Summer 2017

Randomized encoding of combinational and sequential logic for resistance to hardware Trojans

Travis Edward Schulze

Department: Electrical and Computer Engineering

RANDOMIZED ENCODING OF COMBINATIONAL AND SEQUENTIAL LOGIC FOR RESISTANCE TO HARDWARE TROJANS

by

TRAVIS EDWARD SCHULZE

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

2017

Approved
Daryl Beetner, Advisor
Minsu Choi
Mihail Cutitaru
Yiyu Shi
Daniel Tauritz

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following three published or to be published papers, formatted in the style used by the Missouri University of Science and Technology, listed as follows:

Paper I on pages 7-31, was first presented at Asian HOST 2016, in Yilan, Taiwan. The extended version presented here has been submitted to Integration, The VLSI Journal.

Paper II on pages 32-58, has been submitted to IEEE DASC 2017 conference in Orlando, FL.

Paper III on pages 59-85, is intended for submission to IEEE Transactions on VLSI.

ABSTRACT

Globalization of micro-chip fabrication has opened a new avenue of cyber-crime. It is now possible to insert hardware Trojans directly into a chip during the manufacturing process. These hardware Trojans are capable of destroying a chip, reducing performance or even capturing sensitive data. To date, defensive methods have focused on detection of the Trojan circuitry or prevention through design for security methods.

This dissertation presents a shift away from prevention and detection to a design methodology wherein one no longer cares if a Trojan is present or not. The Randomized Encoding of Combinational Logic for Resistance to Data Leakage or RECORD process is presented in the first of three papers. This chip design process utilizes dual rail encoding and Quilt Packaging to create a secure combinational design that can resist data leakage even when the full design is known to an attacker. This is done with only a 2.28x-2.33 x area increase and 1.7x-2.24x increase in power. The second paper describes a new method, Sequential RECORD, which introduces additional randomness and moves to 3D split manufacturing to isolate the secure areas of the design. Sequential RECORD is shown to work with 3.75x area overhead and 4.5x power increase with a 3% reduction in slack. Finally, the RECORD concept is refined into a Time Division Multiplexed (TDM) version in the third paper, which reduces area and power overhead by 63% and 56% respectively. A method to safely utilize commercial chips based on the TDM RECORD concept is also demonstrated. This method allows the commercial chip to be operated safely without modification at the cost of latency, which increases by 3.9x.

ACKNOWLEDGMENTS

First, a special thank you to Dr. Yiyu Shi for both inspiring this work and for taking a chance on someone who just called him out of the blue one day.

Thank you Dr. Daryl Beetner for agreeing to advise me after the departure of Dr. Shi and for all your help since.

Thanks to the gentlemen of the Air Force Research Lab in Rome, NY, Dr. Kevin Kwiat and Dr. Charles Kamhoua, for your help refining the RECORD ideas.

Thank you to all the members of my committee who took time out of their busy schedules to read and listen to this work; Dr. Minsu Choi, Dr. Mihail Cutitaru and Dr. Daniel Tauritz.

Last but not least thank you to my beautiful wife Christina without whom none of this would have been possible.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

PAPER I

PAPER II

PAPER III

LIST OF TABLES

Page

1. INTRODUCTION

1.1 BACKGROUND

Since the proliferation of computerized electronics in the 1980's, hackers have been trying to gain unauthorized access to these personal computers or cause general mischief by developing malicious software. This software is commonly known by many names: computer virus, worm, Trojan, etc. With the integration of the internet into everyday interactions the threat of a computer virus is a part of daily life. The defenses are many and commercially available. Operating system patches to prevent virus exploits are a regular occurrence. However, there is an underlying assumption common to all the defenses against computer viruses, namely that the hardware running it is safe and operating as intended.

In the early 2000's, that paradigm was upended by the concept of malicious hardware embedded into a chip at the time of manufacturing [1]. These hardware Trojans, as they came to be known, became possible through the globalization of the microchip industry. As the semiconductor technology became smaller and smaller the cost of fabrication facilities became just that much greater. Only the largest manufacturers could continue to operate and maintain modern facilities. This forced most chip designers to outsource their designs to other, cheaper, countries. To manufacture a chip, the complete design must be sent to the fabrication facility, usually in a standard GDSII file format. The chip manufacturer then has an opportunity to alter the design to suit their purposes, creating a hardware Trojan.

Hardware Trojans are generally broken out into two categories: reliability and data leakage. The reliability Trojan aims to disrupt the overall function of the chip in

some way or to reduce its useful life [2, 3]. This can mean that the mean time to failure (MTTF) is significantly reduced or that data is corrupted in some way, making the final output meaningless. Initially the infected chips operate as expected, especially during testing. The Trojan effects are triggered after some long period of time or after a rare sequence of events occurs on chip.

Data leakage Trojans are more complicated and potentially more damaging. The undetected loss of secret information can be devastating. These Trojans will not affect the normal operation of chips. Instead, they scan and capture data, such as an encryption key, as it is processed, or possibly allow privilege escalation on a CPU [4, 5]. The captured data can be leaked out through Wi-Fi [6] or the power emissions could even be harnessed [7]. Introduction of a data leaking Trojan is much more complex than a reliability Trojan. The reliability of a chip can be compromised with little understanding of the overall function of the design. Simply reducing some key wire widths so that failure occurs prematurely or adding a counter to switch a line to ground will cause a chip to fail in the field [2]. Conversely, a data leakage Trojan designer must have a near total understanding of the circuit they wish to infect. Since the data leakage of confidential information is so valuable the extra effort is warranted, so are extra defensive measures. Data leakage Trojans will be the focus of this dissertation. Unless otherwise noted, hardware Trojans will refer to the data leakage type in the following text.

1.2 TROJAN DEFENSES

Hardware designers can either try to detect the Trojan or prevent it. Detection of hardware Trojans is extremely difficult, but efforts have been made to detect them.

Runtime monitoring and post-manufacture testing [8] rely on identifying the differences in chip operation introduced by the Trojan circuit. These methods depend on the tester's ability to trigger the Trojan circuit so the effects of the Trojan can be measured. Triggers are intentionally designed so that testing is unlikely to uncover them, e.g. repeating the same instruction hundreds of times. Once triggered, the effects can be obvious, such as circuit malfunction, or can be subtle. Subtle changes to the chip's operation can sometimes be identified through side channels such as power, temperature, path delay or EM emissions. Runtime monitoring and post manufacture testing usually rely on the golden chip concept. The golden chip requirement is the Achilles heel of these methods. Since the designers are outsourcing the design, the only place this chip can come from is the same facility that produced the suspect chips in the first place. Simulations are not typically accurate enough to detect subtle changes in side channel measurements.

A hardware engineer must then try to prevent the attacker from placing the Trojan on chip using a Design for Security (DFS) method. The currently available methods all try to accomplish the same goal, which is to prevent the attacker from understanding the design. The idea being that if an attacker cannot understand what the chip is doing or how it is laid out then there is no opportunity to find and leak data. Commonly available methods include obfuscation, layout camouflaging and split manufacturing [9]. Obfuscation aims to make the function of the circuit less obvious by using nonstandard designs for common functions. It also includes the technique of logic encryption where the data processed or the function performed in a circuit is encrypted [10]. Obfuscation can also be performed on state machines in the design, additional states are added leading to dead ends or black hole states [9].

Layout camouflaging attempts to disguise the design by making the layouts of each gate indistinguishable from each other. For example, the layout of a NAND or NOR cell can be made to look identical. Extracting the netlist using image based techniques on the layout mask then becomes difficult [11, 12].

Finally, split manufacturing attempts to break up the design into front-end and back-end layers. The front-end consists of the lower silicon layers and first metal layers. The back-end being the remaining metal layers [13]. Splitting the fabrication prevents an attacker in one location from having access to the complete design. This can be extended to 3D ICs as well since the upper chip can naturally be manufactured separately.

Unfortunately, all three methods have weaknesses. Obfuscation and layout camouflaging can both be deciphered given enough effort and time spent to reverse engineer the design files. Split manufacturing is susceptible when multiple production runs are needed. An attacker could exploit industry standards in floorplaning, placement and routing to alter one half of the split chip successfully [13] on the first run, or reverse engineer a finished chip, which can be obtained through legitimate or illegitimate means, and inserting attacks into subsequent production runs which are often needed to meet demand [14].

1.3 RESEARCH CONTRIBUTION

The research presented in the subsequent papers presents a new paradigm in hardware Trojan defense. Using the following DFS methods to defend against data leakage Trojans, designers no longer need to worry about detecting Trojans or even

preventing them. The designers can simply ignore them. The first paper, RECORD, presents a new method of designing a custom ASIC chip for defense. Randomized Encoding of Combinational Logic for Resistance to Data Leakage (RECORD) describes how a combinational logic design can be converted to a randomized dual rail circuit. The encoding, in combination with a split manufacturing process called Quilt Packaging [11] which splits the chip into secure and insecure portions, prevents an attacker from capturing any meaningful data from any outsourced portion of the ASIC design. The RECORD process is effective even if the design is fully known to an attacker and maintains its effectiveness through any number of subsequent production runs. The design is generic, allowing it to be used quickly and easily on any existing combinational design. The only costs to the RECORD process are increased area and power, 2.28x-2.33x and 1.7x-2.24x respectively in sample tests on an Advanced Encryption Standard (AES) Substitution Box (SBox) design.

The second paper addresses the challenges of utilizing the RECORD process in a sequential logic environment. By importing the RECORD process directly into a sequential circuit, new vulnerabilities develop. Data is now available on multiple clock cycles and a clever attacker could infer and decode the randomized dual rail signal. The Sequential RECORD process introduces additional randomness into the dual rail encoding along with additional randomness in the assembly of the final chip. No longer will Quilt Packaging be used; instead 3D split manufacturing will take its place. Careful segregation of the circuit components allows the lower tier of the 3D process to be interchangeable with a large number of generic upper tiers. The Sequential RECORD process ends up being more secure than RECORD with a far greater number of

permutations available to the designer. Sequential RECORD is again generic and viable across multiple production runs. The cost is again in increased area and power, 3.75x and 4.5x respectively, and a slight impact to performance of 3% reduction in slack in a sample Data Encryption Standard circuit design.

The major weaknesses of RECORD and Sequential RECORD are addressed by the third paper, namely the increased area and power and the reliance on ASIC designs. Many if not most companies today rely on commercial of the shelf (COTS) products, not their own custom designs. RECORD would be useless on a COTS chip as it stands. To adapt the RECORD concepts to COTS chips and to address the high cost of area and power, a Time Division Multiplexed (TDM) version of RECORD is introduced for both combinational and sequential designs. It decreases the area overhead of sequential RECORD by 63% and power by 56% at the cost of latency which increases by at least 5x. The TDM concept is then further refined to show how it can be used to operate a COTS product from a second chip. The second chip could be an FPGA or CPU. The COTS process is then proven out in real hardware which demonstrates the process and the RECORD principles. The cost of the COTS process is in latency which increases by 3.9x. The RECORD method and its derivatives represent an entirely new way for both the ASIC designer and the COTS integrator to protect their designs free from the worry of data leakage from hardware Trojans.

PAPER

I. PROTECTING INTEGRATED CIRCUITS FROM HARDWARE TROJANS WITH RANDOM DATA ENCODING AND SPLIT MANUFACTURING

Travis E. Schulze[*], Kevin Kwiat[+], Charles Kamhoua[+], Daryl Beetner[*], and Yiyu Shi[Δ]
[*] Dept. of ECE, Missouri University of Science and Technology, Rolla, MO, USA, 65401
[+] Air Force Research Lab., Information Directorate, Cyber Assurance Branch, Rome, NY, USA, 13441
[Δ]Dept. of CSE, University of Notre Dame, Notre Dame, IN, USA, 46556

ABSTRACT

Many companies outsource manufacturing of their chips. Untrustworthy manufacturers may add hardware Trojans which cause data leakage. Existing defensive methods can be compromised if attackers can physically access the chip. A technique, called *RECORD* (*R*andomized *E*ncoding of *CO*mbinational Logic for *R*esistance to *D*ata leakage) is proposed which uses Quilt Packaging and data randomization to prevent attackers from interpreting data even when data leakage exists. Experiments on a 45 nm 8-bit Advanced Encryption Standard (AES) Substitution Box (Sbox) show RECORD can effectively hide information with approximately 2.3x increase in area, 1.7x in dynamic power and 1.06x in delay.

1. INTRODUCTION

The ever-increasing cost of technology scaling has forced many design houses to outsource their semiconductor fabrication process to lower cost facilities in other countries. Chip manufacturing has become a global enterprise. Outsourcing presents a problem when sensitive designs must be surrendered to the manufacturer before production. These manufacturers may not have secure facilities or processes, and their trustworthiness remains unknown. The opportunity exists for malicious parties, or attackers, to re-engineer the original design and to insert malicious hardware known as *hardware Trojans*.

The original functionality of the chip is maintained after Trojan insertion, with little to no increase in area or power consumption, making it very difficult to detect the attack during testing. At runtime, the Trojans are triggered externally or by a specific sequence of internal signals.

Trojan circuits generally target *reliability* or *data leakage*. The reliability Trojan aims to damage the chip in some way or otherwise make it non-functional [1-2]. These Trojans can significantly reduce the mean time to failure (MTTF) or corrupt the data to make the final output meaningless. Data leakage Trojans are more complicated and potentially more damaging. They will not affect the normal operation of chips. Instead, they scan and capture data or give an unauthorized user control of the system, for example by leaking an encryption key or allowing privilege escalation [3]. The technique proposed in this paper defends against data leakage Trojans.

Successful execution of a data leakage Trojan circuit relies on the attacker's ability to understand the chip design. This can be accomplished before fabrication by

analyzing the netlist and layout, or afterwards by reverse-engineering a fabricated chip from the open market [4]. Methods of combating hardware Trojans include runtime monitoring, post-silicon testing [5] or design for security (DFS) [6]. Runtime monitoring and post-silicon testing try to detect abnormal chip behaviors when hardware Trojans are triggered. They are ineffective against data leakage Trojans which do not change the chip's behavior. DFS attempts to make it harder for the attacker to understand the design through obfuscation, layout camouflaging, and split manufacturing [7]. Obfuscation aims to make the function of the circuit less obvious by using nonstandard designs for common functions or by adding states to state machines which lead to dead end or black hole states [7]. Camouflaging attempts to disguise the design by making the layouts of each gate indistinguishable, for example by making a NAND or NOR cell look identical, so that extracting the netlist from the layout becomes difficult or impossible [7,8]. Split manufacturing attempts to break up the design so that the lower silicon and metal layers are manufactured with one company and the remaining metal layers with another [9], preventing either fabricator access to complete design information. All three methods can be compromised when an attacker procures a fabricated chip and reverse-engineers the design. Hardware Trojans can then be designed and injected in the manufacturing process.

A new technique is proposed which prevents leakage of useful data from an established Hardware Trojan. This technique, called RECORD *(R*andomized *E*ncoding of *CO*mbinational Logic for *R*esistance to *D*ata leakage) [10], uses Dual-rail encoding to randomize the information within the chip, and Quilt Packaging [11] to protect a small portion of critical information that is needed to decode the data on-chip. This scheme

prevents an attacker from interpreting leaked data, even if they have full access to the outsourced design and data within the outsourced portion. Simulations of a 45 nm 8-bit Advanced Encryption Standard (AES) Substitution Box (Sbox) show that RECORD can effectively hide the information being processed while incurring an acceptable increase in area, power, and delay.

The remainder of the paper is organized as follows. Section 2 describes the dual rail randomized encoding with Quilt Packaging for data leakage hardware Trojan resistance. Design examples are discussed in Section 3. Concluding remarks are presented in Section 4.

2. FRAMEWORK OF RECORD

The basic framework to randomize the on-chip information in hardware is discussed in Section 2.1. The utilization of Quilt Packaging is discussed in Section 2.2. The associated vulnerability analysis is revealed in Section 2.3 and the design overhead is discussed in Section 2.4.

2.1 RANDOMIZED ENCODING

The key idea of RECORD is to introduce randomness in information processing. To accomplish this, non-overlapping codes are defined for logic values. To allow randomness to be introduced, at least two bits (i.e. dual-rail logic) are needed to encode a logic zero and logic one. The dual-rail combination 00 and 11 were defined to represent a logical zero, and 01 and 10 to represent a one. One of the two rails will be generated from a random number generator and will be held in the "secure" section of the chip. This rail will be called the random rail. The second rail will contain a value which depends on the single-rail logic value and the value on the random rail, and will be available to the inner combinational logic.

Conversion between the single-rail logic and the corresponding dual-rail logic is straightforward. Consider a single-rail binary logic value *x*. One of the dual rails will be given a random logic value *r*. The other rail will be given a logic value, *t*, determined from *x* and *r* as

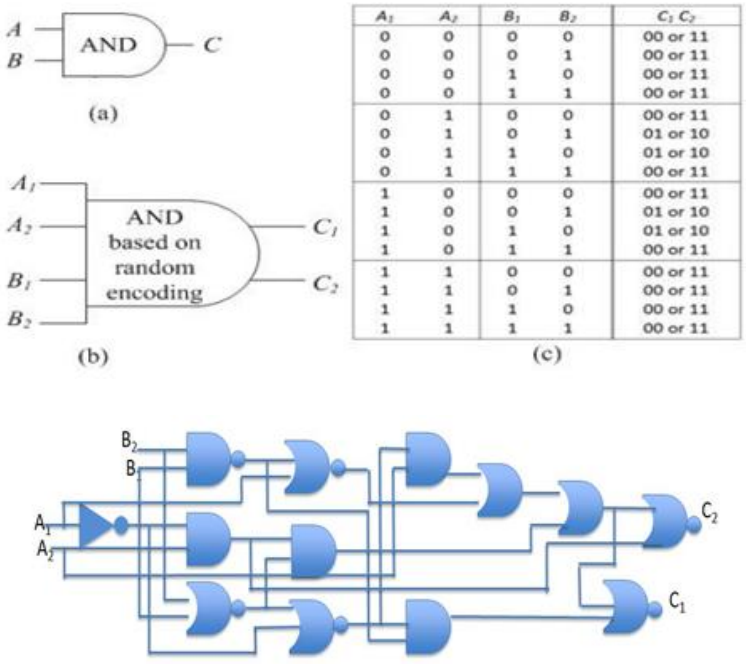
$$t = x \oplus r . \qquad (1)$$

One must then know both *r* and *t* to determine *x*:

$$x = t \oplus r. \qquad (2)$$

Protecting the data, *x*, from Hardware Trojans can then be accomplished by preventing the attacker from accessing the value *r* on the random rail.

A simplistic implementation of the dual-rail approach is shown in Figure 2.1, where each single-rail logic gate is replaced with a corresponding dual-rail gate. The values of A and B correspond to the values of the data, *x*, and A_1, A_2, B_1 and B_2 correspond to the associated values of *t* and *r*. This dual-rail implementation requires an unacceptable increase in area and power over the single-rail gate.



