LED Block Cipher

Team - RSA

Department of CS
Indian Institute of Technology Bhilai

December 8, 2024

Outline

Introduction to LED Cipher

- **Optimization for Lightweight Applications:** AES and its derivatives are primarily optimized for high-speed software performance but struggle to deliver lightweight hardware implementations.
- **Relevance in IoT and Resource-Constrained Devices:** The rapid growth of the Internet of Things (IoT) and resource-constrained devices has created a demand for block ciphers that are both hardware-compact and software-efficient.
- **Significance of LED Cipher:** The LED cipher addresses this demand by being lightweight while achieving hardware-compactness and software efficiency.

Introduction to LED Cipher

- **Key Objectives of LED Cipher:**
  1. An **ultra-light key schedule**, minimizing computational overhead.
  2. **Resistance to related-key and single-key attacks**, ensuring robust security.
- **Comparison with Other Lightweight Ciphers:**
  - Many lightweight ciphers are susceptible to key-related attacks:
    - HIGHT cipher: Vulnerable to a known related-key attack (K+-2010).
    - Hummingbird-1 and KTANTAN: Compromised by practical related-key attacks (S-2011 and A-2011, respectively).
  - LED cipher demonstrates resistance to such attacks despite having an almost negligible key-scheduling mechanism.

Outline

Specs

- **Two Variants:** The LED cipher has two primary variants:
    - A 64-bit block cipher with a 64-bit key.
    - A 64-bit block cipher with a 128-bit key.
- **Focus on 64-bit Key Variant:** For simplicity, this discussion focuses solely on the 64-bit key variant.
- **State Structure:** The LED cipher operates on a 4×4 state matrix, where each element (nibble) is derived from $GF(2^4)$.
- **Field Polynomial:** The polynomial used for field multiplication is $X^4 + X + 1$.

Similarities With AES

1. **AddConstants:**
   - In each round, XORs a round-dependent constant with the first two columns of the state.
   - Round constants are initialized with all zeroes and shifted left cyclically. Example: $\mathrm{rc}_0 = \mathrm{rc}_5 \oplus \mathrm{rc}_4 \oplus 1$.
2. **SubCells:**
   - Substitutes each nibble in the state using the LED S-box, which is identical to the S-box used in PRESENT.
3. **ShiftRows:**
   - Permutes the nibbles in the state by rotating each row to the left by i positions, where i is the row index.
4. **MixColumnsSerial:**
   - Applies the MixColumns operation for diffusion, differing from AES.
   - This process can be interpreted as applying a hardware-friendly matrix A four times to derive the MDS matrix M

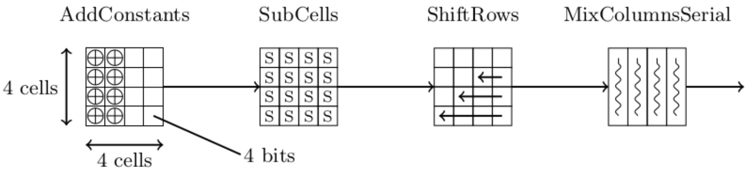
Differences From AES

- **Introduction of Step:**
  - The operation `Step(STATE)` consists of four encryption rounds.
  - Each round performs the following operations in sequence: `AddConstants`, `SubCells`, `ShiftRows`, and `MixColumnsSerial`.
- **MDS Matrix:**
  - LED uses a hardware-friendly MDS matrix optimized for serial implementation.
  - $$(A)^4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 1 & 2 & 2 \end{pmatrix}^4 = \begin{pmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{pmatrix} = M$$

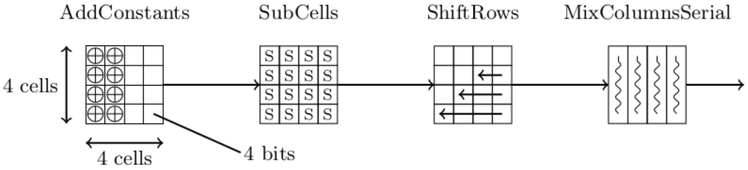
Differences From AES

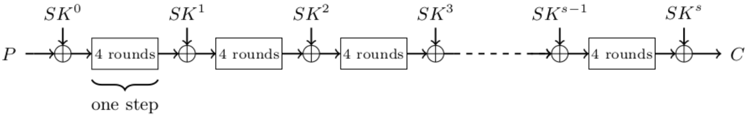
- **Loading the State:**
  - The state is loaded **row-wise**, unlike AES, which loads the state **column-wise** (more **hardware-friendly**).
- **Ultra-light Key Schedule:**
  - LED eliminates the traditional key scheduling process.
  - The user-provided key is directly reused across rounds without modification.

