Detection of Counterfeit ICs using Public Identification Sequences and Side-Channel Leakage

Peter Samarin and Kerstin Lemke-Rust

Abstract—**In this work we present a new approach for counterfeit protection against remarked, overproduced, and out-of-spec integrated circuits (ICs). Our approach uses identification sequences that are regularly published by the original chip manufacturer and hidden in the electromagnetic (EM) leakage of authentic chips. A portion of the chip area is dedicated to a crypto engine running in stream cipher mode that is initialized with a unique key and initialization vector stored in one-time-programmable antifuse memory. IC identification is carried out in the field, by obtaining EM measurements of deployed ICs and by proving the presence of the genuine identification sequences in the measurements. We evaluate our approach using a proof-of-concept implementation on three FPGA boards by capturing EM leakage of the FPGAs under test at their decoupling capacitors. The results show that the user can carry out IC identification on-site, using standard lab equipment in short amount of time.**

Index Terms—**Counterfeit protection, IC identification, Side channels, EM leakage, Leakage circuits, Stream cipher, Antifuse memory, Chip ID.**

I. Introduction

COUNTERFEIT integrated circuits (ICs) cause monetary loss, damage the reputation of the original chip manufacturers (OCMs) and impose safety, security, and reliability risks on the users. IC counterfeiters have created a whole industry that is supported by the asymmetry of the IC identification process—counterfeits can be mass-produced cheaply, however, IC identification needs high effort to authenticate a small sample of chips. The number of counterfeit ICs is steadily increasing each year.

There are many ways how counterfeits are created and inserted into the supply chain [1], [2]. The most common type of counterfeits are ICs that have been *remarked* as a higher grade [3]. They have their die taken from another IC. For example, the IC is marked as industrial grade instead of commercial grade, so that the counterfeiters can sell cheap and old chips at a higher price. ICs are remarked by removing the original package and wrapping the die in a new package with professionally-looking markings. Package removal is done in a way that can be potentially harmful to the die, so that the ICs sold as new might be defective, or have shortened life span.

Another type of counterfeit ICs are *recycled* ICs that have been used and are sold as new. They may function properly, but are closer to reaching their end of life due to prior use. A foundry can *overproduce* ICs and sell them through the common distribution channels. These ICs are authentic and do not bring any risk for the users. However, the OCM suffers monetary loss because the foundry will sell the ICs at a lower price. During the manufacture, some ICs that are *out-of-spec* do not pass the tests and should be discarded. However, a malicious foundry can insert them into the supply chain by selling them as new. IC designs can be *cloned* either through reverse engineering, or by illegally obtaining the design of the original chip.

Contributions: We present a new approach for detection of remarked, overproduced, and out-of-spec ICs. Our approach requires the original chip manufacturer to regularly publish binary identification sequences that can be used to identify ICs by proving correlation between the sequences and the electromagnetic (EM) leakage of the chips. IC counterfeit protection is realized by writing a unique secret key and initialization vector (IV) pair into antifuse memory for each IC. Each key/IV pair is used by a crypto engine to generate keystream. The keystream is leaked over the EM side channel by using leakage circuits—special circuits used for amplifying side channel leakage.

The advantage of our approach is that IC identification can be performed on deployed systems using standard lab equipment, i.e., a digital oscilloscope and an EM probe. Our approach gives any user the ability to quickly identify ICs without sending them to a specialized lab, and use larger batches of ICs when performing IC identification, which increases the chances of detecting the counterfeits.

Paper overview: This paper is organized as follows. The next section reviews other works that are related to ours. Section 3 introduces our approach and discusses the background information necessary for its understanding. Section 4 describes our experimental setup and the experiments performed to evaluate our approach. Section 5 discusses possible attack avenues that IC counterfeiters might try and their countermeasures. Section 6 concludes the paper and gives directions for future work.

Manuscript submitted March 8, 2018. This work was supported by the German Federal Ministry of Education and Research (BMBF) through the project DePlagEmSoft, FKZ 03FH015I3.

P. Samarin and K. Lemke-Rust are with the Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany (e-mail: {peter.samarin,kerstin.lemke-rust}@h-brs.de).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes VHDL source code and bitstreams. Contact {peter.samarin,kerstin.lemke-rust}@h-brs.de for further questions about this work.

II. Related Work

A simple method to track an IC is to write a unique ID together with some information about the chip (e.g. serial number, manufacture date, foundry, speed grade, etc.) into non-volatile one-time-programmable memory [4]. IDs can be read out over a pin, which makes it easy to perform IC identification. However, the IDs can be copied into overproduced ICs, tampered with, and removed. Miller et al. have proposed mixing a unique DNA sequence into the ink that is used to mark the IC packages, which allows the OCM to track a design/mask [5]. Design identification requires sending a small sample of ICs to a special lab, where the DNA on the package can be matched with the expected sequence. DNA validation is costly so that design/mask identification can only be carried out for a small sample of ICs. Several other approaches for non-electronic physically unclonable functions (PUFs) exist in the literature [6]. For example, DeJean and Kirovski [7] have proposed a non-electronic PUF based on radio frequency called *RF-DNA*. It uses a set of thin and short randomly bent copper wires that are held together by a dielectric. By measuring the near EM field of the wires, it is possible to obtain a unique ID. When soldered or baked into a printed-circuit board, ICs are exposed to heat, and for many non-electrical PUF approaches it has not been tested whether the high temperature alters the value carried by the PUF. An approach by Kuemin et el. [8] grows an array of short gold nanorods using an elaborate process. After their growth, the nanorods can be printed onto the IC package, which can then be identified by comparing the visual properties of a nanorod to a database.

Electrical PUFs rely on variations in manufacturing process and have been used for IC identification and on-chip key generation [9]–[12]. The chip ID is obtained by collecting challenge-response (C/R) pairs for each IC. During authentication, a challenge can be only used once and has to be removed from the database after its use. The database must be stored securely, and the ICs can be identified one at a time. The C/R pairs should be collected by a trusted party, for each individual IC. PUFs require a large amount of data to be saved. Here is an example taken from [13] p. 252. A ring oscillator PUF requires $n * log_2(n+1)$ bits of challenge to produce an n bit response. Assuming that there are 100 C/R pairs for each PUF in the database containing one million ICs, we get 1000000*100*128*log(129) bits = 10 GB for ring oscillator PUFs. Arbiter PUFs require $n \cdot k$ bits of challenge to produce n-bit response, which, for a million of ICs will require 100 GB of storage space. Thus, the OCM can only save a very limited number of C/R pairs. No further generation of C/R pairs is possible once the IC is sold to the user. The OCM has to be very selective with the requests for C/R pairs. If all requests are processed, the adversary can potentially be able to exhaust the C/R database by issuing a large number of requests, because the database cannot be replenished with new C/R pairs after shipping the ICs.