A_1	A_2	B_1	B_2	C_1 C_2
0	0	0	0	00 or 11
0	0	0	1	00 or 11
0	0	1	0	00 or 11
0	0	1	1	00 or 11
0	1	0	0	00 or 11
0	1	0	1	01 or 10
0	1	1	0	01 or 10
0	1	1	1	00 or 11
1	0	0	0	00 or 11
1	0	0	1	01 or 10
1	0	1	0	01 or 10
1	0	1	1	00 or 11
1	1	0	0	00 or 11
1	1	0	1	00 or 11
1	1	1	0	00 or 11
1	1	1	1	00 or 11

Figure 2.1. (a) A conventional AND gate; (b) The equivalent AND gate based on the random encoding; (c) Its possible truth tables; and (d) One corresponding implementation

To reduce overhead while maintaining the randomness needed for security, all the gates in a combinational logic block can share the same random rail. Doing so allows use of only one random number generator and allows the logic implementation to become simpler.

With all input signals sharing the same random bit, any Boolean function $f(x_1, x_2, x_3, \dots)$ with $x_1, x_2, x_3 \dots$ as Boolean variables can be converted to the corresponding dual-rail representation as

$$f(x_1, x_2, x_3, \dots) \rightarrow (f(t_1, t_2, t_3, \dots) \oplus r)$$
$$= (f(x_1 \oplus r, x_2 \oplus r, x_3 \oplus r, \dots) \oplus r, r) = (g, r) \qquad (3)$$

where the dual-rail output is represented by r and g, where r is the random logic value on the common random rail, g is the dual-rail representation of the function output, and $t_1, t_2, \dots$ are the logic values on the input dual rail corresponding to input signals $x_1, x_2 \dots$, respectively, i.e., $x_i = t_i \oplus r, \ ^{\forall i}$. While calculation of g still seems complicated, it is worthwhile to note the following logic equivalency which uses Shannon expansion.

$$f(x_1 \oplus r, x_2 \oplus r, x_3 \oplus r, \dots) \oplus r \qquad (4)$$

$$= rf(x_1 \oplus 1, x_2 \oplus 1, x_3 \oplus 1, \dots) \oplus 1 +$$
$$\bar{r}f(x_1 \oplus 0, x_2 \oplus 0, x_3 \oplus 0, \dots) \oplus 0 \qquad (5)$$
$$= r\overline{f(\bar{t}_1, \bar{t}_2, \bar{t}_3 \dots)} + \bar{r}f(t_1, t_2, t_3 \dots)$$

The dual-rail output of a function $f(x_1, x_2, x_3, \ldots)$ can thus be determined from the values of inputs *t*, the random rail *r*, and the combinational function $f(\cdot)$ and can further be implemented using a multiplexer (MUX) as shown in Figure 2.2.

Such a MUX-based implementation has an area and power overhead of approximately 2x compared to the single-rail approach, and applies to any Boolean function. In addition to reducing overhead, it has the benefit that the random signal *r* is clearly separated from the rest of the calculation. As long as this signal and the final MUX are hidden from the attacker, then the information obtained from any other portion of the circuit cannot be directly decoded.

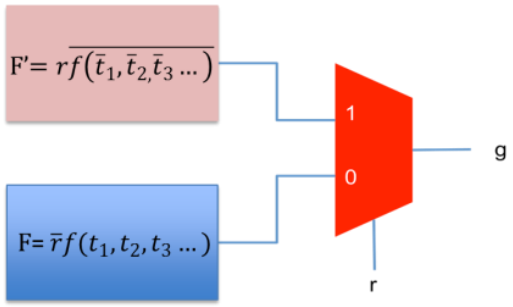


Figure 2.2. A MUX-based implementation of randomized dual-rail encoding scheme

RECORD can be implemented in various ways. For example, it is not necessary to convert all the inputs to dual-rail. Changing just the first input, x_1, of a function to dual rail gives the same effect as converting all the inputs. The corresponding dual-rail representation of a function *f* is as follows

$$f(x_1, x_2, x_3, \ldots) \rightarrow (f(x_1 \oplus r, x_2, x_3, \ldots) \oplus r, r) \qquad (6)$$

The first rail can be re-cast as

$$f(x_1 \oplus r, x_2, x_3, \dots) \oplus r = \tag{7}$$

$$rf((x_1 \oplus 1, x_2, x_3, \dots) \oplus 1 + \bar{r}f((x_1 \oplus 0, x_2, x_3, \dots) \oplus 0$$

$$= r\overline{f(\bar{t}_1, x_2, x_3, \dots)} + \bar{r}f(t_1, x_2, x_3, \dots) \tag{8}$$

The MUX-based implementation is shown in Figure 2.3. In this figure, only input *x1* is converted to dual-rail representation (t_1, r), and all the remaining inputs are single-rail. The output g is also in dual-rail representation with the random rail r, (i.e., the final single-rail logic value will be,$g \oplus$ r). Compared with the implementation in Figure 2.2, this implementation will result in different power and area overhead as shown by the design examples in Section 3.

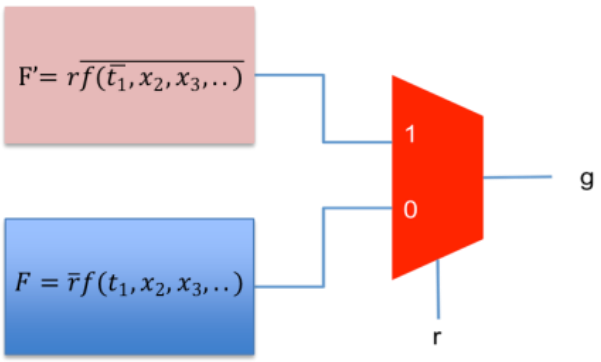


Figure 2.3. An alternative implementation of the randomized dual-rail encoding scheme

For RECORD to be effective it must protect the random rail *r* as well as the final MUX, such that any data obtained from elsewhere on the chip cannot be directly interpreted. So long as this data is protected, the user cannot interpret the data obtained from any other part of the circuit. The next section will explain how random rail and MUX can be protected from hardware Trojans.

2.2 QUILT PACKAGING

To determine the meaning of data within the unsecure portion of the IC, the attacker must know the value of *r*. To get this value, a hardware Trojan must monitor the values of *f* and *f'* and the output, *g*, of the MUX in Figures 2.2 or 2.3, so that *r* can be inferred, or must directly monitor the random rail. These values can be protected using Quilt Packaging [12-14]. Quilt Packaging allows two dies of different sizes and technologies to be fabricated separately and then joined. The process creates a high-speed, low-loss connection with measured insertion losses of only 1 dB at 110 GHz and 2.25dB at 220 GHz [13]. The dies can be attached using several methods including Sn immersion plating and pin transfer of solder paste [14].

To utilize Quilt Packaging, the RECORD design can be partitioned so that a secure area of the chip input/output (I/O) is designed and fabricated separately. The secure I/O area includes the random number generator, the XOR gates for conversion between single-rail and dual-rail, and the output selection MUXes. These components require a small area relative to the remainder of the design. The two dies can then be combined using Quilt Packaging in a trusted facility. An illustration of the layout partition is shown in Figure 2.4, where x_1 is converted to dual-rail (t_1, r), and the output is

converted back to single rail using the same random rail *r*. The random bit would not exist anywhere on the outsourced die, thus prohibiting direct monitoring of this bit's value.

Consider the fact that these I/O elements are replicated many times for each design, once for every input and output bit, and are interchangeable between designs. It is then possible to pre-fabricate them as standardized circuits in a secure facility for universal applications (as long as the locations of the I/O's are pre-defined as a standard). Figure 2.5, shows an example layout. The random number generator is shown in red, as is the random rail. Note that these modules and the layout pattern are independent of the combinational function being implemented, or the number of inputs that are converted to dual-rail. RECORD envisions that these circuits would just need to be fabricated once for many different designs. For instance, the design in Figure 2.4 can now be generated by one random number generator module, one dual-rail input module, and one output module.

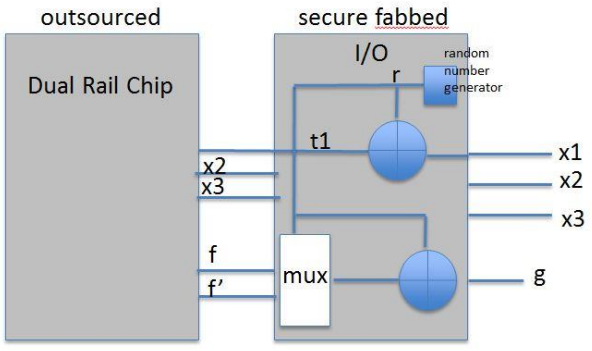


Figure 2.4. Partition of layout for Quilt Packaging

2.3 VULNERABILITY ANALYSIS

RECORD was developed to eliminate vulnerabilities that might occur when an attacker adds a hardware Trojans to a chip for the purpose of leaking privileged information from the IC in the final product. RECORD protects against such Trojan's by ensuring the outsourced chip is never given information about the random bit. The random rail and the random number generators are pre-fabricated for universal applications and incorporated into the outsourced design through Quilt Packaging. The attacker has no access to the random data. They will only see the "randomized" inputs, t_i, and that the outsourced portion of the chip yields two outputs, f and f'. The attacker will know that one of the outputs is correct, but will not know which. Simply choosing one of the outputs f or f' at random would not generate any meaningful results. To decode the output, which is itself a dual rail signal, the chosen signal would need to be XOR'd against the inaccessible random bit.

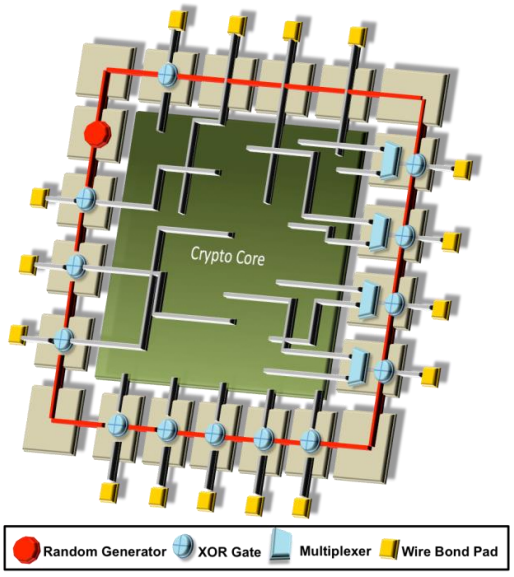


Figure 2.5. Example of Quilt Packaging Layout. The random number generator and random rail are shown in red. The center core can be safely outsourced. The smaller dies on the periphery are pre-fabricated securely and used interchangeably

While RECORD does not allow the outsourced chip access to information about the random bit, and thus prevents an attacker from identifying useful information within the outsourced design, there are other side channels that might assist attackers in obtaining protected information. One important type of side channel attacks is the power analysis attack, including differential power analysis (DPA) and correlation power analysis (CPA) [15-16]. These methods rely on the subtle power difference between a 0 to 1 transition and a 1 to 0 transition of a logic gate, and allow an attacker to guess the data operated on within the IC.

RECORD is naturally resistant to DPA or CPA attacks, since it uses a dual-rail encoding scheme with uniform switching, which hides asymmetries in data processing. The power usage difference between data input patterns in a RECORD design is much smaller than in a conventional design so that a stronger resistance to power analysis attacks can be achieved. Specifically, through randomized encoding (i.e., with 0 being coded as 00 and 11, and 1 being coded as 10 and 01, each with equal probability), the number of wires with a 0→1 or a 1→0 switching in the entire circuit is nearly identical, regardless of the number of logic blocks with a true or false output. The total power consumption reveals little to no information about the processed data.

2.4 OVERHEAD ANALYSIS

The protections afforded by RECORD comes at the cost of an approximately 2x overhead in power and area as each function must be implemented twice, and there are additional costs associated with Quilt Packaging. More accurate data is reported in Section 3, which includes the XOR gates and the random number generator needed for

the conversion between single-rail and dual-rail representations. As should be expected, as the size of the design increases, the overhead will get closer to 2x as the number of components in the protected portion of the design become small compared with the overall design. A 2x overhead is still smaller than many of the existing techniques for reliability enhancement, such as the triple modular redundancy (TMR) [17], and is a cost that should be expected to achieve enhanced security. A 2x overhead should be acceptable for many security applications, where concerns about area and power are usually secondary to the need for trusted hardware.

In addition to power and area, there is also some overhead due to added processing delays. There is minimal difference in the timing of the combinational logic required by the two rails compared to the timing in the original design. Additional delay may be introduced by the inverters that generate the dual-rail inputs and the MUXes at the output, but the added delay should be small compared with the delay of the overall combinational logic block. This assertion will be demonstrated in the design examples in Section 3.

3. DESIGN EXAMPLES

To demonstrate the capabilities and overhead associated with RECORD in real designs, an 8-bit Advanced Encryption Standard (AES) Substitution Box (Sbox) was implemented with RECORD. The AES Sbox was implemented with the same structure as described in [18]. The random bit is generated using a Linear Feedback Shift Register (LFSR) [19], though any valid random number generator would work equally well. The design was synthesized using the FreePDK 45 nm process [20] and Cadence Encounter RTL Compiler v.13.10 to obtain power, area and timing data. One example was then laid out using Cadence Encounter RTL to GDSII v.13.23.

As an illustration of functionality, for RECORD with two inputs converted to dual rail, Figures 3.1 - 3.6 show a series of waveforms for the AES Sbox design. Figure 3.1 shows output bit 3, chosen at random, from the standard AES Sbox. Figure 3.2 shows a randomly selected internal signal from RECORD design. The Figure 3.3 waveform shows the corresponding internal signal from the duplicated function block, F', of the RECORD design. Note the waveforms in Figure 3.2 and Figure 3.3 are not equal as expected, and even if the values in the two waveforms were identical, it is not possible to infer the corresponding logic value unless the random bit is known. Figure 3.4 and 3.5 show the third output bits from each of the two function blocks, F and F'. These are still encoded with the random bit. Finally, Figure 3.6 shows the final bit 3 output of the record process after choosing between 3.4 and 3.5. Note Figure 3.6 is identical to 3.1 as expected.

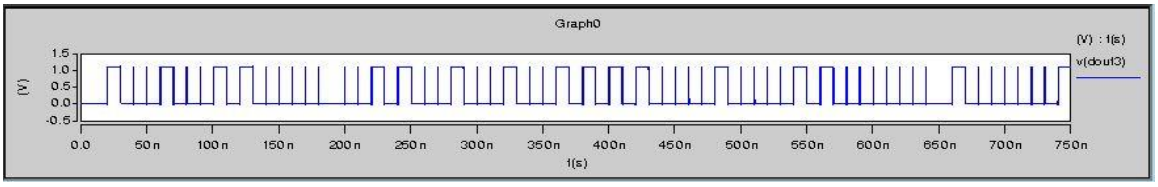


Figure 3.1. Output bit 3 from standard AES Sbox

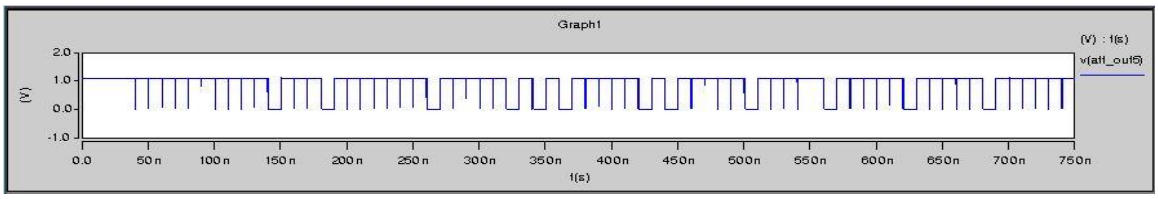


Figure 3.2. Internal signal from first function block, F, of RECORD designed Sbox

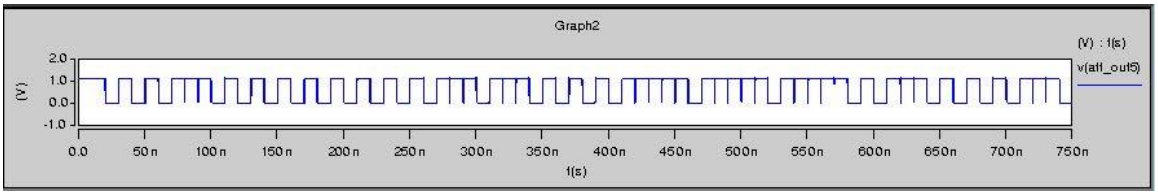


Figure 3.3. Same internal signal as in Figure 8b but from second function block, F', of RECORD designed Sbox

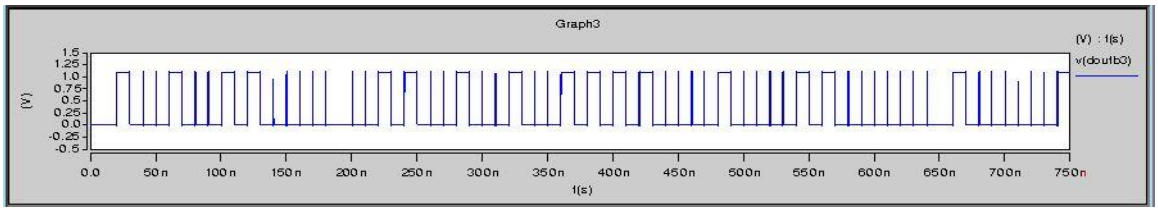


Figure 3.4. Output bit 3 from first function block, F, before demuxing

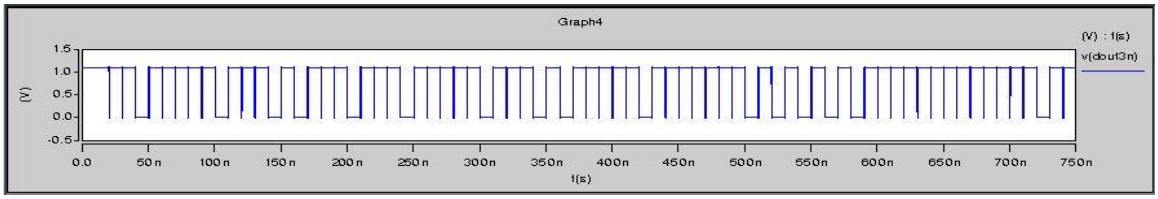


Figure 3.5. Output bit 3 from second function block, F', before demuxing

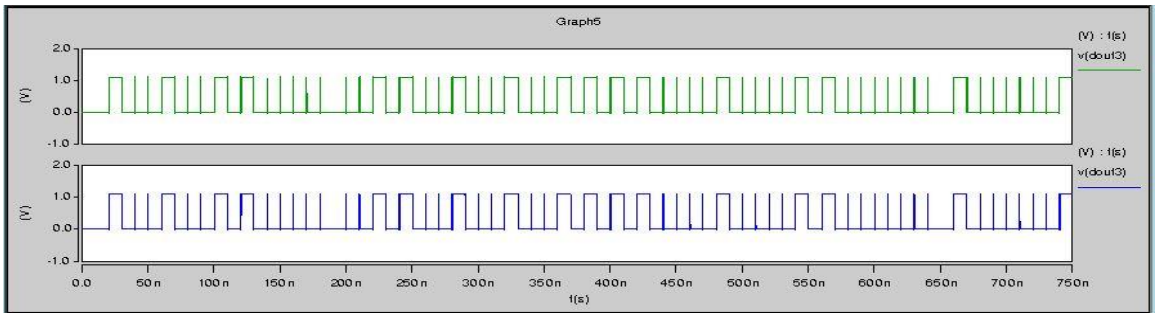


Figure 3.6. Top waveform is a repeat of 8a, standard Sbox output bit 3. Bottom waveform is the final output bit 3 from a RECORD Sbox

Since a designer has the option to choose which inputs are converted to dual rail there are 256 possible designs for the 8-bit AES Sbox. All 256 possibilities have been implemented for comparison of power, area and delay. The increase in total area is shown in Figure 3.7. This figure displays the area for each of the 256 possible design variations sorted by the number of bits converted to dual rail. First one bit then two and so on until all bits are converted to dual rail. As expected the area slowly increases as more bits are converted to dual rail, with the most area efficient designs using only one input converted to dual rail and the designs using the most area converting all bits to dual-rail. It is important to note, however, that there is only a 2% difference between the most and least area efficient designs. The most efficient design, one input bit converted

to dual rail, shows a 2.28x increase in area over the standard non-RECORD sbox. Whereas converting all inputs to dual rail shows only 2.33x area increase. The differences in area result primarily from the added inverters at the inputs and MUXs at the outputs, not from a significant change in the size of the logic.

