ciphertext is stored.
- **Decryption Process:** The same counter generation process is followed as in encryption, using the IV and the secret key. The counter value is encrypted using the block cipher algorithm, generating the keystream block. The keystream block is XORed with the corresponding ciphertext block, retrieving the original plaintext. Each bit or byte of the ciphertext is combined with the corresponding bit or byte of the keystream using the XOR operation. The resulting plaintext is stored.

SESSION DESCRIPTION

MODES OF OPERATION



OFB Mode of Operation

Copy rights of this diagram belongs to original author

SESSION DESCRIPTION

MODES OF OPERATION

Advantages and Limitations of CTR

- The advantages are
- Can do parallel encryptions in h/w or s/w
- Can preprocess in advance of need
- Good for bursty high speed links
- It is simple and secure like other modes
- The disadvantage is the counter values must be made available at receiver for decryption.

SELF-ASSESSMENT QUESTIONS

1. CFB mode, how are the ciphertext blocks generated ?

- a. By encrypting the previous ciphertext block
- b. By encrypting the current plaintext block
- c. By encrypting the IV (Initialization Vector)
- d. By encrypting the secret key

2. Which of the following is a characteristic of CFB mode?

- a. Random access encryption and decryption
- b. Error propagation
- c. Parallel encryption and decryption
- d. All of the above

SELF-ASSESSMENT QUESTIONS

3. In CFB mode, what is the role of the XOR operation?

- a. It combines the ciphertext block with the plaintext block
- b. It combines the ciphertext block with the IV
- c. It combines the ciphertext block with the secret key
- d. It combines the keystream with the plaintext block or ciphertext block

4. What happens if there is an error in a ciphertext block in CFB mode?

- a. The error propagates to the subsequent blocks
- b. The error affects only the corresponding bits of the plaintext
- c. The encryption process fails
- d. The error affects the entire ciphertext

SUMMARY

Throughout the session, students will gain hands-on experience with implementing and analyzing various block cipher modes of operation through practical exercises and programming assignments. By the end of the course, students will have a solid understanding of different block cipher modes and be able to make informed decisions regarding their selection and use in real-world cryptographic applications.

TERMINAL QUESTIONS

1. Demonstrate CFB Mode of Operation.
2. List advantages & limitations of CFB
3. Illustrate OFB Mode of Operation with a neat diagram
4. List the advantages & limitations of OFB
5. Demonstrate CTR Mode with a neat diagram
6. List advantages & limitations of CTR mode.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 5th edition.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier, Second Edition, John Wiley & Sons, Inc., 2015.
3. Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, by Hamid R. Nemati and Li Yang, IGI Global, 2011
4. Forouzon B, "Cryptography and Network Security," Indian Edition, TMH (2010).

THANK YOU



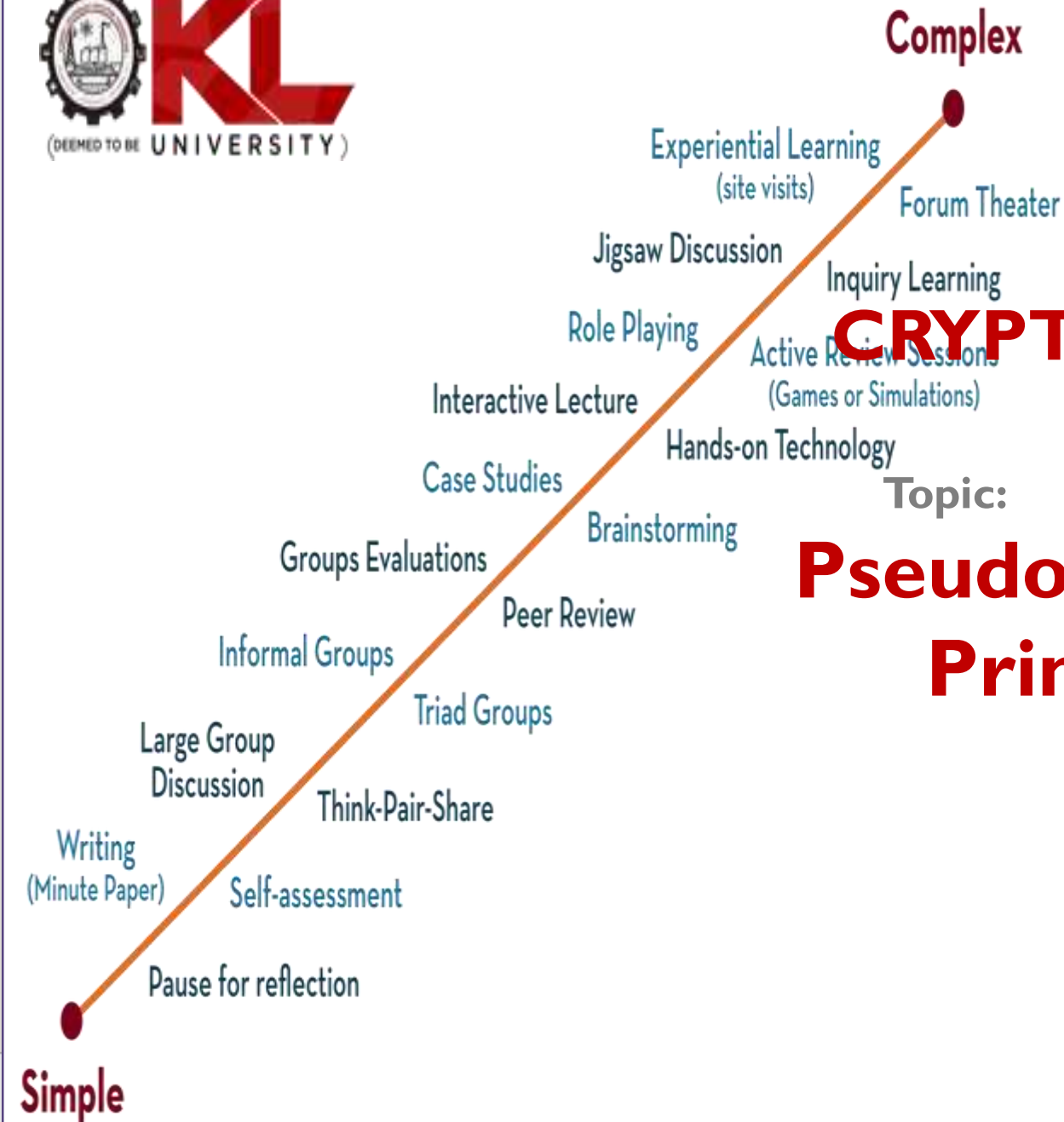
Team – CACD

CRYPTANALYSIS & CYBER DEFENSE 21CS304IRA

Topic:

Pseudorandom Number Generation Principles and Pseudorandom Number, Generators

Session - I0

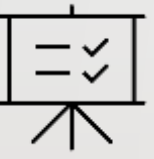


AIM OF THE SESSION




To make the students understand with the basic concepts of Pseudorandom Number Generation and its Principles..

INSTRUCTIONAL OBJECTIVES



The objective of this session is to introduce the basic principles of Random Number Generation and types of Random Number Generators.

LEARNING OUTCOMES



At the end of this session, students are expected to know

1. List principles of Random Number Generation
2. Demonstrate types of Random Number Generators

This module demonstrates block cipher modes of operation. It also helps to identify which ciphers operate as block ciphers and stream ciphers. Principles of Pseudorandom Number Generators are listed and types of Pseudorandom Number Generators are discussed. Finally, Stream Ciphers and its example RC4 algorithm are demonstrated.

A pseudorandom number generator (PRNG) is a deterministic algorithm that produces a sequence of numbers that appear to be random but are actually generated using a mathematical formula. PRNGs are widely used in cryptography, simulations, and other applications where random numbers are required.

