Assignment 2
CSE 667

Ryan Holthouse, Matt O'Connor, Manthan Patel, Caleb Alexander, Huy Tran

1. Key Length

- **Initial 32-bit key length was selected**
- We selected a 32-bit key length as it was the largest within the bounds of the assignment, and thus the most cryptographically secure at its basis
- From there, we can run the limited 32-bit key through SHA-256 to buffer and pad it to a greater degree

1 0 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 1 0 0 0
1 0 1 1 0 1 1 1

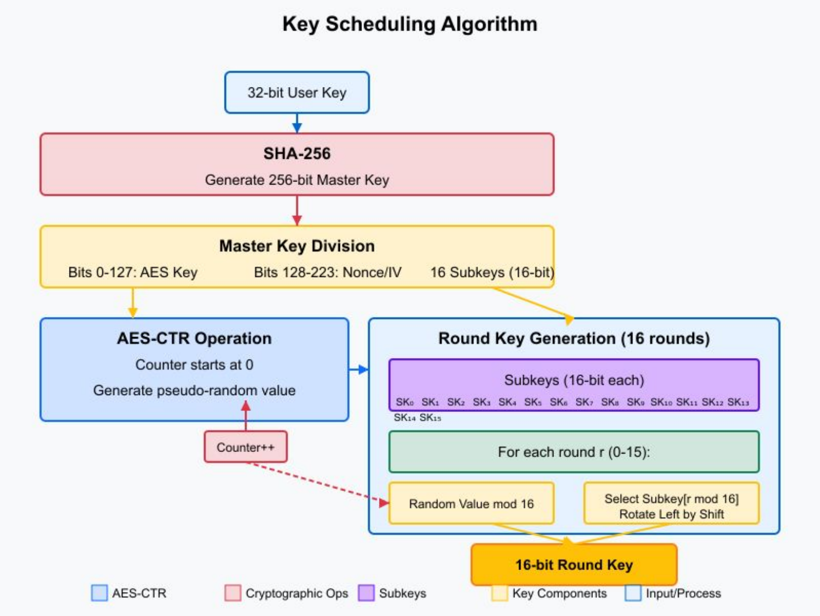
32 - bits

2. Key Scheduling

- Master key: 32-bit key is passed through SHA-256 to produce 256-bit master key.

- The 256-bit master key is divided into 16 subkeys, 16 bits each

- Round key generation: For each of the 16 rounds:
  - AES-CTR generates a pseudo random value using the current counter.
  - This value modulo 16 determines a shift amount (0-15)
  - The appropriate subkey (round mod 16) is **rotated left** by this shift amount
  - The resulting 16-bit value serves as the round key
  - The counter is incremented after each round key generation