Active hardware metering provides ways to actively control the number of produced ICs by locking them initially, and requiring the foundry to activate each chip before it can be tested [14]–[17]. The IC is locked by either using a finite-state machine that starts in a locked state and has to be unlocked by a sequence of correct inputs. Another approach is to embed XOR gates into the design at random locations. The chip can be unlocked by providing a correct key. Active hardware metering aims to stop overproduction by untrusted foundries. It also makes it possible to enforce user (but not IC) authentication for security sensitive systems [15]. However, active hardware metering does not prevent remarking, and some approaches introduce a large chip area overhead due to the presence of public crypto (e.g. [16]).

Several approaches in the literature make use of side channels, such as power [18]–[20], EM [21], [22], and temperature [23] to leak secret data unique to the IC or IP core in order to identify cloned ICs and illegal uses of IP cores. These works are trying to solve the proof of ownership problem, where the goal is to protect the IP cores from unlicensed use. These methods address the problem by leaking a watermark over the chosen side channel and correlating the side channel with a secret code that is known only to the verifier.

Kean et al. have also discussed the possibility of using the temperature side channel to add an ID, called *tag code*, to a design/mask [23]. A unique tag code is written into non-volatile one time programmable memory of a design, so that IC designs can be distinguished. However, since the tag code is the same for all ICs of a design/mask, it is not possible to identify individual ICs. On the IC, the tag defines the initial state of a linear-feedback shift register (LFSR) or a stream cipher to generate a bit sequence based on the tag. To identify a design, the temperature of the chip is measured and correlated to a known sequence. One drawback of this approach is that a successful attack on one IC makes all other ICs of the same design vulnerable because a malicious foundry can overproduce ICs and write correct tag code into all of them. Another problem is that the sequences are stored in a secure web-based database, which means that only a trusted entity can identify a design. Should a user suspect a counterfeit, he cannot carry out IC design identification on his own and needs to work closely with the OCM. To compute the correlation, the same secret bit sequence is used each time, which opens the approach for an attack on the bit sequence generated by the LFSR or keystream. Knowledge of the sequence allows the attacker to produce own ICs and embed the sequence in them, which will make that IC design be falsely recognized as authentic. Because it is a commercial product, the implementation details of this approach are not disclosed.

III. Counterfeit Detection using Public Identification Sequences

Our new approach is based on a crypto engine initialized with unique key/IV pair in each IC. When turned on, each IC computes the keystream and in this way leaks it over the electromagnetic (EM) side channel (SC). The OCM releases parts of keystreams with intentionally added errors—

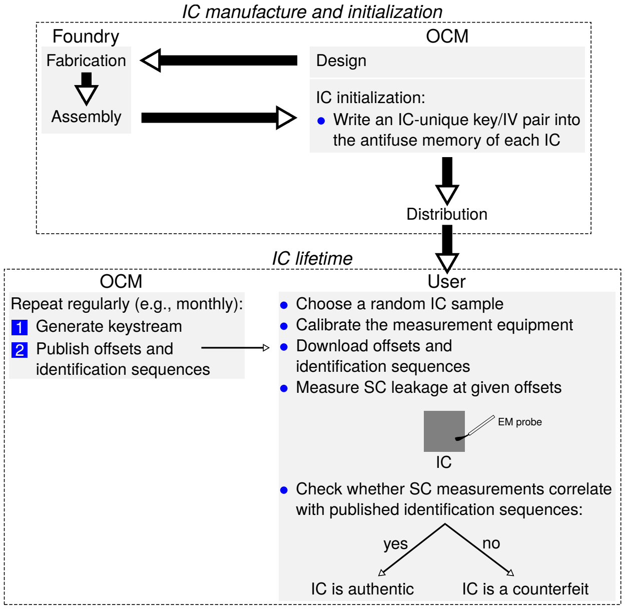

IC manufacture and initialization
Foundry
OCM
Fabrication
Design
Assembly
IC initialization:
Write an IC-unique key/IV pair into the antifuse memory of each IC
Distribution
IC lifetime
OCM
User
Repeat regularly (e.g., monthly):
1 Generate keystream
2 Publish offsets and identification sequences
Choose a random IC sample
Calibrate the measurement equipment
Download offsets and identification sequences
Measure SC leakage at given offsets
EM probe
IC
Check whether SC measurements correlate with published identification sequences:
yes
no
IC is authentic
IC is a counterfeit

Fig. 1: IC identification using public identification sequences.

henceforth called *identification sequences*—online on a regular basis. IC identification is done by proving significant correlation between the identification sequences and SC measurements.

To realize this IC protection scheme non-standard steps during IC manufacturing are required. Our approach is illustrated in Figure 1. Instead of sending the ICs for distribution after their assembly, the foundry sends the ICs for initialization to the OCM, where unique key/IV pairs are written into tamper-resistant one-time programming memory of each IC. After initialization, the OCM starts publishing identification sequences and the offsets at which they have been excerpted from the keystream.

The motivation for using side channel leakage to transmit the keystream instead of simply using a data output pin is twofold. The first reason is to enable IC identification in the field—deployed ICs can be identified without the need to remove them from the circuit board. An extra pin for IC identification might not be connected in a deployed system, however, side channel measurements can always be recorded. The use of EM probes has the advantage that the printed circuit board design does not have to be altered. On large ICs, localized leakage carries a stronger signal than leakage in the power channel. Thus, IC identification can be carried out on deployed systems in the field. The second reason for not using a pin is to enhance resistance to cryptanalytic and fault attacks that aim to recover the internal state of the stream cipher.

Our approach makes IC remarking useless, because the SC leakage from the remarked ICs will not correlate with released identification sequences. Overproduced and out-of-spec ICs will not have the matching keys, so that their SC leakage will not correlate either.

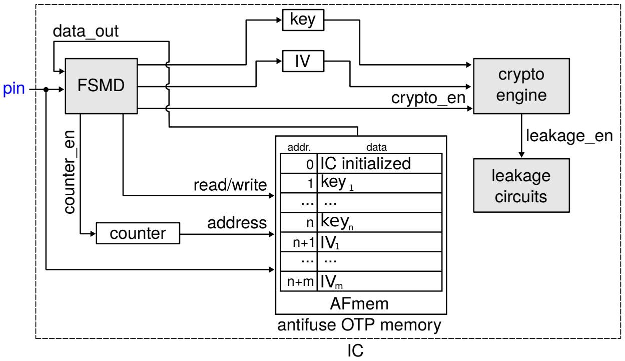


Fig. 2: Counterfeit protection circuit.