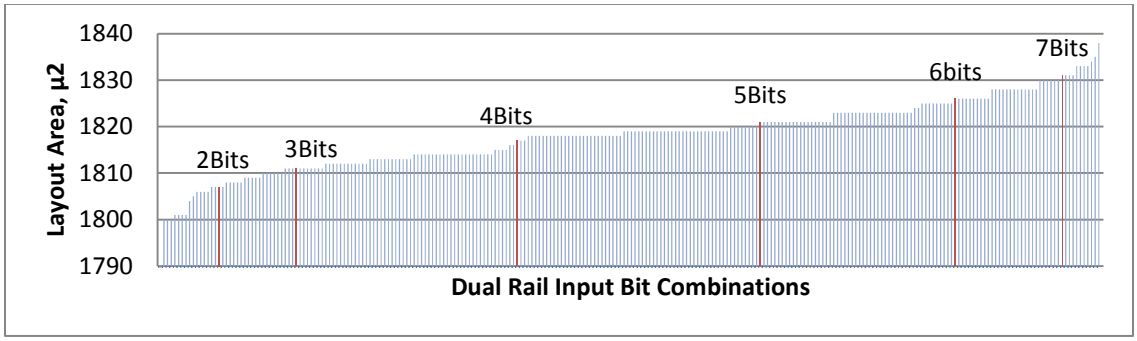


Figure 3.7. Layout area for each possible input dual rail combination, sorted by number of bits converted to dual rail

Figure 3.8 shows the leakage power for all combinations of converted bits, sorted in the same manner as Figure 3.7. A 2.3x increase in leakage power is shown for three of the design options: when bits (7,5,4,3,2), bits (7,5,2,1,0) and bits (5,4,3,2,0) are converted to dual rail. These combinations represent the smallest increase in leakage power. The largest increase in leakage power (2.34x) is found when bits (7,6,5,4,2) are converted to dual rail. Interestingly, both the least and largest increases are found with five bits converted to dual rail. Figure 3.9 shows a similar plot for dynamic power. The least dynamic power increase (1.7x) is found when bits (7,6,5,4,2) are converted which is also the option with the largest leakage power. The largest dynamic power increase of 2.24x

is found for the design option where bits (7,6,4,0) are converted. Finally, Figure 3.10 shows the increase in critical path delay for each design option organized as in the previous tables. The standard Sbox delay was found to be 2.3 us. The delay increases by only 1.06x or 137ps for the option where bits (7,5,2) are converted to dual rail and by 1.12x or 291ps for the option where bits (3,0) are converted. Table 3.1, summarizes these impacts.

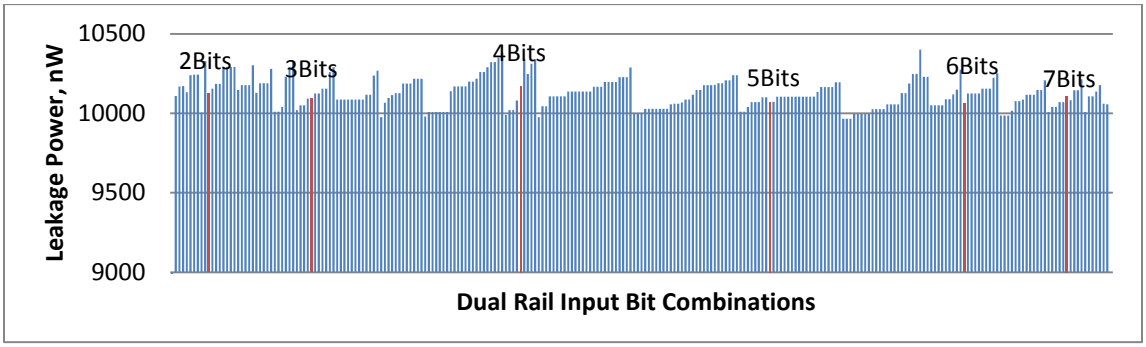


Figure 3.8. Leakage power for each possible input dual rail combination, sorted by number of bits converted to dual rail

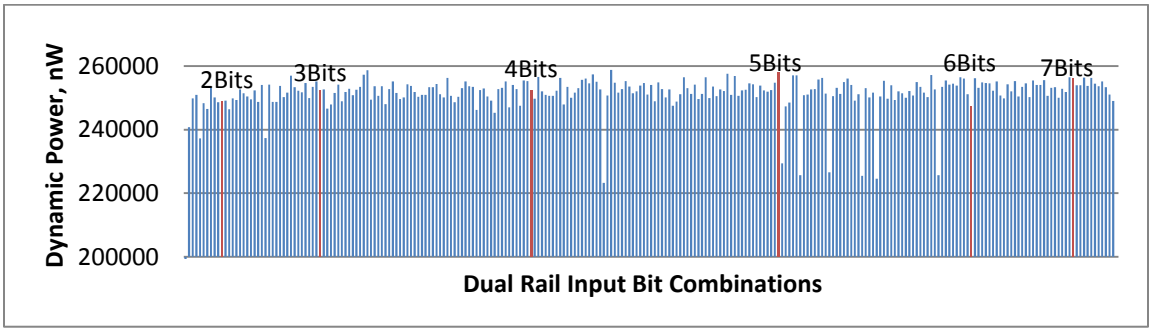


Figure 3.9. Dynamic power for each possible input dual rail combination, sorted by number of bits converted to dual rail

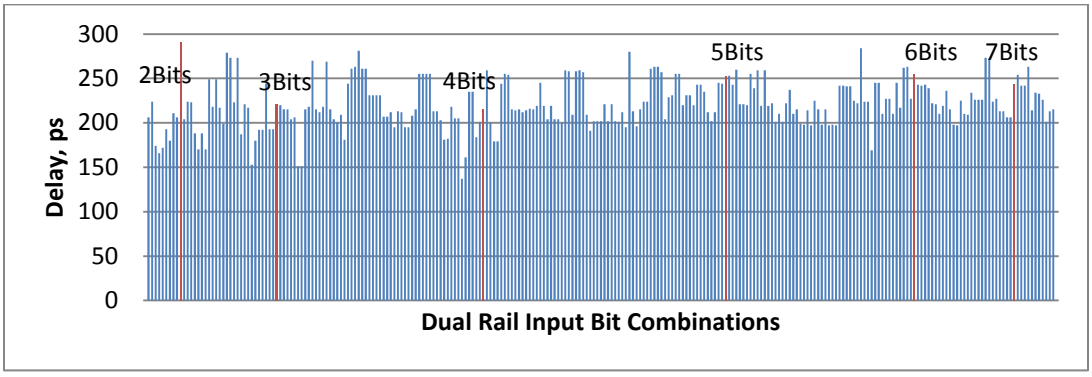


Figure 3.10. Increase in critical path delay for each possible input dual rail combination, sorted by number of bits converted to dual rail

Table 3.1 Summary of RECORD overhead impact

	Area, μm^2	Dynamic, nW	Leakage, nW	Delay, ps
Baseline	788	115169	4305	2311
RECORD Overhead	2.28x-2.33x	1.7x-2.24x	2.3x-2.34x	.06x-.12x

It is also worthwhile to point out that since the different dual rail options are all equally secure. The Quilt Packaging process and the pre-fabricated I/O modules allow the individual final chips to be assembled differently each time. For example, the first chip could have the first input bit converted and the next could have all input bits converted and so on. Thus preventing any reverse engineering attempts from being successful and greatly frustrating any potential attacker.

Figure 3.11 shows an example layout of the RECORD design with all inputs converted to dual-rail. We only demonstrate the portion that can be outsourced (i.e., the core in Figure 2.5). The layout of universal I/O elements is simple and not shown.

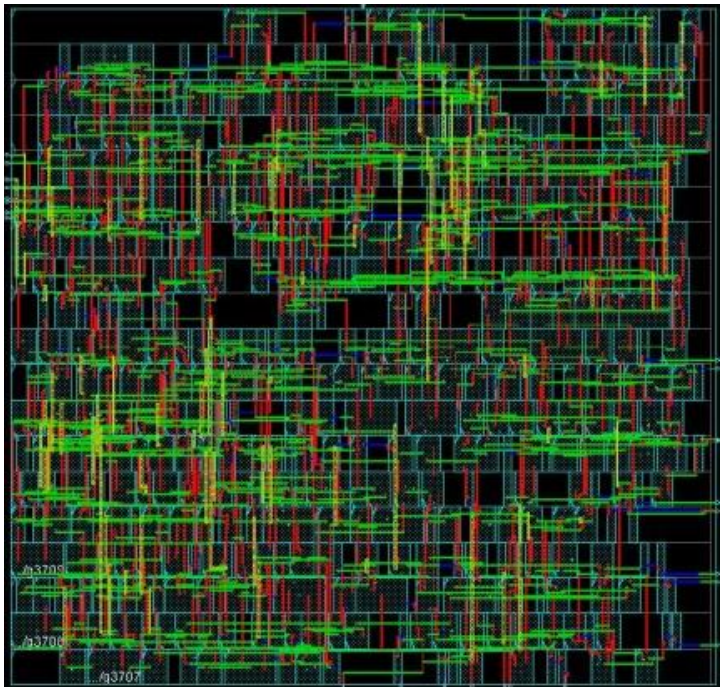


Figure 3.11. Layout of the RECORD design to be outsourced with all inputs converted to dual rail

4. CONCLUSIONS AND FUTURE WORK

A novel scheme was proposed to prevent attackers from interpreting leaked data from hardware Trojans by randomizing data within outsourced combinational logic and providing "secure" chip areas with the aid of Quilt Packaging. Simulation results on a 45 nm 8-bit Advanced Encryption Standard (AES) Substitution Box (Sbox) showed that RECORD can randomize information and separate the dual rail encoding/decoding with a 2.28x-2.33x increase in area, a 1.7x-2.24x increase in dynamic power and an 1.06x-1.12x increase in delay. This overhead can be considered necessary to achieve enhanced security in sensitive applications. To the best of the authors' knowledge, this is the first circuit-level technique that can resist data leakage after hardware Trojans are established. The technique is particularly suitable for secure designs with multiple fabrication runs [21].

Experiments were conducted in the AES design to demonstrate the impact of converting 1 or all 8 bits of the design to dual-rail, or any combination in between. For this logic circuit, there was negligible difference among the area, power usage, or delay added by any of the 256 possible designs.

REFERENCES

[1] A. Sreedhar, S. Kundu and I. Koren, "On Reliability Trojan Injection and Detection," J. Low Power Electronics , 674-683 (2012).

[2] R. Karri, J. Rajendran, K. Rosenfeld and M. Tehranipoor, "Trustworthy Hardware: Identifying and Classifying Hardware Trojans," Computer, 43, 10, 39-46, 2010.

[3] B. Swarup, M. S. Hsiao, M. Banga and S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," Proc. of the IEEE,102, 8,1229-1247, Aug. 2014.

[4] R. Dura, S. Hidalgo, R. Quijada, A. Raventos and T. Francesc, "The Use of Digital Image Processing for IC Reverse Engineering.," Multi-Conference on Systems, Signals & Devices (SSD), 2014 11th International, 1-4, Feb. 2014.

[5] C. Kamhoua, M. Rodriguez and K. Kwiat, "Testing for Hardware Trojans: A Game-Theoretic Approach," Decision and Game Theory for Security. Lecture Notes in Computer Science, 8840,360-369, 2014.

[6] S. Narasimhan and S. Bhunia, "Hardware Trojan Detection," Introduction to Hardware Security and Trust, 339-364, 2012.

[7] M. Rostami, F. Koushanfar and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," Proc. of the IEEE, 102, 8,1283-1295, 2014.

[8] J. Rajendran, M. Sam, O. Sinanoglu and R. Karri, "Security Analysis of Integrated Circuit Camouflaging.," Proc. Of the 2013 ACM SIGSAC Conference on Computer & Communications Security,709-720, 2013.

[9] J. Valamehr, T. Sherwook, R. Kastner, D. Marangoni-Simonsen, T. Huffmire, C. Irvine and T. Levin, "A 3-D Manufacturing Approach to Trustworthy System Development," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32, 4, 611-615, 2013.

[10] T. Schulze, Y. Shi, C. Kamhoua, K. Kwiat and S. Chang, "RECORD: Temporarily Randomized Encoding of Combinational Logic for Resistance to Data Leakage from Hardware Trojan," Asian HOST 2016, Yilan, Taiwan, Dec. 2016.

[11] G.H. Bernstein, Q. Liu, M. Yan, Z. Sun, D. Kopp, W. Porod, G. Snider and P. Fay, "Quilt Packaging: High-Density, High-Speed Interchip Communications," IEEE Transactions on Advanced Packaging, 30, 4, 731-740,2007.

[12] Indiana Integrated Circuits, Home Page, accessed June 2017 from http://www.indianaic.com/.

[13] P. Fay, D. Kopp, T. Lu, D. Neal, G. Bernstein and J. Kulick, "Ultrawide Bandwidth Chip-to-Chip Interconnects for III-V MMICs," IEEE Microwave and Wireless Components Letters, 24, 1, 29-31,2014.

[14] Q. Zheng, D. Kopp, M. Khan, P. Fay, A. M. Kriman and G. Bernstein, " Investigation of Quilt Packaging Interchip Interconnect With Solder Paste," IEEE Transactions on Components, Packaging and Manufacturing Technology, 4, 3, 400-407,2014.

[15] Eric Peeters, "Advanced DPA Theory and Practice Towards the Security Limits of Secure Embedded Circuits," Springer Science + Business Media, New York, NY, 2013.

[16] E. Brier, C. Clavier and F. Olivier, "Correlation Power Analysis with a Leakage Model," Lecture Notes in Computer Science, 3156, 16-29, 2004.

[17] H. Kim, H. Jeon, K. Lee and H. Lee, "The Design and Evaluation of All Voting Triple Modular Redundancy System," IEEE Proceedings Annual Reliability and Maintainability Symposium, 439-444, 2002.

[18] J. Wolkerstorfer, E. Oswald and M. Lamberger, "An ASIC Implementation of the AES SBoxes," Topics in Cryptology, UCT-RCA 2002, 29-52, 2002.

[19] W.A.A. Wijesinghe, M.K. Jayananda and D.U.J. Sonnandara, "Hardware Implementation of Random Number Generators'" Institute of Physics-Sri Lanka Proceedings of the Technical Sessions, 22, 26-36, 2006.

[20] https://www.eda.ncsu.edu/wiki/FreePDK, June 2017.

[21] D. Dilger. (2016, Jan 14). Apple A9 chip fab TSMC reports record earnings, casting further doubt on 'Peak iPhone'. Available: http://appleinsider.com/articles/16/01/14/apple-a9-chip-fab-tsmc-reports-record-earnings-casts-doubt-on-peak-iphone.

II. COMBATING DATA LEAKAGE TROJANS IN SEQUENTIAL CIRCUITS THROUGH RANDOMIZED ENCODING

Travis E. Schulze[*], Kevin Kwiat[+], Charles Kamhoua[+], Daryl Beetner[*], Laurent Njilla[+], and Yiyu Shi[Δ]
[*] Dept. of ECE, Missouri University of Science and Technology, Rolla, MO
[+] Air Force Research Lab., Information Directorate, Cyber Assurance Branch, Rome, NY
[Δ]Dept. of CSE, University of Notre Dame, Notre Dame, IN

ABSTRACT

Globalization of micro-chip fabrication has opened a new avenue of cyber-crime. It is now possible to insert hardware Trojans directly into the chip during the manufacturing process. These hardware Trojans are capable of destroying a chip, reducing performance or even capturing sensitive data. This paper presents a modification to a recently presented method of Trojan defense known as RECORD: Randomized Encoding of COmbinational Logic for Resistance to Data Leakage. RECORD aims to prevent data leakage through a randomized encoding and split manufacturing scheme. Its weakness, however, it that it is only applicable to combinational circuits. Sequential RECORD proposes a method to extend RECORD concepts to sequential designs. Experimental work with Sequential RECORD on a Data Encryption Standard circuit show that it is effective with the cost of a 3.75x area overhead, 4.5x power overhead and only a 3% decrease in performance.

1. INTRODUCTION

Since the turn of the century many IC design houses have outsourced the production of their chips to other countries [1]. This has created a new opening for cyber-attacks. When a firm sends out a sensitive design to be manufactured overseas, the trustworthiness of the manufactured IC can no longer be guaranteed. Hardware Trojans can be injected into the netlist or the layout altered to perform a variety of malicious activities. There are generally two types of hardware Trojans: reliability Trojans that stop the chip from functioning or data capture Trojans that capture the data being processed by the chip [2]. A reliability hardware Trojan is placed on chip with the goal of damaging the chip at some later, unexpected, date. For example, adding a simple circuit to increase power consumption [3] rendering the system un-usable in the field or a counter that counts down to switching the whole circuit off [4].

