ECE 3170 Cryptographic Hardware for Embedded Systems

Fall 2025

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

Homework 10, 105 pts.

Due Monday Nov. 3 prior to 11:55pm

(please turn in homework electronically on Canvas)

As always this semester, you are required to solve this homework alone.

1) (5 pts.) Read the paper by L. Goubin and J. Patarin, "DES and Differential Power Analysis: The 'Duplication' Method," Cryptographic Hardware for Embedded Systems (CHES) conference, 1999. Please indicate that you have read the article by writing an appropriate sentence, e.g., "I hereby certify under the Georgia Tech Honor Code that I have read the CHES 1999 article assigned in this homework for problem 1."

2) (10 pts.) One approach to masking is to replace a multi-bit variable V (e.g., V could have 8 bits) with k variables $V_1,\ldots,V_k$. Condition 1 as described in lecture discusses V_i, $1 \le i \le k$. Why does Condition 1 stress that V_i should have no discernable relationship to V?

If V_i has no discernable relationship to V, then in effect one removes the attacker's ability to choose a plaintext value V and know how a subset of V is used in a cryptographic calculation at a specific point in time and for which the power side channel is being measured.

3) (15 pts.) Consider DES. In this question we are going to focus on the S-box.

Table 12.6
S-Boxes

S-box 1:

14,	4,	13,	1,	2,	15,	11,	8,	3,	10,	6,	12,	5,	9,	0,	7,
0,	15,	7,	4,	14,	2,	13,	1,	10,	6,	12,	11,	9,	5,	3,	8,
4,	1,	14,	8,	13,	6,	2,	11,	15,	12,	9,	7,	3,	10,	5,	0,
15,	12,	8,	2,	4,	9,	1,	7,	5,	11,	3,	14,	10,	0,	6,	13,

S Box																
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	14	8	12	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

a. (5 pts.) For S-box 1 (shown above in two different formats) calculate the following quantities: S(0x13), S(0x31) and S(0x13 ⊕ 0x31). Give your answers in hexadecimal format.

S(0x13) = S(0b010011) = row 01, column 1001 = 0x6 = 0110

S(0x31) = S(0b110001) = row 11, column 1000 = 0x5 = 0101

S(0x31 ⊕ 0x13) = S(0b100010) = 0x1 = 0001

b. (5 pts.) Give a proper definition of linearity of a function.

A function f() is linear if both of the following properties hold:

1. $f(x+y) = f(x) + f(y)$ where both x and y are proper inputs to f().

2. $f(ax) = a*f(x)$ where x is a proper input to f() and a is a scalar.

Note that the above defines linearity in terms of addition and multiplication. We can also substitute other logic functions. For example, we can also say the following must hold for f() to be linear in Boolean logic:

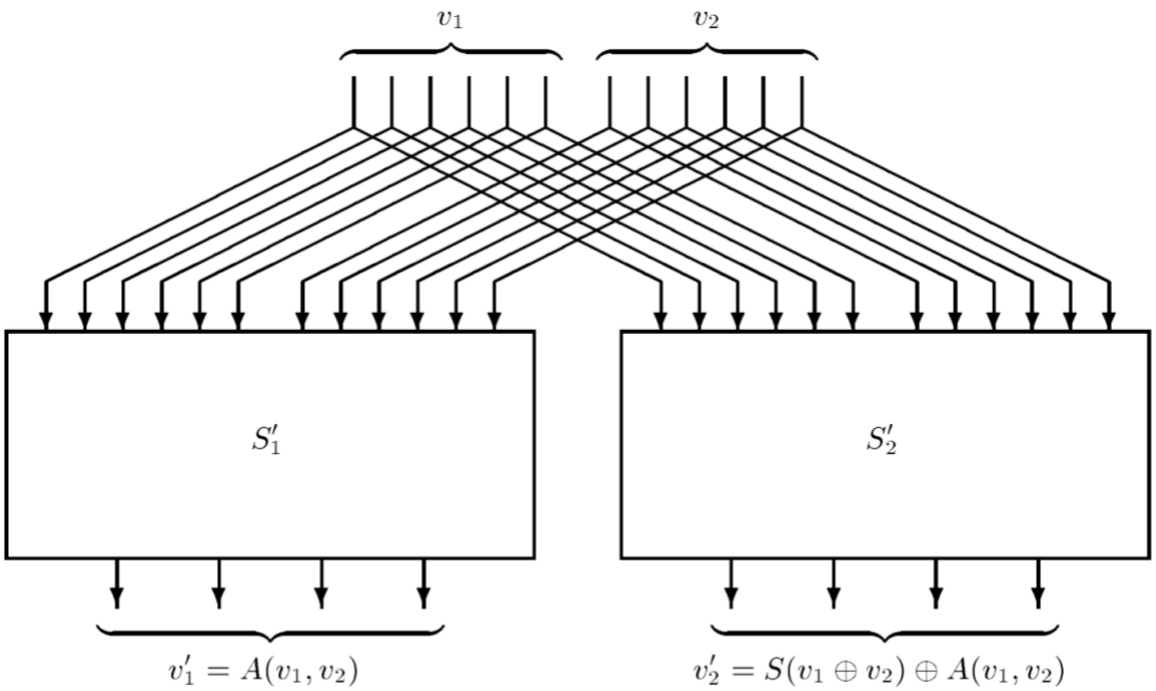
$f(x \oplus y) = f(x) \oplus f(y)$

c. (5 pts.) Is S-box 1 a linear function? If so, why? If not, why not?

S-Box 1 is not linear. $S(0x31 \oplus 0x13) = S(0b10010) = 0x1 = 0001$

$S(0x31) \oplus S(0x13) = 0x5 \oplus 0x6 = 0101 \oplus 0110 = 0011 = 0x3$

$S(0x31 \oplus 0x13) \neq S(0x31) \oplus S(0x13)$



Modified implementation: the values $v = v_1 \oplus v_2$ and $v' = v'_1 \oplus v'_2$ never explicitely appear in RAM

4) (20 pts.) Consider S'_2 in the figure above. Recall from lecture that S'_2 is used in a masked implementation of DES.

a. (5 pts.) Suppose S'_2 above is implemented in a memory look-up table. How many kilobits (Kb) of memory are required to implement the look-up table for S'_2? Please give your answer kilobits (Kb). For example, 64 KB (kilobytes) is equal to 64*8 Kb = 512 Kb = 512*1024 bits = 524,288 bits. A correct answer in KB or any units other than Kb (= 2^{10} bits) will receive zero points. (NOTE: an alternative definition for 2^{10} bits is Kibibits or Kib.)

S'_2 requires $2^{12} * 2^2$ bits $= 2^{14}$ bits $= 16{,}384$ bits $= 2^4 * 2^{10}$ bits
2^4Kb $= 16$Kb

b. (5 pts.) You are asked to give your estimate of the time it will take to calculate (i.e., look up) S_1' and S_2' in parallel in the masked DES implementation as compared to a standard S-Box calculation (look-up) in the standard (non-masked) DES implementation in an Application-Specific Integrated Circuit (ASIC)? Is the time needed (i) approximately the same, (ii) between 50% and 2X larger or (iii) greater than 2X longer? Please explain the reason(s) for your answer clearly and unambiguously. You may assume a standard SRAM design for the look-up table implementation in the ASIC. Please assume that the ASIC clock is as fast as possible.

Exact SRAM delay is used for an ASIC!
The standard S-Box in DES has a 6-bit input and a 4-bit output, hence $2^6 * 2^2 = 2^8 = 256$ bits are needed.
However, S_1' and S_2' each require $2^{12} * 2^2 = 2^{14} = 16{,}384$ bits. Since the number of bits which need to be stored and looked up greatly exceed 2X – in fact, the increase is greater than fifty times or 50X – then the time needed to calculate (i.e., look up) S_1' and S_2' in a standard SRAM design is likely to (iii) exceed 2X.
Another way to argue this is that the 12 address bits versus 6 require approximately double the logic delay to process, plus the SRAM array itself will take longer to access, resulting in a greater than 2X increase in delay.

c. (5 pts.) In order to calculate S_2' off-line (i.e., prior to implementing S_2' in the masked DES implementation), is it necessary to calculate $v = v_1 \oplus v_2$? Why or why not? Please explain the reason(s) for your answer clearly and unambiguously.

Yes, it is necessary to calculate $v = v_1 \oplus v_2$ off-line to calculate S_2' off-line. This is because the S-box output $S(v_1 \oplus v_2)$ needs to be looked up and then exclusive-ored with $A(v_1 \oplus v_2)$ to calculate S_2'. If $v = v_1 \oplus v_2$ is not calculated, then the look-up of $S(v_1 \oplus v_2)$ cannot be carried out which means that S_2' cannot be calculated.

d. (5 pts.) In order to calculate A, S_1' or S_2' off-line (i.e., prior to implementing the masked DES implementation), is it necessary to calculate $v' = v_1' \oplus v_2'$? Why or why not? Please explain the reason(s) for your answer clearly and unambiguously.

