Simplified AES Under Attack: A Peer into Differential Cryptanalysis

Aiden Rivera

December 10, 2024

Contents

1 Abstract

This paper presents the implementation and cryptanalysis of a 2-round Simplified Advanced Encryption Standard (S-AES) cipher using C++. S-AES, a pedagogical variant of the Advanced Encryption Standard (AES), was analyzed for its strengths and vulnerabilities through brute force and differential cryptanalysis. The implementation encompassed key encryption and decryption routines, with correctness verified against expected outputs.

Brute force cryptanalysis exploited the cipher's small 16-bit key space, systematically recovering the key within feasible computational limits. Differential cryptanalysis revealed significant vulnerabilities by leveraging high-probability differentials derived from the S-box, allowing the key to be deduced with fewer attempts than brute force. The study highlighted S-AES's susceptibility to attacks due to its limited rounds, weak diffusion, and predictable non-linear transformations.

These findings underscore the utility of S-AES as an educational tool while emphasizing its limitations in providing robust security. Recommendations include increasing the number of rounds, enhancing S-box design, and expanding key sizes to improve security. This study offers valuable insights into cryptographic principles and serves as a foundation for further research in lightweight encryption and cryptanalysis.

2 Summary

This paper explores the implementation and analysis of the Simplified Advanced Encryption Standard (S-AES) using C++. The work involves three primary tasks: implementing the 2-round S-AES algorithm for encryption and decryption, performing brute force cryptanalysis, and theoretically applying differential cryptanalysis to evaluate the cipher's security.

S-AES, a version of AES created for educational purposes, operates on 16-bit blocks and uses a 16-bit key with two encryption rounds. Its simplicity makes it an ideal candidate for exploring cryptanalysis methods.

Performing brute force cryptanalysis by iterating through the key space of 2^{16} (65,536 possible keys) highlights the vulnerabilities such as the limited diffusion of bits and the small key size within the cipher. Key collisions and weak key scheduling were identified as factors contributing to its susceptibility.

Differential cryptanalysis, a more nuanced attack method, is a method of cryptanalysis in which patterns in input differences are studied as they propagate through the cipher's operations. A Differential Distribution Table (DDT) can be generated for the S-box, and high-probability differentials can be identified through this table. Using these differentials, a systematic approach can be followed to deduce round key candidates by matching predicted and observed ciphertext differences. Statistical analysis of multiple plaintext pairs enabled narrowing down the possible keys.

3 Introduction

Cryptography is a cornerstone of secure communication, with block ciphers playing a pivotal role in encrypting sensitive information. The Simplified Advanced Encryption Standard (S-AES) is a reduced, pedagogical version of the Advanced Encryption Standard (AES), designed to help students and researchers explore fundamental cryptographic principles. S-AES operates on 16-bit blocks, uses 16-bit keys, and typically employs two encryption rounds. Its simplicity makes it an excellent candidate for understanding the mechanics of encryption and the vulnerabilities that cryptanalysis can exploit.

This paper focuses on implementing and analyzing a 2-round S-AES cipher using C++. The project has three objectives:

- Implementing the S-AES encryption and decryption process
- Performing brute force cryptanalysis to identify vulnerabilities inherent in the small key size and weak diffusion
- Exploring differential cryptanalysis on S-AES

The scope of this project is limited to simply exploring Differential Cryptanalysis and gaining a base level understanding of the attack and its theoretical implementation on S-AES, chosen for its tractability in educational settings. The attack is not implemented only because it would be significantly more computationally expensive to implement the attack compared to brute force methods.

4 Discussion

The implementation and analysis of the Simplified Advanced Encryption Standard (S-AES) in this project were divided into three main components: the development of the encryption and decryption processes, brute force cryptanalysis, and differential cryptanalysis.

4.1 Implementation

The S-AES cipher was implemented in C++ with a focus on modularity and clarity. The encryption process follows the standard S-AES operations:

- AddRoundKey
- SubNyb
- ShiftRows
- MixCols

Each operation was implemented as a standalone function to ensure clarity and reusability. The decryption process inverses these operations in the appropriate sequence.

4.2 Brute Force Cryptanalysis

Brute force cryptanalysis was applied to exploit the small key space of S-AES. With a 16-bit key, the cipher has $2^{16} = 65,536$ possible keys, making it feasible to test all possible keys systematically. The key space's small size is a significant vulnerability, as a modern computer can easily exhaustively search this range.

Key collisions, weak key scheduling, and limited diffusion further weaken the cipher. The two-round structure does not adequately disperse plaintext information throughout the ciphertext, leaving the cipher susceptible to systematic attacks. These findings illustrate the importance of key size and diffusion mechanisms in designing robust cryptographic systems.

4.3 Differential Cryptanalysis

Differential cryptanalysis was studied as a more sophisticated attack method to evaluate S-AES. This approach leverages predictable patterns in how input differences propagate through the cipher to deduce information about the secret key. While a method was not implemented in this paper, a theoretical method of attack is outlined below.

- **Analyze the S-box:** Generate A Differential Distribution Table (DDT) to map how input XOR differences translate into output XOR differences. Identify High-probability differentials to construct potential attack vectors.
- **Choose a target differential** Select an input difference (ΔP) based on high-probability differentials from the DDT.
- **Encrypt plaintext pairs:** Encrypt plaintext pairs differing by ΔP and record the ciphertext differences.
- **Deduce round key candidates:** Compare observed ciphertext differences to predictions based on the differential trail. Potential round key candidates will be identified.
- **Narrow down candidates:** Use statistical analysis to focus on keys appearing consistently across multiple plaintext pairs.