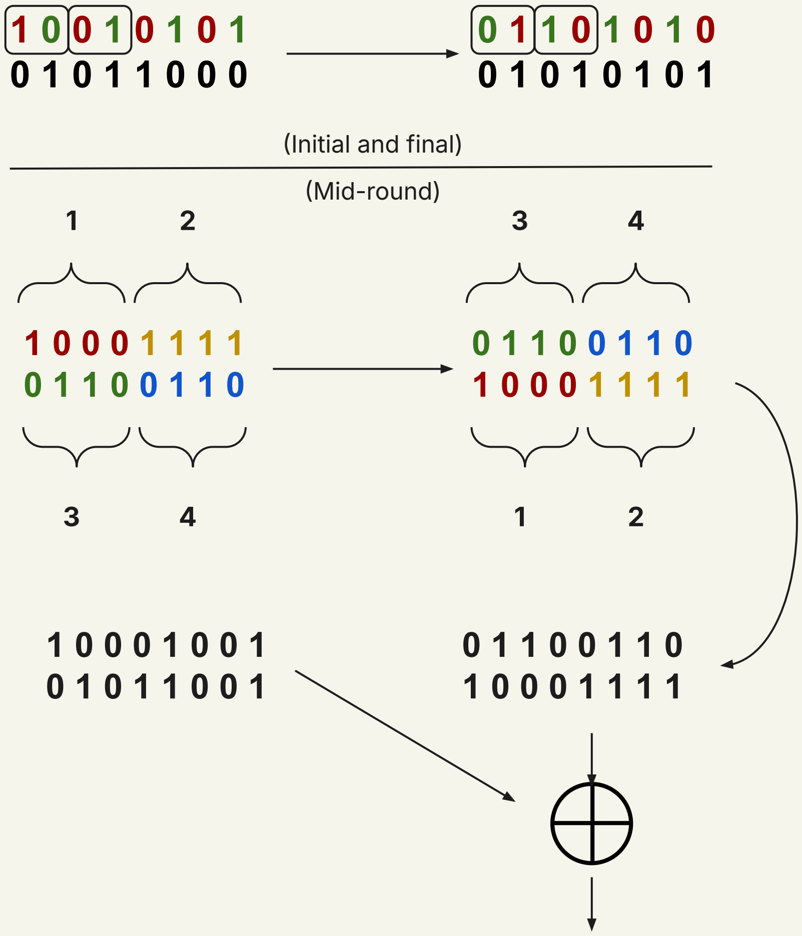
3. Permutations

- The first/initial shift or permutation involves swapping each pair of bits in the original right side of the first key. 1 & 2, 3 & 4, etc.

- Permutations during each round are as follows:
  - First, the remaining/resulting 16-bit output is broken down into 4 sections of 4 bits. These are swapped, as groups 1 & 3 and groups 2 & 4

  - Lastly, the result is put through an XoR function with the original left side of the key

- The final permutation takes place after all 16 rounds, and is a repeat of the initial permutation approach



4. S-Box Construction

Bent functions

Input	Output
00	0
01	1
10	1
11	0

Linear Function Truth Table

Input	Output
00	1
01	0
10	0
11	1

Bent Function Truth Table

S-Box Construction: Choice of functions

We will use functions of the form

$$\textbf{if} \quad x = (x_1, x_2, x_3, \ldots, x_n) \quad \textbf{then} \quad f(x) = x_a x_b \oplus x_c x_d \oplus \cdots \oplus x_i x_j$$
$$\textbf{where} \quad a, b, c, d, \ldots, i, j \in \{1, \ldots, n\} \wedge a \neq b \neq c \neq \cdots \neq j$$

$$f_1(x) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_7x_8$$
$$f_2(x) = x_1x_3 \oplus x_2x_4 \oplus x_5x_7 \oplus x_6x_8$$
$$f_3(x) = x_1x_4 \oplus x_2x_3 \oplus x_9x_{10} \oplus x_{11}x_{12}$$
$$f_4(x) = x_5x_6 \oplus x_7x_8 \oplus x_{13}x_{14} \oplus x_{15}x_{16}$$
$$f_5(x) = x_9x_{10} \oplus x_{11}x_{12} \oplus x_1x_5 \oplus x_2x_6$$
$$f_6(x) = x_3x_7 \oplus x_4x_8 \oplus x_{13}x_{15} \oplus x_{14}x_{16}$$
$$f_7(x) = x_1x_9 \oplus x_2x_{10} \oplus x_3x_{11} \oplus x_4x_{12}$$
$$f_8(x) = x_5x_{13} \oplus x_6x_{14} \oplus x_7x_{15} \oplus x_8x_{16}$$
$$f_9(x) = x_1x_6 \oplus x_2x_5 \oplus x_3x_8 \oplus x_4x_7$$
$$f_{10}(x) = x_9x_{14} \oplus x_{10}x_{13} \oplus x_{11}x_{16} \oplus x_{12}x_{15}$$
$$f_{11}(x) = x_1x_{14} \oplus x_2x_{13} \oplus x_3x_{16} \oplus x_4x_{15}$$
$$f_{12}(x) = x_5x_{10} \oplus x_6x_9 \oplus x_7x_{12} \oplus x_8x_{11}$$
$$f_{13}(x) = x_9x_6 \oplus x_{10}x_5 \oplus x_{11}x_8 \oplus x_{12}x_7$$
$$f_{14}(x) = x_1x_{11} \oplus x_2x_{12} \oplus x_3x_9 \oplus x_4x_{10}$$
$$f_{15}(x) = x_5x_{15} \oplus x_6x_{16} \oplus x_7x_{13} \oplus x_8x_{14}$$
$$f_{16}(x) = x_{13}x_2 \oplus x_{14}x_1 \oplus x_{15}x_4 \oplus x_{16}x_3$$

- The input size must be even.
- Each “term” must contain exactly 2 bits.
- The concatenation operation is multiplication **mod** 2 (AND).
- The additive operation is addition **mod** 2 (XOR).

