Uniformly-Switching Logic for Cryptographic Hardware

Igor L. Markov, Dmitri Maslov

Uniformly-switching Logic for Cryptographic Hardware

Igor L. Markov
Univ. of Michigan, Department of EECS
imarkov@eecs.umich.edu

Dmitri Maslov
Univ. of Victoria, Department of CS
dmaslov@uvic.ca

Abstract

Recent work on Differential Power Analysis shows that even mathematically-secure cryptographic protocols may be vulnerable at the physical implementation level. By measuring energy consumed by a working digital circuit, one can glean enough information to break encryption. Thwarting such attacks requires a new approach to logic and physical design. In this work, we seek to equalize switching activity of a circuit over all possible inputs and input transitions by adding redundant gates and increasing the overall number of signal transitions. We introduce uniformly-switching (U-S) logic, and present a doubling construction that equalizes power dissipation without requiring drastic changes in CAD tools.

1 Introduction

Demands for tighter security are rapidly growing at banks and ATMs, airports, industrial sites, military installations, large entertainment complexes and power stations. These demands are often addressed using electronic cryptographic hardware. However, such hardware may not be as secure as it seems. Researchers proposed and demonstrated several types of physical attacks on relevant hardware, including Differential Power Analysis (DPA) [2], keystroke capture [4], recording electromagnetic-emission [1] and even acoustic attacks [3].

In this work, we seek to complicate attacks against cryptographically secure hardware that are based on DPA. DPA technique uses measurements of power consumption of a circuit in different circumstances in the hope that energy patterns may correlate with signal patterns. We propose to thwart DPA using the new concept of *uniformly-switching (U-S) circuits* that leak less side-channel information than conventional logic circuits. We also study U-S versions of common logic blocks, such as adders and comparators. Given the overhead of uniformly-switching logic, we do not expect that complete secure systems will be entirely based on such circuits. For example, some cryptographic algorithms use published look-up tables that do not compromise secure information. Such look-up tables do not necessarily need to be protected from information leakage. Our work also describes a design technique called the doubling construction, which, applied to a U-S circuit results in construction of a highly DPA resistant circuit.

2 Definitions, the doubling construction

As we seek to equalize power dissipation in whole circuits, we might want to start with single gates and model energy dissipated on high-fanout nets by inserting buffers. Explicitly accounting for simultaneous transitions on multiple inputs is a known obstacle in circuit analysis, and we therefore resort to a standard modelling assumption where only one input is allowed to transition at a time. Multiple-input transitions are modelled as shortest sequences of single-input transitions, which statistically gives a reasonable approximation of circuit physics. In practice, input transitions that are not synchronized do not have to arrive simultaneously and can be deliberately separated in time by gate sizing, buffering and clever layout. When selecting gate libraries, we will require that every gate has the same number of switching outputs for every possible single-output transition. Therefore, when a multiple-input transition is decomposed into a shortest sequence of single-bit transitions, the overall result (in terms of power) does not depend on the specific sequence chosen. Formally, we define U-S gates as follows.

Definition 1. A gate $f(x_1, x_2, ..., x_n) = (y_1, y_2, ..., y_k)$ with n inputs and k outputs is *uniformly-switching* (**U-S**) iff there is a constant $0 < M_f \leq k$ with the following property. For any input combination $(x_1, x_2, ..., x_n)$, changing the value of any single bit in it will lead to changing exactly M_f output bits.

We next define two variants of U-S circuits.

Definition 2. A circuit is called *weak U-S* if for each input wire x_i there exists a constant d_i such that for any one-bit input transition on x_i, the circuit experiences d_i output transitions.

Definition 3. A circuit is called *strong U-S* with a fixed parameter C if for any one-bit input transition, it experiences C gate switches. [1]

We now revisit (as compared to [1]) the conversion of an arbitrary weak U-S circuit $\mathcal{C}$ into a strong U-S circuit, and also consider extending an arbitrary strong U-S circuit so that it experiences the same number of transitions on every clock cycle, regardless of the number of input transitions. The latter clearly requires sequential gates, otherwise no transitions can happen when input values stay constant. Below, we present a *doubling construction* that effectively performs both tasks.

