Stealthy Dopant-Level Hardware Trojans

Tiit Pikma
Supervised by Dominique Unruh

Research Seminar in Cryptography
University of Tartu, Fall 2013

1 Introduction

With the increase in outsourcing manufacturing of integrated circuits to different countries, the topic of trust and security becomes more and more important. The built circuits could contain some malicious modifications introduced during production, known as hardware trojans. Common ways of detecting these modification include optical inspection with electron-microscopes, comparing side-channel information to a known trojan-free "golden" chip, and just plain functional testing.

This seminar paper, based on [1] by Becker, Regazzoni, Paar, and Burleson, describes a method of implementing hardware trojans on the dopant-level without being detectable by the aforementioned methods. The trojans circuits look physically the same as a trojan-free one, with modifications in only the physical composition of the materials used. The attack is demonstrated on two case-studies: reducing the quality of random numbers generated by Intel's Ivy Bridge processors random number generator, and introducing a power side-channel to an AES SBox implemented in side-channel resistant logic.

To understand the operation of transistors and how these dopant-level hardware trojans work, it is first necessary to have a basic understanding of semiconductor doping—this will be briefly introduced in Section 2. Section 3 explains the construction and operation of most transistors used today and how logic gates are built from them. Section 4 describes the dopant-level attack on a simple component and Section 5 gives an overview of the two case studies to see practical applicability of the attack. The last section summarizes the results and gives conclusions.

2 Semiconductor doping

Doping involves adding a dopant agent to a semiconductor, introducing impurities to the material and giving it different electrical properties than a pure semiconductor. A n-type dopant introduces a higher electron concentration to the semiconductor, producing a *n-type semiconductor*—"n-type" refers to the

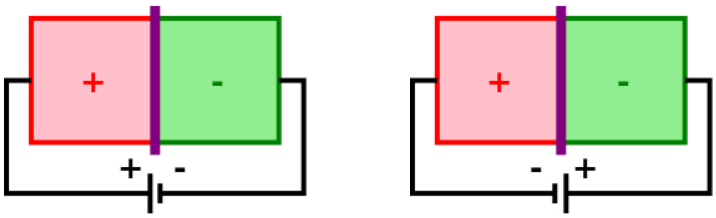


Figure 1: A diode with the p-n-junction illustrated in purple. On the left figure, electricity is flowing freely, as the extra positively charged carriers on the left side are repelled by the positive charge of the power source through the p-n-junction (the same goes for the negatively charged electrons in the other direction). On the right however, no current flows through the junction, because the extra carriers are attracted to the oppositely charged terminals of the power source instead.

negative charge of electrons. A common n-type semiconductor is silicon doped with phosphorous. A p-type dopant introduces a higher hole concentration to the semiconductor, producing a *p-type semiconductor*—"p-type" refers to the positive charge of holes. A common p-type semiconductor is silicon doped with boron[2].

So a n-type semiconductor is just a semiconductor with extra negatively-charged electrons, and a p-type semiconductor is a semiconductor with extra positively-charged holes. If we put a n-type and p-type semiconductor together side-by-side, then we get a barrier where the materials meet: electric current can flow from the p-type semiconductor to the n-type semiconductor, but not in the reverse direction.[1] This barrier is called the *p-n-junction*. This effect is caused by the fact that the extra charge carriers of both semiconductors are attracted to each other and they meet in the middle neutralizing each other, creating a region with no extra carriers. This region is called the *depletion layer*. No extra carriers can pass this layer on their own (even though they are still attracted by the extra carriers on the other side), because if they manage to do it, they will meet another carrier and become part of the layer. At one point, the layer will grow large enough that no carriers pass through it. Now, if we connect the positive terminal of a power supply to the p-type semiconductor and the negative terminal to the n-type semiconductor, then it will provide power to overcome the barrier and push the extra carriers through it. But if we connect the power supply in the reverse direction, then it will just attract the extra carriers from the semiconductor to the power supply terminals and no carriers pass the p-n-junction. Therefore current will only flow through the device in one direction (see Figure 1)[2].

[1] This simple device consisting of two different types of semiconductors allowing current in only one direction is known as a *diode*.