After fabrication, the Trojan circuit remains dormant during testing. The Trojan is usually very small relative to the remaining logic and therefore adds negligibly to the area and power consumption of the final circuit while dormant. Current methods of detecting hardware Trojans after production involve runtime monitoring and post-manufacture testing [5]. Runtime monitoring and post-manufacture testing rely on identifying the differences in chip operation introduced by the Trojan circuit. These methods depend on the tester's ability to trigger the Trojan circuit so the effects of the Trojan can be measured. The effects can be obvious, such as circuit malfunction, or subtle. Subtle changes to the chip's operation can sometimes be identified through side channels such as power, temperature, path delay or EM emissions. Runtime monitoring and post manufacture testing all rely on the golden chip concept. The golden chip

requirement is the Achilles heel of these methods. The only place this chip can come from is the same facility that produced the suspect chips in the first place. Simulations are not typically accurate enough to detect subtle changes in side channel measurements. Because of this problem design for security (DFS) [6] methods are more reliable.

For a data leakage Trojan to be successful the attacker needs to understand, at a minimum, the logic that is being monitored. This information can be gleaned from the netlist and layout or by obtaining a finished chip from a previous production run and reverse engineering it [7]. These types of Trojans can leak sensitive encryption keys out through Wi-Fi [8] or even through the power signature itself [9]. Current DFS methods try to prevent the attacker from understanding what the circuit is doing through obfuscation, layout camouflaging, or split manufacturing [10]. A new method, known as the RECORD [11], takes this technique one step further by preventing the attacker from understanding the meaning of logic signals.

Obfuscation [12], layout camouflaging [10,13] and split manufacturing[14] rely on the inability of the attacker to see what has been done to disguise the circuit. However, with enough time and effort obfuscation and layout camouflaging can be deciphered. Split manufacturing breaks up the design into lowest level silicon and upper level metals. The two pieces are manufactured separately preventing an attacker in one location from having access to a complete design. Split Manufacturing can be overcome by exploiting industry standards in floorplaning, placement and routing to alter one half of the split chip successfully [14], or reverse engineer a finished chip, which can be obtained through legitimate or illegitimate means, and inserting attacks into subsequent production runs which are often needed to meet demand [15].

RECORD is more robust than these techniques since it starts out with the assumption that the design is already known to the attacker. Through the use of a randomized dual-rail encoding and careful selection of split manufacturing the RECORD process prevents hardware Trojans from directly obtaining meaningful information from any part of the protected circuit.

Though RECORD is resistant to attacks even after the design is compromised, it is only defined for combinational logic. It is less than ideal for sequential logic designs as will be explored in Section 2. In a sequential circuit, the attacker has access to information not available in a combinational design: the information created in previous or upcoming clock cycles. As shown in subsection 2.2 this creates a vulnerability in the RECORD process by allowing attackers to infer the random encoding bit.

Sequential RECORD is a modification of the basic RECORD design that expands the dual rail encoding scheme to use two random bits that can change independently on each clock cycle. A change to the split manufacturing scheme that makes use of 3D IC technology is also proposed. These new techniques increase the difficulty to decode the dual rail encoding and allow the RECORD concept to be used successfully in a sequential design. The effectiveness of the Sequential RECORD method is demonstrated by implementing it in a Data Encryption Standard (DES) circuit. The technique requires a 3.75x area overhead and 4.5x power overhead and caused a 3% impact on performance.

The remainder of this paper is organized as follows: Section 2 presents a short review of the RECORD design process. Subsection 2.2 discusses the challenges of utilizing RECORD in sequential designs.

Section 3 describes the Sequential RECORD process in detail. Section 4 presents a design example using a Data Encryption Standard circuit. Conclusions are presented in Section 5.

2. BACKGROUND

2.1 BACKGROUND OF RECORD

With a RECORD design the opponent must still decipher the design as with all the DFS methods. However RECORD makes the assumption that this will be successful and adds additional protection which knowledge of the circuit design does not overcome.

A chip designed using RECORD uses a combination of split manufacturing and randomized dual-rail signaling to foil attackers. As seen in Figure 2.1, the outer ring is composed of prefabricated modules that perform the dual-rail conversion and generate a random signal. Since these sections are prefabricated (possibly years in advance) and later attached to the outsourced logic, they are inaccessible to an attacker at the time of fabrication. The individual chip wafers are joined using the Quilt Packaging process [16, 17].

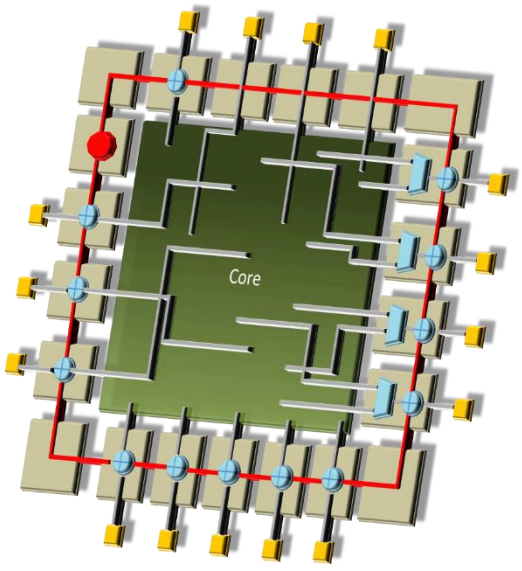


Figure 2.1. Block overview of a RECORD chip

Within these pre-fabricated I/O blocks the signals are converted to dual rail signals and back to single rail at the output. Conversion is done by XORing the input bit, *x*, with a randomly generated signal, *r*, which then becomes the second bit of the dual rail signal, *t*, as shown in (1).

$$t = x \oplus r \tag{1}$$

This bit, *r*, randomly changes each time the chip is activated. The dual rail signals represent a logic 0 as either 00 or 11. Logic 1 can be represented as either 01 or 10.

The two bits are represented by *t* & *r* in (1). The original input value, *x*, is dropped. Note that the second bit, *r*, of the dual-rail signal is always the random signal bit. This bit is never routed outside of the I/O blocks. Only the first bit, *t*, of the new dual-rail inputs is ever routed to the inner combinational logic see Figure 2.2.

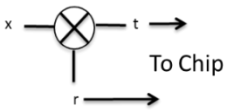


Figure 2.2. Conversion of input data to dual rail

The internal combinational logic needs to be altered to accommodate the dual-rail signaling. Any combinational function *f* can be converted to a dual rail function using (2).

$$f(x_1, x_2, x_3, \ldots) \rightarrow (f(t_1, t_2, t_3, \ldots) \oplus r) \tag{2}$$

Further manipulations using Boolean algebra show that a dual rail encoded function can be represented as in (3).

$$f(x_1 \oplus r, x_2 \oplus r, x_3 \oplus r, \ldots) \oplus r = rf(x_1 \oplus 1, x_2 \oplus 1, x_3 \oplus 1 \ldots) \oplus 1 + \quad \bar{r}f((x_1 \oplus 0, x_2 \oplus 0, x_3 \oplus 0 \ldots) \oplus 0 \tag{3}$$

$$= r\overline{f(\bar{t}_1, \bar{t}_2, \bar{t}_3 \ldots)} + \bar{r}f(t_1, t_2, t_3 \ldots) \tag{4}$$

The result of these manipulations is that the combinational function is duplicated and multiplexed with the random bit, *r*, used as a select signal. Each functional block is then sent only the first rail, *t*, of any dual-rail converted input bits. Never the second rail, *r*. This results in two outputs, *f* and *f'*, see Figure 2.3.

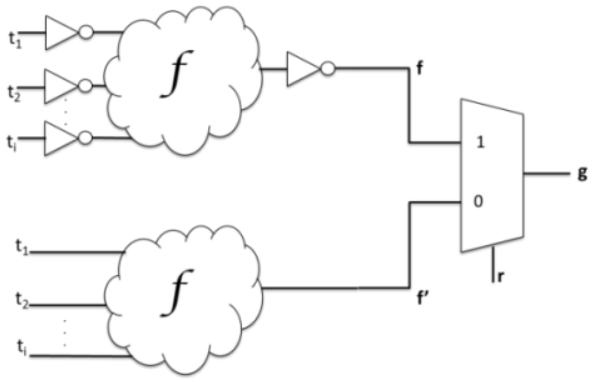


Figure 2.3. Conceptual diagram of RECORD circuit of logic function F

These two outputs are then routed back to the outer I/O modules where they are re-joined by the random bit, *r*. The two outputs are de- muxed using the random bit as a select signal to create output, *g*, which is itself in dual rail encoding with, *r*. Finally, *g* is converted back to single rail with another XOR gate before being routed off chip. The attacker is then presented with a design that is only ever routed with half of a randomly generated dual-rail data signal which is meaningless without the random bit. The random bit does not exist in the outsourced portion of the design. For a more detailed explanation of RECORD see [7].

2.2 MOTIVATION FOR SEQUENTIAL RECORD

The RECORD design scheme adds an unprecedented protection layer beyond traditional DFS, since it assumes that the attacker has already broken the first line of defense. The first line being simply understanding the design. After an attacker deciphers a RECORD design, they are presented with two outputs, *f* and *f'*, each of which is a dual rail encoded signal. The attacker does not have any way of determining which should be chosen since both combinational blocks are identical. Neither does the attacker know what the random bit value is, which is required to decode the final output. Additionally, these choices change randomly each time the circuit is activated and each chip need not be assembled identically.

If RECORD were used directly in a sequential design, however, the attacker would then have additional information: the two outputs, *f* and *f'*, and now a returning signal, *g*, which represents the output of a register. This signal, *g*, can be used to infer the

random bit value and decode the signals on any line. This concept is discussed in detail in Section 3.2.

Furthermore, RECORD does not discuss what is to be done with the random bit on each clock transition. Should it be stored or allowed to change? If stored, then control logic would be required. RECORD is a generic design scheme that can be used indiscriminately on any combinational design. It is desirable to maintain generality in Sequential RECORD. Control logic would destroy this generality.

Sequential RECORD expands on RECORD by allowing the random bit to change on each clock and by adding a second random bit to overcome the vulnerability introduced by the returning signal, g.

3. FRAMEWORK OF SEQUENTIAL RECORD

3.1 HANDLING THE RANDOM BIT CHANGES

RECORD does not discuss what to do with the random bit, *r*, on subsequent clock cycles. There are two options, store it or allow it to change independently. If the designer was to extend RECORD directly to sequential designs without altering the RECORD method, the entire sequential design would be duplicated (registers and combinational logic) just as the combinational function, *F*, in RECORD. Only one rail of the dual rail signal would be stored in the registers on each clock cycle. The random bit would then need to remain the same on the next cycle for the dual rail signal to still be valid. Storing the random bit also implies additional control logic.

Imagine a simple pipeline. Each set of data would have its own random bit which would need to be stored and then recalled at the appropriate time to decode the correct output from the pipeline. Not only does storing the random bit add area to the pre-fabricated I/O blocks but it also hinders the pre-fabrication process. The control logic would not be universal but would need to be customized to properly correlate to each circuit. The plug and play generality of RECORD would be lost.

Sequential RECORD allows the random bit to change independently and randomly on each clock and allows the I/O logic to remain simple and independent of the sequential logic. To accomplish this independence, when the random second rail changes, the data stored in the registers will be evaluated and updated to the new random bit on each clock cycle.

To simplify the design only the intermediate combinational logic, seen in Figure 3.1, is duplicated, as in RECORD. Then only a single, shared, set of registers is required. The outputs of the duplicated combinational logic blocks are de-muxed, see Figure 3.2.

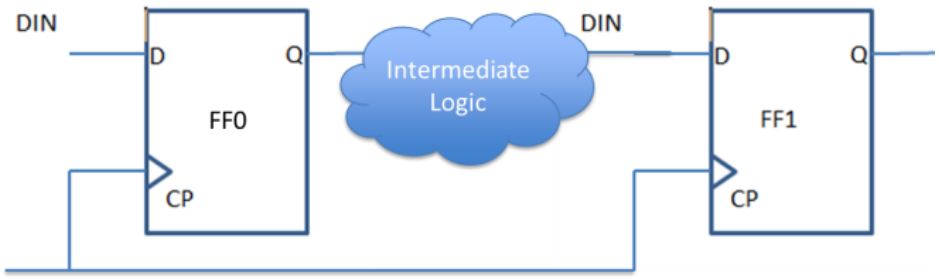


Figure 3.1. Standard un-altered sequential logic

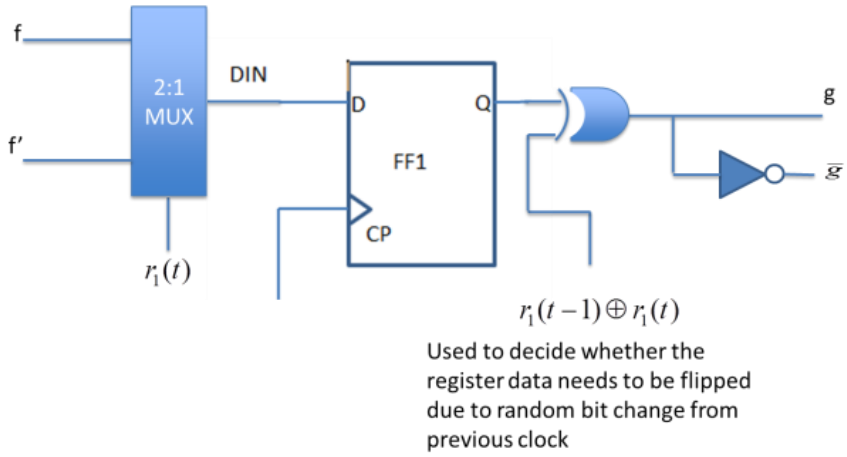


Figure 3.2. Logic needed to keep the dual rail signal in synch with the random bit

Demuxing is done on each clock instead of only at completion of the calculation. The de-muxed output is then stored in the shared set of registers. On the next clock the output of the register must be re-indexed using XOR with the update signal based on

whether the random bit has switched or not. This update signal is the XOR of the current clock cycle's random bit, *r(t),* and the previous clock cycles random bit, *r(t-1),* see (5).

$$update = r(t) \otimes r(t-1) \tag{5}$$

The update process requires one additional register added to the pre-fabricated I/O to store one random bit from the previous cycle.

If the design is laid out as specified in RECORD then each register would need to be prefabricated as an I/O block and quilted to the outsourced core. This would quickly overwhelm the Quilt Packaging process, since it is limited to the perimeter of the core. Imagine a core limited design as opposed to a pad limited design. In order to keep the random bit, *r*, secure, we need to modify the split manufacturing process. Instead of using Quilt Packaging to combine the secure and insecure portions of the chip, Sequential RECORD uses a secure top tier and 3D manufacturing to combine the two tiers using Through Silicon Vias (TSVs) [18]. The split manufacturing will be discussed further in Section 3.3.

3.2 A SECOND RANDOM BIT

The design can now be segregated as in RECORD keeping the random second rail confined to the upper tier of a 3D IC process. The attacker must now try to infer the random bit with the information available in the outsourced lower tier, as there would be no way to directly monitor it. If the design is encoded using a single random bit, as in

RECORD, an inserted Trojan can monitor the signals passed between tiers, *f, f'* and *g(t+1)*. It would then be possible to infer the random bit under certain conditions.

- If *f(t)=f'(t)*, and *g(t+1)=f(t)*, then *r* has not flipped from cycle *t* to cycle *t+1*
- If *f(t)=f'(t)*, and *g(t+1)=*$\overline{f'(t)}$, then *r* has flipped from cycle *t* to cycle *t+1*
- If *f(t)=* $\overline{f'(t)}$, and *g(t+1)=f(t)*, then *r(t+1)*=1
- If *f(t)=*$\overline{f'(t)}$, and *g(t+1)=* $\overline{f'(t)}$, then *r(t+1)*=0

Using these criteria an attacker could discover the random bit and keep track of its changes cycle to cycle. Once the random bit is discovered the entire scheme fails.

The solution is to use an additional random bit. Using the concepts presented in (2)-(4), the following solution can be obtained for adding a second random bit, r_2, to a function, f , see (6).

$$f(x_1, x_2, x_3 \ldots) \rightarrow f(x_1 \oplus r_1, x_2 \oplus r_2, x_3 \oplus r_1 \ldots) \oplus r_1 \tag{6}$$

$$\begin{aligned} &f(x_1 \oplus r_1, x_2 \oplus r_2, x_3 \oplus r_1, \ldots) \oplus r_1 = \\ &r_1 r_2 f(t_1, t_2, t_3, \ldots) + r_1 \bar{r}_2 f(t_1, \bar{t}_2, t_3, \ldots) \\ &+ \bar{r}_1 r_2 \overline{f(\bar{t}_1, t_2, \bar{t}_3, \ldots)} + \bar{r}_1 \bar{r}_2 \overline{f(\bar{t}_1, \bar{t}_2, \bar{t}_3, \ldots)} \end{aligned} \tag{7}$$

In this new implementation r_1 and r_2 are independently switching random numbers. As seen in (7) the two random bit implementation can be represented as four identical functions with different input vectors and outputs inverted, see Figure 3.3.

$$f = f(t_1, t_2, t_3, t_4 \ldots) = F_1$$

$$f' = f(t_1, \bar{t_2}, t_3, \bar{t_4} \ldots) = F_2$$

$$f'' = \overline{f(\bar{t_1}, t_2, \bar{t_3}, t_4 \ldots)} = F_3$$

$$f''' = \overline{f(\bar{t_1}, \bar{t_2}, \bar{t_3}, \bar{t_4} \ldots)} = F_4$$

Figure 3.3. Input vectors for each intermediate combinational block

Some of the inputs will refer to r_1 and some will refer to r_2, the exact combination can be changed but here we have set half to refer to r_1 and the other half to r_2. The intermediate outputs will refer to just one of the random bits. Here we have chosen r_1 but it could just as easily be r_2, as determined by the physical wiring of the 4-1 multiplexer. Figure 3.4 shows how (7) could be implemented in hardware. The combinational logic functions $F_{1\text{-}4}$, are de-muxed using the two random bits r_1 and r_2 as select signals before being stored in a register. This allows the stored signal to be re-indexed against a new random bit in the next clock cycle.

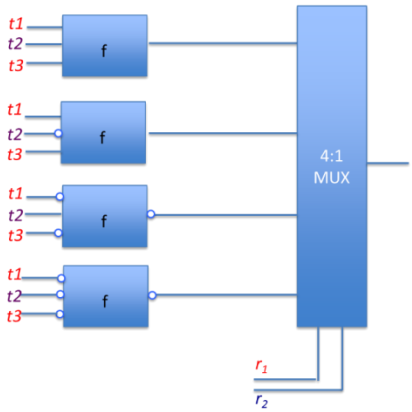


Figure 3.4. Implementation of two random bit Sequential RECORD

The new upper tier design is shown in Figure 3.5. To re-index each bit before it is sent back to the lower tier, the bit must be XOR'd with one of two update signals based on random bit $r_{1(t+1)}$ or $r_{2(t+1)}$. For example, if bit 5 of the input signal was paired with r_2, bit 5 of the intermediate output vector is then referenced to r_1, due to wiring of the multiplexer. When bit 5 is read out of the register on the next clock cycle it must be re-indexed since the random bit may or may not have changed. It may be re-indexed against either r_1 or r_2.