Algorithm 1: IC initialization.

```
Input: a stream of bits transmitted over the pin
if AFmem[0]==0 then // initialize IC
    wait until pin==1
    for counter in 1:$(n+m)$ do // write key and IV
        AFmem[counter] := pin
    AFmem[0] := 1 // finalize IC initialization
else // load key/IV, and start the crypto engine
    for counter in 1:$n$ do // load key
        key[counter] := AFmem[counter]
    for counter in $(n+1)$:$(n+m)$ do // load IV
        IV[counter] := AFmem[counter]
    crypto_en := 1
```

A. OCM: Writing Unique Key/IV Pairs into ICs

Each IC is programmed with a unique secret written into the one-time programmable antifuse (AF) memory. AF memory has the advantage of being tamper-resistant, and withstands current state of the art reverse-engineering attacks aimed to read out its content [24]. It does not require additional masks during fabrication and can be written in the field over a pin. Compared to other one-time-programmable memory technologies, AF memory consumes less power to program and read [25]. Figure 2 illustrates the IC protection circuit. It consists of a finite state machine with datapath (FSMD), an AF memory block, and a crypto engine. The FSMD is programmed over a pin and is used to steer IC initialization. The IC initialization routine of the FSMD is shown in Algorithm 1. After reset, the FSMD checks whether the IC has been initialized by checking the first bit stored in the AF memory. Uninitialized ICs wait until a ’1’ is applied to the pin, and write the $n + m$ bits of the key/IV pair that follow into the AF memory at consecutive addresses. To finalize IC initialization, a ’1’ is written at the first address of the AF memory. After reset, initialized ICs load the stored key/IV pair and start the crypto engine.

B. IC: Leaking Keystream over the EM Side Channel

The uniqueness of key/IV pairs guarantees a unique keystream for each IC, which can be used to identify it. The crypto engine computes one bit of the keystream per clock cycle. It can be realized by using a stream cipher or

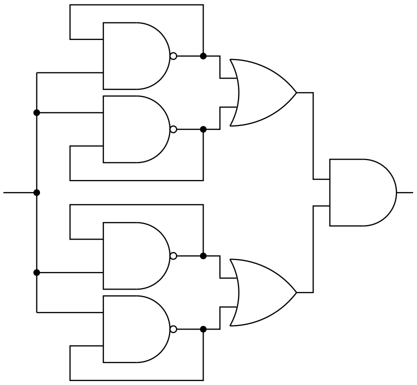


Fig. 3: Leakage circuit consisting of 4 NANDs, 2 ORs, and 1 AND.

a block cipher running in counter mode. By computing the keystream, the IC leaks it over the EM side channel. The leakage is weak and hard to detect because keystream is output one bit at a time. To amplify the signal, we use leakage circuits proposed by Parrilla et. al [26][1] shown in Figure 3. Each leakage circuit consists of 4 NANDs, 2 ORs, and one AND gate. When active, the NAND gates with feedback become ring oscillators that feed the OR gates. However, their outputs do not reach the OR gates at the same time, which results in several glitches per oscillation that propagate through the OR and the AND gates. This results in different EM leakage between the active and non-active leakage circuits. This difference can be amplified by embedding several leakage circuits in parallel. Thus, the keystream is transmitted over the EM side channel by feeding it into the leakage circuits. To reduce the power consumption of the protection circuit, it can be turned off after a certain amount of time. If we assume that from the 2^{64} possible bits of Trivium keystream the OCM only uses the first 50 million bits, from which 10k bits are released to the public each month, it leaves us with $\frac{50000000}{10000 \cdot 12} \approx 416$ years of sequences, which we can safely assume will not be exhausted during the lifetime of an IC. If the keystream circuit runs at 50 MHz, the IC protection circuit can be turned off after one second.

C. OCM: Publishing Identification Sequences

An identification sequence S_n of the n—th IC given its corresponding key_n and IV_n, is any subsequence of L bits of the keystream with artificial errors introduced at random bit positions with a fixed error rate:

$$S_n = \text{keystream}_n[t_r : (t_r + L - 1)] \oplus \text{err}_n,$$

where t_r and $t_r + L - 1$ define the beginning and the end of the subsequence extracted from the keystream, err is a binary error string randomly generated for each identification sequence by swapping bits at random positions, and $\oplus$ is the bitwise XOR operation that takes two bitvectors of equal length as input and produces a bitvector of the same length with their bits pairwise XOR-ed. The number of ones in the error string is equal to $\lfloor \varepsilon \cdot L \rfloor$, where ε is the error rate—a real number in the interval $\varepsilon \in [0; 1]$; and $\lfloor \ \rfloor$ is a *floor* operation.

TABLE I: An example database of public identification sequences for N ICs and 3 releases.

IC/Serial Nr.	Release 0	Release 1	Release 2
IC_0	$S_0^0 = 101\ldots$	$S_0^1 = 100\ldots$	$S_0^2 = 000\ldots$
IC_1	$S_1^0 = 000\ldots$	$S_1^1 = 100\ldots$	$S_1^2 = 110\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\text{IC}_{\text{N-1}}$	$S_{N-1}^0 = 111\ldots$	$S_{N-1}^1 = 101\ldots$	$S_{N-1}^2 = 011\ldots$
Offset (bits)	$t_0 = 2305$	$\text{t}_1 = 4091$	$\text{t}_2 = 6712$

Each identification sequence is taken at a random offset in a way that it does not overlap with previously extracted identification sequences. The identification sequences and their offsets are published online so that the users can access them and determine whether their ICs are authentic. Errors in the keystream increase the effort of attacks on the internal state of the cipher. Instead of taking consecutive bits of the keystream, the OCM can also take each keystream bit at a random non-overlapping offset. This will further increase the difficulty of cryptanalytic attacks on the cipher that rely on the knowledge of consecutive keystream bits.

To prevent an attacker from embedding some of the released identification sequences into cloned ICs, the OCM releases new sequences at regular time intervals, e.g. once a month. To have constant memory usage, the OCM keeps a fixed number of identification sequences per IC in the published database—after adding the new release, the oldest release is removed. Table I shows an example database for N chips with three releases of identification sequences and their offsets.

The length of identification sequences depends on the number of leakage circuits and on the motivation of the chip manufacturer to increase the effort of an attacker who wants to copy the latest sequence into a cloned IC. A small number of leakage circuits results in a weak leakage signal requiring a longer identification sequence in order to be detectable. The minimal sequence length will depend on the IC design and can be determined experimentally.