3 Transistors

A transistor is a semiconductor device which can either amplify or switch electronic signals and electrical power. It has at least three terminals for connecting to external components: these terminals are called the *gate*, *source*, and *drain*.[2] The current between the source and drain can be controlled by applying voltage to the gate terminal. By changing the output power to be higher than the input power, a signal can be amplified. By allowing the current to flow freely or by cutting it off, a signal can be switched either on or off.

For the purposes of this paper we are only looking at transistors acting as a switch in digital circuits. Furthermore, we are specifically looking at the metal-oxide-semiconductor field-effect transistor which is the most widely used type of transistor in integrated circuits[2].

3.1 MOSFET

This section explains the construction and operation of metal-oxide-semiconductor field-effect transistors.

3.1.1 Construction

A *metal-oxide-semiconductor field-effect transistor* or *MOSFET* consists of a doped semiconductor *substrate*[3] into which two separate areas or *diffusions* are doped oppositely from the substrate—these diffusions are called the *source* and the *drain* and have the corresponding terminals connected to them. Note that current can not flow between the source and the drain, because there are p-n-junctions between the substrate and the diffusions, so current can flow from the diffusions to the substrate or from the substrate to the diffusions (depending on the doping of the substrate and the diffusions), but not both at once.

An insulating oxide layer is grown on top of this substrate. The name comes from the fact that this layer is made from silicon dioxide, i.e. silica, but different materials can be used when working with smaller voltages. The purpose of this layer is to insulate the gate from the substrate.

And finally a gate layer is deposited on top of the oxide layer, which, as the name MOSFET implies, used to be made from metal, but now is made from polysilicon. The gate terminal is connected to this gate layer[3].

The actual manufacturing process contains a lot more steps, because the different layers can not be precisely grown or deposited, so first large layers are created and then actual shapes are etched out of them, but the end result is what was described in this section and illustrated on Figure 2.

As one can see, there are two types of MOSFETs depending on the types of doping used: the *nMOSFET*, where the substrate is a p-type semiconductor with

[2]The *gate*, *source*, and *drain* are respectively labelled *base*, *collector*, and *emitter* in older bipolar transistors.

[3]Also known as the *wafer* or the *base*.

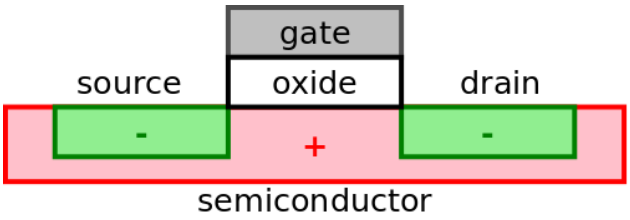


Figure 2: A metal-oxide-semiconductor field-effect transistor. More specifically, a nMOSFET; a pMOSFET is built exactly the same, except the source, drain and semiconductor are oppositely doped, i.e. the substrate is a n-type semiconductor and the source and drain have p-type dopant applied.

n-type diffusions, and *pMOSFET*, where the substrate is a n-type semiconductor with p-type diffusions.[4]

3.1.2 Operation

Let's look at a nMOSFET first. We connect the source terminal to the supply voltage and the drain terminal to the ground. Recall that in a nMOSFET the source and drain diffusions are n-type, which means that they have a higher concentration of negatively-charged electrons, and the substrate is p-type, meaning it has a higher concentration of postively-charged holes. As mentioned in the previous section, current can not normally flow between the source and drain because of the p-n-junctions. In this state, the nMOSFET acts as a switch which is open (off).

If we apply a high voltage to the gate, it generates a positive electric field, which radiates through the oxide layer and starts to repel the positively-charged holes from the surface of the substrate. If the voltage on the gate terminal exceeds a threshold, the extra holes are repelled far enough from the surface of the substrate to allow the extra electrons from the source and drain diffusions to form an inversion layer. This creates a *channel* through which current can now flow from the source to the drain (see Figure). Once the channel is wide enough to allow the supply voltage to flow through, the transistor acts as a switch which is closed (on). This is the called the *field effect*, from which the second part of the name MOSFET is derived[4]. The transistor is called a nMOSFET, because the channel is formed by negatively-charged electrons.

