SAND2016-2064C

Secure Embedded System Design Methodologies for Military Cryptographic Systems

High Integrity Software Systems

Gary N. McGovney

Sandia National Laboratories
PO Box 5800
Mail Stop 0860
Albuquerque, NM 87185

Sandia National Laboratories

Exceptional service in the national interest

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Overview

Sandia National Laboratories

- Embedded System Software Failure
- Access Control vs. Enablement
- Dual-Redundant Processor Comparator Design Architecture
- Comparator Base Element Design
- Covered Output Data Generation
- Authenticated Output Example
- Failure Calculations
- Summary

Embedded System Software Failure

Sandia National Laboratories

- Secure cryptographic embedded systems
  - Safeguard access to both data and control of external systems
  - Grant access to authorized users during normal operation, but also…
  - Shouldn't release data or control because of internal flaws or damage
- Internal or externally-induced failures do occur
  - Formalized failure type and rate calculations exist for hardware
  - Analyzing software failure rates and behavior is much more difficult
    - **What is the probability that failures will cause unauthorized output?**
- Dual-redundant processor comparator designs
  - Protect against random failures, but…
  - Common mode failures are still a matter of concern
  - Both failure types are addressed with this approach

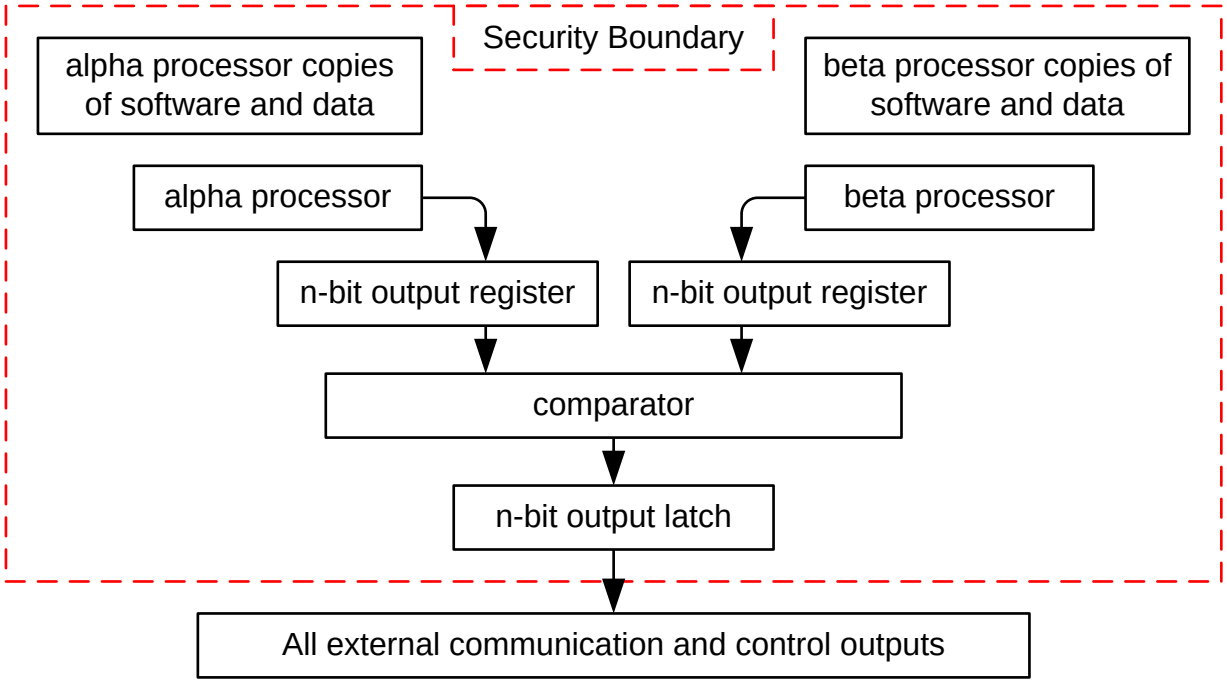
Access Control vs. Enablement

Sandia National Laboratories

- Access control requires authenticated requests
  - Requests prove source authentication via encryption and/or signature
  - Operations are performed or denied based upon authentication
  - Internal failures could lead to unauthenticated operations
- Enablement requires information from outside the system
  - Rather than executing a hard-coded algorithm, controlling an external system uses the comparator to make a series of blind data writes
  - Each processor has a unique, different copy of the output data
  - The output data is not stored plaintext – it can only be uncovered with a combination of external information and processor-unique data
- If the uncovered output data in each processor is not identical the comparator will alarm and the system will be locked