The OCM has to store two databases: a public database containing identification sequences, offsets, and serial numbers of the ICs; and a private database with key/IV pairs used for generation of identification sequences. If we assume that OCM orders a production of 1 million ICs, uses 32 bits to store a serial number, 32 bits to store an offset, keeps 3 releases in the online database and publishes identification sequences of 1000 bits per IC, the amount of memory required to store published sequences will be

$$\frac{1000000 \cdot (3 \cdot 1000 + 32) + 32 \cdot 3}{8 \cdot 1024^2} \approx 361 \text{ megabytes.}$$

[1] We note that the leakage circuit of Bossuet et al. [21], [22] is estimated to be more efficient in terms of area and power consumption.

Assuming that the OCM uses 128 bits to store one key and 128 bits to store one IV, the database with key/IV pairs will need around 31 megabytes.

D. User: IC Identification

Before IC identification can be carried out, the user has to calibrate the measurement equipment in order to find a suitable position for EM-probe to measure the EM leakage of the IC. For this purpose, the first L clock cycles after reset are used to transmit a calibration sequences—a fixed sequence of alternating ones and zeros—over the EM side channel using the same leakage circuits that are used to transmit the keystream afterwards. We assume that the frequency of the clock signal driving the IC protection circuit is public. Calibration is achieved by placing the EM probe in different positions on the IC and at the same time computing the correlation between the calibration sequence and the EM measurements on the fly. The best position to measure is the one where the highest absolute correlation is computed according to Equation (1).

In the next step, IC identification takes place. First, the user downloads the identification sequences and their corresponding offsets associated with the IC under test by querying the OCM with the serial number written on the package of the IC. Next, the user measures the EM leakage of the given IC with a digital oscilloscope and compresses the measurement by averaging all samples of a clock cycle. Figure 4 shows an example of compressed EM measurements for three ICs. The identification sequences are correlated to the compressed measurement at given offsets after reset. We use Pearson's correlation coefficient to compute whether an identification sequence is leaked over the side channel:

$$\rho(S_n^r, M_n^r) = \frac{\sum_{i=0}^{L-1}(s_{n,i}^r - \hat{s}_n^r)(m_{n,i}^r - \hat{m}_n^r)}{\hat{\sigma}_{S_n^r}\hat{\sigma}_{M_n^r}}, \tag{1}$$

where r is the current release number, S_n^r is the r-th released identification sequence of n-th IC, $s_{n,i}^r$ is the i-th bit of the identification sequence S_n^r, $m_{n,i}^r$ is the i-th value in the compressed side channel measurement M_n^r starting from the offset t^r of the IC recorded after reset, L is the length of identification sequences, $\hat{s}_n^r$ and $\hat{m}_n^r$ are the sample means of S_n^r and M_n^r, respectively, and $\hat{\sigma}_{S_n^r}, \hat{\sigma}_{M_n^r}$ are their respective sample standard deviations.

To compute the significance of correlation coefficient, a threshold value is defined using a stated confidence level $1-\alpha$ and the given sample size L. Let $z_{\alpha/2}$ be the point on the standard unit normal distribution exceeded with probability $\alpha/2$. We use the upper bound of the two-sided confidence interval that is computed according to Bonett and Wright [27]. Assuming the sample correlation coefficient is zero, one obtains the threshold:

$$\text{threshold} = \frac{e^{2K_2} - 1}{e^{2K_2} + 1}, \tag{2}$$

with

$$K_2 = \frac{z_{\alpha/2}}{\sqrt{L-3}} \tag{3}$$

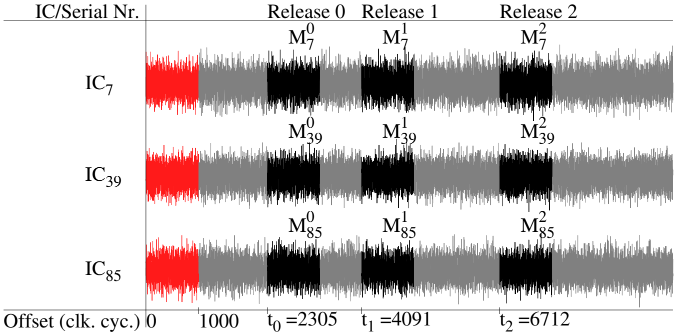


Fig. 4: An example of EM measurements compressed to one sample per clock cycle for 3 ICs and 3 releases. Parts of the measurements that are relevant for IC identification are shown in black; irrelevant parts in gray. The first $L = 1000$ clock cycles (shown in red) of each measurement are used for calibration.

In our experiments, $z_{\alpha/2} = 7$ turned out to be an appropriate threshold for separating distributions of correlated and uncorrelated measurements and identification sequences. Our approach computes the correlation coefficient between a Gaussian distribution (side channel measurements) and a uniformly distributed Bernoulli distribution (identification sequences), whereas Equation (2) is used as a lower bound for approximating the confidence level, i.e., the theoretical confidence level of $z_{\alpha/2} = 7$ for two Gaussian distributions is higher than the confidence level of the threshold observed in our experiments.

The threshold is used to identify ICs as follows:

$$\text{IC}_n \text{ is} \begin{cases} \text{genuine,} & \text{if } |\rho(S_n^r, M_n^r)| \geq \text{threshold} \\ \text{a counterfeit,} & \text{if } |\rho(S_n^r, M_n^r)| < \text{threshold} \end{cases} \tag{4}$$

E. IC Identification Statistics

The presented approach opens up a feedback channel from users querying serial numbers to the OCM. The OCM can estimate the number of counterfeits in circulation and infer the effectiveness of the counterfeit protection method in use.

IV. Experimental Evaluation

The main purpose of our experiments is to determine whether or not, and under which conditions, ICs in deployed systems can be identified by measuring electromagnetic emanation of the chips.

A. Experimental Setup

We evaluate our approach using three different FPGA boards: Spartan-3 Starter Kit board [28], Altera DE2-70 board [29], and Sasebo-GII board [30]. Each board has a different FPGA of different size, as well as the quantity of available periphery. The Spartan-3 board has the smallest FPGA, and Altera DE2-70 board the largest. In addition, the DE2-70 board has the largest number of periphery devices.

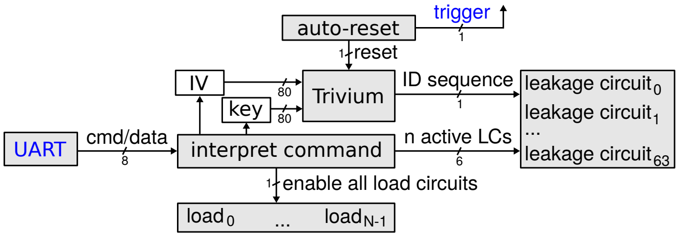


Fig. 5: Top-level entity used in experiments on all three FPGA boards.

