SYMCUBE Low-Power Optimization Strategy for Embedded Platforms

1. Introduction

This document outlines strategies for deploying SYMCUBE on resource-constrained embedded platforms, specifically targeting ARM Cortex-M0+ microcontrollers. The focus is on minimizing power consumption while maintaining an acceptable level of security. This involves architectural modifications, data reduction, and careful consideration of energy-performance trade-offs.

2. Reduced RCU/SSVE Architecture for ARM Cortex-M0+

The standard SYMCUBE architecture, with its 100-unit RCU and complex SSVE, is too resource-intensive for Cortex-M0+ MCUs. We propose a reduced architecture optimized for these platforms:

Simplified RCU:

- Reduced Number of Units: Instead of 100 units, the RCU is scaled down to 48 or 64 units. This significantly reduces the memory footprint and the number of operations required for each rotation.
- Optimized Interconnect: The interconnect between units is simplified to reduce routing complexity and power consumption. A less densely connected topology can still provide sufficient diffusion for security.

Streamlined SSVE:

- Reduced Symbolic Alphabet: The symbolic alphabet is compressed (see Section 3), decreasing the size of the lookup tables and the complexity of the symbolic transformation logic.
- Simplified Transformation Functions: The symbolic transformation functions are optimized for efficient execution on the Cortex-M0+ architecture, minimizing the number of clock cycles and memory accesses.

Cortex-M0+ Specific Optimizations:

- Leverage low-power modes (Sleep, Deep Sleep) aggressively. The SYMCUBE unlock process should be designed to allow the MCU to enter low-power modes whenever possible.
- Minimize memory accesses. Store frequently used data in registers or SRAM to reduce power consumption associated with flash memory access.
- Optimize code for the ARM instruction set. Use efficient coding techniques and compiler optimizations to reduce code size and execution time.

3. Symbol Set Compression

Reducing the number of symbols directly impacts the size of the state space and the complexity of the symbolic transformations.

1. **Compression Target:**The symbol set is reduced from 100 to 48 or 64 unique symbols.
2. **Encoding Strategy:**An efficient encoding scheme is used to represent the reduced symbol set, minimizing the number of bits per symbol.
3. **Security Considerations:**The reduced symbol set must still provide sufficient entropy and diffusion to resist attacks. Cryptographic analysis is performed to ensure that the reduced set does not significantly weaken the security of the system.

4. Unlock Sequence Length Reduction Strategies

The length of the unlock sequence directly affects the energy required for the unlock operation. We explore strategies to reduce this length:

4. **Optimized IRN Topology:**A carefully designed IRN topology can achieve the required security with a shorter unlock sequence. We investigate topologies that provide faster diffusion and mixing of the input state.
5. **Adaptive Unlock Sequence:**The length of the unlock sequence can be made adaptive, based on the security context. For example, a shorter sequence can be used for low-security applications, while a longer sequence is used for high-security applications.
6. **Parallel Processing:**Where possible, parts of the unlock validation process can be parallelized to reduce the overall time taken, even if the sequence length remains the same. This can be beneficial on MCUs with some parallel processing capabilities.

5. Energy-Performance Trade-Off Analysis

There is an inherent trade-off between energy consumption and performance (unlock time) in SYMCUBE.

7. **Voltage and Frequency Scaling:**The Cortex-M0+ allows for dynamic voltage and frequency scaling (DVFS). Lowering the clock frequency and supply voltage reduces power consumption but increases the unlock time.
8. **Profiling and Optimization:**Extensive profiling is performed to identify the most energy-intensive parts of the SYMCUBE implementation. These parts are then optimized for power efficiency, potentially at the cost of some performance.
9. **Application-Specific Tuning:**The energy-performance trade-off can be tuned for specific IoT applications. For example, a battery-powered sensor might prioritize extremely low power consumption, even if it means a longer unlock time.

6. Expected Impact on Temporal Unlock Accuracy

The proposed optimizations may affect the temporal accuracy of the unlock process.

10. **Increased Timing Variability:**The reduced RCU and symbol set may lead to increased timing variability, as there are fewer operations in the unlock sequence.
11. **Calibration and Compensation:**The unlock validation logic is carefully calibrated to account for this increased variability. Techniques such as adaptive timing windows and statistical analysis are used to