Round Functions



Encryption


AddConstants
SubCells
ShiftRows
MixColumnsSerial
4 cells
4 cells
4 bits
S


SK^0
SK^1
SK^2
SK^3
SK^{s-1}
SK^s
P
C
4 rounds
one step

Outline

Differential Cryptanalysis

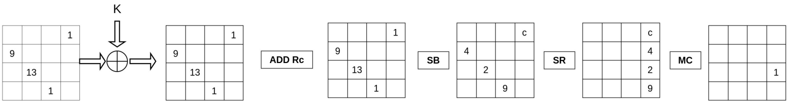
- We Performed Differential cryptanalysis for One step(i.e. 4 Rounds) of LED Cipher
- The maximum differential probability of the Present Sbox is 2^{-2}
- The total differential probability for four round is 2^{-50}

Differential Cryptanalysis

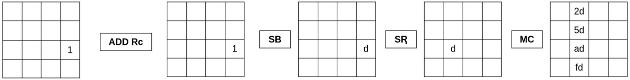
Maximum Differential Prob=$(1/2)^2$

Total Differential Probability=2^{-50}

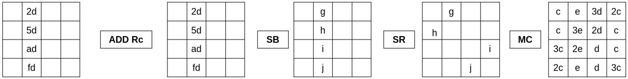
Round 1 *$Pr[\Delta S_1^I \text{--->} \Delta S_1^{SB}]=(4/16)^4 \Rightarrow 2^{-8}$*



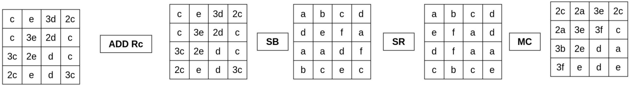
Round 2 *$Pr[\Delta S_2^I \text{--->} \Delta S_2^{SB}]=(4/16) \Rightarrow 2^{-2}$*



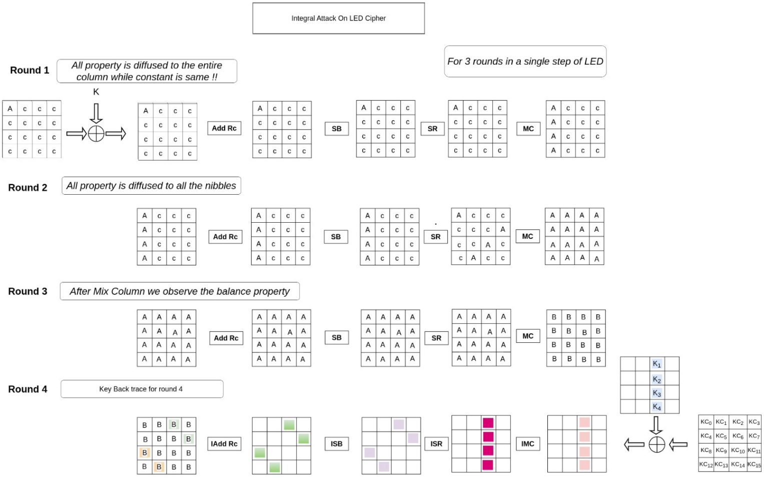
Round 3 *$Pr[\Delta S_3^I \text{--->} \Delta S_3^{SB}]=(4/16)^4 \Rightarrow 2^{-8}$*



Round 4 *$Pr[\Delta S_4^I \text{--->} \Delta S_4^{SB}]=(4/16)^{16} \Rightarrow 2^{-32}$*



Integral Cryptanalysis



Outline

Hash Construction

- **Hash Function Implementation:**
  - A hash function was implemented using the LED encryption function.
  - The implementation is based on the **Davies-Mayer construction**.
- **Attacks Performed:**
  - Attempted the following attacks:
    - **Pre-image Attack**
    - **Second Pre-image Attack**
    - **Collision Detection Attack**
- **Attack Results:**
  - All attacks were unsuccessful.
  - The primary reason was the **high-order complexity** involved.

Decrpytion

- **Unexpected Challenge:**
  - Discovered the absence of an existing implementation for the **decryption function** of LED.
- **Key Operation:**
  - Successfully incorporated the `InverseMixColumns` operation for decryption.
- **Outcome:**
  - Developed a fully functional and verified decryption function for LED.

Outline

Conclusion

- The most significant difference between AES and LED Block cipher is their Key-Schedule Algorithm.
- The adaptability of LED to various encryption needs, whether it's low-resource devices or high-throughput systems, demonstrates its versatility.
- Compared to Speck and Simon, LED offers a unique balance of speed, security, and simplicity, making it ideal for low-energy devices that need fast encryption without compromising security.

FB19E4FA8DC1E084D57D08F6724B62B6

Team Members

- Rishit Agarwal (12141380)
- Sourabh Dadaore (12141580)
- Abhishek Singh Kushwaha (12140050)

Implementation Info

- Github Link: https://github.com/ASK-03/LED-Cipher.git