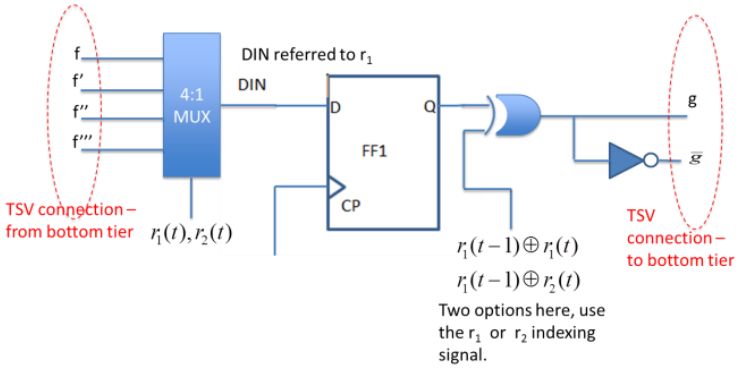


Figure 3.5. Upper tier for two random bit implementation

3.3 SPLIT MANUFACTURING

As discussed in Section 3.1, the split manufacturing for Sequential RECORD will be performed using a 3D process with a lower, outsourced, die and an upper, secure, die joined using TSV, see Figure 3.6. The upper tier will include all of the components previously included in the prefabricated blocks used in RECORD: random bit generator, dual-rail generating XOR gates, multiplexers and single rail generating XOR gates. The

upper tier will now also include all of the components found in Figure 3.5, including 4-1 multiplexers, registers and an additional random bit generator.

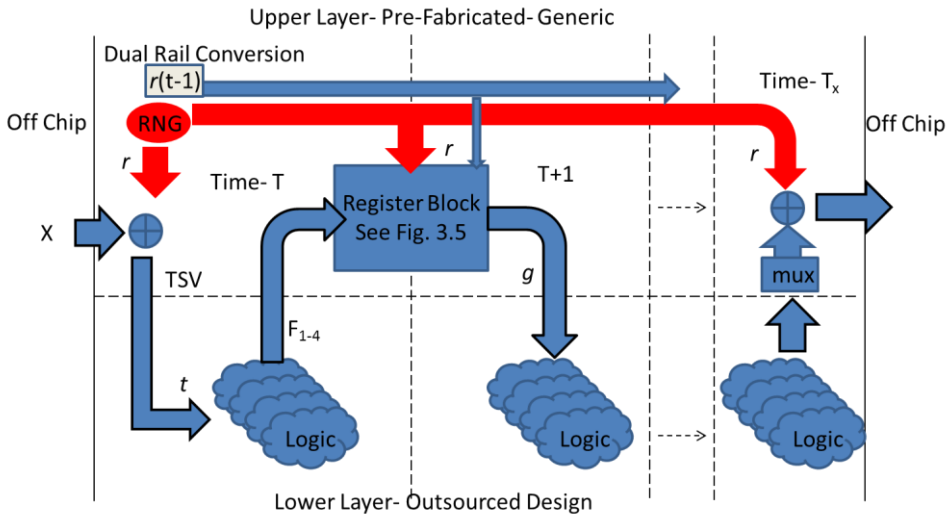


Figure 3.6. Conceptual data flow diagram for Sequential RECORD. Showing how data flows between the two layers

The upper tier can be prefabricated in an array structure as seen in Figure 3.7, where each block in the figure contains the components of Figure 3.5: the multiplexor, register and XOR gate needed for the dual rail design. Not all the top structures need be identical. Several variations could be used to obfuscate the design and frustrate the attacker. Possibilities include switching the dependence of the output bits from r_1 to r_2, or switching the dependence of the input bits.

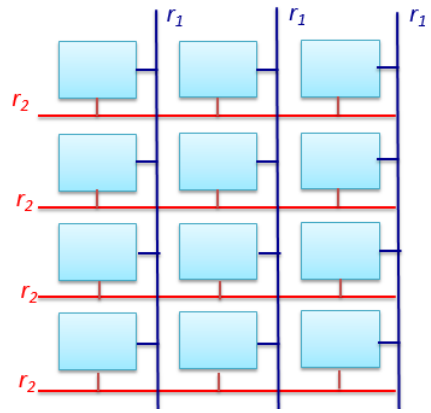


Figure 3.7. Array structure of pre-fabricated top tier

A possible layout for these standard register blocks is shown in Figure 3.8. This layout uses 45 nm technology standard cells. Each TSV is 1 µm in diameter and the total area of each block is 55.6 µm^2.

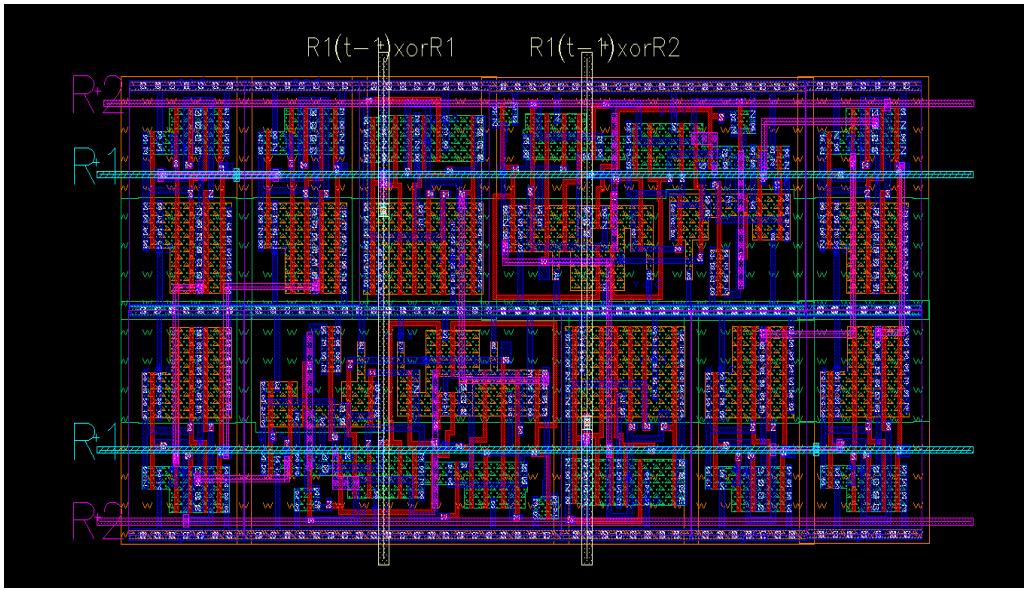


Figure 3.8. A possible layout for the upper tier register blocks used in Figure 3.7

3.4 VULNERABILITY ANALYSIS

Looking at the possible input/output scenarios using the same methodology described in Section 3.2 to determine if the random bit can be inferred, see Table 3.1. By including a second random bit the number of possible output combinations has quadrupled and reduced the number of scenarios for direct random bit inference to 25%. Table 3.2 shows similar results for outputs referred to r_2.

Table 3.1. Possible input/output combinations and potential for discovery of r_1 and r_2 when F_x output is referred to r_1

F_1	F_2	F_3	F_4	g	Conclusions	
0	0	0	0	0	$r_1(t+1) = r_1(t)$	Partially identifiable
0	0	0	0	1	$r_1(t+1) = \overline{r_1(t)}$	
0	0	1	1	0	$r_1(t+1) = 0$	
0	0	1	1	1	$r_1(t+1) = 1$	Fully identifiable
1	1	0	0	0	$r_1(t+1) = 1$	
1	1	0	0	1	$r_1(t+1) = 0$	
1	1	1	1	0	$r_1(t+1) = \overline{r_1(t)}$	Partially identifiable
1	1	1	1	1	$r_1(t+1) = r_1(t)$	

Table 3.2. Possible input/output combinations and potential for discovery of r_1 and r_2 when F_x output is referred to r_2

F_1	F_2	F_3	F_4	g	Conclusions	
0	0	0	0	0	$r_2(t+1) = r_1(t)$	Partially identifiable
0	0	0	0	1	$r_2(t+1) = \overline{r_1(t)}$	
0	0	1	1	0	$r_2(t+1) = 0$	
0	0	1	1	1	$r_2(t+1) = 1$	Fully identifiable
1	1	0	0	0	$r_2(t+1) = 1$	
1	1	0	0	1	$r_2(t+1) = 0$	
1	1	1	1	0	$r_2(t+1) = \overline{r_1(t)}$	Partially identifiable
1	1	1	1	1	$r_2(t+1) = r_1(t)$	

Using the scenarios in Tables 3.1 and 3.2, an attacker could determine the random bit in four scenarios. However, these tables assume several things. First the attacker must know which physical logic block is which, F_1, F_2, F_3 and F_4. As can be seen from the tables, the order matters. The attacker must also know which bit the intermediate outputs are being referenced to prior to being stored in the registers. This can be either r_1 or r_2, and is easily changed by re-wiring the select signals on the multiplexers. Finally the attacker must know which bit the returning signal $g(t+1)$ is referenced to.

Since the entire register block is located on a secure upper tier, each upper layer could be different. For each register block there are 96 different permutations of input wiring and output references. the same lower insecure wafer could be bonded to any number of different possible upper layers, indiscriminately. Conversely, the upper tiers can be used for any lower tier design allowing bulk manufacturing of the upper tier for cost savings.

A different inference table is needed for each possibility and there is no way for an attacker to know which table is needed for any given chip. The possible number of upper tier variations is limited only by the number of registers in the total design and the fabrication cost of an upper tier. Only a small number of upper tier variations need actually be manufactured. The attacker would not know which ones and would need to take all into account.

4. DESIGN RESULTS

To demonstrate the performance of Sequential RECORD, a 64 bit Data Encryption Standard (DES) design was implemented using this technique [19]. The baseline design was implemented in VHDL and synthesized using a 45 nm process with Cadence Encounter RTL Compiler v.13.10. The Sequential RECORD modifications were performed directly to the synthesized Verilog netlist of the baseline code. Encounter RTL Compiler provided the power and timing slack analysis. Both designs were then laid out using Cadence Encounter RTL to GDSII v.13.23, where the areas of the designs were measured.

The Sequential RECORD design method was synthesized and a full layout performed. The layout is shown in Figure 4.1. The area of the core was 3.75x larger than the baseline DES design. At first glance it might be assumed that the area overhead should be closer to four times, given the four times duplication of the logic blocks, however recall that the registers are not duplicated in the Sequential RECORD design process. The simulation results are summarized in Table 4.1. Dynamic power increased 4.5x, 3.1x for leakage and 4.5x overall. A timing analysis was performed at 31 MHz. The clock speed was chosen based on the clock speed given in the DES datasheet [19]. The slack was reduced by only 3% due to Sequential RECORD. This overhead may seem excessive but consider the use of triple modular redundancy (TMR) which carries 3x area overhead and is a common design method used to increase reliability [20]. The extra 0.75x area increase over TMR is a small price to pay for added security.

Also, note that the dual rail layout values include area for unnecessary multiplexors and XOR gates that occur resulting from pre-fabrication. Due to the nature

of the DES design there is a 64 bit input word that is held in the first bank of registers. These register values are not operated on in subsequent clock cycles and therefore do not need to be multiplexed and referenced back to a random bit as seen in Figure 3.5. The synthesizer recognized that the multiplexers and XOR gates were not used and deleted them. Since we are pre-fabricating the top layer these extra components would be present and would contribute area overhead. The area was manually added back into the numbers you see in the table.

Comparing Sequential RECORD to RECORD we see that Sequential RECORD increases the area overhead over baseline from 2.3x to 3.75x or 63%. Power overhead similarly increases from 2.8x to 4.5x or 61%.

Table 4.1. Power, Area and Slack Comparison for DES design and Sequential RECORD DES

	Dynamic Power	Leakage Power	Pre-Layout Area	Post Layout Area	Slack
Baseline DES	0.96 mW	0.044 mW	7,619 μm^2	9,662 μm^2	29.7 ns
Sequential RECORD DES	4.35 mW	0.138 mW	28,948 μm^2	36,285 μm^2	28.9 ns

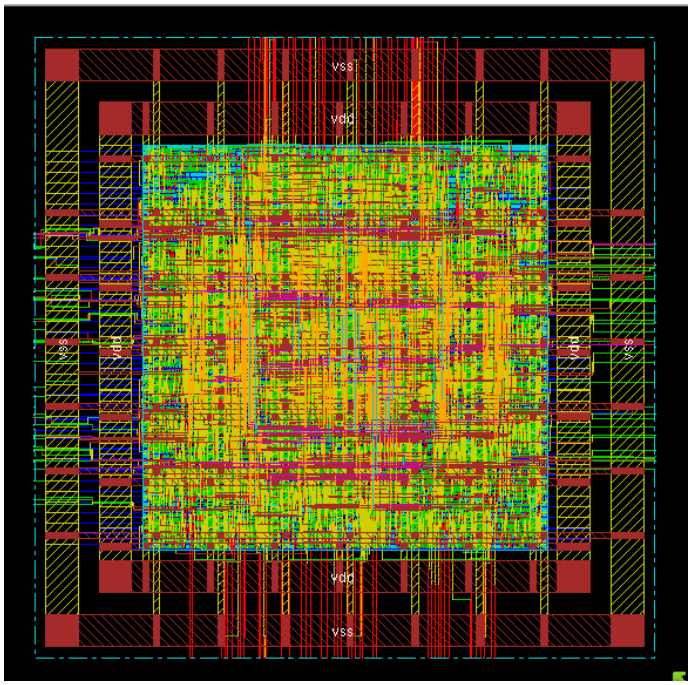


Figure 4.1. Layout of DES design utilizing Sequential RECORD process

5. CONCLUSION

We have presented Sequential RECORD, an extension to the RECORD design process to protect hardware from data leakage induced by insertion of hardware Trojans. Sequential RECORD extends the basic principles of RECORD, random dual-rail signaling and split manufacturing, to sequential designs. This is done with the addition of a second random bit and a move from Quilt Packaging to 3D manufacturing. Experimental design analysis utilizing a DES circuit as a baseline yielded a 3.75x area overhead and 4.5x power overhead. The timing slack was reduced only 3% from baseline by the addition of the Sequential RECORD components. Sequential RECORD maintains the random signaling of RECORD and its generic plug and play design method. Sequential RECORD also greatly increases the number of design variations thereby making decoding of a Sequential RECORD circuit difficult.

REFERENCES

[1] “China just surpassed USA in semiconductor manufacturing-and the trend is likely to continue” http://qz.com/72542/china-just-surpassed-the-us-in-semiconductor-manufacturing-and-the-trend-is-likely-to-accelerate/, Sep. 2013.

[2] R. Karri, J. Rajendran, K. Rosenfeld and M. Tehranipoor, “Trustworthy Hardware: Identifying and Classifying hardware Trojans,” Computer, vol. 43, no. 10, pp. 39-46, 2010.

[3] Jin, Y., Makris, Y, “Hardware Trojan detection using path delay fingerprint,” IEEE Int. Workshop on Hardware-Oriented Security and Trust, HOST 2008, pp. 51–57, 9 June 2008.

[4] Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B., “Trojan detection using IC fingerprinting,” IEEE Symp. on Security and Privacy, IEEE Computer Society, pp. 296–310, 2007.

[5] C. Kamhoua, M. Rodriguez and K. Kwiat, “Testing for hardware Trojans: A Game-Theoretic Approach. Decision and Game Theory for Security”, Lecture Notes in Computer Science, vol. 8840, pp. 360-369, 2014.

[6] S. Narasimhan and S. Bhunia, “Hardware Trojan Detection,” Introduction to Hardware Security and Trust, pp.339-364, 2012.

[7] R. Dura, S. Hidalgo, R. Quijada, A. Raventos and T. Francesc, “The Use of Digital Image Processing for IC Reverse Engineering,” Multi-Conference on Systems, Signals & Devices (SSD), 2014 11th International, pp. 1-4, Feb. 2014.

[8] Y. Jin and Y. Makris, “hardware Trojans in Wireless Cryptographic ICs,” IEEE Design & Test of Computers, pp. 26-35,Jan/Feb 2010,.

[9] Lin L., Kasper M., Güneysu T., Paar C., Burleson W., “Trojan Side-Channels: Lightweight hardware Trojans through Side-Channel Engineering,” Cryptographic Hardware and Embedded Systems - CHES 2009. Lecture Notes in Computer Science, vol. 5747, 2009.

[10]M. Rostami, F. Koushanfar and R. Karri, “A Primer on Hardware Security: Models, Methods, and Metrics,” Proc. of the IEEE, vol. 102, no. 8, pp.1283-1295, Aug. 2014.

[11]T. Schulze, Y. Shi, C. Kamhoua, K. Kwiat and S. Chang, “RECORD: Temporarily Randomized Encoding of Combinational Logic for Resistance to Data Leakage from Hardware Trojan,” Asian HOST 2016, Yilan, Taiwan, Dec. 2016.

[12] Chakraborty, R.S., Bhunia, S., "Security against hardware Trojan through a novel application of design obfuscation," Proc. of the 2009 Int. Conf. on Computer-Aided Design, ICCAD'09, New York, NY, USA, pp. 113–116, 2009.

[13] J. Rajendran, M. Sam, O. Sinanoglu and R. Karri, "Security Analysis of Integrated Circuit Camouflaging," Proc. Of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp.709-720, 2013.

[14] J. Rajendran, O. Sinanoglu and R. Karri, "Is Split Manufacturing Secure?," Proceedings of the Conference on Design, Automation and Test in Europe, DATE '13, pp. 1259-1264, 2013.

[15] D. Dilger. (2016, Jan 14). *Apple A9 chip fab TSMC reports record earnings, casting further doubt on 'Peak iPhone'*. Available: http://appleinsider.com/articles/16/01/14/apple-a9-chip-fab-tsmc-reports-record-earnings-casts-doubt-on-peak-iphone.

[16] G.H. Bernstein, Q. Liu, M. Yan, Z. Sun, D. Kopp, W. Porod, G. Snider and P. Fay, "Quilt Packaging: High- Density, High-Speed Interchip Communications," IEEE Transactions on Advanced Packaging, 30, 4, pp. 731-740, 2007.

[17] Indiana Integrated Circuits, Home Page, accessed July 17, 2015 from http://www.indianaic.com/.

[18] Motoyoshi M., "Through-Silicon Vias (TSV)," Proceedings IEEE, 30, 1, pp.43-48, Jan. 2009.