This analysis revealed key dependencies between the S-box properties and the cipher's resistance to differential attacks. S-AES's limited rounds and weak diffusion made it especially vulnerable, as the trails from high-probability differentials could be directly exploited.

5 Results

The implementation and cryptanalysis of the Simplified Advanced Encryption Standard (S-AES) produced the following key findings:

5.1 Implementation Results

- The 2-round S-AES encryption and decryption processes were successfully implemented in C++, with correct outputs verified against expected values.
- Key components, including **AddRoundKey**, **SubNyb**, **ShiftRows**, **MixCols**, and their inverses, performed as designed.
- Modular implementation enabled seamless integration of encryption, decryption, and brute force functions.

5.2 Brute Force Cryptanalysis

- The brute force attack tested all 2^{16} (65,536) possible keys to decrypt a known plaintext-ciphertext pair.
- The correct key was successfully recovered within a feasible computational time, accentuating the vulnerability of S-AES's small key space.
- Key collisions and weak diffusion were observed, with multiple plaintexts mapping to overlapping ciphertext patterns due to the limited number of rounds as well as a small key size.

5.3 Differential Cryptanalysis

- Implementing differential cryptanalysis proves to be a moot exercise, as the computational cost of generating multiple plaintext pairs and encrypting them would be far greater than the cost of brute forcing this cipher.

5.4 Comparison of Cryptanalysis Methods

Method	**Strength**	**Weakness**	**Outcome**
Brute Force	Guaranteed to find the key; simple.	Inefficient for larger key spaces; requires exhaustive search.	Inefficient for larger key spaces; requires exhaustive search.
Differential Analysis	Exploits cipher design flaws; scalable.	Requires detailed understanding of cipher structure.	Did not implement.

Table 1: Comparison of Cryptanalysis Methods for S-AES

5.5 Limitations of S-AES

- The small key size of 16 bits and the limited two-round structure make S-AES highly susceptible to both brute force and differential cryptanalysis.
- The low diffusion of plaintext bits into ciphertext after two rounds significantly reduces the cipher's resistance to statistical attacks.

6 Conclusions

This study implemented and analyzed the 2-round Simplified Advanced Encryption Standard (S-AES) using C++. Through brute force and differential cryptanalysis, the strengths and vulnerabilities of this simplified cipher were thoroughly examined.

- **Vulnerabilities:** The cipher's 16-bit key size and 2-round structure make it highly insecure against modern cryptographic attacks. Brute force cryptanalysis effectively exploited the small key space, while differential cryptanalysis demonstrated significant vulnerabilities due to weak diffusion and predictable S-box properties.
- **Effectiveness of Differential Cryptanalysis:** Whilst an implementation of differential cryptanalysis was not shown, the power of differential cryptanalysis as an attack method remains unparalleled as an effective attack on block ciphers.
- **Limitations of S-AES:** The simplicity of S-AES, while beneficial for educational purposes, limits its applicability as a secure encryption standard. The observed weaknesses, including key collisions and insufficient rounds for effective diffusion, are inherent to its design.

The above key conlusions demonstrates the practical application of cryptanalysis methods and reinforces the need for strong design principles in cryptography, such as larger key sizes, robust non-linear transformations, and sufficient rounds for diffusion.

7 Recommendations

Based on the findings from the implementation and cryptanalysis of the 2-round Simplified Advanced Encryption Standard (S-AES), the following recommendations are proposed:

- **Expand S-AES Rounds:** Increasing the number of encryption rounds can improve diffusion and significantly enhance the cipher's resistance to cryptanalysis methods such as differential cryptanalysis. Testing S-AES with additional rounds would provide deeper insights into the trade-offs between security and computational efficiency.

- **Explore Alternative S-box Designs:** The current S-box in S-AES was found to have predictable differential properties, making it vulnerable to cryptanalysis. Designing and testing alternative S-boxes with improved differential uniformity could strengthen the cipher against such attacks.

- **Analyze Larger Key Spaces:** While the 16-bit key size in S-AES is pedagogically convenient, it is unrealistic for practical cryptographic applications. Implementing and analyzing versions of S-AES with larger key sizes (e.g., 64-bit or 128-bit keys) would align the study more closely with modern encryption standards.

- **Develop Automated Tools for Cryptanalysis:** Building automated tools for generating differential distribution tables, selecting high-probability differentials, and statistically analyzing key candidates would streamline the cryptanalysis process and make it more scalable for studying other ciphers.

- **Apply Cryptanalysis to Full AES:** To bridge the gap between educational tools and real-world encryption, applying the methods theorized in this study to a small-scale implementation of the full AES algorithm would provide valuable comparative insights.

References

[1] Heys, Howard M. *A Tutorial on Linear and Differential Cryptanalysis.* Cryptologia, vol. 26, no. 3, July 2002, pp. 189–221. DOI: `https://doi.org/10.1080/0161-110291890885`.

[2] Schneier, Bruce. *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C.* Wiley, Indianapolis, IN, 2015.

[3] Trappe, Wade, and Lawrence C. Washington. *Introduction to Cryptography: With Coding Theory.* Pearson Education, Hoboken, New Jersey, 2020.