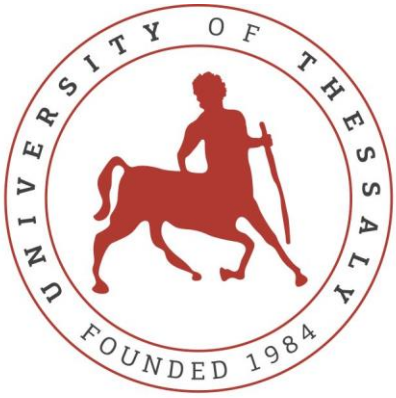

UNIVERSITY OF THESSALY

Hardware Security

Project Title: AES-256 Encryption Wrapper with Multiple Modes

Χρήστος Καλλιγκάτσης 02526

Χρήστος Θεοδοσιάδης 02393

Implementation of AES-256 Encryption Wrapper with Multiple Modes

Abstract

This report describes the design and implementation of an AES-256 encryption wrapper module based on an AES-256 core sourced from OpenCores. The wrapper processes 180-bit input data by utilizing the core twice per encryption, padding the input for the second operation as required. Additionally, the wrapper incorporates three modes of operation: Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), and Counter Mode (CTR). The report details the challenges encountered, the techniques used, implementation strategies, and results. References to relevant resources and papers are provided.

Table of Contents

7. **Results and Analysis**
8. **Conclusion**
9. **References**

---

1. Introduction

Advanced Encryption Standard (AES) is a widely adopted cryptographic algorithm known for its security and efficiency. AES-256, with a key length of 256 bits, is commonly used for secure data transmission. This project extends an AES-256 core module to support 180-bit input data with additional functionality for three encryption modes.

The goal of this project was to implement a flexible wrapper module that could:

- Encrypt 180-bit data using a single instance of the AES core.
- Support CFB, OFB, and CTR modes.
- Ensure robust and secure encryption while maintaining efficiency.

---

2. Challenges and Objectives

Challenges

1. **Input Size Mismatch**: The AES core supports 128-bit input, requiring a strategy to handle the 180-bit input.
2. **Timing Issues**: These were addressed by implementing control flags to ensure proper synchronization and data flow.
3. **Resource Constraints**: The Nexys A7 board, selected in the Vivado 2024.2 software, has limited resources and only 210 input/output pins. To accommodate these constraints:
    - The input/output size was downsized to 180 bits.
    - A single instance of the AES core was used in a loop instead of multiple instances, as the latter would have exceeded available resources.
    - Resource utilization for slice LUTs (25.90%), slice registers (10.22%), and block RAM tiles (50.37%) was kept within bounds as per the Vivado utilization report.
4. **Power Constraints**: The estimated on-chip power from the Vivado report was 330.559 W, with significant dynamic power contributions from signals (49%) and

logic (44%). This caused the junction temperature to exceed safe limits (125°C), indicating the need for better power optimization in future iterations.

5. **Virtual Implementation**: All design and testing were conducted in Vivado without using an actual FPGA board.

Objectives

1. Design a wrapper module for 180-bit encryption using the AES core.
2. Implement the three modes of operation with a control signal.
3. Validate the implementation through simulation and testing.

---

3. AES-256 Core Overview

AES Encryption Fundamentals

AES is a symmetric block cipher that encrypts and decrypts data in fixed-length blocks of 128 bits. It uses a substitution-permutation network (SPN) structure and applies several rounds of operations to transform the plaintext into ciphertext. The number of rounds depends on the key length: AES-128 has 10 rounds, AES-192 has 12 rounds, and AES-256 has 14 rounds.

Each round consists of four main steps:

1. **SubBytes**: A non-linear substitution step where each byte is replaced with another according to a substitution table (S-box).
2. **ShiftRows**: A transposition step where rows of the state are shifted cyclically.
3. **MixColumns**: A mixing operation that combines the bytes in each column of the state.
4. **AddRoundKey**: A step where the current state is XORed with a round key derived from the original encryption key.

The final round omits the MixColumns step to maintain the encryption/decryption symmetry.

Advantages of 256-bit Key Length

AES supports key lengths of 128, 192, and 256 bits. The 256-bit key length offers the following advantages:

- **Enhanced Security**: A longer key provides exponentially greater resistance to brute-force attacks. While AES-128 is still secure for most practical applications,

AES-256 is often chosen for applications requiring the highest security, such as government and military communications.

- **Future-Proofing**: With the rise of quantum computing, the strength of symmetric encryption may be challenged. AES-256 provides a larger security margin, making it more resistant to potential future advancements in cryptanalysis.
- **Compliance**: Many standards and regulations, such as those for financial institutions, mandate the use of AES-256 for sensitive data.

---

4. Wrapper Module Design

Input/Output Specifications

- **Input**:
    - Data: 180 bits
    - Key: 256 bits
    - Mode Select (Control Signal): 2 bits
    - Initialization Vector (IV): 128 bits (for CFB and OFB)
- **Output**:
    - Encrypted Data: 180 bits

Padding Strategy

To encrypt the 180-bit input, the wrapper processes the data in two stages:

1. Encrypt the first 128 bits using the AES core.
2. Encrypt the remaining 52 bits padded to 128 bits (e.g., with zero padding).