We clone $\mathcal{C}$ and connect its identical copy $\mathcal{C}_2$ in such a way that the input wire x'_i of $\mathcal{C}_2$ transitions if and only if the input wire x_i of $\mathcal{C}$ does *not* transition. To accomplish this, each input of $\mathcal{C}_2$ is computed as an XOR of an input of $\mathcal{C}$ and an output of a shared toggle (T) flip-flop, as shown in Figure 1. Given a multi-bit input transition, the number of signal transitions in $\mathcal{C}$ is the sum of d_i over indices of transitioning input wires. Similarly, the number of signal transitions in $\mathcal{C}_2$ is the sum of d_i over complementary indices. Therefore, the total is Σd_i, and thus the overall energy dissipation is always the same. The outputs of $\mathcal{C}_2$ are considered “garbage wires” and grounded. If the resulting short-circuit power is a concern for DPA, dual-rail logic can be used to equalize energy dissipation of logic 0 and 1.

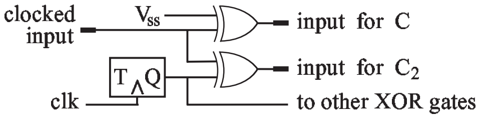

Figure 1: Complementary inputs.

A potential problem with the doubling construction above is that when $\mathcal{C}$ is large and none of its inputs switch, each input of $\mathcal{C}_2$ must

[1] Note that strong U-S circuits are precisely weak U-S circuits with zero *variation* V, where $V := \sum_{1 \leq i \leq n}(\max\{d_1, d_2, ..., d_n\} - d_i)$. This observation can be used to harden a weak U-S circuit into a strong U-S circuit by decreasing variation to zero while preserving the weak U-S property.

Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'05)
1530-1591/05 $ 20.00 IEEE

IEEE COMPUTER SOCIETY

AND_{u-s}	OR_{u-s}	XOR_{u-s}
$xy,\ \bar{x}\,\bar{y}$	$x \vee y,\ \bar{x} \vee \bar{y}$	$x \oplus y$
$xy,\ a \oplus \bar{x}\,\bar{y}$	$x \vee y,\ a \oplus\ \bar{x} \vee \bar{y}$	$x \oplus y,\ \bar{a}$
$xy,\ a \oplus b \oplus \bar{x}\,\bar{y}$	$x \vee y,\ a \oplus b \oplus\ \bar{x} \vee \bar{y}$	$x \oplus y,\ a \oplus b$

Table 1: U-S variants of AND, OR and XOR with primary inputs x and y and garbage inputs a and b.

switch, forming a detectable hot-spot if $\mathcal{C}_2$ is laid out separately from $\mathcal{C}$. This concern can be addressed by the following *layout interleaving* technique. Since we are dealing with two copies of the same circuit, we match each gate in $\mathcal{C}$ with its unique copy in $\mathcal{C}_2$ and place matched pairs side by side. To accomplish interleaving with existing layout tools, we artificially double the width of every gate in $\mathcal{C}$, place the bloated gates with existing software, then shrink the gates to their original size and use the remaining room to place each gate of $\mathcal{C}_2$ next to its original from $\mathcal{C}$.

3 Uniformly-switching gates

A complete set of gates is usually required for synthesis. The following U-S two-input two-output gates with parameter $M_f = 1$ are derived from the AND and NOT gates, which shows that they alone form a complete gate library: $(x_1, x_2) \mapsto (x_1 x_2,\ x_1 \vee x_2)$ and $(x_1, x_2) \mapsto (\bar{x_1}, \bar{x_2})$. Observe that the former gate computes both AND and OR functions. Further, it is possible to directly construct U-S extensions of other common gates, such as NAND and NOR, and multi-input gates such as MAJ and AOI.

Lemma 1. Every n-input k-output Boolean function can be extended to an n-input $2k$-output U-S function with parameter $M_f = k$.

Non-traditional U-S gates can be assembled from more common CMOS gates, and may be optimized at the transistor level. However, care should be taken to ensure that all new gates with equal M_f dissipate approximately equal amounts of energy at every transition. This can be achieved by varying widths of individual transistors.

