CMOS Structures Suitable for Secured Hardware

Sylvain Guilley, Philippe Hoogvorst, Yves Mathieu, Renaud Pacalet and Jean Provost
GET / Télécom Paris, CNRS LTCI,
Département communication et électronique
46 rue Barrault, 75634 Paris Cedex 13, France.
{sylvain.guilley, philippe.hoogvorst, yves.mathieu, renaud.pacalet, jean.provost}@enst.fr

Abstract

Unsecured electronic circuits leak physical syndromes correlated to the data they handle. Side-channels attacks, like SPA or DPA, exploit this information leakage. We provide balanced and memoryless CMOS structures for a 2-input secured NAND gate.

1. Introduction

SPA or DPA [1] exploit physical information leaked by a cryptographic device to retrieve its secret key.

The counter-measures reported so far against such attacks consist in lowering the correlation between the data and the information leakage. Techniques usually resort to a data randomization, like cleartext or key *blinding*. To our knowledge, the only solution addressing the problem of reducing the information leakage by a dedicated microelectronic design is asynchronous logic with a dual rail datapath encoding [2, 5] and RTZ (Return to Zero) protocol.

The rest of this article is organized as follows. A NAND gate that does not leak information is specified and an adequate micro-architecture is proposed. Regular and secured NAND performances are evaluated and compared.

2. Secured NAND Gate Specifications

The difference between the rising and the falling transitions symptoms ($P_\uparrow \neq P_\downarrow$) is the main phenomenon making side-channel attacks possible [5]. As a result, an electronic gate can only be claimed to be secured against side-channels attacks provided every logical change of its output corresponds to the same type (either rising or falling) of transitions. One solution consists in encoding every data A on two wires, a_0 and a_1, and to start every computation with the condition $a_0 = a_1$ satisfied. We consider in the rest of this article the RTZ protocol shown in (1).

$$\begin{array}{l} \texttt{NULL} \rightarrow \texttt{VALID} \rightarrow \texttt{NULL} \rightarrow \texttt{VALID} \rightarrow \texttt{NULL} \rightarrow \cdots \\ \hline (0,0) \nearrow (0,1) \searrow (0,0) \nearrow (0,1) \searrow (0,0) \nearrow \cdots \\ \quad\;\; \searrow (1,0) \nearrow \quad\;\; \searrow (1,0) \nearrow \quad\;\; \searrow \cdots \end{array} \quad (1)$$

Moreover, the NAND gate must meet the following three conditions:

1. The electrical network seen from input a_0 must be identical to the one seen from a_1. If it was not the case, an event on a_0 would probably consume a different amount of energy than an event on a_1.
2. The gate shall not produce anticipated outputs. If the gate evaluates before all the inputs are in the same state, either all VALID or all NULL, the computation duration depends on the input data.
3. The gate must be memoryless. Every CMOS gate (but the INV) contains internal nodes that are not always driven. Undriven nodes store a state in their capacitance; the memorizing state is usually referred to as *high-impedance* or *Z* state. The memorization of an electrical charge from one computation to another leads to an observable conditional discharge. This discharge is correlated to the gate input data sequence and is for this reason exploited by side-channel attacks.

3. Secured NAND Gate Micro-Architecture

The first condition is straightforward on balanced truth tables gates, as for instance INV, XOR, HA, FA, MUX21. As shown later in Fig. 1a, dummy loads can be used to balance the NAND electrical network.

The second condition can be implemented by a *rendez-vous* of the inputs, so as to make sure the gate processes the inputs (either VALID or NULL) only when they have all arrived. The inputs are thus first decoded by C-Element gates [4]. For instance, a 2-input (a_0, a_1), (b_0, b_1) gate performs the rendez-vous between every couple (a_i, b_j). Only one of these rendez-vous produces an event. As soon as

1530-1591/04 $20.00 (c) 2004 IEEE

this event occurs, the gate can evaluate. Fig. 1a (inspired from [2]) shows a structure for a NAND gate that satisfies the first two conditions. The 3OR gate on the path to y_1 is dummy: its only purpose is to balance the y_0 and y_1 ways[1].

The third condition can be fulfilled by adding circuitry to remove any undriven nodes.

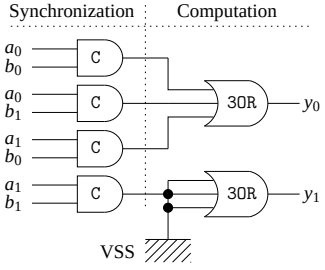

Figure 1a. A secured NAND gate schematic.