SESSION DESCRIPTION

- It is important to note that while PRNGs can produce numbers that appear random, they are not truly random and are subject to certain vulnerabilities.
- For example, if an attacker can determine the seed value and the algorithm used by a PRNG, they may be able to predict the entire sequence of numbers it will produce.
- To address these vulnerabilities, cryptographic PRNGs use a combination of a secret key and a seed value to produce a sequence of numbers that is both random and unpredictable.
- These cryptographic PRNGs are used in applications such as generating encryption keys, authentication tokens, and random numbers for use in cryptographic protocols.

SESSION DESCRIPTION

Pseudorandom Number Generation

Random Numbers

A random number is a value that is generated unpredictably and lacks any discernible pattern or relationship to previously generated values. Random numbers play a crucial role in various fields, including cryptography, simulations, statistical analysis, gaming, and more.

SESSION DESCRIPTION

Types of Random Numbers:

True Random Numbers (TRNG):

- True random numbers are generated from unpredictable physical processes, such as atmospheric noise, radioactive decay, or thermal noise.
- TRNGs provide genuinely random and unbiased numbers, as they are derived from natural phenomena.
- However, generating true random numbers can be challenging and often requires specialized hardware.

SESSION DESCRIPTION

Types of Random Numbers:

Pseudorandom Numbers (PRNG):

- Pseudorandom numbers are generated by deterministic algorithms that use mathematical formulas and a seed value to produce a sequence of numbers that appears random.
- PRNGs are repeatable, meaning that with the same seed value, they will produce the same sequence of numbers.
- However, the generated numbers are not truly random, but rather exhibit statistical randomness and pass various tests for randomness.

SESSION DESCRIPTION

Pseudorandom Number Generation

Criteria:

1. Uniform distribution: The distribution of bits in the sequence should be uniform.
2. Independence: No one subsequence in the sequence can be inferred from the others.

SESSION DESCRIPTION

Pseudorandom Number Generation

Application of Random Numbers:

Random numbers find applications in various areas, such as

- Generating encryption keys
- Conducting statistical sampling
- Simulating real-world scenarios
- Ensuring fairness in games
- Enhancing security in cryptographic systems.

SESSION DESCRIPTION

Pseudorandom Number Generation

Types of Random Numbers (IN DETAIL):

☐ True Random Number Generators (TRNGs):

True Random Number Generators (TRNGs) are devices or algorithms that generate random numbers from unpredictable physical processes or phenomena. Unlike pseudorandom number generators (PRNGs), which rely on deterministic algorithms, TRNGs utilize inherent randomness in the physical world to produce truly random and unbiased numbers.

SESSION DESCRIPTION

Pseudorandom Number Generation

Here are some key points about TRNGs:

- ☐ **Unpredictable Sources:** TRNGs extract randomness from various sources, such as atmospheric noise, radioactive decay, thermal noise, electronic noise, or even quantum phenomena. These sources are considered unpredictable and provide a source of true randomness.
- ☐ **Hardware-based TRNGs:** Hardware-based TRNGs use physical components or circuits to capture random physical processes. They can include sensors, amplifiers, analog-to-digital converters, and post-processing modules to refine the obtained raw random data.
- ☐ **Physical Phenomena:** TRNGs exploit physical phenomena that exhibit inherent randomness, such as variations in voltage, temperature, or radioactive decay events. These phenomena provide a basis for generating random bits.

SESSION DESCRIPTION

Pseudorandom Number Generation

- ☐ Testing and Certification: TRNGs undergo rigorous testing and evaluation to verify their random properties and ensure they meet specific standards. Organizations like the National Institute of Standards and Technology (NIST) provide guidelines and certification processes for TRNGs.
- ☐ Non-deterministic Output: TRNGs generate numbers that are not derived from any algorithmic or mathematical formula. The output of a TRNG at a given moment is not dependent on previous outputs or the current state of the device.
- ☐ Applications: TRNGs are essential in applications where true randomness is critical, such as cryptography, key generation, gambling, lotteries, and scientific simulations. They provide a high level of security and eliminate the predictability associated with pseudorandom numbers.

SESSION DESCRIPTION

Pseudorandom Number Generation

- It's worth noting that TRNGs may have limitations, such as slower generation speeds compared to PRNGs and potential biases or correlations in the generated random numbers.
- Thus, careful design, post-processing techniques, and statistical analysis are employed to ensure the quality and reliability of the generated random numbers.

SESSION DESCRIPTION

Pseudorandom Number Generation

☐ Pseudorandom Number Generators (PRNGs):

Pseudorandom Number Generators (PRNGs) are algorithms or software routines that generate sequences of numbers that appear to be random, but are actually determined by a deterministic process. PRNGs use mathematical formulas and an initial seed value to produce a sequence of numbers that exhibits statistical randomness properties

SESSION DESCRIPTION

Pseudorandom Number Generation

Here are some key points about PRNGs:

- ☐ **Deterministic Algorithms:** PRNGs use deterministic algorithms to generate random-like sequences. Given the same seed value, a PRNG will produce the same sequence of numbers. The output is entirely determined by the algorithm and the seed value.

- ☐ **Seed Value:** The seed value serves as the initial input to the PRNG algorithm.

By changing the seed value, different sequences of numbers can be generated. If the same seed value is used, the same sequence will be produced.

- ☐ **Periodicity:** PRNGs have a period, which is the length of the sequence before it repeats itself. The period is determined by the algorithm and the internal state of the PRNG. A good PRNG should have a long period to avoid predictability and repetition.

SESSION DESCRIPTION

Pseudorandom Number Generation

- ❑ **Reproducibility:** PRNGs offer reproducibility, meaning that given the same seed value, the sequence of numbers can be replicated. This property is useful in simulations, testing, and debugging scenarios.
- ❑ **Statistical Randomness:** PRNGs strive to generate sequences that exhibit statistical randomness properties, such as uniform distribution, independence, and Unpredictability. They are designed to pass various statistical tests for randomness.
- ❑ **Cryptographically Secure PRNGs (CSPRNGs):** In cryptographic applications, special PRNGs called Cryptographically Secure PRNGs (CSPRNGs) are used. CSPRNGs are designed to withstand cryptographic attacks and provide a high level of randomness suitable for encryption, key generation, and other security-sensitive operations.
- ❑ **Efficiency:** PRNGs are computationally efficient and can generate a large number of random-like sequences quickly. They are commonly used in applications like simulations, gaming, numerical analysis, and modeling.

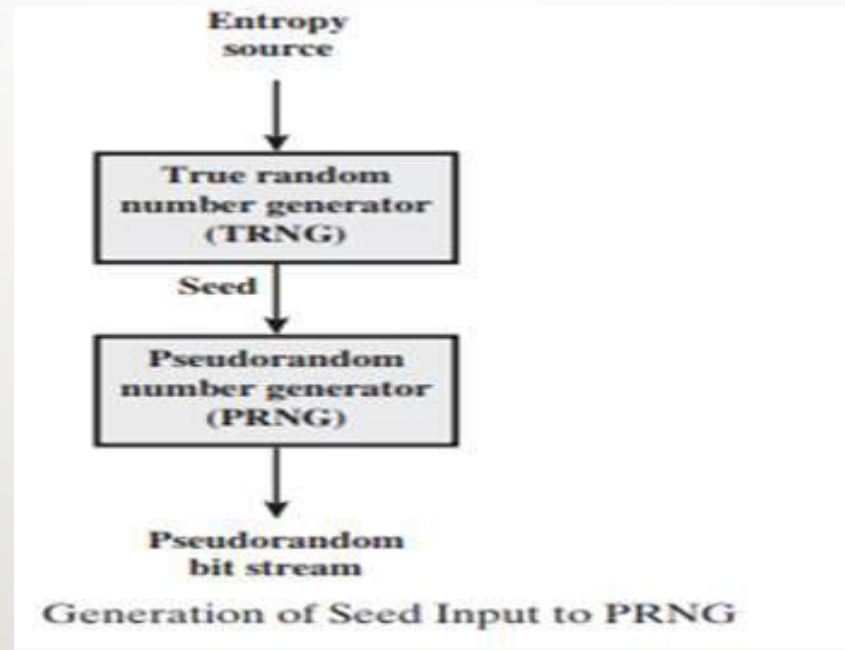
SESSION DESCRIPTION

Pseudorandom Number Generation

- It's important to note that PRNGs, unlike true random number generators (TRNGs), are not inherently random.
- They rely on their algorithmic design and the seed value for their randomness.
- As a result, PRNGs may exhibit patterns or correlations in their output over long periods or if the seed values are not chosen carefully.

