Hardware-Oriented Security and Trust
ECE 4156 HST / ECE 6156 HST
Spring 2024

Assoc. Prof. Vincent John Mooney III
Georgia Institute of Technology

Homework 8, 75 pts. (ECE 4156) 80 pts. (ECE 6156)
Due Friday April 5 prior to 11:55pm

1) (15 pts.) For each of the following PUFs, use a few (e.g., between one and three) sentences to describe the source of entropy (randomness). Make sure to discuss both the physical as well as the functional aspects. (Note1: by physical what is meant are sources based in physical properties of materials and associated structures.) (Note2: by functional what is meant are sources due to logic design, e.g., Exclusive-OR logic functionality.)

   a. (5 pts.) Arbiter PUF

Solution

Physical Source of Randomness: For the Arbiter PUF, the physical sources of randomness are the transistor and wire delays in the switches – but the transistor delays dominate.

Functional Source of Randomness: The functional control over these physical sources are the challenge bits selecting which paths the signals will follow through the switches.

   b. (5 pts.) Ring Oscillator PUF

Solution

Physical Source of Randomness: The main physical entropy source in an RO PUF is the inverter. In the dominant form of chip technology – CMOS – an inverter is designed using a p-type transistor (a MOSFET, a FINFET or an all-around FET) and an n-type transistor. The manufacturing variations in transistor fabrication are the dominant source of randomness.

Functional Source of Randomness: The functional control over which ROs are selected for comparison through selection (challenge) bits fed to multiplexers results in a particular random bit through comparison of two RO delays.

   c. (5 pts.) Metal Resistance PUF

Solution

Physical Source of Randomness: The Metal Resistance PUF uses three types of conducting materials as a source of entropy: metal (typically Aluminum), polysilicon (amorphous silicon), and vias (typically Aluminum or Tungsten). In addition, an n-type transistor (MOSFET) in the pull-down path adds to the entropy.

Functional Source of Randomness: The main functional control is to choose which specific metal resistance network to select; once chosen, current flows across the metal delay elements through the "shorting" pull-down n-type transistor results in a specific voltage value which is measured. In particular, two voltages values are converted to digital numbers and then compared, with the comparison result expressed in a single bit value.

2) (25 pts.) Consider the bit sequence 0 1 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 0 0 and apply the following tests. For each test, please show your work regarding how you calculated the numerical result; provision of only a numerical answer without showing the steps and calculations used will result in zero credit.

a. (5 pts.) Entropy

Solution

We know that the current bit sequence has 9 0's and 11 1's.

P (1) = 11/20 = 0.55, P (0) = 1- P (1) = 9/20 = 0.45

Entropy can be calculated using Shannon's Entropy Formula:

$H(x) = -\Sigma^{n}_{i=1} [P(x_i) * \log_b P(x_i)] = \Sigma^{n}_{i=1} [P(x_i) * \log_b(1 / P(x_i))]$

$= -(0.55 \times \log_2(0.55) + 0.45 \times \log_2(0.45))$

= 0.9927

b. (5 pts.) MinEntropy

Solution

P(1)=0.55, P(0)=0.45

Minimum Entropy can be calculated using the following:

$H(x) = \log_2 (\max P(x))$

$= -\log_2(0.55)$

= 0.8625

c. (15 pts.) Conditional MinEntropy using pairs of two bits:
0 1 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 0 0

Solution

There are four combinations of Bits.

P(00) = 0.1

P(01) = 0.5

P(10) = 0.2

P(11) = 0.2

Probability of a second bit being 1 is P(01) + P(11) = 0.7

The probability of a second bit being 0 is P(00) + P(10) = 0.3

$H(x) = -\log2 (\max (P(x)/P(y))$

$= -\log2(\max(P(00)/0.3, P(01)/0.7, P(10)/0.3, P(11)/0.7)$

$= -\log2(0.5/0.7)$

= 0.4854

3) (5 pts.) What defines an intrinsic PUF?