Inversely, a pMOSFET forms an inversion layer of positive holes if we apply a low voltage to the gate which repels the electrons from the surface of the substrate. An important fact to note is that when we connect the gate terminal of a pMOSFET to a digital circutry, then the gate has a low voltage if it is in the digital state 0. This means that a pMOSFET switch is closed (on) by default, and will open (switch off) once we apply a signal to the gate. This is

[4]The names nMOSFET and pMOSFET are not references to the type of dopant used to create the diffusions, but the charge of the carriers in the channel, which will be explained in the next section.

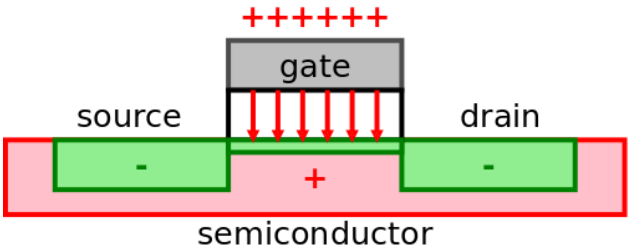


Figure 3: The field effect in a nMOSFET: the high voltage on the gate terminal is generating a positive electric field, which repels the extra positive holes of the substrate from the surface and forms a channel between the source and drain. A pMOSFET has the exact same effect, except all the dopants and charges are reversed, so the channel forms when a negative charge is applied to the gate.

logically opposite of the nMOSFET, which was open (off) by default and closed (switched on) if a high voltage, i.e. 1 signal, was applied to the gate.

3.2 CMOS logic

Complementary metal-oxide-semiconductor or *CMOS* is a technology for creating integrated circuits. It combines nMOSFETs and pMOSFETs in a complementary fashion to implement logic gates: all nMOSFETs in the circuit must have an input from either ground or from another nMOSFET and all pMOSFETs must have an input from either the voltage source or from another pMOSFET. Every nMOSFET is complemented with a pMOSFET such that both drains and gates are connected together. A low voltage (0 signal) on the gate will cause the nMOSFET to swith off and the pMOSFET to switch on and a high voltage (1 signal) will cause the opposite. Since the pMOSFET is (either directly or indirectly) connected to the voltage source and the nMOSFET is connected to the ground, this means that there is no connection between the source and ground except momentarily when switching states. This greatly reduces power consumption and is one of the main reasons for the popularity of CMOS logic[5, 6].

Let us take an inverter or NOT gate as an example of CMOS logic: it is the simplest CMOS device, consisting of only a single nMOSFET and a single pMOSFET in the described complementary configuration (see Figure 4). If A has a 0 signal, there is a connection between the voltage source and Q, but no connection between Q and ground, meaning that Q will have high voltage, i.e. a 1 signal. If A has a 1 signal, there is no connection between the voltage source and Q, but there is a connection between Q and ground, meaning that Q will have low voltage, i.e. a 0 signal.

More complex logic gates are created by just adding more transistors in specific configurations, but the two basic principles—nMOSFETs are connected to ground and pMOSFETs are connected to source, and every nMOSFET is

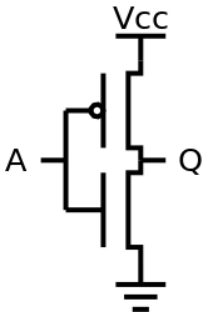


Figure 4: A CMOS inverter or NOT gate. The top transistor is a pMOSFET (denoted by the circle) connected to the power source (Vcc) and the bottom transistor is a nMOSFET (without the circle) connected to ground. A is the input to the inverter, connected to the gates of the transistors, and Q is the output.

complemented with a pMOSFET—remain.