SESSION DESCRIPTION

Pseudorandom Number Generation



Generation of Seed Input to PRNG

Copyrights of this diagram are reserved with original author

SESSION DESCRIPTION

Pseudorandom Number Generation

□ Pseudo-Random Function (PRF)

A Pseudo-Random Function (PRF) is a deterministic function that takes an input key and an input message or data and produces an output that appears random and unpredictable, even though it is computed by a deterministic algorithm. PRFs are commonly used in cryptography for various purposes, including key generation, data integrity verification, message authentication codes (MACs), and pseudorandom number generation.

SESSION DESCRIPTION

Pseudorandom Number Generation

Here are some key points about PRFs:

- ☐ **Deterministic Function:** A PRF is a deterministic function, meaning that given the same input key and message, it will always produce the same output. This property allows for reproducibility and consistency in cryptographic operations.
- ☐ **Pseudo randomness:** PRFs generate output that exhibits pseudorandomness, meaning that the output appears random and unpredictable to an observer who does not have knowledge of the key. However, since the output is computed by an algorithm, it is not truly random.
- ☐ **Keyed Function:** PRFs rely on a secret key that is known only to the parties involved in the cryptographic operation. The key enhances the security of the PRF and ensures that the output is dependent on both the input message and the key.

SESSION DESCRIPTION

Pseudorandom Number Generation

- Security Properties: A secure PRF should exhibit properties like unpredictability, resistance to key recovery, and computational indistinguishability. These properties ensure that an adversary cannot distinguish the output of the PRF from truly random output or gain knowledge of the key based on the observed output.
- PRF vs. PRNG: While both PRFs and Pseudorandom Number Generators (PRNGs) generate pseudorandom output, PRFs are typically designed for specific cryptographic purposes, such as generating keys or providing integrity and authentication, whereas PRNGs are focused on generating random-like sequences of numbers for general-purpose applications.

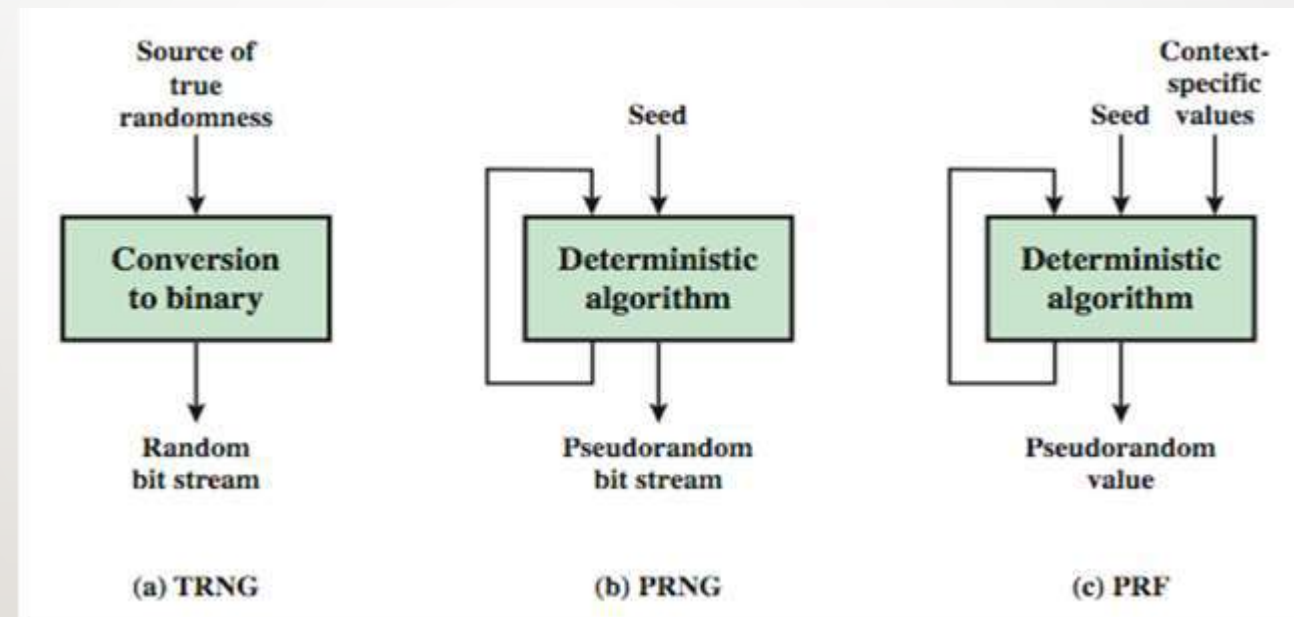
SESSION DESCRIPTION

Pseudorandom Number Generation

- ❑ Common PRF Constructions: HMAC (Hash-based Message Authentication Code) is a widely used construction for PRFs. It combines a cryptographic hash function with a secret key to produce a pseudorandom output. Other constructions, such as block ciphers in various modes, can also serve as PRFs.
- ❑ PRFs play a crucial role in ensuring the security and integrity of cryptographic systems. They are fundamental building blocks for many cryptographic protocols and applications, providing the necessary pseudo randomness and security properties required for secure communications and data protection.

SESSION DESCRIPTION

Pseudorandom Number Generation



Schematic diagram of TRNG, PRNG and PRF.
Copyrights of this diagram are reserved with original author

SELF-ASSESSMENT QUESTIONS

1. Which of the following statements is true about random number generators (RNGs) ?

- a) RNGs generate truly random numbers.
- b) RNGs generate predictable sequences of numbers.
- c) RNGs require a seed value to generate random numbers.
- d) RNGs are not used in cryptography.

2. What is the main purpose of a random number generator in cryptography?

- a) To generate random keys for encryption algorithms.
- b) To generate random plaintext messages.
- c) To generate random ciphertext messages.
- d) To generate random IVs for cryptographic modes of operation.

SELF-ASSESSMENT QUESTIONS

3. Which of the following is a key principle of a secure random number generator?

- a) Determinism
- b) Predictability
- c) Reproducibility
- d) Unpredictability

4. Which type of random number generator uses physical phenomena to generate randomness?

- a) Pseudorandom number generator (PRNG)
- b) Hardware random number generator (HRNG)
- c) Deterministic random bit generator (DRBG)
- d) Software-based random number generator

SUMMARY

This session provides an overview of the topics typically covered in a course on PRNGs. The duration and depth of each topic may vary depending on the level and intended audience of the course. Additionally, practical exercises, assignments, and assessments can be incorporated throughout the course to reinforce learning and practical application of PRNG concepts.

TERMINAL QUESTIONS

1. Summarize PRNG concepts
2. Illustrate TRNG with neat diagram.
3. Demonstrate PRNG with neat diagram.
4. Illustrate PRF with neat diagram.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 5th edition.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier, Second Edition, John Wiley & Sons, Inc., 2015.
3. Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, by Hamid R. Nemati and Li Yang, IGI Global, 2011
4. Forouzon B, "Cryptography and Network Security," Indian Edition, TMH (2010).

THANK YOU



Team – CACD

Complex

Department of CSE(HONORS)

CRYPTANALYSIS & CYBER DEFENSE 21CS304IRA

Topic:

Types of Pseudorandom Number, Generators

Session - I I

Simple

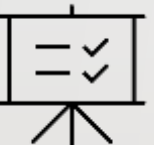
Experiential Learning
(site visits)
Forum Theater
Jigsaw Discussion
Inquiry Learning
Role Playing
Active Review Sessions
(Games or Simulations)
Interactive Lecture
Hands-on Technology
Case Studies
Brainstorming
Groups Evaluations
Peer Review
Informal Groups
Triad Groups
Large Group Discussion
Think-Pair-Share
Writing
(Minute Paper)
Self-assessment
Pause for reflection

AIM OF THE SESSION




To make the students understand with the basic concepts of Pseudorandom Number Generation and its Principles.