S-Box Construction: The S-Box Matrix

For an n-length input we can consider the output of our S-Box to be →

$$S(x) = (f_1(x), f_2(x), f_3(x), \ldots, f_{16}(x)).$$

But there is a problem, this construction will not necessarily give a *balanced* output. Some inputs could go entirely to 0, others entirely to 1. Outputs could also have obvious patterns such as 0101...01

S-Box Construction: Achieving Balance

To achieve balance, we must pick an appropriate nxn matrix A, and nx1 vector b. Our final output will be →

$$S'(x) = A \cdot S(x) + b$$

b should be pseudo-random and should have no obvious patterns.

A should be invertible and have good cryptographic properties such as high diffusion

S-Box Construction: Our Choice of A

$$
\begin{bmatrix}
0&1&0&0&1&0&0&0&1&1&0&1&0&1&1&0\\
0&1&1&0&0&1&0&0&1&0&1&1&0&1&0&1\\
0&0&1&1&0&0&1&0&0&1&0&1&1&0&1&0\\
1&0&0&1&0&0&0&1&1&0&1&0&1&1&0&1\\
1&0&0&0&0&1&0&0&0&1&1&0&1&1&0&1\\
0&1&0&0&0&1&1&0&0&1&0&1&1&0&1&1\\
0&0&1&0&0&0&1&1&1&0&1&0&0&1&0&1\\
0&0&0&1&1&0&0&1&1&1&0&1&1&0&1&0\\
1&1&0&1&0&1&1&0&0&1&0&0&1&0&0&0\\
1&0&1&1&0&1&0&1&0&1&1&0&0&1&0&0\\
0&1&0&1&1&0&1&0&0&0&1&1&0&0&1&0\\
1&0&1&0&1&1&0&1&1&0&0&1&0&0&0&1\\
0&1&1&0&1&1&0&1&1&0&0&0&0&1&0&0\\
0&1&0&1&1&0&1&1&0&1&0&0&0&1&1&0\\
1&0&1&0&0&1&0&1&0&0&1&0&0&0&1&1\\
1&1&0&1&1&0&1&0&0&0&0&1&1&0&0&1
\end{bmatrix}
$$

Credits to M. Tolga Sakallı and Bora Aslan

Found at http://www.singacom.uva.es/~edgar/cactc2012/trabajos/CACT2012_Sakalli.pdf