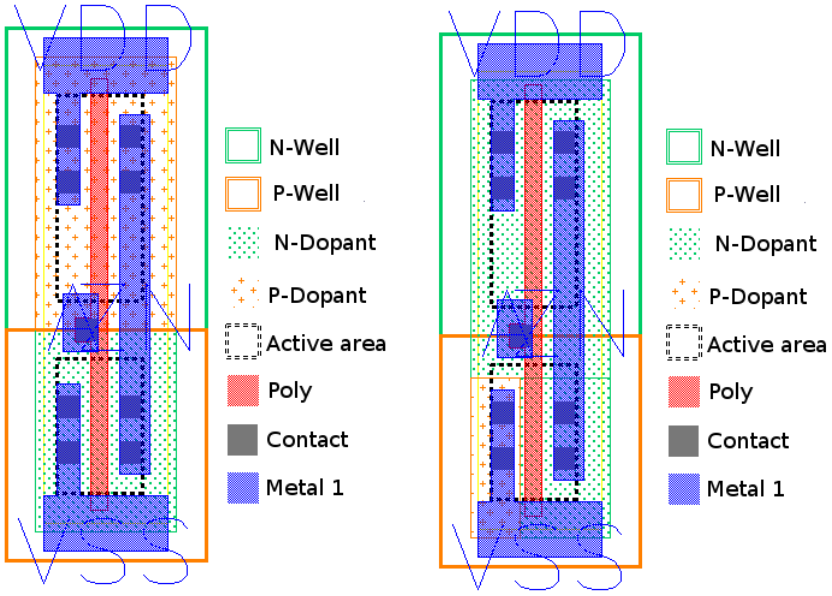
4 Dopant-level trojans

Becker, Regazzoni, Paar and Burleson proposed a technique for creating hardware trojans by making changes to integrated circuits on the dopant level. By changing the types of dopants used in the MOSFETs, one can change the behaviour of the circuit to their will. Since the wiring and layout of the circuit is exactly the same as an unmodified device and the only difference is in the atomic makeup of the semiconductor substrate, these kinds of changes are extremely difficult to detect: conventional means, e.g. optical inspection and comparing to known good "golden" chips, are not able to uncover the differences introduced[1].

We first demonstrate how this type of attack could be done on the inverter or NOT gate introduced in the previous section, by creating an inverter which outputs 1 regardless of the input. A top-down view of the physical representation of a normal inverter can be seen on Figure 5a [5][6]. If the p-type dopant used to create the diffusions in the pMOSFET is exchanged with a n-type dopant, then there are no p-n-junctions between the pMOSFETs diffusions and substrate. This means that the source and drain terminals connect directly to the substrate and each other, and since the source terminal of a pMOSFET is connected to the voltage source, so is the drain terminal, regardless of the voltage on the pMOSFETs gate and thus forcing the switch to be always closed

[5] As we can see from the figure, instead of using different wafers for the transistors in the CMOS design, large doped wells are created into a single substrate: these areas are called the n-well and p-well.

[6] Note that dopant is applied only inside the active area even though the dopant mask is larger. In addition, no dopant is applied under the polysilicon gate (even though the image depicts it such) and that area is the same polarity as the underlying well.



(a) Unmodified gate (b) Trojan gate

Figure 5: An unmodified NOT gate (5a) and a trojan NOT gate (5b) with contant output 1. The top part contains the pMOSFET with the bottom containing the nMOSFET. The small metal rectangle center-left is the input (A) and the large metal on the right is the output (N). The polysilicon gate (Z) can be seen in red, with the power source (VDD) at the top and ground (VSS) at the bottom. Both images are taken from [1].

(on). As a second step, the n-type dopant used to create the source diffusion in the nMOSFET is replaced with a p-type dopant (the drain diffusion is left untouched). This forms a permanent p-n-junction between the two diffusions—even if a high voltage on the gate creates a channel between them—and forces the switch to be always open (off). Therefore the output of the inverter is always connected to the voltage source and never connected to ground, making it output a constant 1 (see Figure 5b). This general technique can be applied to other types of gates in a similar way.

In addition to fixing the output of logic gates to specific values, the strength of a transistor can be modified in a similar manner. The strength of a transistor in CMOS logic is defined by its width: by reducing the width of the diffusions in a transistor we decrease the strength of a transistor, as it can let less charge carriers through. This kind of diffusion width modification will play an important role in the second case study discussed in Section 5.2.

A similar method of modifying dopants is already in use by commercially deployed code-obfuscation methods to prevent optical reverse-engineering[7],

meaning that this technique is practically feasible.

5 Case Studies

This section covers the two case-studies that Becker et al used to demostrate the applicability of their dopant trojans.

5.1 Intel Ivy Bridge Random Number Generator

The first case study targets Intel's new cryptographically secure random number generator (RNG) used in the Ivy Bridge platform and is likely to be used in future designs. They chose this target as it is very common and has detailed information available about its design and testing[8].