INSTRUCTIONAL OBJECTIVES



The objective of this session is to introduces types of Psuedorandom Number Generators

LEARNING OUTCOMES



At the end of this session, students are expected to know

1. Demonstrate types of Random Number Generators.
2. Apply Psuedorandom Number Generators

This module demonstrates block cipher modes of operation. It also helps to identify which ciphers operate as block ciphers and stream ciphers. Principles of Pseudorandom Number Generators are listed and types of Pseudorandom Number Generators are discussed. Finally, Stream Ciphers and its example RC4 algorithm are demonstrated.

A pseudorandom number generator (PRNG) is a deterministic algorithm that produces a sequence of numbers that appear to be random but are actually generated using a mathematical formula. PRNGs are widely used in cryptography, simulations, and other applications where random numbers are required.

SESSION DESCRIPTION

- It is important to note that while PRNGs can produce numbers that appear random, they are not truly random and are subject to certain vulnerabilities.
- For example, if an attacker can determine the seed value and the algorithm used by a PRNG, they may be able to predict the entire sequence of numbers it will produce.
- To address these vulnerabilities, cryptographic PRNGs use a combination of a secret key and a seed value to produce a sequence of numbers that is both random and unpredictable.
- These cryptographic PRNGs are used in applications such as generating encryption keys, authentication tokens, and random numbers for use in cryptographic protocols.

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

Linear Congruential Generator (LCG):

A Linear Congruential Generator (LCG) is a type of pseudorandom number generator (PRNG) that generates a sequence of random numbers based on a linear equation. It follows the recurrence relation:

$$X_{n+1} = (a * X_n + c) \bmod m$$

where:

X_n is the current state or the previously generated number in the sequence.

a , c , and m are constants chosen for the LCG algorithm.

The mod m operation calculates the remainder when $(a * X_n + c)$ is divided by m .

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

- The LCG algorithm starts with an initial seed value (X_0) and uses the recurrence relation to generate subsequent numbers in the sequence. Each new number is obtained by multiplying the previous number by a constant, adding another constant, and then taking the modulo of the result.
- The generated numbers are considered pseudorandom because they appear to be random, but they are actually determined entirely by the initial seed value and the chosen constants. The quality of randomness in an LCG depends on the values of a , c , and m . With carefully chosen constants, LCGs can produce a long period of seemingly random numbers before the sequence repeats.

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

- However, LCGs have limitations and weaknesses. If poorly chosen constants are used, the generated sequence may exhibit patterns or have a short period, making it predictable. LCGs are not suitable for cryptographic applications that require high-quality random numbers due to their deterministic nature and vulnerability to attacks.
- To improve the randomness and security of LCGs, it is recommended to carefully select the constants and periodically reseed the generator with a high-quality random source. In modern applications, more advanced random number generators, such as cryptographic random number generators (CSPRNGs), are preferred as they offer better randomness and stronger security guarantees.

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

Blum Blum Shub Generator (BBS):

The Blum Blum Shub (BBS) generator is a cryptographic pseudorandom number generator (CPRNG) that was introduced by Lenore Blum, Manuel Blum, and Michael Shub in 1986. It is designed to generate random numbers based on the quadratic residues modulo a Blum integer. The BBS generator is considered secure, relying on the assumption that the integer factorization problem is computationally difficult.

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

Here's how the BBS generator works:

Key Generation:

Two large prime numbers, p and q , are chosen such that $p \equiv q \equiv 3 \pmod{4}$. These primes are kept secret and are typically of equal length to enhance security.

The Blum integer, n , is calculated as the product of p and q , which serves as the modulus for the generator.

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

Seed Generation:

A seed value, X_0 , is selected, which should be relatively prime to n . It is advisable to use a seed generated from a high entropy source to ensure randomness.

Random Number Generation:

Each random bit or byte is generated through the following steps:

Compute $X_{i+1} = (X_i)^2 \bmod n$, where X_i represents the previously generated value.

The least significant bit or byte of X_{i+1} is extracted and used as the random output.

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

- The BBS generator generates a sequence of random bits or bytes by repeatedly squaring the previous value and taking the modulo n . The resulting output is the least significant bit or byte, depending on the desired output size.
- The security of the BBS generator relies on the difficulty of factoring the Blum integer n into its prime factors. If an attacker can efficiently factorize n , it would enable them to predict future outputs of the generator. Thus, the security of the BBS generator hinges on the secrecy of the prime numbers p and q .

SESSION DESCRIPTION

Types of Pseudorandom Number Generator (PRNGs)

Although the BBS generator is known for its simplicity and ease of implementation, it tends to be slower compared to other pseudorandom number generators due to the modular exponentiation operation involved. As a result, it is commonly used in applications where security is prioritized over speed, such as cryptographic key generation.

SELF-ASSESSMENT QUESTIONS

1. The LCG equation is defined as $X_{n+1} = (a * X_n + c) \text{ mod } m$. What does "mod" represent in this equation?

- a) Modulus operation
- b) Multiplication operation
- c) Addition operation
- d) Exponentiation operation

2. Which of the following is a potential drawback of using an LCG?

- a) Short period length
- b) Poor statistical properties
- c) Predictability of the generated sequence
- d) All of the above

SELF-ASSESSMENT QUESTIONS

3. Which of the following is an essential component of the Linear Congruential Generator (LCG)?

- a) Seed value
- b) Modulus
- c) Multiplier
- d) All of the above

4. In an LCG, the seed value:

- a) Determines the length of the generated sequence
- b) Determines the period of the generated sequence
- c) Determines the initial state of the generator
- d) None of the above

SUMMARY

This session provides an overview of the topics typically covered in a course on PRNGs. The duration and depth of each topic may vary depending on the level and intended audience of the course. Additionally, practical exercises, assignments, and assessments can be incorporated throughout the course to reinforce learning and practical application of PRNG concepts.

TERMINAL QUESTIONS

1. Suppose an LCG is defined with the parameters:

$$a = 7$$

$$c = 0$$

$$m = 10$$

Find the seed value (X_0) that produces the longest possible period for this LCG.

2. An LCG has the following parameters:

$$a = 25214903917$$

$$c = 11$$

$$m = 2^{48}$$

If the initial seed (X_0) is 7, calculate the 10th number generated by the LCG.

3. Suppose a BBS generator is initialized with the following parameters:

$$p = 103$$

$$q = 107$$

$$X_0 = 33$$

Calculate the next 10 bytes generated by the BBS generator.

4. Consider a BBS generator with the following parameters:

$$p = 17$$

$$q = 29$$

$$X_0 = 2$$

Determine whether the BBS generator will produce a full period or a shorter period, and explain your reasoning.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 5th edition.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier, Second Edition, John Wiley & Sons, Inc., 2015.
3. Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, by Hamid R. Nemati and Li Yang, IGI Global, 2011
4. Forouzon B, "Cryptography and Network Security," Indian Edition, TMH (2010).

THANK YOU



Team – CACD

Complex

Department of CSE(HONORS)

CRYPTANALYSIS & CYBER DEFENSE 21CS304IRA

Topic:

Types of Pseudorandom Number, Generators

Session - I2

Simple

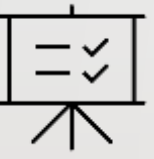
Experiential Learning
(site visits)
Forum Theater
Jigsaw Discussion
Inquiry Learning
Role Playing
Active Review Sessions
(Games or Simulations)
Interactive Lecture
Hands-on Technology
Case Studies
Brainstorming
Groups Evaluations
Peer Review
Informal Groups
Triad Groups
Large Group Discussion
Think-Pair-Share
Writing
(Minute Paper)
Self-assessment
Pause for reflection

AIM OF THE SESSION




To make the students understand the basic concepts of Pseudorandom Number Generation and its Principles

INSTRUCTIONAL OBJECTIVES



The objective of this session is to introduce the basic principles of Random Number Generation and types of Random Number Generators.

