جامـــــعة المـــــستقبل
AL MUSTAQBAL UNIVERSITY

قســـم الامـــــن الـــــــــسيبرانـــــــي

Department of Cyber Security

Subject:

Serpent Algorithm

Class:

Second

Lecturer:

Asst. lect. Mustafa Ameer Awadh

Lecture: (8)

Serpent Algorithm

Introduction The Serpent algorithm is a symmetric-key block cipher that was one of the five finalists in the **Advanced Encryption Standard (AES) competition**. Developed by **Ross Anderson, Eli Biham, and Lars Knudsen**, Serpent was designed to be highly secure and efficient. Although it was not selected as the final AES standard (Rijndael was chosen instead), Serpent remains a strong encryption algorithm used in various security applications.

Key Features of Serpent:

- **Block size**: 128 bits
- **Key sizes**: 128, 192, or 256 bits
- **Number of rounds**: 32 rounds
- **Structure**: Substitution-Permutation Network (SPN)
- **Designed for high security**, resisting differential and linear cryptanalysis

Serpent Algorithm Structure

Serpent follows a **Substitution-Permutation Network (SPN)** structure. It processes 128-bit plaintext blocks through a series of transformations that include **key expansion, substitution, linear transformation, and XOR operations.**

1. Key Expansion

Serpent supports **128, 192, or 256-bit keys**. The key expansion process generates **33 subkeys** of 128 bits each, which are used in the 32 rounds of encryption.

2. Encryption Process

Each encryption round in Serpent consists of the following steps:

- **Key Mixing**: The current round subkey is XORed with the block.
- **S-Box Substitution**: Serpent uses 8 different S-boxes, each applied in a round-wise pattern.
- **Linear Transformation (LT)**: A bitwise linear transformation increases diffusion.
- **Final Permutation**: The final round omits the linear transformation and instead applies an additional key mixing step.

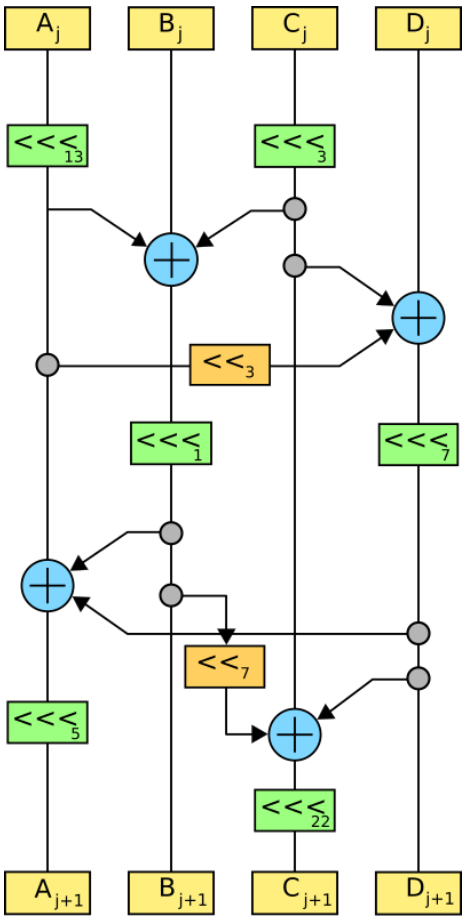

Figure 1: one round of the Serpent encryption algorithm.

The structure in the figure (1) represents one round of the Serpent encryption algorithm. Here's a breakdown of its components:

1. Input Variables (Aj, Bj, Cj, Dj)

- These are four 32-bit words that make up the 128-bit block.
- They are processed through different transformations in each round.

2. Rotations and XOR Operations

- **Bitwise rotations (<<<)**
  - The green boxes indicate left circular shifts by different amounts (e.g., 13, 3, 7, etc.).
  - These shifts help in diffusion, ensuring that small changes in input affect the entire ciphertext.
- **XOR operations (+ in blue circles)**
  - XOR operations are performed between rotated values to increase complexity and improve security.

3. Linear Transformations (LT)

- The orange boxes represent linear transformations.
- They enhance diffusion, spreading small input changes across the entire block.

4. Output Variables (Aj+1, Bj+1, Cj+1, Dj+1)

- After all transformations, the modified 128-bit block (split into four words) is passed to the next round.

Purpose of This Structure

- This step is repeated for 32 rounds in Serpent.
- The combination of rotations, XORs, and linear transformations makes it highly resistant to cryptanalysis.

3. Decryption Process

Decryption follows the same steps as encryption but in **reverse order**, using the same subkeys.

Security Analysis

Serpent was designed to maximize security. Some of its strengths include:

- **High resistance to differential and linear cryptanalysis**
- **32 rounds provide strong protection** (compared to AES's 10, 12, or 14 rounds)
- **Strong S-box design**, ensuring non-linearity and security against various attacks

However, Serpent is slower than AES in software implementations, which is one reason AES (Rijndael) was chosen instead.

Advantage	Disadvantage
Highly secure due to 32 rounds	Slower than AES in software
Resistant to most known cryptanalysis attacks	More complex structure
Supports 128, 192, and 256-bit keys	Less widely adopted than AES