[19] TCDG, TETRAEDRE S.A.R.L, white paper, 1999, Used with permission.

[20] H. Kim, H. Jeon, K. Lee and H. Lee, "The Design and Evaluation of All Voting Triple Modular Redundancy System," IEEE Proceedings Annual Reliability and Maintainability Symposium, pp.439-444, 2009.

III. COMBATING DATA LEAKAGE TROJANS IN COMMERCIAL AND ASIC APPLICATIONS WITH TIME DIVISION MULTIPLEXING AND RANDOM ENCODING

Travis E. Schulze[*], Daryl Beetner[*], Kevin Kwiat[+], Charles Kamhoua[+], and Yiyu Shi[Δ]
[*] Dept. of ECE, Missouri University of Science and Technology, Rolla, MO, USA, 65401
[+] Air Force Research Lab., Information Directorate, Cyber Assurance Branch, Rome, NY, USA, 13441
[Δ]Dept. of CSE, University of Notre Dame, Notre Dame, IN, USA, 46556

ABSTRACT

Globalization of micro-chip fabrication has opened a new avenue of cyber-crime. It is now possible to insert hardware Trojans directly into a chip during the manufacturing process. To date, defensive methods have focused on detection of the Trojan circuitry or prevention through design for security methods. One recent DFS method has attempted to shift the focus from prevention and detection by creating ASICs that are inherently secure and require no testing to detect Trojans. Randomized Encoding of Combinational Logic for Resistance to Data Leakage (RECORD) and its Sequential logic variant, present processes that can prevent hardware Trojans from leaking meaningful information even when the entire design is known to the attacker. Both of these methods have significant area and power overhead, apx. 4x area and 4.5x power for the sequential version. In this paper, the fundamental ideas of RECORD are re-imagined to create a Time Division Multiplexed version of the RECORD design process which reduces area overhead by 63% and power by 56%. This TDM concept is further refined to allow Commercial Off The Shelf (COTS) products and IP cores to be safely operated from a separate chip. These new methods trade off latency (5.3x for TDM and 3.9x for COTS) and energy use to accomplish the area and power savings and achieve greater security than the original RECORD process.

1. INTRODUCTION

Detection and prevention of hardware Trojan attacks has become a major concern for any company wishing to outsource its hardware manufacturing. The threat that a hardware Trojan could lie dormant on a new chip for years before crippling it, is a very real one. These Trojans could also steal encryption keys, passwords or other sensitive information, compromising the entire enterprise. These data leakage Trojans can leak secret information out through Wi-Fi [1] or even through the power signature itself [2]. Detecting hardware Trojans has proven very difficult if not impossible. Some Trojans have been demonstrated to successfully operate with as little as five transistors [3]. Most current methods of defense would be ineffective against such an attack. Of course this problem extends beyond custom ASIC designs and the firms creating them. Most companies buy commercial chips and are in no way involved in the design and production processes. Protecting a Commercial Off The Shelf (COTS) chip has so far been a mostly un-explored area of hardware Trojan defense methods.

Current methods of detecting hardware Trojans after production involve runtime monitoring and post-manufacture testing [4]. Runtime monitoring and post-manufacture testing rely on identifying the differences in chip operation introduced by the Trojan circuit. These methods depend on the tester's ability to trigger the Trojan circuit so the effects of the Trojan can be measured. In order for runtime monitoring and post-manufacture testing to work a golden chip is usually needed so that there is something to compare the manufactured chip to. Simulation of the chip functionality is generally not accurate enough to show the very small changes introduced by the additional Trojan circuitry. How the golden chip is to be produced is left to the reader. Neither of these

broad categories would be relevant for a third party company buying the chip commercially.

A more reliable method of protection is to design the chip for security from the beginning. The concept of Design For Security (DFS) encompasses many possible ideas and methods; Obfuscation[5,6], Layout Camouflaging[5,7], and Split Manufacturing[5,8] for example. However, obfuscation and layout camouflaging are both susceptible to reverse engineering [9] given enough time and effort. Split manufacturing is susceptible if multiple fabrication runs are used, which is often the case [10]. This paper focuses on a new DFS method known as RECORD (Randomized Encoding of Combinational Logic for Resistance to Data Leakage) [11] and more specifically the Sequential RECORD variant [12] which can protect sensitive information from being leaked even when the full design is known to the attacker and multiple fabrication runs are needed.

Sequential RECORD took the initial concept of RECORD, which was only specified for combinational logic, and extended it to sequential circuits. The general idea uses two randomly generated numbers to temporarily encode incoming data into a dual rail signal. Wherein, the random numbers represent one of the rails. Through Boolean manipulation combined with split manufacturing, the Sequential RECORD process is able to effectively prevent any data leakage Trojans from capturing meaningful data from anywhere on the chip. This process does not try to detect or even prevent hardware Trojans on the chip. There is no longer any need. Any data captured by the attacks would be meaningless.

The Sequential RECORD process suffers from two drawbacks. First the RECORD algorithm increases the area by almost 4x and the power by about 4.5x. And

since it is a DFS method it must be designed into the chip from the start making it useless for COTS applications. In this paper two modifications to the RECORD process are presented. The first, Time Division Multiplexing (TDM) RECORD, will show how the RECORD process can be modified to reduce the area and power overhead by 63% and 56%, respectively, at the expense of only processing time and total energy used. Secondly, a scheme to use the RECORD concepts off chip to allow safe operation of COTS products is presented. This method is shown to work effectively without modification of the COTS product and can be implemented on an FPGA or other processor which may already be present in the design with approximately 4x increase in processing time.

The remainder of this paper is laid out as follows. Section 2 includes a discussion of the Sequential RECORD design process. Section 3 describes the TDM RECORD modified process. Section 4 shows how RECORD concepts can be used to safely operate infected COTS chips. Section 5 will show the experimental results and Section 6 will conclude.

2. BACKGROUND OF SEQUENTIAL RECORD

The Sequential RECORD design process is itself a modification of the initial RECORD design process which allows the concepts to be used successfully on sequential designs. For a full discussion of the RECORD process see [11].

The first step in the Sequential RECORD process is to create dual rail representations of the incoming data vectors. The conversion is done by XORing the input bits, *x*, with one of two randomly generated signals, r_1 & r_2, which then becomes the second bit of the dual rail signal,*t*, as shown in (1). The exact order of which bit is XOR'd with which random signal is up to the designer and can be changed chip to chip.

$$t = x \oplus r_1 \text{ or } t = x \oplus r_2 \tag{1}$$

The bits ,r_1 & r_2, randomly change on each clock. The dual rail signals represent logic 0 as either 00 or 11. Logic 1 can be represented as either 01 or 10. The two bits are represented by *t* & *r* in (1). The original input value, *x*, is dropped. Note that the second bit, *r*, of the dual-rail signal is always the random signal bit. This bit is never routed to the sequential logic. Only the first bit, *t*, of the new dual-rail inputs is ever routed to the inner combinational logic see Figure 2.1.

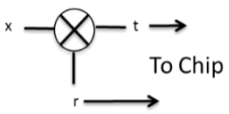


Figure 2.1. Conversion of input data to dual rail

The new input, *t*, vectors are routed to the logic according to the following equations. Equation 2 shows how the original function *f* can be converted to dual rail with two random bits, r_1 & r_2. Equation 3, shows how the concept can be implemented in real hardware after some simple Boolean manipulation.

$$f(x_1, x_2, x_3 \ldots) \rightarrow f(x_1 \oplus r_1, x_2 \oplus r_2, x_3 \oplus r_1 \ldots) \oplus r_1 \tag{2}$$

$$\begin{aligned} & f(\mathrm{x}_1 \oplus r_1, \mathrm{x}_2 \oplus r_2, \mathrm{x}_3 \oplus r_1, \ldots) \oplus r_1 = \\ & r_1 r_2 f(\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3, \ldots) + r_1 \bar{r}_2 f(\mathrm{t}_1, \bar{t}_2, \mathrm{t}_3, \ldots) \\ & + \bar{r}_1 r_2 \overline{f(\bar{t}_1, \mathrm{t}_2, \bar{t}_3, \ldots)} + \bar{r}_1 \bar{r}_2 \overline{f(\bar{t}_1, \bar{t}_2, \bar{t}_3, \ldots)} \end{aligned} \tag{3}$$

Equation 3, describes four identical functions each being sent a slightly altered version of the same input vector. The two random bits are used as select signals to demux the four outputs and select the correct output. Note that the output selected by the de-muxing process is still in dual rail representation, with r_1 in this case. The second random bit could be used instead with simple reworking of the equations. Figure 2.2 shows what these equations would look like in hardware.

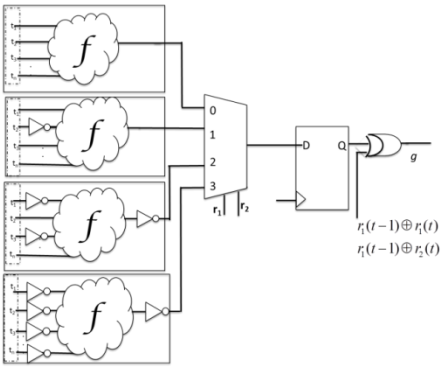


Figure 2.2. Implementation of two random bit Sequential RECORD

Since the random bits are allows to change independently on each clock cycle a method to keep track of the changes and update the, *t*, signals accordingly is needed. First, it can be seen in Figure 2.2 that the registers are not duplicated only the intermediate logic. The intermediate logic ouputs are de-muxed according to (3) and the output stored as normal in the register. On the next clock cycle the output is read out and the re-indexed against one of the two random bit signals. Since the output of the register is already in a dual rail representation with r_1, the $r_1(t\text{-}1)$ signal is needed to create an update signal, (4). The new value, *g*, can be indexed against either random bit at the designers discretion before being sent on to the next set of four intermediate logic blocks, see Figure 2.3.

$$update = r(t) \otimes r(t-1) \tag{4}$$

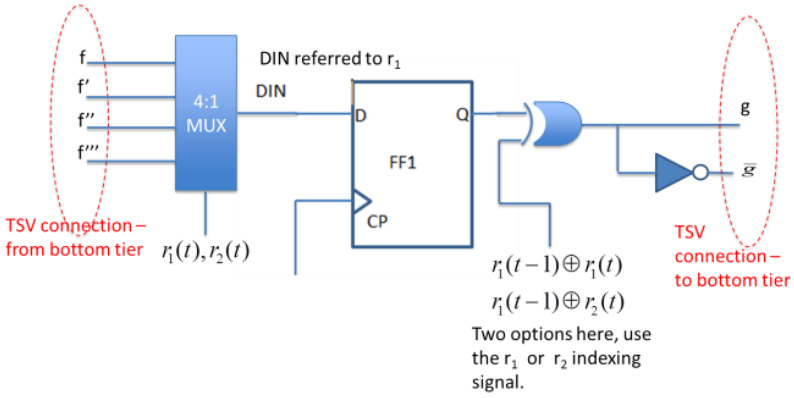


Figure 2.3. Circuitry needed for each register to update the random bit changes

The last major step in the Sequential RECORD process is to segregate the design into an upper and lower tier for 3D manufacturing. This is done to keep the random bits secret and to allow for multiple variations of the same chip to be produced simultaneously. The contents of Figure 2.3, are placed on the upper tier along with the random bit generator/s. Figure 2.3 can now be referred to as a register block which can be laid out in an array structure, Figure 2.4. The upper tier is then connected with Through Silicon Vias (TSV) to the lower tier. The lower tier contains only the four copies of all the intermediate logic. This lower tier can now be outsourced. The upper tier is generic. These register blocks are compatible with any design as long as there are enough registers. The upper tier should be pre-fabricated so that the exact logic it will be bonded with is unknown and it should be done in a secure environment or at the very least a totally unaffiliated facility than the lower tier. Not all the top structures need be identical. Several variations could be used to obfuscate the design and frustrate the attacker. Possibilities include switching the dependence of the output bits from r_1 to r_2, or switching the dependence of the input bits.

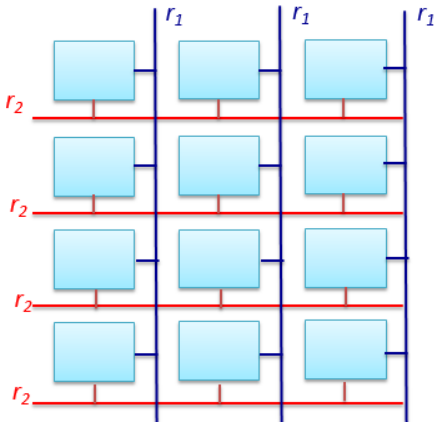


Figure 2.4. Array structure of pre-fabricated top tier

The final fully assembled design is represented by Figure 2.5, which shows the data flow from chip input, between the layers, and the final output routed off chip.

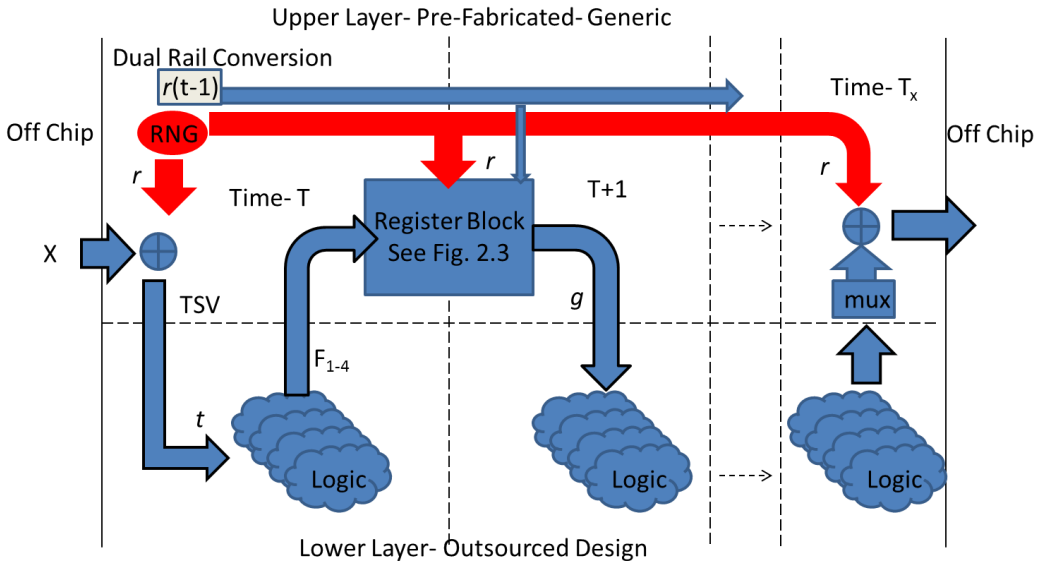


Figure 2.5. Conceptual data flow diagram for Sequential RECORD. Showing how data flows between the two layers

A design prepared using the Sequential RECORD method resists data leakage very effectively, however there are a very small percentage of cases in which an attacker could de-code the design with a Trojan placed on the lower tier. Table 2.1 shows the possible combinations of outputs from the lower tier (F1, F2, F3 and F4) and the returning input to the lower tier, *g*, which would allow the attacker to infer the random bit. Note these five signals are the only signals an attacker would have access to on the lower tier. This equates to 25% of cases in which the random bit could be inferred. However this is not as simple as it first appears. First the attacker must know which

physical logic block is which, F1, F2, F3 and F4. As can be seen from the table, the order matters. The attacker must also know which bit the intermediate outputs are being referenced to prior to being stored in the registers. This can be either r_1 or r_2, and is easily changed by re-wiring the select signals on the multiplexers. Finally the attacker must know which bit the returning signal g(t+1) is referenced to.

Table 2.1. Possible input/output combinations and potential for discovery of r_1 and r_2 when Fx output is referred to r_1

F_1	F_2	F_3	F_4	g	Conclusions	
0	0	0	0	0	$r_1(t+1) = r_1(t)$	Partially identifiable
0	0	0	0	1	$r_1(t+1) = \overline{r_1(t)}$	
0	0	1	1	0	$r_1(t+1) = 0$	
0	0	1	1	1	$r_1(t+1) = 1$	Fully identifiable
1	1	0	0	0	$r_1(t+1) = 1$	
1	1	0	0	1	$r_1(t+1) = 0$	
1	1	1	1	0	$r_1(t+1) = \overline{r_1(t)}$	Partially identifiable
1	1	1	1	1	$r_1(t+1) = r_1(t)$	

Since the entire register block is located on a secure upper tier, the upper layer should be designed to take advantage of the different possibilities. For each register block there are 96 different permutations of input wiring and output references. The same lower, insecure, wafer could be bonded to any number of different possible upper layers, indiscriminately. Additionally, the upper tiers can be used for any lower tier design allowing bulk manufacturing of the upper tier for cost savings. A different inference table is needed for each possibility and there is no way for an attacker to know which table is needed for any given chip. The possible number of upper tier variations is limited only by the number of registers in the total design and the fabrication cost of an upper

tier. Furthermore, only a small number of upper tier variations need actually be manufactured. The attacker would not know which ones had been used and would need to take all into account. For a much more thorough explanation of Sequential RECORD please refer to [12].

3. TIME DIVISION MULTIPLEXING

RECORD provides excellent protection for custom ASIC designs both for combinational or sequential circuits. It does however still carry large area and power overhead, approximately 3.75x area and 4.5x power for Sequential RECORD. RECORD also benefits from a very simple and generic implementation that can be easily adapted to any design with no circuit specific logic or control. Consequently, when area or power is at a premium a different approach is needed. Time Division Multiplexing offers a perfect solution to reduce the power and area overhead. The Sequential RECORD process contains four copies of the same logic. This serves two purposes; it allows for quick parallel processing of the four input vectors and helps to confuse attackers. TDM RECORD eliminates the duplication while at the same time opening up possibilities for an even more secure protection scheme.

The intermediate combinational logic is copied four times and operated on in parallel. The only difference between the blocks is that each block of logic is sent a different input vector. In TDM RECORD, the four variant input combinations can be sent in sequence to just one copy of the intermediate logic instead of four. The outputs of the intermediate logic are then stored in one of four register blocks before being de-muxed to determine the final correct output.

The register blocks are not the same as in Sequential RECORD. In Sequential RECORD each register stores the de-muxed output of the four input vectors before being re-indexed to the random bit on the next clock, Figure 2.3. In TDM RECORD all four outputs are not available in the same clock cycle. It now takes four periods to accumulate all four outputs. During this time the random bit would be changing on each clock. After

the first output is ready the subsequent outputs would be indexed against different random bits.