LEARNING OUTCOMES



At the end of this session, students are expected to know

- Demonstrate Psuedorandom Number Generators using Block Ciphers.
- Demonstrate X9.17 ANSI Ring

This module demonstrates block cipher modes of operation. It also helps to identify which ciphers operate as block ciphers and stream ciphers. Principles of Pseudorandom Number Generators are listed and types of Pseudorandom Number Generators are discussed. Finally, Stream Ciphers and its example RC4 algorithm are demonstrated.

A pseudorandom number generator (PRNG) is a deterministic algorithm that produces a sequence of numbers that appear to be random but are actually generated using a mathematical formula. PRNGs are widely used in cryptography, simulations, and other applications where random numbers are required.

SESSION DESCRIPTION

- It is important to note that while PRNGs can produce numbers that appear random, they are not truly random and are subject to certain vulnerabilities.
- For example, if an attacker can determine the seed value and the algorithm used by a PRNG, they may be able to predict the entire sequence of numbers it will produce.
- To address these vulnerabilities, cryptographic PRNGs use a combination of a secret key and a seed value to produce a sequence of numbers that is both random and unpredictable.
- These cryptographic PRNGs are used in applications such as generating encryption keys, authentication tokens, and random numbers for use in cryptographic protocols.

SESSION DESCRIPTION

PRNG Using Block Cipher Modes of Operation

- The Output Feedback (OFB) mode is recommended in standards such as X9.82 and RFC 4086.
- Figure 12.1 provides an illustration of the two methods used in OFB mode. In both methods, the seed consists of two components: the encryption key and a value that undergoes updates after generating each block of pseudorandom numbers.
- For instance, when using AES-128, the seed comprises a 128-bit key and a 128-bit value.

SESSION DESCRIPTION

PRNG Using Block Cipher Modes of Operation

In the Counter (CTR) case, the value is incremented by 1 after each encryption operation. On the other hand, in the OFB case, the value is updated to be equal to the value of the preceding pseudorandom number generator (PRNG) block.

In both scenarios, pseudorandom bits are generated one block at a time. For example, with AES, the PRNG bits are generated in chunks of 128 bits at a time.

It is important to note that the above information is based on the mentioned standards and the specific mode of operation being discussed.

SESSION DESCRIPTION

PRNG Using Block Cipher Modes of Operation

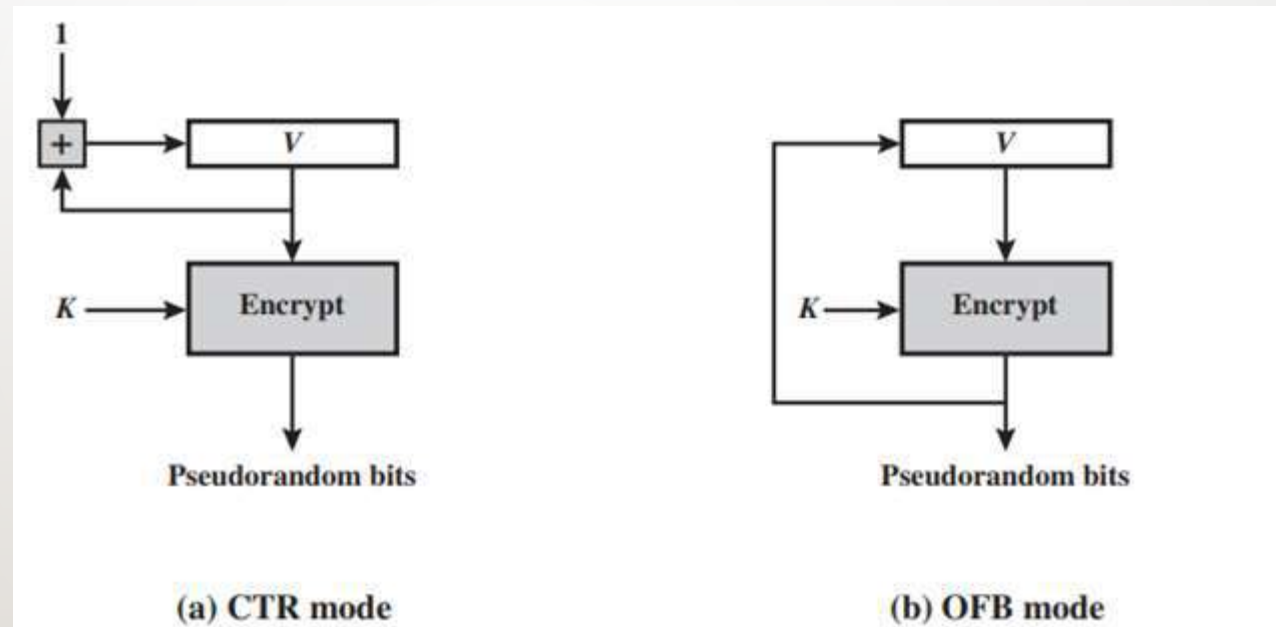


Figure 12.1: PRNG Mechanism Based on Block ciphers
Copyrights of this diagram are reserved for the original author

SESSION DESCRIPTION

ANSI Ring X9.17

The ANSI X9.17 standard specifies a PRNG that is considered to be one of the strongest in terms of cryptographic security. This technique is utilized in various applications, including financial security and PGP

Figure 12.2 provides a depiction of the algorithm employed in this PRNG, which utilizes triple DES for encryption.

The algorithm relies on two pseudorandom inputs as its driving force. The first input is a 64-bit representation of the current date and time, which is updated each time a number is generated. The second input is a 64-bit seed value that is initially set to an arbitrary value and subsequently updated during the generation process.

SESSION DESCRIPTION

ANSI Ring X9.17

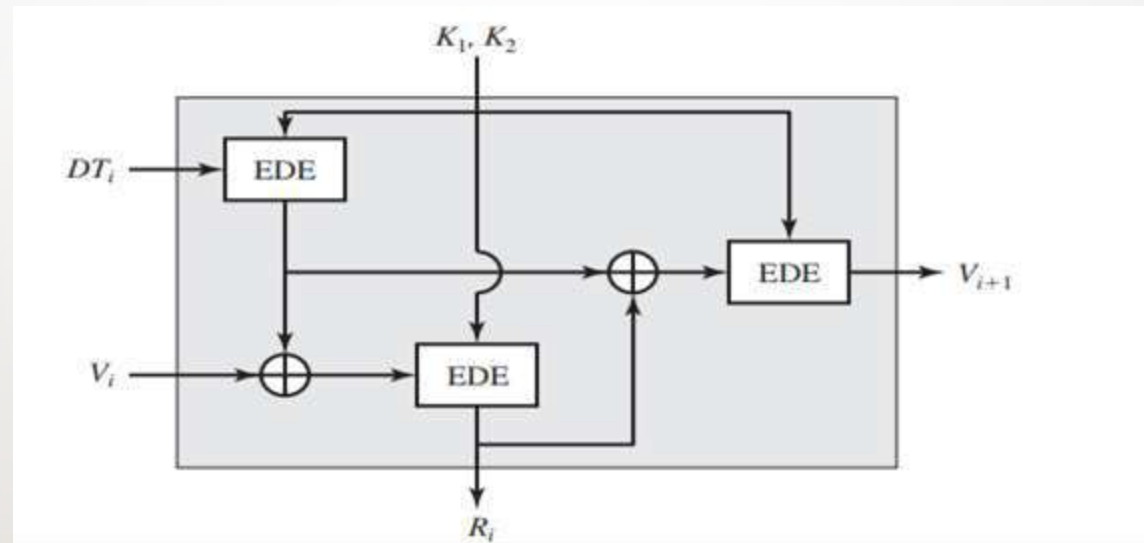


Figure 12.2: ANSI X9.17 Pseudorandom Number Generator
Copyrights of this diagram are reserved for the original author

SESSION DESCRIPTION

ANSI Ring X9.17

Key Usage: The PRNG utilizes three triple DES encryption modules, with each module employing the same pair of 56-bit secret keys. These keys are specifically designated for pseudorandom number generation purposes and should remain confidential.