It is not necessary to calculate $v' = v_1' \oplus v_2'$ to calculate A, S_1' or S_2' off-line. This is because v' is not an input to any of A, S_1' or S_2'.

5) (5 pts.) Read the paper by T. Popp and S. Mangard, "Masked Dual-Rail Pre-charge Logic: Differential Power Analysis Resistance Without Routing Constraints," Cryptographic Hardware for Embedded Systems (CHES) conference, 2005. Please indicate that you have read the article by writing an appropriate sentence, e.g., "I hereby certify under the Georgia Tech Honor Code that I have read the CHES 2005 article assigned in this homework for problem 5."

Table 2. Transitions of the value d_m of a masked node

Line no.	d_{t-1}	m_{t-1}	$d_{m_{t-1}}$	d_t	m_t	d_{m_t}	Energy	Probability
1	0	0	0	0	0	0	E_{00}	$\frac{1}{4}p_{00}$
2	0	0	0	1	1	0	E_{00}	$\frac{1}{4}p_{01}$
3	1	1	0	0	0	0	E_{00}	$\frac{1}{4}p_{10}$
4	1	1	0	1	1	0	E_{00}	$\frac{1}{4}p_{11}$
5	0	0	0	0	1	1	E_{01}	$\frac{1}{4}p_{00}$
6	0	0	0	1	0	1	E_{01}	$\frac{1}{4}p_{01}$
7	1	1	0	0	1	1	E_{01}	$\frac{1}{4}p_{10}$
8	1	1	0	1	0	1	E_{01}	$\frac{1}{4}p_{11}$
9	0	1	1	0	0	0	E_{10}	$\frac{1}{4}p_{00}$
10	0	1	1	1	1	0	E_{10}	$\frac{1}{4}p_{01}$
11	1	0	1	0	0	0	E_{10}	$\frac{1}{4}p_{10}$
12	1	0	1	1	1	0	E_{10}	$\frac{1}{4}p_{11}$
13	0	1	1	0	1	1	E_{11}	$\frac{1}{4}p_{00}$
14	0	1	1	1	0	1	E_{11}	$\frac{1}{4}p_{01}$
15	1	0	1	0	1	1	E_{11}	$\frac{1}{4}p_{10}$
16	1	0	1	1	0	1	E_{11}	$\frac{1}{4}p_{11}$

6) (10) Consider Table 2 shown above. Why does the first row have probability $\frac{1}{4}p_{00}$?

The first row has a probability of $\frac{1}{4}p_{00}$ because given that the mask bit m is uniformly distributed, there is an equal probability that each of the four outcomes where $d_{t-1} = 0$ and $d_t = 0$ can occur; as a result, p_{00} is simply divided by the number of equally likely outcomes.

Table 3. Truth table of an MDPL AND gate

Line no.	a_m	b_m	m	q_m	$\overline{a_m}$	$\overline{b_m}$	$\overline{m}$	$\overline{q_m}$
1	0	0	0	0	1	1	1	1
2	0	0	1	0	1	1	0	1
3	0	1	0	0	1	0	1	1
4	0	1	1	1	1	0	0	0
5	1	0	0	0	0	1	1	1
6	1	0	1	1	0	1	0	0
7	1	1	0	1	0	0	1	0
8	1	1	1	1	0	0	0	0

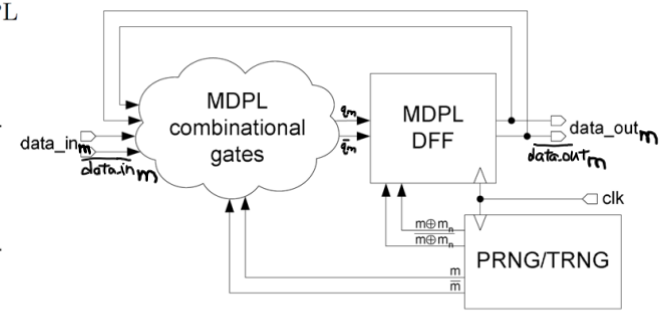


Fig. 5. Architecture of an MDPL circuit

7) (20 pts.) Suppose that "MDPL combinational gates" shown above in Fig. 5 consists of a single AND gate implemented in MDPL. We are going to consider specific cases. Note that $data\_in_m$ and $\overline{data\_in_m}$ come from an MDPL flip-flop not shown where the MDPL flip-flop not shown has inputs $data\_in$, $\overline{data\_in}$, mask bit m and complemented mask bit $\overline{m}$.

   a. (5 pts.) Suppose $data\_in = 0$, $\overline{data\_in} = 1$, $data\_out = 0$, $\overline{data\_out} = 1$, mask bit $m = 1$ and complemented mask bit $\overline{m} = 0$. Given these input values, what are the values of the "MDPL combinational gates" outputs q_m and $\overline{q_m}$?