5.1.1 Intel's Ivy Bridge RNG design

The RNG consists of an *entropy source* followed by digital post-processing. The entropy source is a metastable circuit built from two cross-coupled inverters which uses thermal noise within the silicon to output a random stream of bits at a rate of 3 Gbps.

This stream is passed to the *conditioner* which is the first step in the two-step digital post-processing. It collects two 256-bit raw entropy samples and reduces them to to a single more concentrated sample using AES-CBC-MAC.

This 256-bit value is passed on to the second step of the digital post-processing: the deterministic random bit generator or the *rate matcher* (see Figure 6). It's role is to stretch the 256-bit value into a large set of random numbers. The rate matcher has a 128-bit internal *state* register and a 128-bit *key* register. Half of the 256 bits from the conditioner are used to randomize the state register, and the other half to randomize the key register. The rate matcher uses its state as input to the AES encryption function with key being used as the AES key; it does this three times, incrementing the value of state by 1 between each time. The first 128-bit output is the cryptographically secure random number that is provided to the user. The next two 128-bit values are used to randomize the state and key registers before the next cycle (see Figure 7). The conditioner reseeds the rate matcher whenever it has collected enough entropy, but the rate matcher will not generate more than 511 128-bit random numbers before blocking until it has been reseeded.

Another part of the digital post-processing is an online health test unit which measures the quality of the entropy coming from the entropy source. It tests the bit patterns of samples against expected bit distributions specified by a mathematical model of the entropy source. If a sample fails this test, it is marked as "unhealthy", which the conditioner can use to ensure that atleast two "healthy" samples are included in the seed.

The design also includes built-in self tests that verify the health of the RNG each startup before making it available to software. These include known-answer-tests on the digital post-processing, which pass deterministic values into

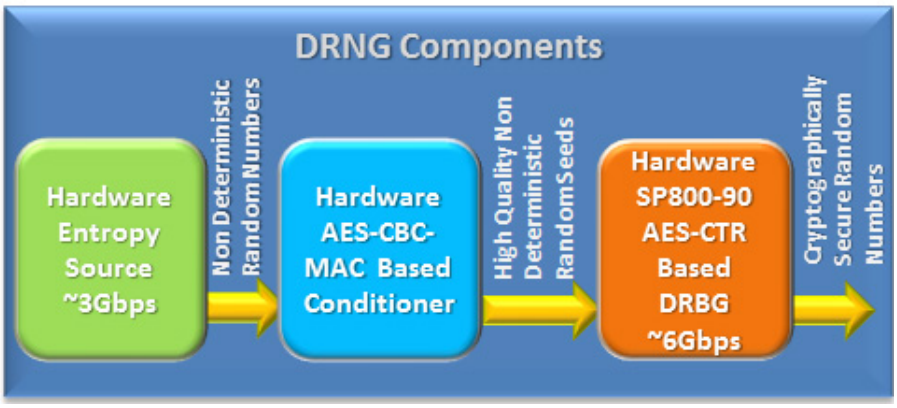


Figure 6: An overview of Intel's Ivy Bridge RNG components. Image taken from [8].

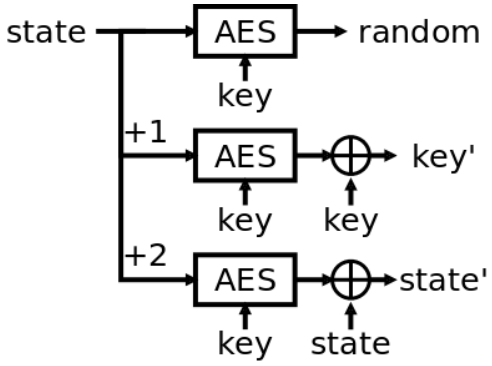


Figure 7: Generalized operation of the rate matcher in Intel's Ivy Bridge RNG. The generated *random* is passed to the user, while key' and $state'$ are used as the key and state in the next cycle.

the post-processing and verify the output, and statistical tests on the random numbers, which simultaneously tests the entropy source, seeds the conditioner and rate matcher for normal operation, and fills the output queue with random numbers.