4 Adapting conventional logic circuits

Since we now have a universal U-S gate library, our next step is adapting conventional logic circuits to the weak U-S form. As an input we take a NOT-AND-OR-XOR circuit. We first substitute every gate in the circuit with a U-S variant using Lemma 1, leaving new gate outputs (garbage) unconnected. Eventually, the main output of each gate and the newly-constructed garbage output will connect to the same downstream gate. However, such a downstream gate will have to accommodate three or four inputs, including one or two garbage inputs that do not affect the main output. Such gates can be constructed by adding disconnected inputs to two-input one-output gates and applying Lemma 1 to define the additional garbage output (see examples in Table 1). In the constructed circuit, all wires except primary inputs are paired up — the main gate output and the garbage output. In each such pair, exactly one wire switches after a one-bit input transition at their driving gate. If inputs of every gate transition one bit at a time, then the resulting circuit is weak U-S because every one-bit input transition sensitizes all signal paths originating from it. Figure 2 illustrates how (a) conventional circuit for a full adder can be transformed into (b) weak U-S. U-S variants of AND, OR and XOR gates (see Table 1) are marked with "u-s". Analysis of transistor layouts suggests that the overhead of this circuit transformation is between 2 and 5 times by area, and much smaller in terms of signal delay.

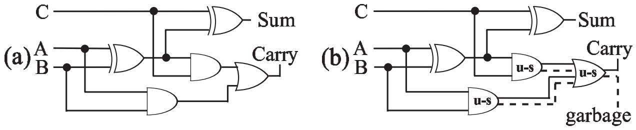

Figure 2: Three-bit full adders.

5 Comparisons to related work

The work most related to ours includes the *SABL* logic family [5] and the recent *Wave Dynamic Differential Logic (WDDL)* [6]. A major advantage of WDDL is that it can be handled by a traditional EDA tool flow. Our work goes further in the sense that we show how to reuse existing tools for synthesis and layout. However, we pursue a somewhat different task — equalizing energy dissipation, not total power consumption. Indeed, in CMOS power is consumed mostly during 0-1 transitions, but both 0-1 and 1-0 transitions dissipate energy.

We observe that empirical results in [6] require careful interpretation. For example, path delay overhead in Table 1 does not account for the use of every second cycle in WDDL circuits for "pre-charging waves", which halves data rate for the same cycle time. Our techniques do not affect data rate. Area and delay overhead may strongly depend on the logic function. Indeed, the WDDL logic requires re-expressing each Boolean function using AND/OR/NOT gates, while the U-S logic extends existing circuits. The former results in a large overhead when many XOR operations are required, e.g., in the Kasumi algorithm [6]. Our techniques adapt existing circuits and do not impose significant restrictions on the gates used. They incur the smallest overhead on circuits consisting entirely of XOR/XNOR/NOT gates. Since in WDDL all inputs are pre-charged to zero, the use of AND/OR gates instead of NAND/NOR seems unavoidable, implying additional 50% area overhead in CMOS (however, a convenient LUT-based implementation is proposed in [6]). This and the pairing of AND/OR gates in WDDL implies a lower bound of 3x on area overhead, which agrees with empirical data in [6, Table 1] and sharpens the lower bound of 2x advertised in [6]. The use of WDDL may require complete re-synthesis, while we adapt existing circuits and preserve the structure of critical paths.

6 Conclusion

We propose a new architecture for cryptographic applications to mitigate side-channel information, especially dissipated energy. Our *uniformization* and *doubling* constructions equalize energy dissipation for all inputs, states, as well as input and state transitions.

References

[1] W. Knight, "Computer Chip Noise May Betray Code". *New Scientist*, May 2004. http://www.newscientist.com/news/news.jsp?id=ns99994979.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", *Lecture Notes in Comp. Sci.*, 1666:388–397, Jan. 1999.

[3] A. Shamir and E. Tromer, "Acoustic Cryptanalysis", http://www.wisdom.weizmann.ac.il/˜tromer/acoustic/,'04.

[4] M. Stroh, "Loose Clicks Sink Computers", *Baltimore Sun*, 07/19/2004, p. 6A.

[5] K. Tiri, M. Akmal, I. Verbauwhede, "A Dynamic and Differential CMOS Logic With Signal-Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards" *IEEE Eur. Solid-State Circ. Conf. (ESSCIRC)*, pp. 403-406, 2002.

[6] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation", *DATE 2004*, pp. 246–251.