ECE 459/559
Secure & Trustworthy Computer Hardware Design

Metrics and Security Properties (Online)

Garrett S. Rose

Spring 2021

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Recap

- Hardware security primitives discussed:
  - Random Number Generators
    - True Random Number Generator (TRNG)
    - Pseudo Random Number Generator (PRNG)
  - Physical Unclonable Function (PUF)
    - Arbiter PUF (A-PUF)
    - Ring Oscillator PUF (RO-PUF)
  - Logic Locking Techniques
    - XOR-based key gate *insertion*
    - LUT-based logic *replacement*
    - Boosted Finite State Machine (BFSM)
- Security primitives do not offer security in and of themselves

need HW – sensitive to noise / radioactive decay

e.g. LFSR

359 x

459 x

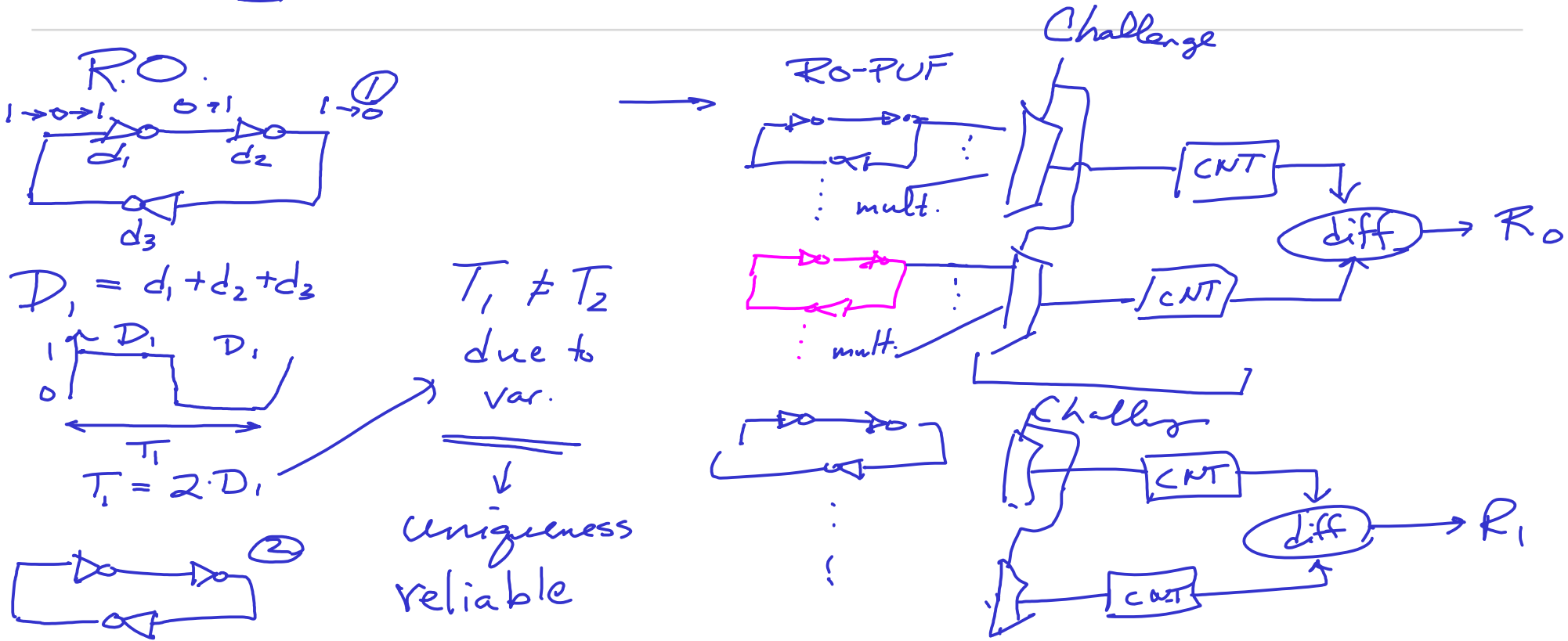
Recap

- Hardware security primitives discussed:
  - Random Number Generators
    - True Random Number Generator (TRNG)
    - Pseudo Random Number Generator (PRNG)
  - Physical Unclonable Function (PUF)
    - Arbiter PUF (A-PUF)
    - Ring Oscillator PUF (RO-PUF)
  - Logic Locking Techniques
    - XOR-based key gate *insertion*
    - LUT-based logic *replacement*
    - Boosted Finite State Machine (BFSM)
- Security primitives do not offer security in and of themselves