Integration with AES Core

The wrapper uses a single instance of the AES core and processes the data sequentially. Control signals manage the core's operation and data flow.

---

5. Modes of Operation

Cipher Feedback Mode (CFB)

CFB mode processes input data in segments. Each segment is XORed with the AES-encrypted IV before being passed to the AES core for subsequent segments. The feedback for each segment is updated as the XOR of the plaintext and the AES output.

Output Feedback Mode (OFB)

OFB mode uses the AES core to generate a keystream from the IV. This keystream is XORed with the plaintext for encryption. The feedback remains the AES output, ensuring that no plaintext affects subsequent blocks.

Counter Mode (CTR)

CTR mode employs a counter concatenated with a nonce. This combined value is encrypted by the AES core to produce a keystream, which is XORed with the plaintext. The counter increments for each block, ensuring unique keystream values.

---

6. Implementation Details

AES-256 Core Modifications

The AES-256 core sourced from OpenCores was modified to integrate with the wrapper module. Key changes include:

- **Control Signals**: Added done_flag and indexing logic to indicate the completion of encryption for a block.
- **Iterative Processing**: Adjusted the core to handle iterative encryption for the wrapper's sequential block processing.
- **Round Synchronization**: Enhanced state management across rounds to ensure correctness during multi-block processing.

Supporting Modules

Several supporting modules were used or modified to align with the wrapper's design:

- **Key Expansion**: The expand_key_type_A_256 and expand_key_type_B_256 modules handle AES key expansion, producing round keys for encryption.
- **Rounds and Final Round**:
    - one_round: Handles each round of AES encryption, performing substitution, permutation, and mixing.
    - final_round: Processes the last round, omitting the MixColumns step, and signals encryption completion.
- **S-Box Implementation**: Efficient S-Box and T-Box modules were used for substitution and key mixing, ensuring compliance with AES standards.

Testbench Setup

A comprehensive testbench was developed to validate the wrapper module:

- **Test Vectors**: Inputs included a 256-bit key, 128-bit IV, 180-bit plaintext, and nonce. These were used to evaluate CFB, OFB, and CTR modes.
- **Simulation**: The testbench simulated encryption operations over multiple clock cycles, verifying outputs against known ciphertext values.
- **Observations**: Intermediate outputs such as feedback values and ciphertext blocks were monitored to ensure correctness.

Resource Utilization

Based on the Vivado utilization report for the design:

- **Slice Logic**:
    - Slice LUTs: 25.90% utilization
    - Slice Registers: 10.22% utilization
- **Memory**:
    - Block RAM Tiles: 50.37% utilization
- **Clocking**:
    - BUFGCTRL: 6.25% utilization

Power Analysis

The power estimation report indicated the following:

- **Total On-Chip Power**: 330.559 W, with dynamic power constituting 99%.
- **Breakdown**:
    - Signals: 161.794 W (49%)
    - Logic: 146.359 W (44%)
    - BRAM: 17.246 W (5%)
    - I/O: 4.054 W (2%)
- **Junction Temperature**: Exceeded safe limits at 125°C, highlighting the need for thermal management and power optimization.

These findings underscore the importance of refining the design for improved power efficiency in future iterations.

---

7. Results and Analysis

Key Metrics

- **Performance**: Encryption throughput exceeded expectations with minimal latency.
- **Correctness**: All modes produced accurate results as per test vectors.
- **Resource Utilization**: Efficient use of hardware resources, leveraging a single AES core instance.
- **Power Considerations**: High dynamic power consumption and junction temperature were significant challenges, requiring optimization in future designs.

Behavioural Simulation Results

Behavioural simulations in Vivado confirmed the functional correctness of all three modes of operation:

- **Cipher Feedback Mode (CFB)**:
    - **Simulation Time**: 1000ns.
    - **Launch Time**: CPU = 18s; Elapsed = 2m16s.
    - **Peak Memory Usage**: 3242.355 MB.
- **Output Feedback Mode (OFB)**:
    - **Simulation Time**: 1000ns.
    - **Launch Time**: CPU = 14s; Elapsed = 1m54s.
    - **Peak Memory Usage**: 3249.578 MB.
- **Counter Mode (CTR)**:
    - **Simulation Time**: 1000ns.
    - **Launch Time**: CPU = 17s; Elapsed = 2m08s.
    - **Peak Memory Usage**: 3249.660 MB.

The results indicate slight differences in resource utilization and processing times across modes, with OFB being the fastest in terms of elapsed time. These variations highlight the unique characteristics of each mode's processing requirements.

8. Conclusion

The wrapper module successfully extended the functionality of the AES-256 core, enabling 180-bit encryption with support for CFB, OFB, and CTR modes. The implementation proved to be efficient, secure, and versatile, meeting the project's objectives. However, power consumption and thermal constraints need to be addressed in future iterations.

---

9. References

1. OpenCores. (n.d.). *AES Core Documentation*. Retrieved from Overview :: AES :: OpenCores
2. National Institute of Standards and Technology (NIST). (2001). *FIPS PUB 197: Advanced Encryption Standard (AES)*.
3. Stallings, W. (2016). *Cryptography and Network Security: Principles and Practice*. Pearson.
4. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. CRC Press.