5.1.2 Dopant-trojan for Intel's Ivy Bridge RNG

Similarly to how the output of the inverter was fixed in Section 4, the dopant-level attack on the RNG relies on fixing some bits in the circuit. In the first step, the key register of the rate matcher is fixed to a constant value, i.e. all 128-bits have a constant value. The second step fixes all but n bits in the state register, i.e. $128 - n$ bits have a constant value. Note that the fixed bits in the state register can't be randomly chosen, but have to be chosen such that they pass the known-answer-tests of the built-in self tests. But since the known answers of those tests are hardcoded on the circuit, the attacker can fix the bits in the state register such that they give correct answers on the tests.

Normally the output of the RNG is the 128-bit result of the AES encryption of an unknown 128-bit input with an unknown 128-bit key: therefore the output has a complexity of 128-bits since all the bits are random. But the output of a trojaned RNG is the AES encryption of a 128-bit value, of which the attacker knows all but n bits, with a 128-bit key known to the attacker: so for the attacker, the produced random number has only a complexity of n bits. Note, that for everybody who doesn't know the fixed bits in the input or the fixed key, the output looks legitimately random because AES generates numbers with good random properties even if only n bits in the input differ. Becker et al tested a circuit with $n = 32$, which had good enough randomness to pass all the built-in self tests, online health tests, and the NIST random number test suite, yet can be easily brute forced by someone who knows the fixed values.

Every bit of the registers is stored in a separate flip-flop[7] consisting of 32 transistors. Only 4 transistors of a flip-flop need to be modified to fix the value stored in it: this means that a total of $(128 + 128 - n) \times 4$ transistors need to be modified on the whole circuit for this attack to work. If $n = 32$, then only 896 of the millions of transistors used have to be modified. This is why these types of attacks are infeasible to discover via optical reverse-engineering: while a dedicated setup could eventually discover the changes to dopants used in a transistor, finding the modified transistors in a large integrated circuit is impractical and extremely time-consuming.

5.2 Side-channel resistant AES SBox implementation

The second case study serves to demonstrate the flexibility of the dopant-level trojans: it introduces a hidden side-channel into an AES SBox implemented in *improved Masked Dual-Rail Pre-charge Logic* or *iMDPL*—a technology for

[7] A flip-flop is a circuit which has two stable states and can be used to store a binary bit of information.

building side-channel resistant logic gates—to leak the keys used without modifying the design's logical behavior. Specifically, the case study targets AND gates in the SBox.

5.2.1 iMDPL

As mentioned, improved Masked Dual-Rail Pre-charge Logic is a technology for building side-channel resistant logic gates[9]. It consists of three main ideas:

1. Dual-Rail: for every signal computed, also the inverse of it is computed. Therefore the circuit will compute the same amount of 0's and 1's regardless of input. This helps prevent attacks based on Hamming weight.

2. Pre-charge phase: before each clock cycle is a pre-charge phase in which all iMDPL gates (except data-storing registers) are set to 0, so that they contain no values from the previous cycle. This helps prevent attacks based on Hamming distance.

3. Mask bit: all input to a logic gate has been masked by a random bit, which is passed to a logic gate as an extra input (note that also the inverse of the mask bit is passed because of the dual-rail system). This is needed because there might be differences in power consumption between a signal and its inverse: the random mask bit helps to hide this by randomizing the power-consumption.

An AND gate implemented in iMDPL will have six inputs and two outputs. The inputs are $a_m = a \oplus m$, $\overline{a}_m = a \oplus \overline{m}$, $b_m = b \oplus m$, $\overline{b}_m = b \oplus \overline{m}$, m and $\overline{m}$. The outputs are $q_m = q \oplus m$ and $\overline{q}_m = q \oplus \overline{m}$, where $q = a \wedge b$. Internally, the AND gate consists of two 3-input majority gates[8][9]. The first gate takes a_m, b_m, and m as input and computes q_m: if $m = 0$, then $a_m = a$, $b_m = b$, and the majority gate computes

$$maj(a, b, 0) = a \wedge b = q = q_m,$$

and if $m = 1$, then $a_m = \overline{a}$, $b_m = \overline{b}$, and the majority gate computes