The Virtex-5 FPGA of the Sasebo-GII board uses the smallest technology of 65 nm out of all three boards.

We have developed a proof-of-concept design in VHDL. A high-level view of our design is shown in Figure 5. Some parameters, such as key, IV, number of active leakage circuits, and the enable signal for the load circuits can be controlled at run time over UART interface. In a real-world design, the parameters will be fixed at the design time and stay constant throughout the lifetime of the chip. The system auto-resets itself after a fixed number of clock cycles. Upon reset, the system sets a trigger signal for one clock cycle on an external pin of the board that is used to start the recording of the side-channel traces. In a deployed system, where no trigger signal is available, the reset must be actively issued before the SC measurement.

At the core of the counterfeit protection circuit is a Trivium [31] circuit that is initialized with a key/IV pair upon a chip reset. Trivium is a light-weight hardware-tailored stream cipher that requires 288 flip-flops to hold the state, 3 AND gates, and 11 XOR gates [31]. Trivium uses an 80-bit key, an 80-bit initialization vector, and delivers a long sequence of 2^{64} bits. Its internal state consists of three shift registers. So far, there is no cryptanalytic attack on Trivium that is better than brute-force, so Trivium is considered to be secure [32].

The system has the capacity of turning on 64 leakage circuits in total. We have performed our experiments by letting the counterfeit protection circuit run alone, and under additional load, where the load was generated by AES-128 cores in CBC mode each encrypting random initial plaintexts. To desynchronize the operation of AES cores, each core is paused and enabled by a dedicated 128-bit LFSR, each initialized by a random seed. The load circuits are unaffected by the auto-reset of the Trivium module and continue to run until the board is powered off. Table II shows the resource usage of one leakage circuit, Trivium, and the load circuits for each FPGA. The Spartan-3 FPGA can only fit one load circuit, and the Cyclone-2 FPGA can fit 40 load circuits. The designs of all three boards are driven by a 50 MHz clock, which results in keystream generated at a bit rate of 50 Mbps.

Figure 6 shows our setups. A good position for the EM probe that also results in reproducibly high correlation across all three setups can be found at the main decoupling capacitors of each FPGA, as shown in Figure 6. Though, measuring with an EM probe directly at the IC or on the backside of the boards yields better results in general. We use Langer EM probe RF U 5-2 in combination with operation amplifier Langer PA 303 to measure EM leakage. The probe is designed for measuring magnetic field over surfaces and current in the wires. Side channel measurements were recorded using a digital oscilloscope PicoScope 6402C at a sampling rate of 156 MHz. Figure 7 shows examples of EM measurements recorded for each board, their compressed versions, and the keystream bits transmitted at corresponding clock cycles.

TABLE II: Resource usage.

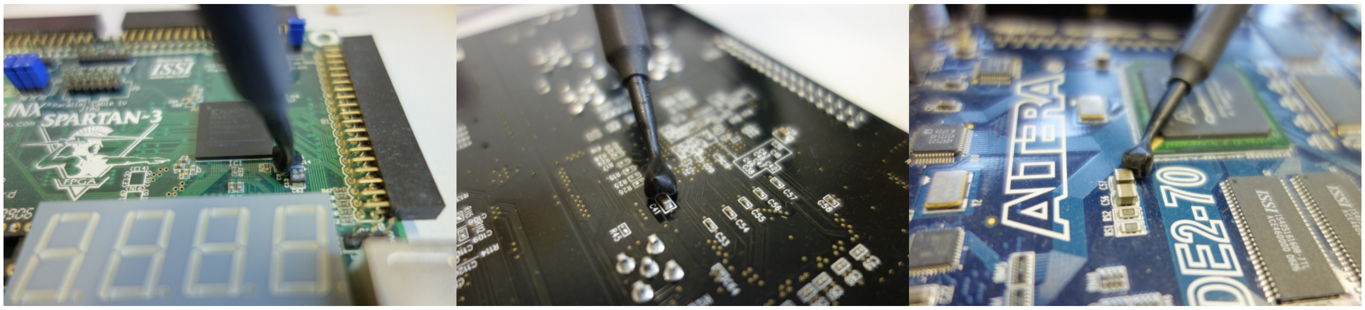
Board	Device	Circuit	Slices	Logic Cells	LUTs
Digilent	Spartan-3 (90 nm),	One leakage circuit	6	-	11 (0.3%)
Spartan-3	XC3S200FT256	Trivium	147	-	166 (4.3%)
Board	total: 3840 LUTs	Load circuits (1)	788	-	1390 (20.5%)
Sasebo-GII	Virtex-5 (65 nm),	One leakage circuit	5	-	7 (<0.1%)
	XC5VLX50-1FFG324	Trivium	76	-	164 (0.6%)
	total: 28800 LUTs	Load circuits (26)	7826	-	24674 (85.7%)
DE2-70	Cyclone-2 (90 nm),	One leakage circuit	-	6	6 (<0.1%)
	EP2C70F896C6N	Trivium	-	292	181 (0.3%)
	total: 68416 LUTs	Load circuits (40)	-	47160	44840 (65%)

B. *Experiments*

To perform experiments, a table of 1000 unique key/IV pairs was randomly generated. For each key/IV pair, 50k bits of Trivium keystream were computed. The key/IV pairs were subsequently loaded onto the three FPGAs boards. Their EM leakage of the first 50k clock cycles was measured and compressed to one measurement sample per clock cycle for each key/IV pair and board, respectively. To reduce the noise, an average of ten measurements was taken for each key/IV pair and FPGA board combination.

In the first experiment, for each key/IV pair we have chosen 10 non-overlapping identification sequences at random offsets. The identification sequences were correlated at all possible offsets (49k offsets in total) with all 1000 compressed EM measurements. For each identification sequence and key/IV pair combination, this resulted in 1000*1*49k correlations at correct offsets, and 1000*999*49k correlations at incorrect offsets. However, only two numbers were retained: the maximum absolute correlations at correct and incorrect offsets. Figure 8 shows that correlation is significant when identification sequences originate from the same key/IV pair that is embedded on the measured FPGA, designated by black dots in the figure. Identification sequences and EM measurements that originate from different key/IV pairs result in correlation approaching zero shown by gray dots in the figure. The lower subfigures of Figure 8 show the experiments using the same identification sequences to distinguish key/IV pairs in FPGAs under load. For the same key/IV pairs, the maximum correlation is lower than in the load-free case.

