Modeling and Analysis of the Impact of Diversity in Digital Circuits on Attackers

Introduction

- Diversity in implementation can eliminate some vulnerabilities and make it uncertain whether a given implementation will have a particular vulnerability [1].
- The only general, quantifiable technique for eliminating a priori unknown vulnerabilities is to introduce design elements that cannot be anticipated by the attacker.
- We focus on voting amongst diverse implementations of the same function

\[ m = \text{A circuit} \]

\[ m_1, m_2, \ldots, m_n \]

• Diversity in implementation can eliminate some vulnerabilities and make it uncertain whether a given implementation will have a particular vulnerability [1].
• The only general, quantifiable technique for eliminating a priori unknown vulnerabilities is to introduce design elements that cannot be anticipated by the attacker.
• We focus on voting amongst diverse implementations of the same function

\[ m = \text{A circuit} \]

\[ m_1, m_2, \ldots, m_n \]

Routing Model

• Data is processed from source to destination through tiers of subcircuits. Several diverse implementations of each tier are available, and a subset of these is used to implement the circuit.
• A single route is realized by selecting one unit from each tier. The attacker succeeds if any unit in the path is subverted. We assume that the attacker can subvert only one node.
• We can also process along several parallel paths and vote on the output.

Composition Structures

• The system is decomposed into subcircuits, each of which must operate correctly for the system to operate correctly.
• Components are vulnerable to some fraction of the input space. We assume that the composition itself does not introduce vulnerabilities. Under this assumption the “composition of voters” architecture is more secure than the “voter of compositions”, but at the cost of more voters.

Analysis of Composition Structures

- Consider a circuit of \( m \) components that are, on average, vulnerable to a fraction \( v \) of the system inputs. Assuming the vulnerabilities are triggered by some \( k \) bit input, \( v=2^k \) where \( k \) is the key length.

\[ \text{Voter of Compositions Probability of Compromise} \]

\[ P(s) = \left( \frac{1}{n} \right)^m \left( 1 - \frac{1}{n} \right) \left( 1 - \frac{1}{n} \right) \left( 1 - \frac{1}{n} \right) \]

\[ \text{Probability of } q \text{ Successes} \]

\[ B(N,p) \]

Moving Target

In the moving target scenario, the attacker must be successful at least \( q \) times, and the system is re-randomized after each attacker try. If the attacker tries \( N \) inputs, each with probability of success \( p \), then the number of successes follows a binomial distribution \( B(N,p) \). Since \( p \) is small, we approximate this as a Poisson with parameter \( A=Np \). Then the attacker has at least \( q \) successes with probability 1 – \[ \sum_{i=q}^{\infty} \left( \frac{e^{-A} A^i}{i!} \right) \]

Future Work

We are currently implementing these approaches in hardware at different levels of coverage to understand the cost in area, operating frequency, and power required to achieve these reductions in attacker probability of success.