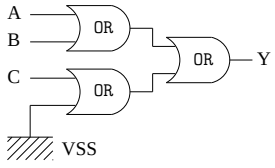

Figure 1b. 3OR whose inputs are equivalent.

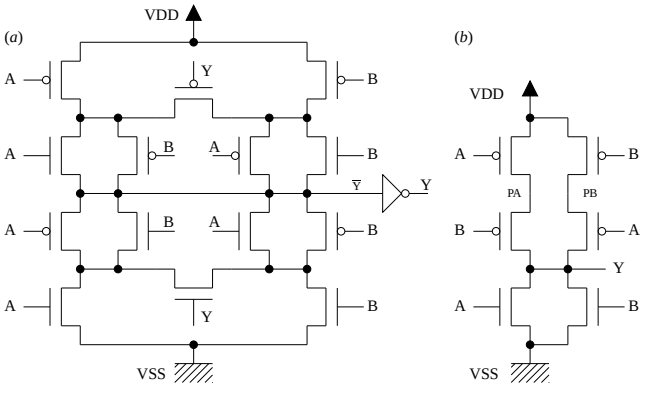

Figure 2. (*a*) Balanced C-Element without internal Z states and (*b*) balanced NOR, without undriven nodes provided A $\neq 1$ and B $\neq 1$.

The 4 C-Elements of Fig. 1a can be implemented by *symmetrical* C-Element [3] modified to be memory-less (Fig. 2 (*a*)). A NOR gate[2] suitable for the implementation of an equilibrated 3OR gate (as depicted in Fig. 1b) is shown in Fig. 2 (*b*). It is symmetric with respect to the input exchange A $\leftrightarrow$ B. But it has internal Z states: PA = PB = Z, when A = B = 1. However, this situation never happens because the dual-rail RTZ protocol (1) is used.

4. Performances

Spice simulations of the power consumption of the secured dual-rail RTZ NAND gate are shown in Fig. 3 (a). There is a slight difference between the traces corresponding to two computations that lead to a different output (y_0 or y_1), as shown in Fig. 3 (b). The difference, null in average, arises from the fact that an internal node shortened to ground by a wire is not equivalent to a node connected to ground by a closed transistor (cf. dummy 3OR of Fig. 1a).

Tab. 1 summarizes the performances of the regular and secured NAND gate.

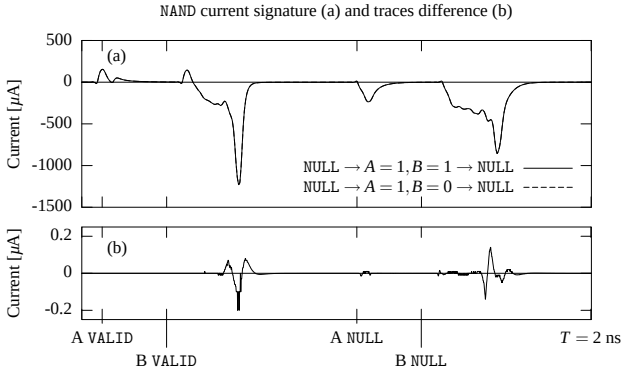

Figure 3. Secured NAND current signature.

Table 1. Standard and secured NAND gate performances.

Performance	Standard gate	Secured gate
Surface	4 transistors	112 transistors
Prop. $0\rightarrow1$ / $1\rightarrow0$	0.03 ns / 0.02 ns	$2\times\{0.19$ ns / 0.19 ns$\}$
$10\log(P_\uparrow/P_\downarrow)$	23.2 dB	~ 0 dB

5. Conclusion

A secured NAND gate is shown to leak little information about the data manipulated. A structure in transistors, balanced and memory-less, is proposed. Performances analysis show that removing side-channels in a gate strongly impacts its area and propagation time.

References

[1] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis: Leaking secrets. *Proc. of CRYPTO'99*, 1666:388–397, 1999.

[2] S. Moore, R. Anderson, P. Cunningham, R. Mullins, and G. Taylor. Improving smart card security using self-timed circuits. *Proc. of Async'02*, pages 211–217, 2002.

[3] M. Shams, J. Ebergen, and M. Elmasry. Modeling and comparing CMOS implementations of the C-element. *IEEE Transactions on VLSI Systems*, 6(4):563–567, 1998.

[4] I. E. Sutherland. Micropipelines. *Communications of the ACM (Turing award)*, 32(6):720–738, 1989.

[5] Z. Yu, S. Furber, and L. Plana. An investigation into the security of self-timed circuits. *Proc. of Async'03*, 2003.

[1] More sophisticated architectures where the additional material detects errors (in the context of fault attacks) have been proposed [2].

[2] A CMOS OR gate is typically build with a NOR followed by an INV.