Out of 48 billion correlations at incorrect offsets, one exceeds the threshold $z = 7$ for the Spartan-3 and Sasebo-GII setups with and without load, respectively. This one correlation suggests that the IC under test is genuine, which is a false positive. However, since there are always several identification sequences available for each IC, it is unlikely that all of them will exceed the threshold. To reduce the



(a) Spartan-3 board, measuring at C62
(b) Sasebo-GII board, measuring at C41
(c) DE2-70 board, measuring at C57

Fig. 6: Measuring the electromagnetic leakage at the main decoupling capacitors of three FPGA boards.

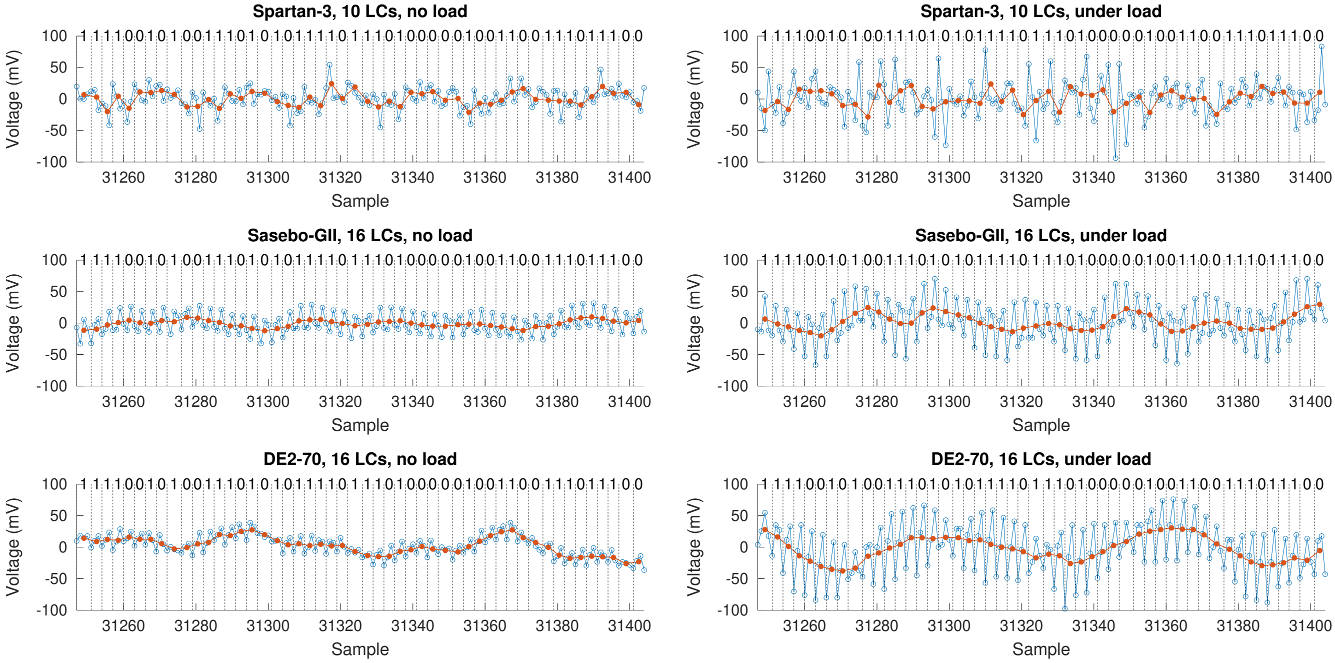


Fig. 7: Examples of EM traces recorded at the decoupling capacitors of the three evaluation boards. The keystream is generated at 50 MHz, while the oscilloscope is sampling at 156 MHz, giving us 3.125 samples per clock cycle. Blue lines and empty circles show the recorded EM data. Red lines and filled circles show the compressed traces with one value per clock cycle. Dashed vertical lines show the beginning of a clock cycle. 30 bits of the keystream are shown at the top of each subfigure, centered around their corresponding clock cycle.

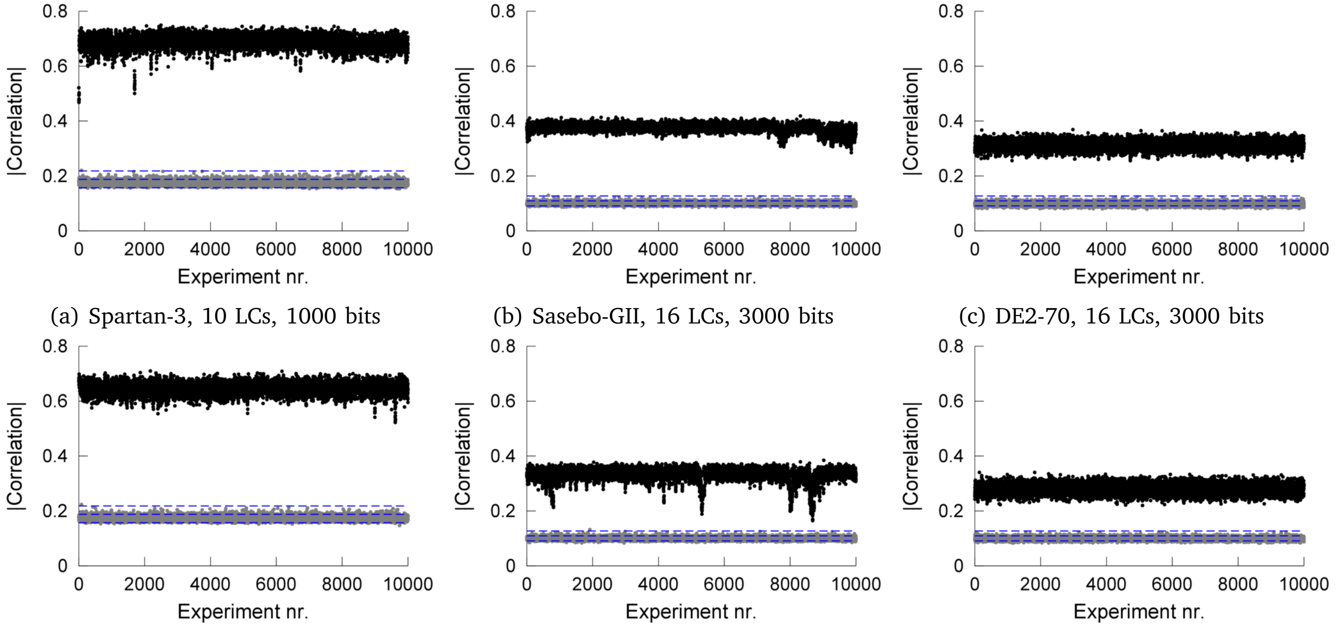
number of false positives, the threshold can be set to higher values of z. It is noted that the Sasebo-GII board was powered over a USB cable, which can explain time-localized drops in maximum correlation coefficient.