- **Output Format:** The PRNG generates a 64-bit pseudorandom number as its primary output. Additionally, it produces a 64-bit seed value, which is an essential component for the subsequent generation of pseudorandom numbers.

SESSION DESCRIPTION

ANSI Ring X9.17

Let us define the following quantities.

DT_i	Date/time value at the beginning of i th generation stage
V_i	Seed value at the beginning of i th generation stage
R_i	Pseudorandom number produced by the i th generation stage
K_1, K_2	DES keys used for each stage

Then

$$R_i = \text{EDE}([K_1, K_2], [V_i \oplus \text{EDE}([K_1, K_2], DT_i)])$$
$$V_{i+1} = \text{EDE}([K_1, K_2], [R_i \oplus \text{EDE}([K_1, K_2], DT_i)])$$

where $\text{EDE}([K_1, K_2], X)$ refers to the sequence encrypt-decrypt-encrypt using two-key triple DES to encrypt X .

Figure 12.3: ANSI X9.17 Psuedorandom Number Generator Equations
Copyrights of this diagram are reserved for the original author

SESSION DESCRIPTION

- Several factors contribute to the cryptographic strength of this method. The technique utilizes a 112-bit key, which ensures a high level of security. The process involves three EDE (Encrypt-Decrypt-Encrypt) encryptions, which effectively amounts to a total of nine DES (Data Encryption Standard) encryptions. This multiple encryption scheme enhances the overall security of the method.
- The generation process of pseudorandom numbers is driven by two inputs: the current date and time value and a seed value. It is important to note that the seed value is distinct from the pseudorandom number generated by the generator. This separation of inputs adds an additional layer of complexity to the system and makes it extremely challenging for an adversary to compromise the security.

SESSION DESCRIPTION

- The generation process of pseudorandom numbers is driven by two inputs: the current date and time value and a seed value. It is important to note that the seed value is distinct from the pseudorandom number generated by the generator. This separation of inputs adds an additional layer of complexity to the system and makes it extremely challenging for an adversary to compromise the security.

SESSION DESCRIPTION

- Even if a pseudorandom number were to be compromised, it would be infeasible for an attacker to deduce the original seed value or the encryption key. This is due to the utilization of an additional EDE operation in the generation process, which further obscures the relationship between the seed value and the pseudorandom number.
- The combination of a large key size, multiple encryptions, separate inputs, and additional encryption steps ensures that the level of material that an adversary would need to compromise is overwhelmingly large. This makes the method highly resistant to attacks and provides a strong cryptographic foundation for generating pseudorandom numbers.

SELF-ASSESSMENT QUESTIONS

1. Which of the following is a block cipher mode of operation commonly used for pseudorandom number generation?

- a) ECB (Electronic Codebook)
- b) CTR (Counter)
- c) CBC (Cipher Block Chaining)
- d) OFB (Output Feedback)

2. Which property makes block cipher modes of operation suitable for pseudorandom number generation?

- a) Randomness
- b) Determinism
- c) Key expansion
- d) Block size

SELF-ASSESSMENT QUESTIONS

3. In which block cipher mode of operation is the previous ciphertext block (or initialization vector) encrypted to generate the pseudorandom keystream?

- a) ECB (Electronic Codebook)
- b) CBC (Cipher Block Chaining)
- c) CTR (Counter)
- d) OFB (Output Feedback)

4. Which block cipher mode of operation allows for parallel encryption and decryption of individual blocks?

- a) ECB (Electronic Codebook)
- b) CTR (Counter)
- c) CBC (Cipher Block Chaining)
- d) OFB (Output Feedback)

SUMMARY

This session provides an overview of the topics typically covered in a course on PRNGs. The duration and depth of each topic may vary depending on the level and intended audience of the course. Additionally, practical exercises, assignments, and assessments can be incorporated throughout the course to reinforce learning and practical application of PRNG concepts.

TERMINAL QUESTIONS

1. Summarize the concept of pseudorandom number generation using block cipher modes of operation.
2. Identify the purpose of using block cipher modes of operation for pseudorandom number generation.
3. Summarize the role of the initialization vector (IV) in block cipher modes of operation for pseudorandom number generation.
4. Enumerate the potential security vulnerabilities and weaknesses associated with using block cipher modes of operation for pseudorandom number generation.
5. Identify the impact of the block size on the efficiency and security of pseudorandom number generation using block cipher modes of operation.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 5th edition.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier, Second Edition, John Wiley & Sons, Inc., 2015.
3. Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, by Hamid R. Nemati and Li Yang, IGI Global, 2011
4. Forouzon B, "Cryptography and Network Security," Indian Edition, TMH (2010).

THANK YOU



Team – CACD

Complex

Department of CSE(HONORS)

CRYPTANALYSIS & CYBER DEFENSE 2ICS304IRA

Topic:

Stream Ciphers and SRC4

Session - I3

Simple

Experiential Learning
(site visits)

Forum Theater

Jigsaw Discussion

Inquiry Learning

Role Playing

Active Review Sessions
(Games or Simulations)

Interactive Lecture

Hands-on Technology

Case Studies

Brainstorming

Groups Evaluations

Peer Review

Informal Groups

Triad Groups

Large Group Discussion

Think-Pair-Share

Writing
(Minute Paper)

Self-assessment

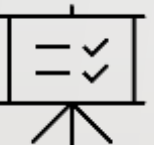
Pause for reflection

AIM OF THE SESSION




To make students understand basic concepts of Stream Ciphers and apply SRC4 algorithm on a given plaintext.

INSTRUCTIONAL OBJECTIVES



The objective of this session is to introduce types of Pseudorandom Number Generators.

LEARNING OUTCOMES



At the end of this session, students are expected to know

- Summarize the concepts of Stream Cipher.
- Apply SRC4 cipher to a given plaintext.

This module demonstrates block cipher modes of operation. It also helps to identify which ciphers operate as block ciphers and stream ciphers. Principles of Pseudorandom Number Generators are listed and types of Pseudorandom Number Generators are discussed. Finally, Stream Ciphers and its example RC4 algorithm are demonstrated.

Stream ciphers are a type of symmetric encryption algorithm that operates on individual bits or bytes of data. Unlike block ciphers, which encrypt data in fixed-size blocks, stream ciphers encrypt data on a continuous stream, hence the name. Stream ciphers are commonly used for real-time applications, such as securing network communications, as they can encrypt and decrypt data in a stream-like fashion without requiring buffering or padding.

One of the most well-known and widely used stream ciphers is RC4 (Rivest Cipher 4). RC4 was designed by Ronald Rivest in 1987 and became popular due to its simplicity and efficiency. It was initially a trade secret but later became publicly available, making it widely adopted.

SESSION DESCRIPTION

Stream Ciphers and SRC4

- Stream ciphers are a type of symmetric encryption algorithm that operates on individual bits or bytes of data.
- Unlike block ciphers, which encrypt data in fixed-size blocks, stream ciphers encrypt data on a continuous stream, hence the name.
- Stream ciphers are commonly used for real-time applications, such as securing network communications, as they can encrypt and decrypt data in a stream-like fashion without requiring buffering or padding.

SESSION DESCRIPTION

Stream Ciphers and SRC4

Features of Stream Ciphers:

- ☐ Processes the message bit by bit.
- ☐ Generates pseudo random key stream.
- ☐ Then performs XOR operation with plaintext bit by bit.
- ☐ randomness of stream key completely destroys statistically properties in message.
- ☐ but must never reuse stream key.