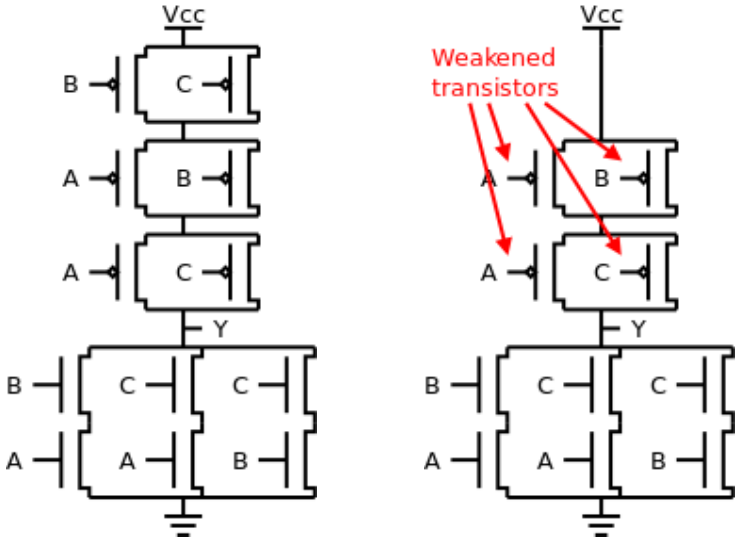
$$maj(\overline{a}, \overline{b}, 1) = \overline{a} \vee \overline{b} = \overline{a \wedge b} = \overline{q} = q_m.$$

The other gate just takes the complemented inputs and computes the complement of the first gate.

Because of the integrated circuit library Becker et al used in this case study did not have a majority gate implemented, the majority gates in the iMDPL AND gate are replaced with AND-OR-INVERT (AOI) gates followed by NOT gates. An AOI gate with three inputs A, B, and C computes $\overline{A \wedge B \vee A \wedge C \vee B \wedge C}$, which when followed by a NOT gate forms a 3-input majority gate. Regardless, the authors also verified that the following attack works with a regular 3-input majority gate[1].

[8] They actually also contain extra circuitry to verify that all signals are complementary, but that is not relevant to this case study.

[9] A 3-input majority gate is a logic gate where if at most one of the inputs is 1, then the gate's output is 0; if at least two of the inputs are 1, then the output is 1.



(a) Unmodified AOI gate (b) Trojan AOI gate

Figure 8: An unmodified AOI gate (8a) and a trojan AOI gate (8b), both configured as not-majority gates with three inputs. A, B, and C are the inputs, Y is the output.

5.2.2 Dopant-trojan for iMDPL

For the attack, the two AOI gates in a single iMDPL AND gate of the SBox are replaced with trojan gates that create an input-dependant power side-channel independant of the mask bit.

Figure 8 shows a normal AOI gate which acts as the not-majority gate and a trojaned AOI gate with the power side-channel. The first modification introduced is the disabling of the two top-most pMOSFETs: their sources and drains are doped such that the transistors act as a constantly closed switch, allowing current through regardless of the gate input. This is done exactly as it was done with the pMOSFET in the inverter in Section 4 and effectively removes those two transistors from the logical circuit (even though they are physically still present): the output is connected to the voltage source even if both B and C are 1's. The second step is to weaken the remaining pMOSFETs by reducing their width as discussed in Section 4.

The trojan gate behaves just like the normal AOI gate except for the input $A = 0$, $B = 1$, and $C = 1$: in this case both gates have the output connected to ground, but in the trojan-free gate the connection from output to the voltage source is cut off, while in the trojan gate it is not. Yet the pMOSFETs are significantly weaker than the nMOSFETs, so the output has a much stronger connection to ground and the output voltage is still near 0. Therefore the trojan gate has the same output as the trojan-free gate, but there is a direct connection

between the source and ground, causing a large power consumption as electricity is flowing to ground unresisted. For all other inputs there is no direct connection between source and ground, causing no extra power consumption in those cases.