Even if all the different random bits were stored for reference the de-muxing process would become invalid. In order to overcome this issue, a slight modification of the process is needed. Now each intermediate output vector, $g_{1\text{-}4}$, will be stored until all four input vectors have completed. Then the four stored outputs will be de-muxed with the two random bits as select signals, as before. The new output will then be re-indexed against two newly generated random bits before being sent on to the next set of intermediate logic, Figure 3.1. This means that the random bits must be stored for four clocks cycles instead of changing on each cycle. It also means that control logic is needed to orchestrate when new random bits are needed and into which register block the intermediate outputs, $g_{1\text{-}4}$, are placed.

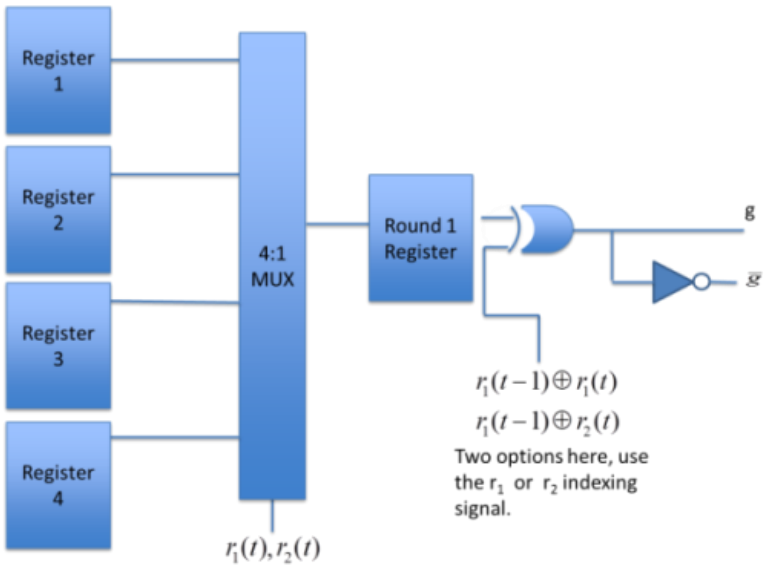


Figure 3.1. Registers for each round of TDM RECORD and the re-indexing logic

The entire process is illustrated in Figure 3.2. At time, T, the incoming data vector is converted to dual rail representation with r_1 and r_2. The new input bits, *t*, are then sent to the first logic block. The output is then stored in register block 1. In time, T+1, the same input bits, t, are sent back to the same logic block but with some of the inputs inverted as described in (3). The outputs are stored in register block 2. This process is repeated two more times with the input bits, t, being inverted or not inverted according to (3). Finally in T+4, the register blocks are de-muxed using r_1 and r_2 as select signals. Here r_1 and r_2 have not been allowed to change from cycle to cycle but have been stored since, T. Then, in T+5 the random bits are allowed to change, the output of the register block is updated and the process begins again.

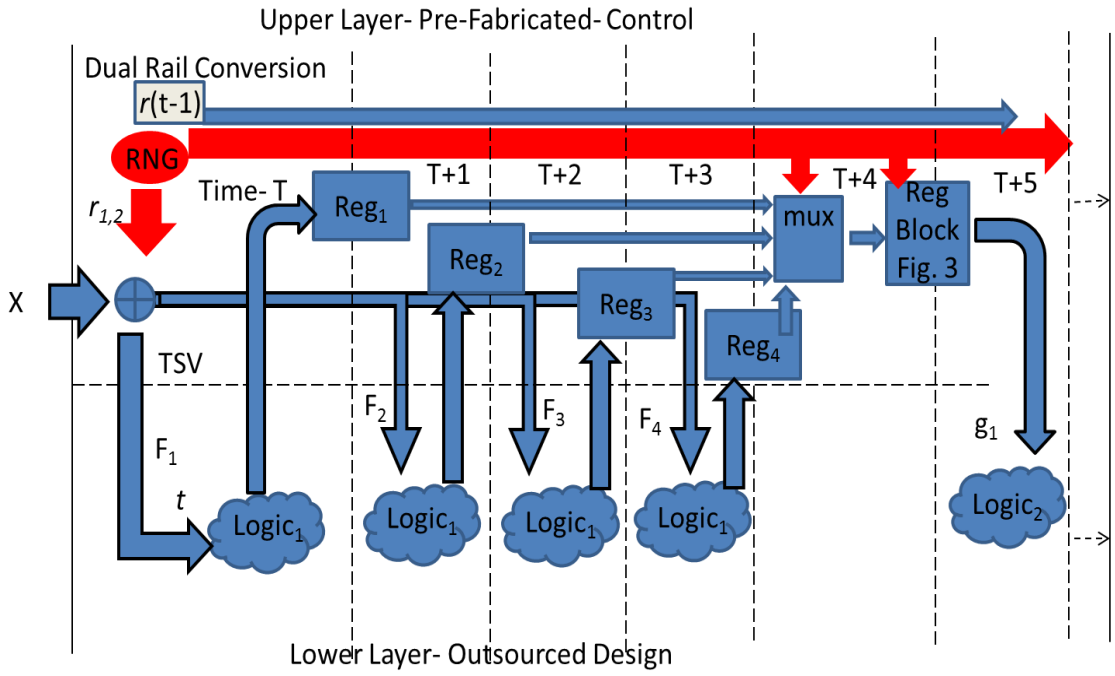


Figure 3.2. Conceptual data flow diagram for TDM RECORD process. Showing the first round of data processing

The security of the random bits is maintained with split manufacturing. The upper tier would contain the control logic, the input multiplexers, registers with XOR gates, output registers and final de-mux and the bottom tier contains all of the intermediate logic. Storing the random bits for four clock cycles does nothing to compromise the security. An attacker would still need to wait until all four input data vectors have processed before attempting to infer the random bit, at which time the bit would change just as in Sequential RECORD.

The complexity of the control logic is determined by the complexity of the application but it does eliminate the generic application of RECORD and Sequential RECORD. While at the same time opening up two very important options. First, Sequential RECORD was resistant to attackers inferring the random bits by reading the signals moving between upper and lower tiers. However, remote possibilities still existed if the attacker knew which combinational block was receiving which input vector, which input bits were indexed against which random bits and finally the indexing of the returning register outputs. By making use of the random number generator/s the control logic can randomly change the order in which the input vectors are presented to the logic. For example, the first iteration could present the input vectors in normal order (1,2,3,4) but the next iteration could be (2,3,1,4) and so on. By doing this the inference table found in Table 2.1 is rendered useless.

Secondly to further frustrate attackers the number of random bits can be increased. Adding a third random bit was previously infeasible due to the excessive area and power overhead. Now, that third random bit would increase the number of input vector variations to eight but at a very small cost to additional area and power. Only four

additional register blocks and alterations to the control logic are needed. At the same time greatly increasing the number of possible dual rail input permutations and upper tier permutations.

The cost of the TDM RECORD process is time. Latency is increased by a minimum of 5x or 9x in the case of the three random bit option. Design time also increases, the TDM RECORD concept must be designed into the chip from the beginning and custom control is needed for each design. Finally, while power does decrease compared to Sequential RECORD overall energy usage increases.

4. COTS APPLICATION

RECORD in all its variations provides excellent protection against hardware Trojan attacks. However, they are applicable only to custom ASIC chips. Quilt-Packaging [13], used in the original RECORD process, and 3D manufacturing are not cheap. The cost of manufacturing a RECORD or Sequential RECORD design would be many times the cost of a standard chip. This may by acceptable in certain cases but may also be cost prohibitive. Many applications, even military ones, are moving to COTS chips to meet specifications and budget. These chips are even more susceptible to hardware Trojan injection since there may not be any hardware security plan in mind when they are designed. A recent paper [14] has identified commercial parts as the number one area needing research and a solution for preventing hardware Trojans attacks. The TDM RECORD process presents an opportunity to do just that.

TDM RECORD requires a custom ASIC solutions since the design must be segregated into two tiers for the 3D manufacturing process and the internal registers must be moved to the upper tier. This of course will not work for a COTS chip which is already manufactured. A way is needed to interface with the COTS chip in such a way that any data leakage Trojans already present on the chips will be ineffective. The solution is to move the TDM RECORD control logic to a separate chip, such as an FPGA.

The registers on the COTS chip cannot be moved nor can they be individually accessed in most cases. Consequently the TDM process of injecting the appropriate data vectors into each intermediate logic block is no longer applicable. The second chip of the COTS RECORD process must now send each of the four input data vectors to the COTS

chip and wait for the chip to fully complete its operation, Figure 4.1. In the case of a DES chip, this would take sixteen cycles to produce a final encoded output. After which that output would be stored in the control chip and the next of four input vectors would be sent. While this is going on, only one set of random bits is used and stored on the control chip. There is no need to ever send them to the COTS chip so they are safe from detection. When all four vectors have been processed, the control chip will de-mux using the random bits as select signals and convert back to single rail before sending the final output to its original destination.

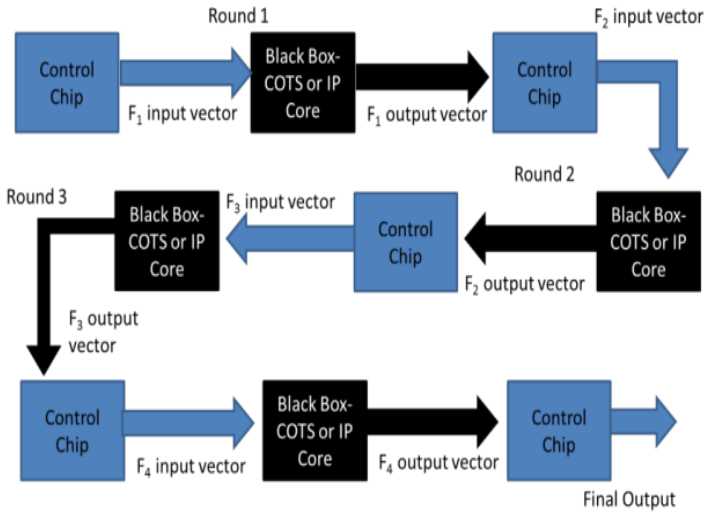


Figure 4.1. COTS RECORD data flow between two chips

The COTS RECORD process is perhaps the most secure application of the RECORD design scheme. At no time does a hardware Trojan present on the COTS chip have any way to monitor or infer the random bits. Since the Trojan has been on the chip

since manufacture there is also no way for it to adapt and recognize that the RECORD process is in use. It will continue to leak data as before, however now the data will be only one half of a dual rail signal. The control chip can make use of the TDM RECORD options to mix up the input order of the input data vectors. The designers can add as many random bits to the process as desired. Again, the cost of the COTS RECORD process is mostly in processing time, and design time. The additional board space of the control chip may also be a concern, especially in mobile devices, as well as the additional power requirements of that control chip. When the design requires the use of a proprietary COTS chip or even IP core, the COTS RECORD process provides an excellent method of safely operating the commercial product.

5. DESIGN RESULTS

The TDM and COTS RECORD processes were both implemented in VHDL to confirm functionality and evaluate impacts to area, power and performance. The TDM process was implemented on a 64 bit Data Encryption Standard (DES) design [15] for direct comparison to Sequential RECORD and simulated using Modelsim Altera edition v.10.1d. The design was synthesized with Cadence Encounter RTL Compiler v.13.10 and Cadence Generic Process Design Kit (GPDK) 45nm to obtain power estimates. The design was then fully laid out using Cadence Encounter RTL to GDSII v.13.23 to obtain an area estimate. The COTS RECORD process was implemented in hardware on an Altera DE2 development board featuring a Cyclone II FPGA. The control code written in VHDL was synthesized using Quartus II 13.0 sp.1. The VHDL code was also synthesized in Cadence Encounter for more direct comparisons with Sequential RECORD and TDM RECORD also using the Cadence GPDK 45nm technology library.

The TDM RECORD design of the 64 bit DES was found to have an area of 15,440 μm^2 compared to 7,619 μm^2 for the original unaltered DES circuit. This is an area increase of 2.02x. Comparing that to the Sequential RECORD design area for the same DES circuit, 28,948 μm^2, TDM RECORD reduces the area overhead by 63%. The power usage for TDM RECORD at 31Mhz has increased by 1.96x over standard DES and has reduced the power overhead of Sequential RECORD by 56%. Table 5.1 summarizes the power and area overhead of TDM RECORD.

The cost of TDM record, as stated previously, is in performance or latency. The original DES design requires 6.72ms to complete its testbench. The new TDM RECORD version takes 35.7ms to complete the same test bench at 31Mhz. Resulting in an increase

Table 5.1. Power and Area Comparison for TDM RECORD

	Standard DES	Sequential RECORD DES	TDM RECORD DES
Area μm^2	7619	28,948	15,440
Dynamic Power, mW	0.96	4.35	1.88
Leakage Power, mW	0.044	0.138	0.097

of 5.3x as was expected in the data flow of Figure 3.2. The additional 0.3x is due to processing time needed for the Sequential RECORD process, such as converting to dual rail, re-indexing and selecting the appropriate data to pass to the next logic stage, etc. Total energy consumption would also increase since the same operation must be completed four times to get the same final result. For example, standard DES would consume 3.6 J of energy per operation, Sequential RECORD DES, 15.71 J for the same operation and TDM DES apx. 37.7 J.

The latency overhead could be further reduced with optimization of the source code. For example, since the control logic knows what the random bits are it could perform a 'smart' selection of the intermediate logic outputs and simply drop the un-needed data. This would have the dual effect of reducing power and area. Latency would also be reduced since the steps in time T+4 in Figure 3.2 would no longer be needed. It may however open up the circuit to power analysis attacks since it would most likely be easy to see which round used more power by storing the data. Further analysis is planned to explore the effect of power analysis on RECORD circuits. Regardless, given today's high clock speeds, often in the Ghz range, this increase in latency should have minimal impact on all but the highest performance applications.

To illustrate the COTS RECORD concept, the FPGA was used as the control chip to operate an Intel 8294A Data Encryption Unit, which represented the Trojan infected COTS chip. The Intel 8294A implements the National Bureau of Standards information processing encryption standard [16,17] more commonly known as DES. This chip was used extensively in banking and other transactions well into the 1990's. The FPGA was loaded with custom logic to control the 8294A. The 8294A was sent the four RECORD input data vectors and it returned four dual rail encoded output vectors, Figure 4.1. The 8294A was never 'aware' of the fact that the input data was dual rail encoded and it also never saw the random bits. The dual rail encoded outputs from the 8294A were then de-muxed on the FPGA and converted back to single rail before being output from the FPGA. Figure 5.1 shows a picture of the actual setup.

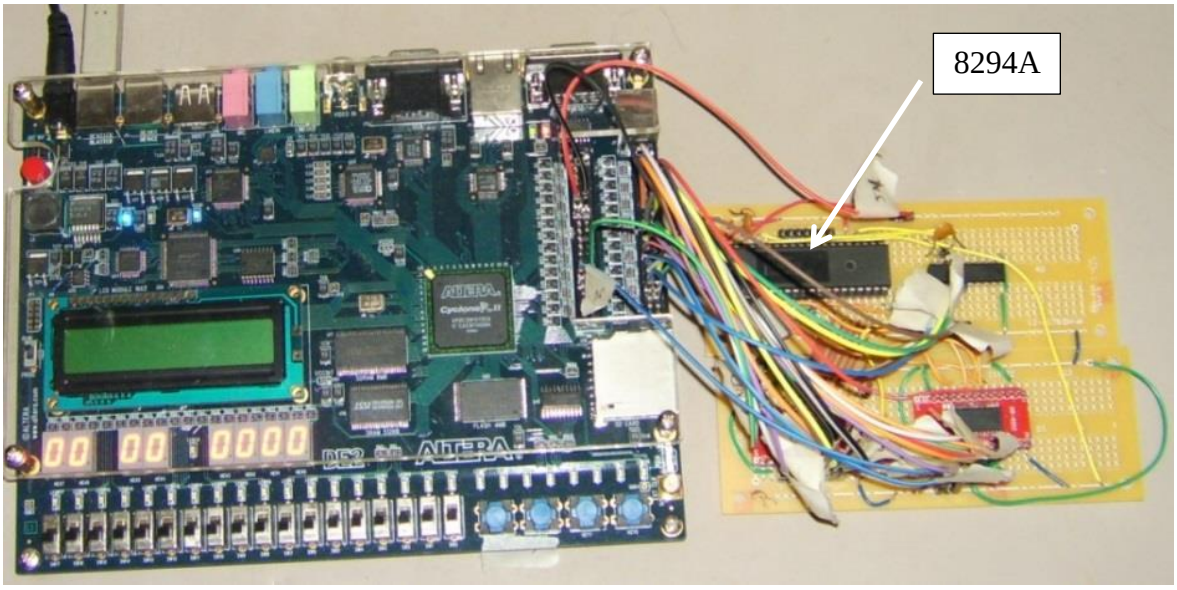


Figure 5.1. Experiment setup showing Altera DE2 development board and Intel 8294A with level shifting interface chips

This real hardware implementation serves not only to demonstrate COTS RECORD but also is the first real hardware implementation of the Sequential RECORD concept. The area requirements are broken down in Table 5.2. The COTS RECORD control logic, which includes a Trivium based random number generator [18,19], increased the 8294A control logic by 2.55x on the Cyclone II. The random number generator (RNG) itself is quite large and other smaller random number generators may be available. Control logic power has increased by 5.3x.

Again, the cost of COTS RECORD is in latency. Processing time was measured with an HP54645D oscilloscope. The standard control logic without RECORD measured 44ms to complete the DES operation process. The COTS RECORD control logic took 172ms to complete the same operation, an increase of 3.9x. The designer must also consider the extra space and cost of an additional chip on the board. However, in some instances the COTS RECORD control could be integrated into a pre-existing microcontroller, FPGA or processor, greatly reduce the additional cost and space.

Table 5.2. Comparison of Area for COTS RECORD

	8294A Control Logic	COTS RECORD Control	RNG
Area FPGA, logic units	563	1436	416
Area from layout, μm^2	2368	8394	2532
Dynamic Power, µW	18.67	98.3	33.67
Leakage Power, µW	0.044	0.175	0.07

6. CONCLUSION

Two modifications to the Sequential RECORD design process have been explored. Both offer area and power savings over the original Sequential RECORD process. First TDM RECORD utilizes time division multiplexing concepts to reduce the area of a Sequential RECORD design by 63% and the power consumption by 56% in the example DES circuit, at the small cost of 5.3x increased latency and total energy consumption. At the same time giving the designer greater flexibility to frustrate potential attacks by randomizing the order of input vector operation and even increasing the number of random bits.

COTS RECORD has presented a method to simply and safely operate potentially infected COTS products, believed to be a first in hardware Trojan defense. The COTS RECORD process splits the RECORD concept into two chips and results in an area increase of 2.55x and a power increase of 5.3x for the control logic. All while increasing latency by only 3.9x in the example circuit. COTS RECORD also gives the designer the option to easily increase the number of random bits and to randomize the order of input data operation. These options, TDM and COTS, allow both the ASIC designer and the systems integrator simple and effective options to safely utilize their designs.