SESSION DESCRIPTION

Stream Ciphers and SRC4

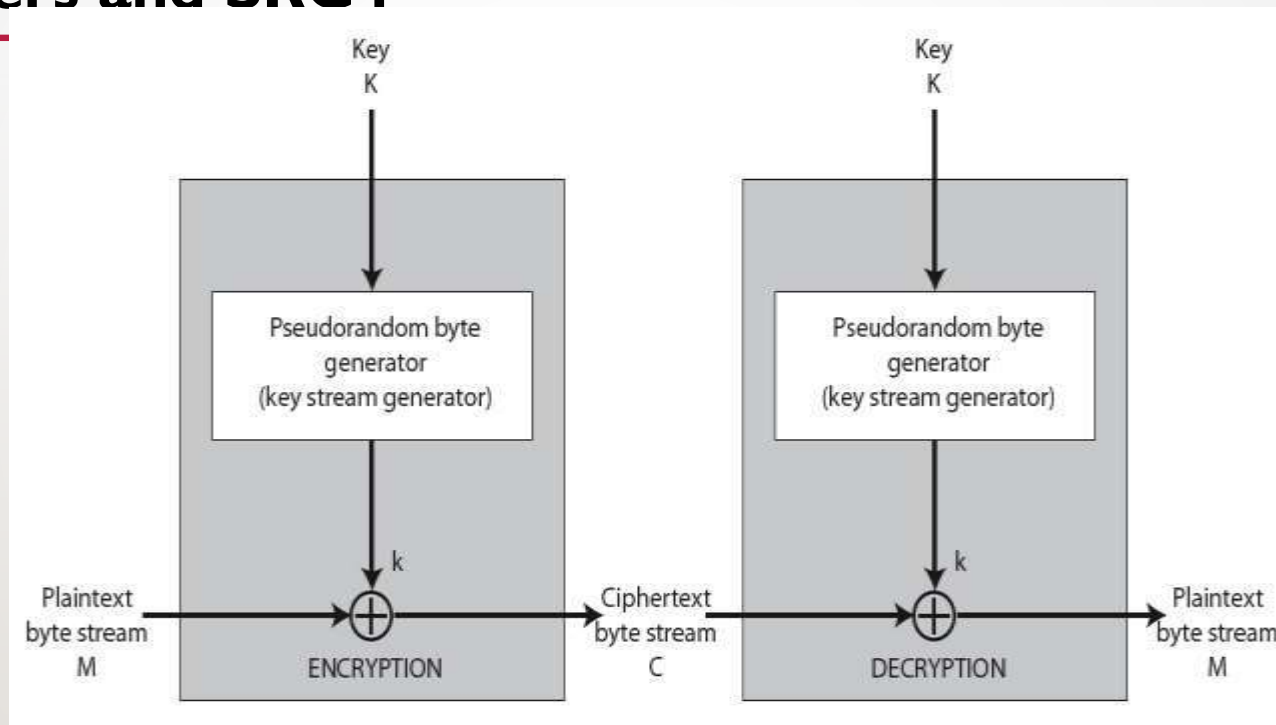


Figure 13.1: Structure of Stream Ciphers

Note: Copyrights of this diagram are reserved for original author

SESSION DESCRIPTION

Stream Ciphers and SRC4

One of the most well-known and widely used stream ciphers is RC4 (Rivest Cipher 4). RC4 was designed by Ronald Rivest in 1987 and became popular due to its simplicity and efficiency. It was initially a trade secret but later became publicly available, making it widely adopted.

- ☐ Some design considerations are:
- ☐ long period with no repetitions
- ☐ statistically random
- ☐ depends on large enough key
- ☐ large linear complexity
- ☐ If properly designed, can be as secure as a block cipher with same size key.
- ☐ Usually simpler & faster.

SESSION DESCRIPTION

Stream Ciphers and SRC4

SRC4:

- The RC4 algorithm, also referred to as Rivest Cipher 4 or Ron's Code 4, is a symmetric stream cipher widely utilized for encryption purposes.
- Developed by Ron Rivest in 1987, RC4 gained popularity due to its simplicity and speed.
- It found application in various protocols, including WEP for Wi-Fi networks and SSL/TLS for secure communications.

SESSION DESCRIPTION

Stream Ciphers and SRC4

A simplified explanation of the RC4 algorithm entails the following steps:

Key Setup:

- Initialization of two 256-byte arrays, S and T, with values ranging from 0 to 255.
- Permutation of the values in array S based on the provided key.
- Repetition of the permutation process for array T until it is fully populated with the key.

SESSION DESCRIPTION

Stream Ciphers and SRC4

Encryption/Decryption:

- Generation of a pseudorandom stream of bytes, referred to as the keystream, by combining the values from arrays S and T.
- XOR operation between each byte of the plaintext/ciphertext and the corresponding byte from the keystream, resulting in the encrypted/decrypted output.

SESSION DESCRIPTION

Pseudo-random Generation Algorithm (PRGA):

- Initialization of two indices, i and j , set to 0.
- Execution of a loop for the pseudo-random generation algorithm:
- Increment of index i .
- Modification of array S by swapping its values based on the current index i .
- Calculation of the pseudo-random index j using array S .
- Swapping of the values at indices $S[i]$ and $S[j]$.
- Retrieval of the pseudo-random value from array S by adding the values at $S[i]$ and $S[j]$ modulo 256.

SELF-ASSESSMENT QUESTIONS

1. Which of the following best describes a stream cipher?

- a) A symmetric encryption algorithm that operates on fixed-size blocks
- b) An asymmetric encryption algorithm used for key exchange
- c) A symmetric encryption algorithm that operates on individual bits or bytes
- d) An encryption algorithm used specifically for secure hash functions

2. In a stream cipher, the keystream is generated by:

- a) XORing the plaintext with a secret key
- b) Adding the plaintext and the secret key
- c) Multiplying the plaintext and the secret key
- d) Generating random bits using a pseudo-random number generator

SELF-ASSESSMENT QUESTIONS

3. Which of the following is a characteristic of stream ciphers?

- a) They are generally slower compared to block ciphers
- b) They require a larger key size for equivalent security compared to block ciphers
- c) They can encrypt data in parallel at high speeds
- d) They are more resistant to cryptanalysis attacks compared to block ciphers

4. The security of a stream cipher mainly relies on:

- a) The size of the plaintext
- b) The strength of the secret key
- c) The length of the ciphertext
- d) The complexity of the encryption algorithm

SUMMARY

In summary, stream ciphers like SRC4 are encryption algorithms that encrypt data in a continuous stream. SRC4, in particular, gained popularity due to its simplicity and efficiency. However, it has since been found to have security vulnerabilities and is no longer considered secure for use in modern cryptographic applications..

TERMINAL QUESTIONS

1. Illustrate Psuedorandom Number Generators using bock ciphers with a neat diagram
2. Demonstrate ANSI X9.17 Psuedorandom Number Generators
3. Elaborate the SRC4 Ciphers
4. Outline the advantage and disadvantage of Stream Ciphers

REFERENCES FOR FURTHER LEARNING OF THE SESSION

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 5th edition.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier, Second Edition, John Wiley & Sons, Inc., 2015.
3. Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, by Hamid R. Nemati and Li Yang, IGI Global, 2011
4. Forouzon B, "Cryptography and Network Security," Indian Edition, TMH (2010).

THANK YOU



Team – CACD

Complex

Department of CSE(HONORS)

CRYPTANALYSIS & CYBER DEFENSE

21CS304IRA

RC4

Topic:

Session - I 4

Experiential Learning
(site visits)

Forum Theater

Jigsaw Discussion

Inquiry Learning

Role Playing

Active Review Sessions
(Games or Simulations)

Interactive Lecture

Hands-on Technology

Case Studies

Brainstorming

Groups Evaluations

Peer Review

Informal Groups

Triad Groups

Large Group
Discussion

Think-Pair-Share

Writing
(Minute Paper)

Self-assessment

Pause for reflection

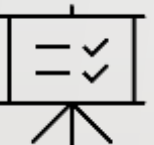
Simple

AIM OF THE SESSION




To make students understand basic concepts of Stream Ciphers and apply SRC4 algorithm on a given plaintext.

INSTRUCTIONAL OBJECTIVES



The objective of this session is to introduce types of Pseudorandom Number Generators.

LEARNING OUTCOMES



At the end of this session, students are expected to know

- Demonstrate RC4 cipher to a given plaintext.