In the context of an iMDPL AND gate, if both AOI gates are replaced with the trojan versions, then the gate will have increased power consumption if the non-complemented AOI gate has input $A = a_m = 0$, $B = b_m = 1$, and $C = m = 1$ or if the complemented gate has input $A = \overline{a}_m = 0$, $B = \overline{b}_m = 1$, and $C = \overline{m} = 1$. In the first case

$$\begin{aligned} a_m = a \oplus m = a \oplus 1 = \overline{a} = 0 \quad &\Rightarrow \quad a = 1, \\ b_m = b \oplus m = b \oplus 1 = \overline{b} = 1 \quad &\Rightarrow \quad b = 0, \end{aligned}$$

and in the second case

$$\begin{aligned} \overline{a}_m = a \oplus \overline{m} = a \oplus 1 = \overline{a} = 0 \quad &\Rightarrow \quad a = 1, \\ \overline{b}_m = b \oplus \overline{m} = b \oplus 1 = \overline{b} = 1 \quad &\Rightarrow \quad b = 0, \end{aligned}$$

so we can see that the iMDPL AND gate will have increased power consumption if $a = 1$ and $b = 0$ regardless of the mask bit m.

Since no logical behavior is changed, this trojan cannot be detected by functional testing. As discussed before, optical inspection also does not detect this as we only modify the dopants. Without optical inspection or functional testing an evaluator also cannot verify a golden chip, i.e. a chip which is known to not contain a trojan. So one also cannot compare the side-channels of a golden chip to a possibly-trojan one. Therefore this dopant side-channel cannot be detected via conventional means.

6 Conclusions

This paper described a new technique for introducing virtually undetectable hardware trojans that only require modification of the dopants applied. The resulting trojan circuits look physically exactly the same as trojan-free ones, making them undetectable via conventional means: optical inspection can find no differences and without optical inspection we cannot verify a "golden" chip. In the two example cases, functional testing was also ineffective as the trojan circuits behaved or at least seemed to behave exactly as a trojan-free one.

This technique was demonstrated on two case studies which involved real-world scenarios and practical applications. The first was Intel's secure random number generator design which is deemed secure by a third-party security company and conforms to several security standards. Hardware trojans were used to compromise the quality of the hardware generated random numbers produced by it without being detectable. The second case study featured weakening transistors in an AES SBox implementation to introduce an input-dependant power side-channel without modifying the output of the circuit—this served to demonstrate the versatility of the attack, showing that entire bits don't need to be fixed but can be modified in a non-digital manner and used to leak, for example, AES

keys. In both cases, only the attacker knows the exact modifications that were introduced to the circuits so they seem completely normal to all by-standers and leak no information to them.

This attack is also practically feasible as a similar method is already in practical use by code-obfuscating solutions commercially-deployed and there are currently no known effective methods known for detecting it.

References

[1] Becker, G. T., Regazzoni, F., Paar, C., Burleson, W. P. "Stealthy Dopant-Level Hardware Trojans". Cryptographic Hardware and Embedded Systems - CHES 2013, pp. 197-214, Springer, 2013.

[2] Neamen, D. A. "Semiconductor Physics and Devices: Basic Principles (3rd ed.)". McGraw-Hill Higher Education, 2003.

[3] Kahng, D. "Electric Field Controlled Semiconductor Device". U. S. Patent No. 3,102,230, filed May 31, 1960, issued August 27, 1963.

[4] Edgar, L. J. "Method and apparatus for controlling electric currents". U.S. Patent No. 1,745,175, filed Oct 8, 1926, issued Jan 28, 1930.

[5] Wanlass, F. M. "Low stand-by complementary field effect circuitry". U.S. Patent No. 3,356,858, field Jun 18, 1963, issued Dec 5, 1967.

[6] Baker, R. J. "CMOS: circuit design, layout, and simulation (Second ed.)". Wiley-IEEE, 2008.

[7] SypherMedia International. "Circuit Camouflage Technology - SMI IP Protection and Anti-Tamper Tenchologies". White Paper Version 1.9.8j, 2012.

[8] Intel Corporation, Hofemeier, G. "Intel Digital Random Number Generator (DRNG) Software Inmplementation Guide". http://software.intel.com/en-us/articles/intel-digital-random-number-generator-drng-software-implementation-guide, 2012.

[9] Popp, T., Kirschbaum, M., Zefferer, T., Mangard, S. "Evaluation of the Masked Logic Style MDPL on a Prototype Chip". Cryptographic Hardware and Embedded Systems - CHES 2007, pp. 81-94, Springer, 2007.