In-Depth Study of Simplified AES (S-AES)

# 🔐 Overview

Simplified AES (S-AES) is a scaled-down version of the full AES algorithm, designed primarily for educational purposes. It operates on 16-bit plaintext and uses a 16-bit key, which allows for manual implementation and analysis. Despite its simplicity, S-AES retains the core structure of AES, including substitution, permutation, and mixing operations, as well as key expansion.

# 🔄 Core Operations in S-AES

## 1. SubNib (Substitute Nibbles)

The 16-bit state is divided into four 4-bit nibbles. SubNib replaces each nibble using a predefined S-Box to introduce non-linearity and confusion. The S-Box for S-AES is smaller (4-bit), allowing for 16 entries, unlike the 8-bit S-Box used in standard AES.

S-AES S-Box Example:

|  |  |
| --- | --- |
| Input (Hex) | Output (Hex) |
| 0 | 9 |
| 1 | 4 |
| 2 | A |
| 3 | B |
| 4 | D |
| 5 | 1 |
| 6 | 8 |
| 7 | 5 |
| 8 | 6 |
| 9 | 2 |
| A | 0 |
| B | 3 |
| C | C |
| D | E |
| E | F |
| F | 7 |

## 2. ShiftRows

ShiftRows in S-AES works on a 2x2 matrix of nibbles. The second row of the state matrix is rotated left by one position, which helps achieve diffusion by spreading byte dependencies across columns.

Example:  
Before:  
[a0 a1]  
[a2 a3]  
After:  
[a0 a1]  
[a3 a2]

## 3. MixColumns

MixColumns transforms each column of the state using matrix multiplication in the finite field GF(2⁴). The goal is to achieve diffusion, so changes in one nibble affect the entire column.

MixColumns Matrix:  
[1 4]  
[4 1]  
  
Operations are performed using GF(2⁴) arithmetic with the irreducible polynomial x⁴ + x + 1.

## 4. AddRoundKey

AddRoundKey XORs the current state with a 16-bit round key derived from the key expansion step. This step ties encryption to the specific key and adds confusion. It is the only operation that directly uses the encryption key.

# 🧠 Key Expansion (Key Scheduling)

The 16-bit cipher key is expanded into three 16-bit round keys using word operations. Each word is an 8-bit value. Key expansion introduces diversity across rounds and defends against attacks targeting repetitive patterns in key usage.

Key Expansion Steps:  
- Split Key into w0 and w1 (each 8 bits)  
- w2 = w0 ⊕ Rcon1 ⊕ SubNib(RotNib(w1))  
- w3 = w2 ⊕ w1  
- w4 = w2 ⊕ Rcon2 ⊕ SubNib(RotNib(w3))  
- w5 = w4 ⊕ w3

Functions:  
- SubNib: Substitutes each nibble using the S-Box  
- RotNib: Swaps the two nibbles  
- Rcon: Round constants like 0x80, 0x30

# 🧩 S-Box and Inverse S-Box

S-Box is a static substitution box that maps each 4-bit input to a unique 4-bit output, introducing non-linearity. The inverse S-Box reverses this mapping during decryption.

## S-Box Mapping

Same as in SubNib section above.

## Inverse S-Box Mapping

|  |  |
| --- | --- |
| Input (Hex) | Output (Hex) |
| 0 | A |
| 1 | 5 |
| 2 | 9 |
| 3 | B |
| 4 | 1 |
| 5 | 7 |
| 6 | 8 |
| 7 | F |
| 8 | 6 |
| 9 | 0 |
| A | 2 |
| B | 3 |
| C | C |
| D | 4 |
| E | D |
| F | E |

# ⚙️ CTR Mode (Counter Mode)

CTR mode transforms a block cipher into a stream cipher. Instead of chaining blocks like CBC, it encrypts a continuously incrementing counter and XORs it with the plaintext.

Encryption:  
Ciphertext = Plaintext XOR Encrypt(Counter)  
  
Decryption:  
Plaintext = Ciphertext XOR Encrypt(Counter)  
  
Properties:  
- Counters must be unique for each block.  
- Enables parallel encryption.  
- Stateless and efficient.

# 🧠 Cryptanalysis & Brute Force Techniques

## 1. Brute-Force Attacks

S-AES has a 16-bit key space, which allows only 65,536 possible keys. This makes brute-force attacks feasible in a matter of seconds using modern hardware. Therefore, S-AES is not secure but is great for demonstrating brute-force cryptanalysis.

## 2. Differential and Linear Cryptanalysis (conceptual)

Differential Cryptanalysis:  
- Tracks differences in plaintext pairs and how they affect output differences.  
- Useful for identifying high-probability differential trails to deduce key bits.  
  
Linear Cryptanalysis:  
- Approximates nonlinear operations using linear equations.  
- Exploits statistical biases to infer information about the key.

## 3. Known Plaintext Attack

If an attacker has access to both plaintext and its corresponding ciphertext, they can derive parts of the key or reduce the search space significantly by analyzing the transformations. This is more practical in S-AES due to the small state and key size.

# 📘 Summary Table

|  |  |
| --- | --- |
| Feature | Description |
| Block Size | 16 bits |
| Key Size | 16 bits |
| Rounds | 2 + initial AddRoundKey |
| Operations | SubNib, ShiftRows, MixColumns, AddRoundKey |
| Key Scheduling | SubNib, RotNib, XOR, Rcon |
| S-Box | 4-bit static lookup |
| Mode of Operation | CTR (Counter Mode) |
| Brute Force Feasibility | Very easy (2¹⁶ keys) |
| Cryptanalysis | Supports differential/linear in theory |