Figure 9 shows the absolute correlation of identification sequences with the EM measurements for sequences of different lengths L at correct (black) and incorrect (gray) offsets for systems under load. For all three boards it is possible to use shorter sequences and still distinguish the two sets—the identification sequences can be successfully matched using a few hundreds of bits when only the counterfeit protection circuit is active on the FPGA, and require several thousands of bits when the FPGA is under additional load. As expected, longer sequences increase the signal-to-noise ratio and reduce the variance of the signal. Increasing the length of identification sequences allows us to identify correct ICs more reliably. The absolute correlation for the same key/IV pair has lower variance with increasing length of identification sequences.

Figure 10 shows the examples of IC identification for systems under load. For each board, we computed the absolute correlation coefficient for 12 identification sequences at all 49k possible offsets. At the correct offsets, the correlation is high and well above the threshold, and low elsewhere.

The effect of adding errors into the keystream is illustrated in Figure 11. This increases the difficulty of differential attacks on the cipher at the cost of lower absolute correlation at correct offsets. For example, an error rate of 0.1 means that 10% of the bits of the identification sequence have been flipped.

The clock signal of the oscilloscope and the system under test might diverge from each other slowly over



(a) Spartan-3, 10 LCs, 1000 bits

(b) Sasebo-GII, 16 LCs, 3000 bits

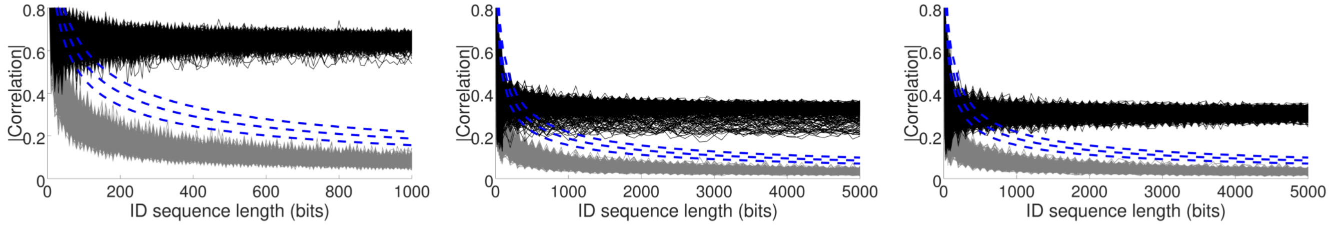
(c) DE2-70, 16 LCs, 3000 bits

(d) Spartan-3, 10 LCs, 1000 bits, load

(e) Sasebo-GII, 16 LCs, 3000 bits, load

(f) DE2-70, 16 LCs, 3000 bits, load

Fig. 8: Absolute maximum correlation of 10k identification sequences at correct offsets (black dots) and incorrect offsets (gray dots) with EM measurements of the three setups. Setup name, the number of used LCs, and the length of identification sequences are given in the captions below each figure. (a,b,c) Systems run without additional load. (d,e,f) Systems run with additional load. Blue horizontal lines show the thresholds for $z = 5, 6, 7$. Only the maximum absolute correlations of identification sequences and the EM traces at incorrect offsets are shown.

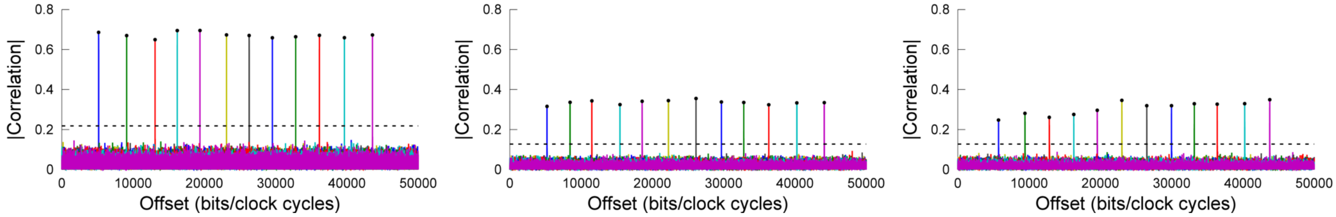


(a) Spartan-3, 10 LCs

(b) Sasebo-GII, 16 LCs

(c) DE2-70, 16 LCs

Fig. 9: Absolute correlation of 1000 identification sequences of varying length with 1000 EM measurements of systems under load at correct offsets (black), and at incorrect offsets (gray). Dashed lines show the thresholds for $z = 5, 6, 7$.



(a) Spartan-3, 10 LCs, 1000 bits

(b) Sasebo-GII, 16 LCs, 3000 bits

(c) DE2-70, 16 LCs, 3000 bits

Fig. 10: Examples of IC identification with 12 identification sequences at random non-overlapping offsets for systems under load. Dashed horizontal lines show the thresholds for $z = 7$.

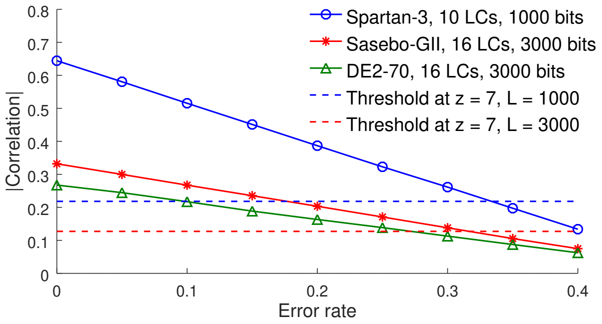


Fig. 11: Mean absolute correlation of 1000 EM measurements of systems under load and erroneous keystream with different error rates at random offsets.

time. Figure 12 shows absolute correlation of identification sequences of 1000 bits at 1, 2, . . . , 10 seconds after reset and the EM measurements at around the same offsets. The system and the measurement clocks run apart from each other by approximately 15 clock cycles (300 ns) per second. This can be accounted for by using a time window of ±200 clock cycles centered around a given offset when computing the correlation coefficient. The correlation should be computed at all offsets in the window. The window approach can be also used in cases when the actual IC reset happens after a short but unknown period of time after the user has triggered the reset of the IC. Note that because we measure the EM leakage at the decoupling capacitor, the correlation peaks last for around 8 clock cycles.

The results in this section, and resource usage from Table II suggest that our protection scheme can be used on relatively small ICs. As demonstrated in the experiments with the XC3S200FT256 FPGA that has 3840 of 4-input LUTs in total, our approach needs 166 LUTs for Trivium plus 10*7=70 LUTs for ten leakage circuits, which makes 236 LUTs in total. Without accounting for the antifuse memory, the protection circuit makes about 7% of total available LUTs on the Spartan-3 FPGA.