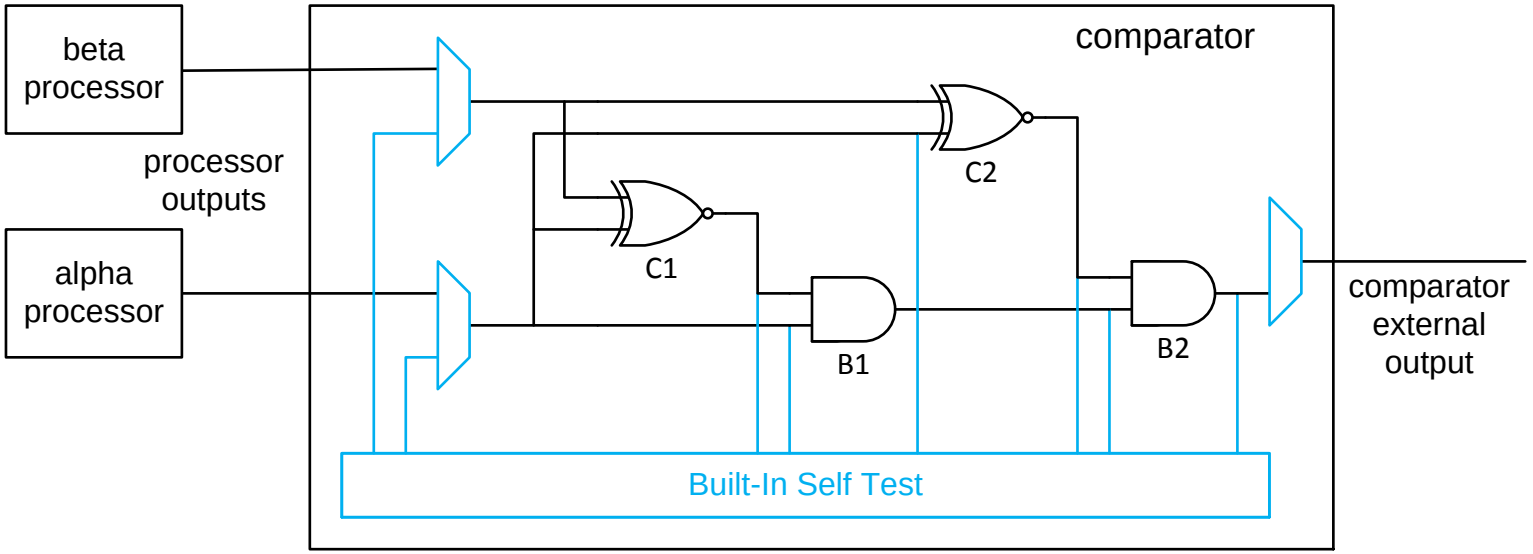
Dual-Redundant Processor Comparator Design Architecture

- Isolated processors and memories within a security boundary
- Comparator arbitration for all data/control leaving boundary
- System alarm and lock on error detection



Comparator Base Element Design

- Parallel redundant compare elements (C1 & C2)
- Serial redundant blocking elements (B1 & B2)
- Comparator hardware can be fully analyzed and tested
  - Fault Tree Analysis (FTA) and Built-In Self Test (BIST)
  - Software validation of hardware BIST result



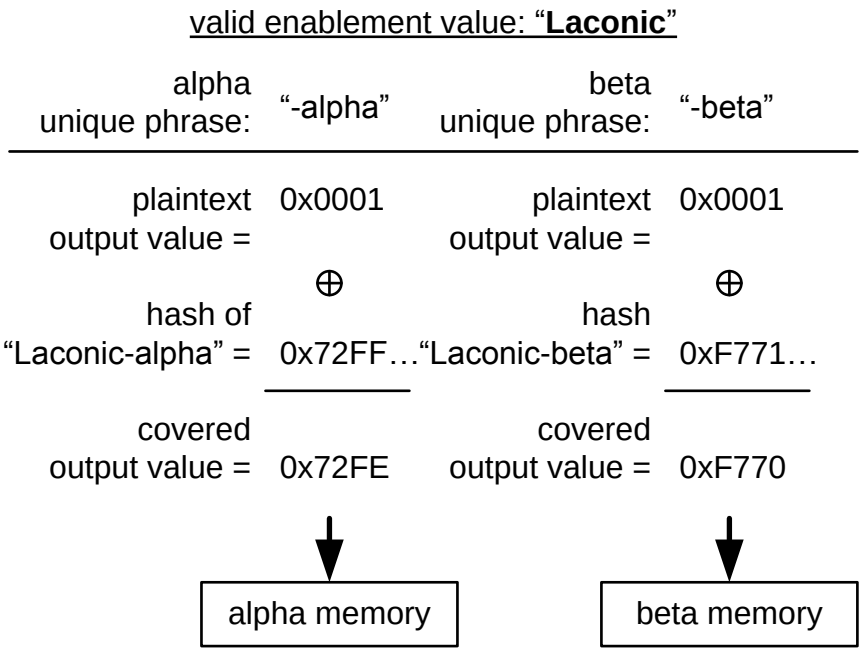
Covered Output Data Generation

Sandia National Laboratories

Simple example with a single 16-bit output value of 0x0001

(One output changed from 0 to 1)