Note: Many students forgot to XOR the AND result with 'm'

$$q_m = \big((data_{in} \oplus m)\ \&\ (data_{out} \oplus m)\big) \oplus m = \big((0 \oplus 1)\ \&\ (0 \oplus 1)\big) \oplus 1 = 0$$
$$\overline{q_m} = \Big(\big(\overline{data_{in}} \oplus \overline{m}\big)\ \&\ \big(\overline{data_{out}} \oplus \overline{m}\big)\Big) \oplus \overline{m} = \big((1 \oplus 0)\ \&\ (1 \oplus 0)\big) \oplus 0 = 1$$

   b. (5 pts.) Recalling that $q_m = q \oplus m$, what are the values of q and $\bar{q}$?

$$q_m = q \oplus m \rightarrow 0 = q \oplus 1 \rightarrow q = 1$$

Since $q = 1$, then $\bar{q} = 0$.

   c. (5 pts.) Now instead suppose $data\_in = 1$, $\overline{data\_in} = 0$, $data\_out = 1$, $\overline{data\_out} = 0$, mask bit $m = 0$ and complemented mask bit $\overline{m} = 1$. Given these input values, what are the values of the "MDPL combinational gates" outputs q_m and $\overline{q_m}$?

$$q_m = \big((data_{in} \oplus m)\ \&\ (data_{out} \oplus m)\big) \oplus m = \big((1 \oplus 0)\ \&\ (1 \oplus 0)\big) \oplus 0 = 1$$
$$\overline{q_m} = \Big(\big(\overline{data_{in}} \oplus \overline{m}\big)\ \&\ \big(\overline{data_{out}} \oplus \overline{m}\big)\Big) \oplus \overline{m} = \big((0 \oplus 1)\ \&\ (0 \oplus 1)\big) \oplus 1 = 0$$

   d. (5 pts.) Given these newly calculated values (see part c above), and recalling that $q_m = q \oplus m$, what now are the values of q and $\bar{q}$?

$$q_m = q \oplus m \rightarrow 1 = q \oplus 0 \rightarrow q = 1$$

Since $q = 1$, then $\bar{q} = 0$.

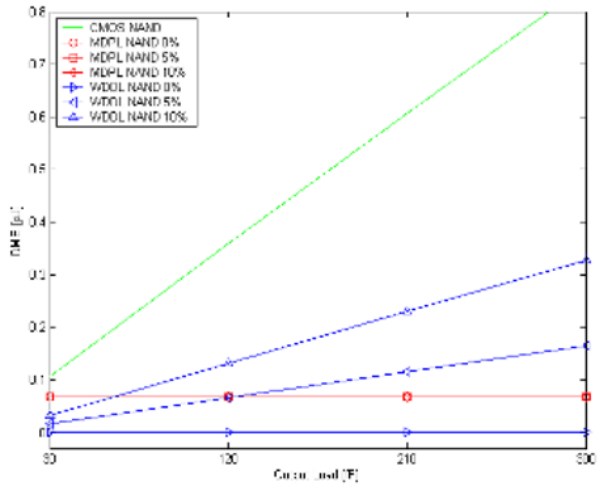


Fig. 6. Comparison of the DME of CMOS, WDDL and MDPL implementations NAND gate

8) (20 pts.) Consider Figure 6 above.

a. (5 pts.) What does the green line represent? Please explain the reason for its slope.

The green line displays the difference-of-mean-energies (DME) as load capacitance increases for a standard CMOS NAND gate. The slope indicates a large linear relation between energy consumption and load.

b. (5 pts.) The top blue line is labeled "WDDL NAND 10%." Why is the slope not flat?

The top blue line does not have a flat slope due to the 10% difference in capacitance of q and $\bar{q}$. As the load increases the 10% difference creates a larger magnitude DME.

c. (5 pts.) The bottom blue line is labeled "WDDL NAND 0%" and is flat. Is this line realistic? Why or why not? Please explain the reason(s) for your answer clearly and unambiguously.

WDDL NAND 0% indicates matched capacitances for q and $\bar{q}$. It is not realistic as real devices will be highly unlikely to exhibit exactly matched capacitances due to manufacturing variations.

d. (5 pts.) Why are all three red lines on top of each other? What does the slope tell you about MDPL?

The three lines are on top of each other due to MDPL providing identical DME regardless of load. This tells us that MDPL does not leak data via DME as the load increases.