5. Encryption & Decryption

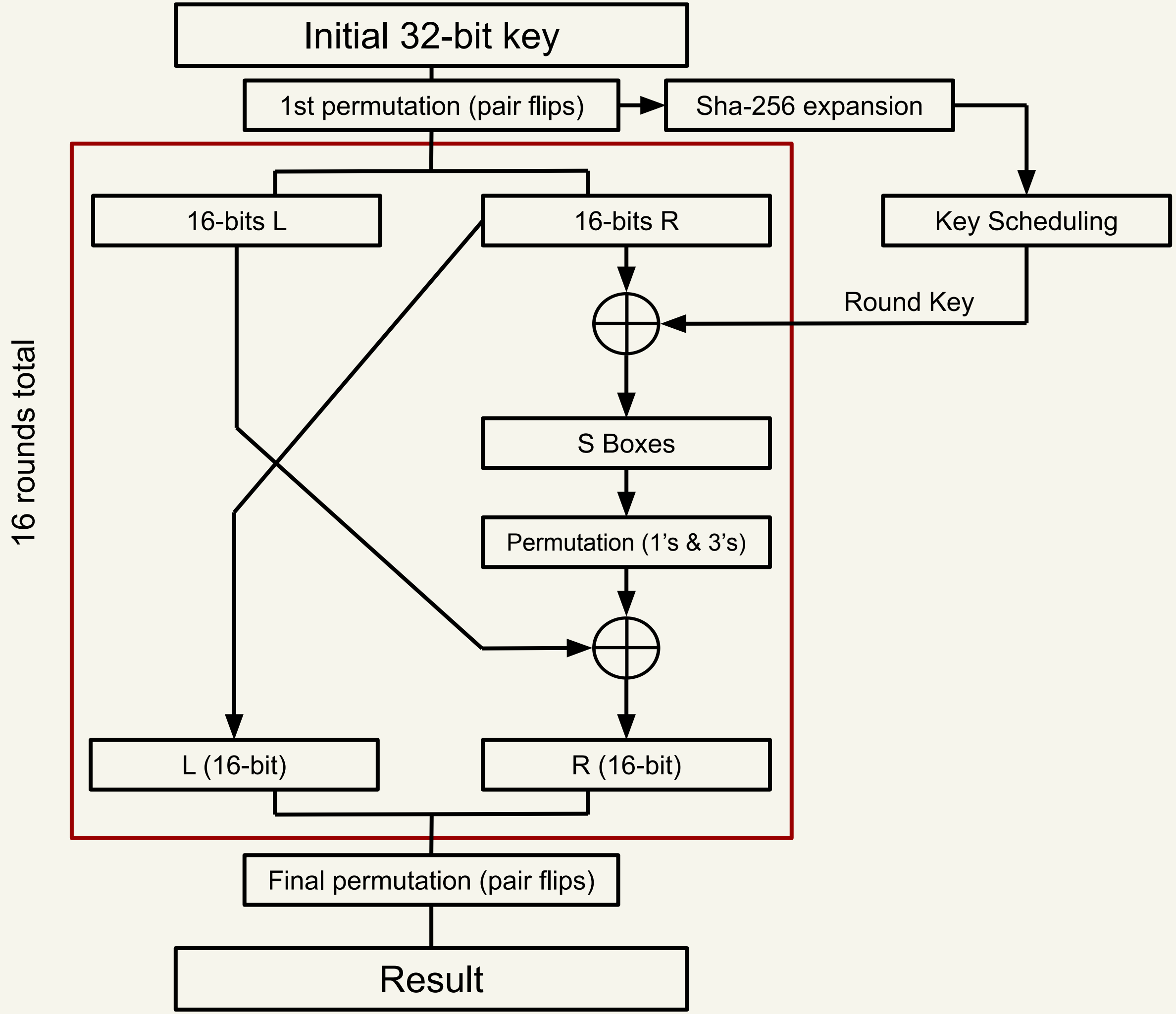
Our algorithm proceeds as below:

1. Upon receiving a 32 bit message, M, we construct a 32 bit key, K.
2. K is expanded to 256 bits, using SHA-256 and then split into 16 different “round” keys K1,...,K16.
3. M is passed through an initial permutation, as previously described.
4. The encryption rounds begin:
    a. Every round begins by splitting the current M into two halves L and R.
    b. We compute R′=S(R XOR K_n)XOR L
        i. R is XORed with the round key, passed through our S box, and XORed with L to become R′.
    c. We say that L′ = R.
    d. L′ and R′ are used as the left and right for next round respectively.
5. After 16 rounds, we pass the modified M through a final permutation which is the inverse of our initial permutation.

Decryption is the reverse of encryption:

1. Clearly IP and FP are invertible and memorized.
2. At each round the round key is obtainable because we have the initial key.
3. At each round the L should require no work because it is the right half of the previous round.
4. Recall that R is S(R_{n-1} XOR K_{n-1})XOR L_{n-1}. So we can retrive L_{n-1} of the previous round by taking L=R_{n-1}, XORing it with our previous round key and passing it through the S-Box. Then L_{n-1}=S(L XOR K_{n-1}) XOR R

Overall View



6. Security Analysis

- Our key length is the longest possible which gives security against brute-force attacks.

- Each of our permutations provides defense against pattern matching.

- Our suitably non-linear S-box provides defense against differential and linear attacks as well as obscuring the original information about the message (number of 0s and 1s etc.).

- 16 rounds means that the reverse engineering of the process and key are rightfully complex, with many points for an attacker to make errors.

Summary

Our algorithm combines several strong features:

- Key expansion using SHA-256
- Permutations
- Cryptographically strong S-boxes based on bent functions
- 16 rounds of operations

Future improvements could include adding salt/nonce to the initial key, exploring alternative S-box constructions and resistance against side channel attack.

Questions