This module demonstrates block cipher modes of operation. It also helps to identify which ciphers operate as block ciphers and stream ciphers. Principles of Pseudorandom Number Generators are listed and types of Pseudorandom Number Generators are discussed. Finally, Stream Ciphers and its example RC4 algorithm are demonstrated.

Stream ciphers are a type of symmetric encryption algorithm that operates on individual bits or bytes of data. Unlike block ciphers, which encrypt data in fixed-size blocks, stream ciphers encrypt data on a continuous stream, hence the name. Stream ciphers are commonly used for real-time applications, such as securing network communications, as they can encrypt and decrypt data in a stream-like fashion without requiring buffering or padding.

One of the most well-known and widely used stream ciphers is RC4 (Rivest Cipher 4). RC4 was designed by Ronald Rivest in 1987 and became popular due to its simplicity and efficiency. It was initially a trade secret but later became publicly available, making it widely adopted.

SESSION DESCRIPTION

RC4

- RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security.
- A variable key-length from 1 to 256 bytes is used to initialize a 256-byte state vector S , with elements $S[0]$ to $S[255]$.
- At all times, S contains a permutation of all 8-bit numbers from 0 through 255.

SESSION DESCRIPTION

RC4

- For encryption and decryption, a byte key k is generated from S by selecting one of the 255 entries in a systematic fashion.
- RC4 operates based on a pseudorandom key stream that is generated using a secret key.
- The key stream is then combined with the plaintext using a bitwise exclusive OR (XOR) operation to produce the ciphertext.
- To decrypt the ciphertext, the same key stream is generated and XORed with the ciphertext, resulting in the original plaintext.

SESSION DESCRIPTION

RC4

/* Initialization of S */

for i = 0 to 255 do

S[i] = i

T[i] = K[i mod keylen]

/* Initial Permutation */

j = 0

for i = 0 to 255 do

j = (j + S[i] + T[i]) (mod 256)

swap (S[i], S[j])

SESSION DESCRIPTION

RC4

RC4 Stream Generation:

Stream generation involves cycling through all the elements of $S[i]$ and for each $S[i]$, swapping $S[i]$ with another byte in S according to scheme dictated by the current configuration of S .

$i = j = 0$

for each message byte M_i

$i = (i + 1) \pmod{256}$

$j = (j + S[i]) \pmod{256}$

swap($S[i]$, $S[j]$)

$t = (S[i] + S[j]) \pmod{256}$

$C_i = M_i \text{ XOR } S[t]$

SESSION DESCRIPTION

RC4

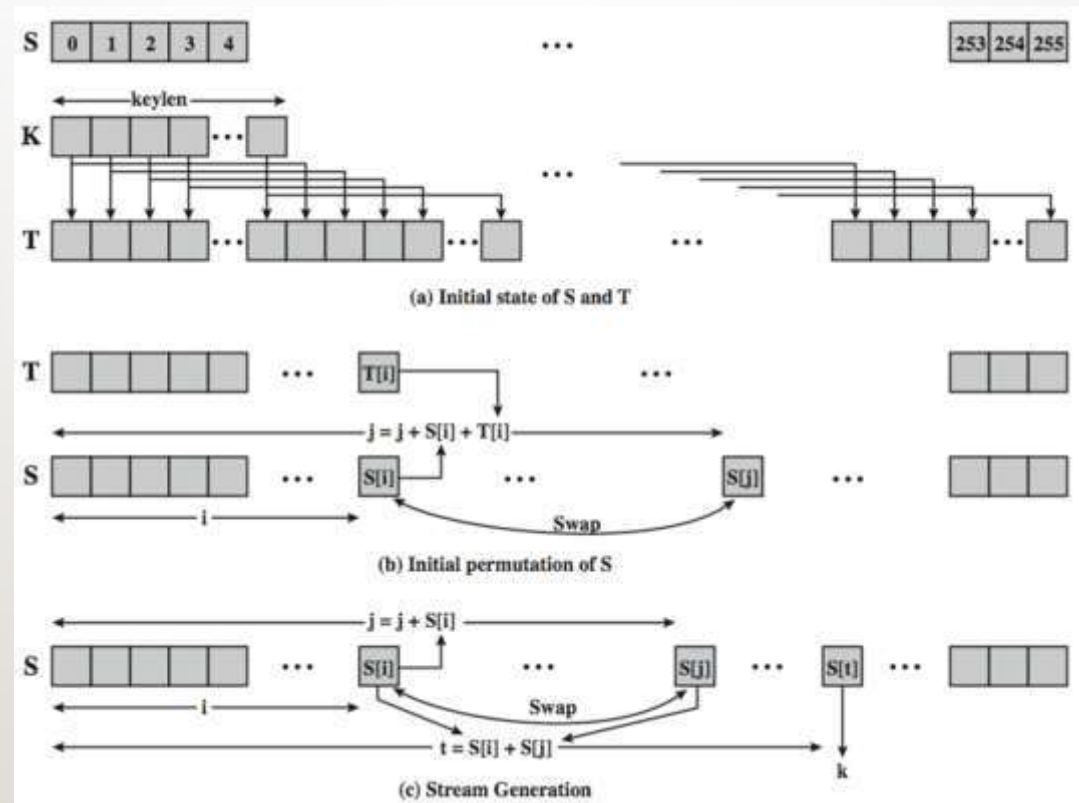


Figure 14.1: RC4 Overview

SESSION DESCRIPTION

RC4

- The key stream generation in RC4 is based on a permutation of all possible bytes (256 values) determined by the secret key.
- It uses a variable-length key ranging from 1 to 256 bytes, making it flexible for different applications.
- RC4 consists of two main components: the key-scheduling algorithm (KSA) and the pseudo-random generation algorithm (PRGA).

SESSION DESCRIPTION

RC4

- The KSA initializes an internal state array (S-box) by permuting the values 0 to 255 based on the key.
- This step ensures that each element of the S-box depends on the entire key.
- The PRGA generates the key stream by repeatedly modifying the state array and outputting a pseudorandom byte.

SESSION DESCRIPTION

RC4

- While RC4 was widely used, it has been discovered to have several security vulnerabilities over time.
- Weaknesses in its key scheduling algorithm and biases in its output stream have been identified, making it susceptible to attacks.
- Due to these vulnerabilities, RC4 is no longer recommended for use in secure systems, and more advanced stream.

SELF-ASSESSMENT QUESTIONS

1. What does RC4 stand for?

- a) Rivest Cipher 4
- b) Ron's Code 4
- c) Random Cipher 4
- d) Reliable Cipher 4

2. Who designed the RC4 algorithm?

- a) Ron Rivest
- b) Bruce Schneier
- c) Phil Zimmermann
- d) Whitfield Diffie

SELF-ASSESSMENT QUESTIONS

3. Which of the following is NOT a common use of RC4?

- a) WEP encryption in Wi-Fi networks
- b) SSL/TLS in secure communications
- c) Disk encryption
- d) Password hashing

4. What type of cipher is RC4?

- a) Symmetric block cipher
- b) Symmetric stream cipher
- c) Asymmetric block cipher
- d) Asymmetric stream cipher

SUMMARY

In summary, stream ciphers like SRC4 are encryption algorithms that encrypt data in a continuous stream. SRC4, in particular, gained popularity due to its simplicity and efficiency. However, it has since been found to have security vulnerabilities and is no longer considered secure for use in modern cryptographic applications..

TERMINAL QUESTIONS

1. Summarize the key setup process in the RC4 algorithm.
2. Explain how the pseudo-random generation algorithm (PRGA) works in RC4.
3. Discuss the potential security vulnerabilities associated with RC4.
4. Compare and contrast RC4 with other symmetric stream ciphers.
5. Summarize the impact of key length on the security of RC4

REFERENCES FOR FURTHER LEARNING OF THE SESSION

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 5th edition.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier, Second Edition, John Wiley & Sons, Inc., 2015.
3. Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, by Hamid R. Nemati and Li Yang, IGI Global, 2011.
4. Forouzon B, "Cryptography and Network Security," Indian Edition, TMH (2010).

THANK YOU



Team – CACD