REFERENCES

[1] Y. Jin and Y. Makris, "Hardware Trojans in Wireless Cryptographic ICs," IEEE Design & Test of Computers, pp. 26-35, Jan/Feb 2010.

[2] L. Lin, M. Kasper, T. Güneysu, C. Paar, W. Burleson, "Trojan Side-Channels: Lightweight hardware Trojans through Side-Channel Engineering," Cryptographic Hardware and Embedded Systems - CHES 2009, Lecture Notes in Computer Science, vol. 5747, pp. 382-395, 2009.

[3] K. Yang, M. Hicks, Q. Dong, T. Austin and D. Sylvester, "A2: Malicious Hardware," 2016 IEEE Symposium on Security and Privacy, San Jose, CA, May 2016.

[4] C. Kamhoua, M. Rodriguez and K. Kwiat, "Testing for hardware Trojans: A Game-Theoretic Approach. Decision and Game Theory for Security," Lecture Notes in Computer Science, vol. 8840, pp. 360-369, 2014.

[5]M. Rostami, F. Koushanfar and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," Proc. of the IEEE, vol. 102, no. 8, pp.1283-1295, Aug. 2014.

[6] R.S. Chakraborty and S. Bhunia, "Security Against Hardware Trojan through a Novel Application of Design Obfuscation," Proc. of the 2009 Int. Conf. on Computer-Aided Design, ICCAD'09, New York, NY, USA, pp. 113–116, 2009.

[7] J. Rajendran, M. Sam, O. Sinanoglu and R. Karri, "Security Analysis of Integrated Circuit Camouflaging," Proc. Of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp.709-720, 2013.

[8] J. Rajendran, O. Sinanoglu and R. Karri, "Is Split Manufacgturing Secure?," Proceedings of the Conference on Design, Automation and Test in Europe, DATE '13, pp. 1259-1264, 2013.

[9] R. Dura, S. Hidalgo, R. Quijada, A. Raventos and T. Francesc, "The Use of Digital Image Processing for IC Reverse Engineering," Multi-Conference on Systems, Signals & Devices (SSD), 2014 11th International, pp. 1-4, Feb. 2014.

[10] D. Dilger. (2016, Jan 14). *Apple A9 chip fab TSMC reports record earnings, casting further doubt on 'Peak iPhone'*. Available: http://appleinsider.com/articles/16/01/14/apple-a9-chip-fab-tsmc-reports-record-earnings-casts-doubt-on-peak-iphone.

[11] T. Schulze, Y. Shi, C. Kamhoua, K. Kwiat and S. Chang, "RECORD: Temporarily Randomized Encoding of Combinational Logic for Resistance to Data Leakage from Hardware Trojan," Asian HOST 2016, Yilan, Taiwan, Dec. 2016.

[12] T. Schulze, K. Kwiat, C. Kamhoua, D. Beetner, L. Njilla and Y. Shi, “Combating Data Leakage Trojans in Sequential Circuits Through Randomized Encoding,” unpublished.

[13] G.H. Bernstein, Q. Liu, M. Yan, Z. Sun, D. Kopp, W. Porod, G. Snider and P. Fay, “Quilt Packaging: High- Density, High-Speed Interchip Communications,” IEEE Transactions on Advanced Packaging, vol. 30, no. 4, pp. 731-740, 2007.

[14] K.Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, “Hardware Trojans: Lessons Learned after One Decade of Research,” ACM Transactions on Design Automation of Electronic Systems, vol. 22, no. 1, Article 6, May 2016.

[15] TCDG, TETRAEDRE S.A.R.L, White Paper, 1999, Used with permission.

[16] “Data Encryption Device Contains NBS Approved Algorithm”, Computer Design, vol. 17, no. 2, p. 186, Feb. 1978.

[17] J. Beaston,“One-Chip Data Encryption Unit Accesses Memory Directly,” Electronics, vol.52, p.126-129, Aug.2,1979.

[18] J. Van Rantwijk, “Pseudo Random Number Generator “Trivium”,” source code, 2016.

[19] De Cannière C., “Trivium: A Stream Cipher Construction Inspired by Block Cipher Design Principles,” Information Security, ISC 2006, Lecture Notes in Computer Science, vol. 4176, pp. 171-186, 2006.

SECTION

2. CONCLUSIONS AND FUTURE WORK

2.1 CONCLUSIONS

Current methods of hardware Trojan defense focus on either detecting the Trojan after production or preventing it from being inserted, all with varying degrees of success. All of the current methods suffer from some key weakness. This dissertation has presented a new way of looking at hardware defense by shifting the paradigm from "detection/prevention" approach to a don't care state. In what is believed to be a first for hardware Trojan defense, the concepts explored here can prevent data leakage even when the full design is known to the attacker. Three papers were presented which developed this idea.

The first paper, RECORD, defined a method by which combinational logic circuits could be altered to prevent data leakage Trojans from capturing meaningful data on chip. The RECORD method first utilizes a randomly generated number to dual rail encode the incoming data bits. These new signals are then split and only one half of the signal is operated on by the combinational logic. Further Boolean manipulation led to duplication of the original combinational logic which allowed the dual rail signal to be processed. The method was implemented on an Advanced Encryption Standard (AES) Substitution Box (Sbox) [15] circuit and synthesized with a 45 nm process. The design incurred a 2.28x-2.33x area overhead and 1.7x-2.24x power overhead. Impact to performance was only 0.06x increase in latency. It is expected that as larger designs are used with the RECORD process, the overhead will approach 2x. The increase in area and

power is acceptable when compared to other, commonly utilized, reliability methods such as Triple Modular Redundancy (TMR), which requires 3x area overhead. It is expected that highly sensitive applications will be more concerned with security than size and power consumption.

The RECORD process has a lower bound on area of approximately 2x. The area cannot be less than that since the original circuit has been duplicated. The precise value of the area overhead will approach 2x as the size of the original circuit increases. The increase that derives from the extra RECORD circuitry, XOR gates, multiplexors, inverters etc., will then be minimized.

The second paper described how the RECORD concepts could be successfully used on sequential logic. Sequential RECORD expanded the Boolean equations of RECORD to include two random bits. This in turn required quadrupling the intermediate combinational logic and adding update logic to the registers. The design scheme moved away from Quilt Packaging to 3D split manufacturing to protect the random bit. To demonstrate this process, a Data Encryption Standard (DES) [16] circuit was simulated using the Sequential RECORD design process. The 45nm design showed an increase in area of 3.75x and an increase in power of 4.5x at 31Mhz. The Sequential RECORD process added increased protection by increasing the possible dual rail combination and by utilizing the split 3D process. The permutations of final assembled chips are limited only by the total number of registers in the design.

The random number generator could present an upper bound for overall clock speed of the design. The sequential RECORD process adds little in the way of additional circuitry, with the exception of the random number generators. The random number

generators can be relatively large and slow. Whichever random number generator the designer chooses should be capable of generating two new random numbers on each clock at the required clock speed and should be capable of generating numbers that are difficult to guess, for example by selecting a seed which depends on some physical parameter and using a long random number sequence, or selecting the random numbers based on a physical random process.

The final paper presented two new ideas to address the shortcomings of RECORD and Sequential RECORD. The first was the TDM RECORD process which drastically reduced area and power consumption, by 63% and 56% respectively over Sequential RECORD. This was accomplished by creating custom control logic to feed each input bit vector to a single logic block in turn, instead of operating the input vectors in parallel. The resulting circuit operated at 5.3x slower latency compared to the Sequential RECORD design. The TDM process was simulated utilizing the same DES design as Sequential RECORD for direct comparisons.

The COTS RECORD process was presented last. Its key contribution is to allow previously manufactured chips to be utilized safely without any alteration. The COTS process breaks out the control logic of TDM onto a separate chip which then sends each input vector to the commercial chip in turn. The commercial chip, and any Trojan that may be on it, is never aware that the input data it is receiving is dual rail encoded. The concept was implemented on real hardware utilizing an Altera DE2 development board as the control chip and an Intel 8294A data encryption unit as the ‘infected’ chip. The control logic on the FPGA showed an increased area of 2.55x and increased power of 5.3x over the baseline 8294A control logic. The cost to the overall system though is

really in latency which increased by 3.9x from 44ms to 172ms to complete the DES operation. This is a small price to pay to open up the opportunity to safely utilize the COTS marketplace in secure environments.

2.2 FUTURE WORK

The RECORD design process can benefit from further investigations into optimizing the choice of dual rail input bits. Each of the RECORD design options include numerous choices on which input bits to convert to dual rail and which inputs are converted using which random bits. Paper I includes extensive data on these options. The TDM and COTS processes also open up the door to utilizing more than two random bits. Developing a process or algorithm to determine the optimum combination for reduced area and power for a given circuit would be very useful. Also needed would be finding the optimum tradeoff between latency and increased randomness of any extra random bits. These options could be explored using a hill climber technique or a specially constructed neural net.

The effect of the 3D manufacturing process on overall area has not been explored. The area overhead presented in Papers II and III is simply add in the increased area of the upper tier as if it were all incorporated into a single die. In reality, the upper tier would be on top of the outsourced lower tier and would not take up as much overall area in the final chip design. This is especially true for TDM RECORD since the lower tier contains no duplication. The area overhead may be significantly reduced when these effects are thoroughly characterized.

Design time and effort can become an issue for very large sequential designs. Ideally the RECORD process would be implemented from the start of the circuit design and incorporated directly into the VHDL or Verilog code. This may not be possible in many cases, and the process of converting a traditional design so it can be used with the RECORD process can be quite tedious if the RECORD designer is working from an already completed chip. Therefore, implementation of any of the RECORD processes should be automated. The best place for such an automation algorithm to start would be the final netlist. The automated process could then easily see what modules should be duplicated or quadrupled. The automated process could also easily identify and segregate all of the internal registers, moving them to a new module which would then become the secure upper tier. Automating the conversion process is not expected to be difficult.

The RECORD process is believed to be secure however, a concerted effort should be made to 'break' the RECORD process from the point of view of a hardware Trojan. Since this method claims only to resist data leakage hardware Trojans that type of Trojan should be used to try to leak meaningful data from a RECORD design. Full knowledge of the chip design should be made available to the attackers, or 'red team'.

An analysis of the RECORD processes' resistance to side channel power analysis attacks, such as Differential Power Analysis (DPA) [17], would also be very interesting. DPA and its variants look for patterns in the power signature of a chip to determine the data that is being processed. It is a different kind of attack from hardware Trojans but equally damaging.

At first glance, it would seem that the RECORD processes would provide some defense against these attacks given the randomization of the internal signaling. Further investigation into DPA and other side channel attacks is worth studying.

APPENDIX

STEP BY STEP IMPLEMENTATION INSTRUCTIONS FOR SEQUENTIAL RECORD

The following is a step by step instruction set for implementing a Sequential RECORD version of a pre-existing design. When a designer is starting from scratch with a design, the RECORD concepts should be written into the hardware description. The split manufacturing, either Quilt Packaging or 3D, will still have to be implemented manually during layout.

1- Synthesize the design using the desired technology library to obtain a netlist.

2- Edit the netlist by creating a new top level module. This module will contain all of the registers (flip flops) in the design.

3- Route all inputs and outputs through the new module. Any sub-modules will now be called from here.

4- Move **ALL** registers from the lower sub-modules to the new top modules. Verify connectivity to the sub-modules and confirm the circuit still functions properly.

5- Add a random number generator of your choice to the top module.

6- Now begin implementation of the RECORD process. First take all incoming data bits and convert them to dual rail with XOR gates. You may choose which bits are indexed to either r_1 or r_2. The inputs to each gate are the input bit, x, and either r_1 or r_2. You will end up with a new input vector of variable t.

7- Take the first sub-module, formerly the top level module, and create four instances of it.

8- The *t* vector will be routed as inputs to each of the four instances. Remember that each instance will receive a different version of *t* inputs. The easiest way to do this is to create a '*t not'* vector by inverting *t*. Then simply select the required bits from either *t* or *t not* and send them to the appropriate instance. So, the first instance gets all *t* inputs. The next gets even bits inverted. The third gets odd bits inverted and the last gets all bits inverted. See figure 3.3 on page 47.

9- Take the outputs of the four instances and invert the third and fourth output vectors.

10- Route the outputs from the first two instances and the inverted outputs of the third and fourth to a four-to-one multiplexor. This must be done for each bit. For example, a 64 bit output requires 64 multiplexors.

11- This step is very **important**. Make sure you understand how your multiplexors are wired and function. Wire r_1 and r_2 as select signals. Wire the multiplexor so that 'r_1=0, r_2=0' selects the first instance, 'r_1=0, r_2=1' selects the second, 'r_1=1, r_2=0' selects the third and 'r_1=1, r_2=1' selects the fourth. If r_1 is the most significant select signal bit then the input to the register will be referenced to r_1. If you switch the bits then the de-muxed register input will be referenced to r_2.

12- Wire the mux output to the appropriate register.

13- Create update signals. First store the previous clock cycle random bit that **you chose** as the de-muxed reference bit in step 11. XOR that signal with the current cycle r_1 and r_2 to create two update signals.

14- On each register, send the output to an XOR gate as one of the inputs and the other input is the update signal of your choice. Make sure not to just use one of

the update signals. Mix it up so that some bits use the r_1 reference and the others use r_2.

15- Create an inverted version of the vector you created in step 14.

16- Route the data to the next set of quadrupled sub-modules, just as in step 8.

17- Repeat until the entire circuit has been processes.

18- When the final stage data returns from the final set of quadrupled sub-modules, de-mux with four to one multiplexors as in step 11. The next step depends on the design. If the data is routed directly off chip, then all output bits must be XOR'd again with the appropriate random bit, see step 11, to convert back to single rail. If the data will be stored for multiple clock cycles then XOR with the appropriate random bit before storing in the register. This way the final output data will be in single rail and will not need to be continually re-indexed.

19- Create the layout making sure to keep all of the top module information on the upper tier.

REFERENCES

[1] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. 2007. Trojan detection using IC fingerprinting. In Proceedings of the IEEE Symposium on Security and Privacy, pp. 296–310, 2007.

[2] A. Sreedhar, S. Kundu and I. Koren. 2012. On Reliability Trojan Injection and Detection. J. Low Power Electronics pp. 674-683, 2012.

[3] R. Karri, J. Rajendran, K. Rosenfeld and M. Tehranipoor. Trustworthy Hardware: Identifying and Classifying Hardware Trojans. Computer, vol. 43, no. 10, pp. 39-46, 2010.

[4] B. Swarup, M. S. Hsiao, M. Banga and S. Narasimhan. Hardware Trojan Attacks: Threat Analysis and Countermeasures. Proc. of the IEEE, vol. 102, no. 8, pp. 1229-1247, 2014.

[5] T. Reece and W. H. Robinson, "Analysis of data-leak hardware Trojans in AES cryptographic circuits," Technologies for Homeland Security (HST), 2013 IEEE International Conference on, Waltham, M, pp. 467-472, 2013.

[6] Y. Jin and Y. Makris, “Hardware Trojans in Wireless Cryptographic ICs,” IEEE Design & Test of Computers, pp. 26-35,Jan/Feb 2010.

[7] Lin L., Kasper M., Güneysu T., Paar C., Burleson W., “Trojan Side-Channels: Lightweight hardware Trojans through Side-Channel Engineering,” Cryptographic Hardware and Embedded Systems - CHES 2009. Lecture Notes in Computer Science, vol. 5747, 2009.

[8] C. Kamhoua, M. Rodriguez and K. Kwiat, “Testing for hardware Trojans: A Game-Theoretic Approach. Decision and Game Theory for Security”, Lecture Notes in Computer Science, vol. 8840, pp. 360-369, 2014.

[9] M. Rostami, F. Koushanfar and R. Karri, “A Primer on Hardware Security: Models, Methods, and Metrics,” Proc. of the IEEE, vol. 102, no. 8, pp.1283-1295, Aug. 2014.

[10] S. Dupuis, P. S. Ba, G. Di Natale, M. L. Flottes and B. Rouzeyre, "A novel hardware logic encryption technique for thwarting illegal overproduction and Hardware Trojans," 2014 IEEE 20th (IOLTS), Platja d'Aro, Girona, pp. 49-54, 2014.

[11] G.H. Bernstein, Q. Liu, M. Yan, Z. Sun, D. Kopp, W. Porod, G. Snider and P. Fay, “Quilt Packaging: High-Density, High-Speed Interchip Communications,” IEEE Transactions on Advanced Packaging, vol. 30, no. 4 , pp. 731-740, 2007.

[12] J. Rajendran, M. Sam, O. Sinanoglu and R. Karri, "Security Analysis of Integrated Circuit Camouflaging.," Proc. Of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 709-720, 2013.

[13] J. Valamehr, T. Sherwook, R. Kastner, D. Marangoni-Simonsen, T. Huffmire, C. Irvine and T. Levin, "A 3-D Manufacturing Approach to Tustworthy System Development," IEEE Transactions on Computer-Aided Design, vol. 32, no. 4, pp. 611-615, 2013.

[14] D. Dilger. (2016, Jan 14). *Apple A9 chip fab TSMC reports record earnings, casting further doubt on 'Peak iPhone'*. Available: http://appleinsider.com/articles/16/01/14/apple-a9-chip-fab-tsmc-reports-record-earnings-casts-doubt-on-peak-iphone.

[15] J. Wolkerstorfer, E. Oswald and M. Lamberger. An ASIC Implementation of the AES SBoxes. Topics in Cryptology, UCT-RCA 2002, pp. 29-52, 2006.

[16] TCDG, TETRAEDRE S.A.R.L, White Paper, 1999, Used with permission.

[17] Kamil Gomina, Jean-Baptiste Rigaud, Philippe Gendrier, Philippe Candelier, Assia Tria, "Power analysis methodology for secure circuits", Design and Diagnostics of Electronic Circuits & Systems (DDECS) 2013 IEEE 16th International Symposium on, pp. 102-107, 2013.

VITA

Travis Edward Schulze was born and raised in Belleville, IL. He attended Belleville Area College, later Southwestern Illinois College, where he earned an Associate of Science and an Associate of Arts degree in 2001. He then transferred to Southern Illinois University, Edwardsville where he earned a Bachelor of Science in Electrical Engineering in 2003. After graduating he began working for the United States Dept. of Defense (DOD) and held various engineering positions within multiple agencies including one overseas tour to the United Kingdom from 2010-2012. While employed with the DOD he earned a Master of Science in Electrical Engineering from Washington University in St. Louis in 2007.

In 2013 Travis left the US Transportation Command in order to pursue a PhD. He attended the Missouri University of Science and Technology starting in Jan. 2014. While there he participated in two summer fellowships (2015 & 2016) with the Air Force Research Lab in Rome, NY. He received a Doctor of Philosophy in Electrical Engineering from Missouri University of Science and Technology in July of 2017.