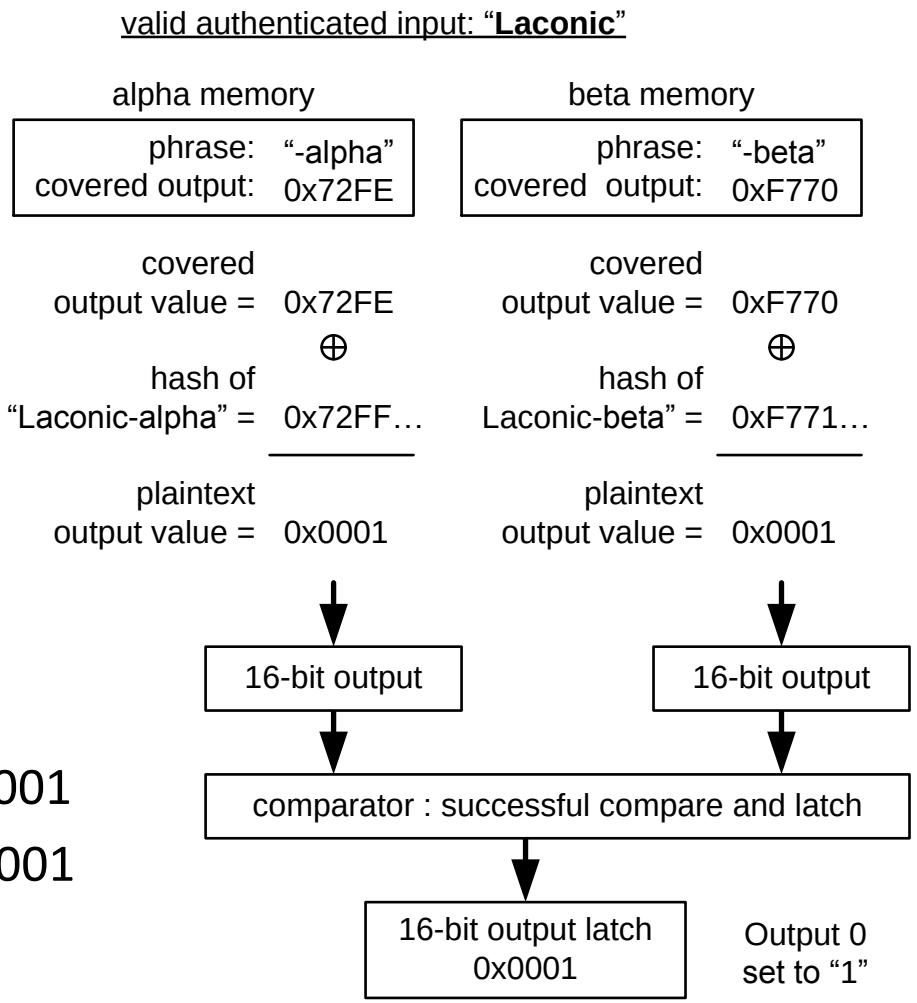
- Concatenate:
  - Enablement value
    - example – “Laconic”
  - Processor unique values
    - example – “-alpha” & “-beta”
- Hash (example – SHA256)
  - “Laconic-alpha” => 0x72FF…
  - “Laconic-beta” => 0xF771…
- XOR hash with output data
  - alpha pattern: 0x72FE
  - beta pattern: 0xF770



Unique covered data is stored in each processor’s memory

Authenticated Output Example

- Concatenate:
  - Authenticated input
    - "Laconic"
  - Processor unique values
    - "-alpha" & "-beta"
- Hash
  - "Laconic-alpha" => 0x72FF…
  - "Laconic-beta" => 0xF771…
- XOR hash with stored data
  - alpha: 0x72FF ⊕ 0x72FE = 0x0001
  - beta: 0xF771 ⊕ 0xF770 = 0x0001
- Write output data
  - Matching data => successful output



Failure Calculations

- If an unspecified fault causes the software to begin driving outputs without authenticated input, it will either:
  - Write the covered data values to the comparator, or
  - Write the covered data XORed with the hash of invalid data
- Fail-safe operation depends upon mismatch detection
  - For strong hash algorithms, such as SHA-256 in the example, the probability of two output bit positions having the same value for different input strings is like flipping coins – ½ of the time they match
- The probability that this fault will be detected and shut down by comparator hardware for 16 and 32-bit output data is:

$$1 - \left(\frac{1}{2}\right)^{16} = 0.99998 \qquad \textbf{16-bit data word}$$

$$1 - \left(\frac{1}{2}\right)^{32} = 0.9999999998 \qquad \textbf{32-bit data word}$$

Summary

Sandia National Laboratories

- Secure software-based systems can guarantee to a quantifiable level of confidence that outputs will not be driven upon software failures
  - Using dual-redundant processor comparator designs
  - The comparator is an analyzable and testable discriminator
  - Output data is uniquely stored/protected in each processor
  - External enablement data is required to create valid output
  - Outputs will not be driven with bad values (no thrashing)
- Desired levels of confidence can be achieved via selection of output drive complexity, sequence steps, and compared output data size