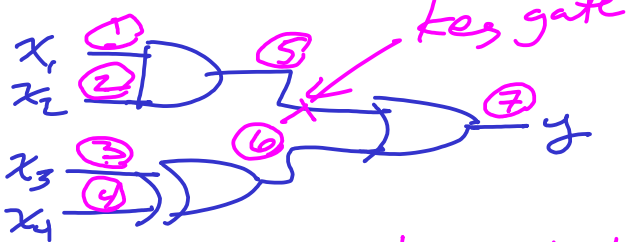
general security metric: time

- They provide properties that can be utilized at higher levels of abstraction
- Security only makes sense when consider user (and attacker) interactions

Review – Ring Oscillator PUF



Review – Rajendran Logic Encryption



1) label all nodes
2) det. testability ← where
for all nodes
3) rank nodes by testability
4) insert key gates at highest nodes

node	test.
5	0.6
3	0.5
4	0.4
5	0.3

testability
→ count the # test vec's that allow a test on that node for both faults
→ alt.: fault sim — HOPE

Other consideration:
# of gates ↔ Hamming Dist

Desired Property: Source of Entropy (Randomness)

- NIST has a suite of tests/metrics:
  "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications"
- TRNGs and PRNGs are needed for many cryptographic algorithms and even hardware security protocols
  - E.g. EPIC active metering technique employees a TRNG for secret key generation
- High-level algorithm must assume the TRNG or PRNG works & works well
- The NIST suite includes 15 tests to determine if a RNG is "random enough"

NIST Test Suite – 15 Tests

Metrics

- The Frequency (Monobit) Test
- Frequency Test within a Block
- The Runs Test
- Tests for Longest-Run-of-Ones in a Block
- Binary Matrix Rank Test
- Discrete Fourier Transform (Spectral) Test
- Non-overlapping Template Matching Test
- Overlapping Template Matching Test
- Maurer's "Universal Statistical" Test
- Linear Complexity Test
- The Serial Test
- Approximate Entropy Test
- Cumulative Sums (Cusums) Test
- The Random Excursions Test
- The Random Excursions Variant Test

Desired Properties: Uniqueness and Reliability

- Recall: PUFs are generally measured by two key security properties:
  - Uniqueness – variability in *space* (differences chip-to-chip)
  - Reliability – variability in *time* (differences in response on one chip)
- These properties are measured by two common metrics:
  - Inter-chip distance (aka. inter-distance) for uniqueness – Ideal: 50 %
  - Intra-chip distance (aka. Intra-distance) for reliability – Ideal: 0 %
- Both are based on Hamming Distance (HD) comparisons between different challenge-response-pairs (CRPs)

Hamming Distance as a Metric

Maiti et al

- Note: this is not a full blown security measure... it helps assess the "goodness" of security primitives
- Basic definition (for inter-distance):

uniqueness

assess over mult. chips → det. average

$$Inter\ HD = \frac{1}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n}$$

$HD(R_i, R_j)$ – Number of bits different between response *i* and response *j*

n = number of bits in response *R*

k = number of chips considered

i and *j* represent two different chips

R_i = response on chip *i* for challenge *C*

R_j = response on chip *j* for challenge *C*

Other Useful Metrics for PUFs

- Bit-Aliasing
  - Measure of bits within a response that may be biased to 0 or 1
- Uniformity
  - Basically, % of bits that are 1 in average response
  - Desire 50 % – on average, we'd like same numbers of 1's and 0's
- Paper with some useful description of PUF metrics:
  "Method to Evaluate Performance of PUFs" – Maiti (on Canvas)

Desired Properties: Output Sensitivity and Output Corruption

Confusion of outputs

- For logic locking, we considered two metrics that can help with design:
  - *Testability* – determine where to place key gates (assuming gate insertion)
  - *Hamming Distance* – determine how many key gates to place

design parameters

- Testability can be thought of as a measure of how sensitive the outputs are to a change in a particular node's logic value
  - Testability is the metric
  - Security property could be thought of as output sensitivity
- Hamming distance is a measure of the average corruption of output bits when an incorrect key is supplied
  - Hamming distance is the metric
  - Security property could be thought of as output corruption

Think about:
- diffusion
- avalanche effect
- vanigueness

Takeaways

- High level cryptographic algorithms and security protocols must be able to *assume* the security primitives are strong, and accomplish the task
  - A TRNG should be assumed to be "random" ← need to show it — use NIST
- However, when these primitives are hardware security primitives, we must design them such that they perform as needed
- Good metrics are important in designing solid hardware security primitives
- With good metrics, we can confidently say the primitive provides the needed security properties for higher level security protocols