Solution

A PUF construction needs to meet at least two conditions to be called an intrinsic PUF:

i. Its evaluations are performed internally by embedded measurement equipment.

ii. Its random instance-specific features are implicitly introduced during its production process.

4) (10 pts.) Use your own words (do **not** copy from the lecture notes or any other source!) to answer the following question: what is the difference between a weak PUF and a strong PUF?

Solution

A weak PUF has insufficient challenge space to withstand a brute force attack from an adversary with prolonged access to the PUF. For example, a PUF with 23-bit challenges in the hands of an adversary for a week (due to a step in the supply chain) could potentially have all possible challenges (the total is 2^{23}) exercised with the responses stored in a database. Another example of a weak PUF is one that is model buildable where with prolonged access enough challenges can be applied to reliably predict all PUF responses even if the challenge space is large, e.g., 256 bits.

A strong PUF, on the other hand, has a large enough challenge space to be able to say with certainty that an adversary can only learn a negligible portion of the challenge-response space even with prolonged PUF access. For example, a PUF with 128-bit challenges and which cannot be machine learned would be strong. If the 56-bit key of DES is taken as a maximum number of challenges an adversary can try (i.e., 2^{56} challenges), then for the 128-bit PUF we find that $2^{56} / 2^{128} = 1 / 2^{72} = 2^{-72}$ which is a negligible chance of the adversary correctly guessing which challenges will be used.

5) (10 pts.) Why are PUFs used for encryption not threatened by model building?

Solution

A PUF used for encryption never has its response revealed to the adversary. Without access to the response bits, the adversary has no known way to model build other than breaking the encryption key (for which the adversary has a negligible chance).

6) (10 pts.) Consider the tamper-evidence property of a PUF.

a. (5 pts.) What is the tamper-evidence property? Please describe the tamper-evidence property of a PUF using your own words.

Solution

An adversary with physical access to the chip on which the PUF is implemented may be able to remove the packaging – or at least open up the top (of course this depends on the package type) – and proceed to use a probe to touch the I/O pads on the chip or even the top layers of metal. Additional reverse engineering techniques exist where the highest levels of the chip can be delayered with appropriate etching chemicals thus revealing lower metal layers and even, eventually, the transistors.

However, all of the aforementioned reverse engineering techniques alter at least in small ways the physical properties of the chip and the PUF including, for example, resistance and capacitance values. As a result, the PUF measurements will certainly be changed, thus altering the PUF responses to challenges and preventing the attacker from learning the original PUF responses. Therefore, a Silicon PUF is tamper-evident in that tampering with the physical instantiation of the PUF alters its behavior and thus makes it evident that tampering has occurred.

b. (5 pts.) What does the tamper-evidence property of PUFs protect against?

Solution

A PUF 's tamper-evidence makes the chances that an adversary can use physical attacks to correctly learn the PUF 's responses negligible. In other words, you can say with certainty than an adversary employing the use of metal probes and other reverse engineering techniques (as are common in failure-mode analysis in chip fabrication facilities) will not succeed in learning anything useful about the actual challenge-response behavior of the PUF.

7) [ECE 6156 only!] (5 pts.) Explain why a compressed file has more entropy than the same file uncompressed.

Solution

This question must be interpreted as **entropy per bit** (as opposed to entropy per file). In terms of entropy per bit, a compressed file uses **fewer bits to represent the same information** (e.g., a sequence of ASCII characters) as the uncompressed file. To carry out compression, many techniques are employed such as replacing common character sequences in the source file with much smaller bit patterns in the compressed file, with the result that a much more complete range of bit values occur with many more bit-level variations. In summary, then, the compressed file uses the **full range** (e.g., for eight bits there are 256 possible combinations of bit values) of bit combinations **with more frequent variations** which has the effect of i**ncreasing the entropy** (randomness) per bit. NIST tests applied to compressed files shows this increase in entropy.