It takes approximately one minute to find a suitable measuring position for an EM probe in a previously unseen setup. Subsequent measurements of identical setups will need less time. Recording one EM measurement after waiting for the offset and computing the correlation between the measurement and three identification sequences takes less than a second. The computations were carried out in Python 2.7.12 on an Intel i5-2435M processor with two cores running at 2.40 GHz.

V. Possible Attacks and Countermeasures

A. Remarked ICs

In case that the attacker remarks an IC that has been protected using our approach into the same IC of a higher grade, he has to know one key of any of the higher-grade ICs, but also needs to be able to change the content of the AF memory of the lesser-grade chip. This attack requires package removal, tampering, and repackaging for each

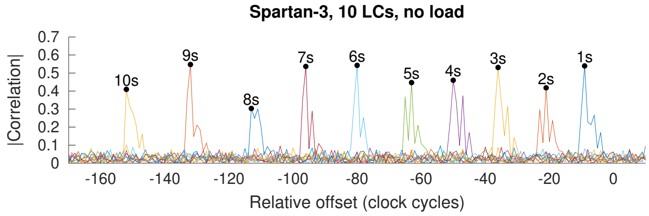


Fig. 12: Analysis of divergence between the clock of the oscilloscope (156 MHz) and the clock of the protection circuit (50 MHz). The x-axis shows the clock cycles relative to the main offets taken at 1, 2, . . . , 10 seconds after reset. The most-right correlation peak shows the highest absolute correlation coefficient between correct identification sequence of 1000 clock cycles and one EM-measurement at one second (50 mil. clock cycles minus approx. 10 clock cycles) after reset. The peak to its left shows the correlation coefficient after 2 seconds (100 mil. clock cycles minus approx. 20 clock cycles).

counterfeited IC. However, AF memory is not only tamper-resistant, but is also able to withstand existing reverse-engineering attacks [24], [33].

B. ICs Remarked to Another Brand

In this scenario, the attacker either buys or in some way obtains a number of legitimate ICs protected by our scheme, and remarks them to have some other company's logo. Then he releases it back into the market. Our approach is suitable for detection of such ICs. Since the ICs have been remarked, the OCM does not know the original serial number, which also means that the corresponding identification sequence is also not known initially. To prove that the IC has been remarked, the OCM records one EM-trace at the correct offset, and performs inverse search by correlating the measurement with all identification sequences of one release in order to find the actual serial number of the IC. On our machine with two Intel i5 cores running Ubuntu, it takes about 30 seconds to compute correlation of one EM-trace with one million identification sequences with the length of 1000 bits. Even though this takes more time than IC identification with known serial numbers, it is clearly possible to identify plagiarized ICs in a reasonable amount of time.

C. Overproduced and Out-of-Spec ICs

A malicious foundry can produce an extra batch of chips without OCM's consent. Overproduced ICs are not subject to the quality control of the OCM, so that some of them might be out-of-spec or faulty. Our approach cannot prevent a malicious foundry from creating overproduced and out of spec ICs. However, since the foundry has no access to the keys and IVs written into genuine ICs, overproduced and out-of-spec ICs will not pass the identification tests.

D. FIB Attacks on Key and Keystream

The attacker can use a focused-ion beam (FIB) workstation and try to read out the internal state or the key during the runtime. As a countermeasure, obfuscations of the chip layout of the cipher and the leakage circuit can be added.

In another attack using FIB, the attacker can try to obtain the keystream by observing the behavior of the leakage circuits. If the attacker knows the output of the leakage circuit ahead of time, he can produce an IC that leaks the same keystream, even if he does not know the key. However, the identification sequences are published at random offsets. Since the attacker does not know which parts of the keystream will be used as identification sequences, he will be forced to reproduce the whole sequence, which will require a large amount of memory on the IC just to store a small portion of the keystream. For example, in our proof-of-concept implementation on three boards, we have used 50 MHz as the clock signal for Trivium, which gives us the data rate of 50 Mbps. Saving the sequence of 1 second losslessly requires $\frac{50000000}{8 \cdot 1024 \cdot 1024}\ bits \approx 5.96$ MB (megabytes) of ROM memory on the IC, whereas, saving 30 seconds requires 179 MB. At this data rate saving several seconds is very cost-inefficient for the attacker. In case that the attacker decides to compress the sequence, he will have to add a compression circuit. However, the compression is not effective on pseudo-random data. For example, the attacker might save only every second bit, and thus cut the amount of required memory in half. The known bits can be sent twice in a row over the EM channel. However, this strategy will also significantly reduce maximum correlation at correct offsets, as can be seen in Figure 11.

FIB attacks have to be done for each IC individually. Due to high costs of a FIB attack it is expected that the extracted key/IV pair of one successfully attacked chip is written into a whole batch of overproduced or cloned ICs. These ICs will be identified as authentic, however, the larger the batch with the same IDs, the more likely they will be detected. Having a unique key/IV pair per IC forces the attacker to invest a large amount of effort to carry out FIB attacks to get several key/IV pairs. Each IC must have the correct ID written on the package and the matching key/IV pair written in the antifuse memory.

E. Fake ICs with Hardcoded Published Sequences

The attacker can take the most recently published identification sequences and embed them into the IC together with some leakage circuit, so that the counterfeited IC will be identified as genuine using our scheme for a period of time between two releases. This will cause the SC measurements of this IC to correlate with this set of identification sequences. The user can detect this attack by waiting for a new release before deploying the IC.

F. Fake ICs With Internet Access

In a more sophisticated version of the above attack, the counterfeit IC is embedded in a system with Internet access. As soon as new batch of identification sequences is released by the chip manufacturer, the counterfeit system downloads the new correct identification sequence and plays it back at the correct offset. This approach might be successful for several releases, however, the sequences must be saved in non-volatile memory and played back at the right time after the reset. Potentially a large number of memory must be allocated to store the sequences. In addition, the user should cut the Internet access of the suspicious system during times when new identification sequences are released. The best countermeasure is to record one long EM measurement that includes a large number of clock cycles right after obtaining the IC, and to match the measurement with new sequences when they are released. In this case, updating the IC with new sequences will be ineffective.

G. Used and Recycled ICs

Used ICs, and ICs that have reached their end of life can be recycled and resold as new. Our method cannot be used to identify recycled ICs, since they will have correct key/IV pair. Our method will be effective only if the markings of the ICs are changed during the recycling step.

H. Attacks on Trivium

Due to its simple and elegant design, Trivium is an attractive candidate for ongoing cryptanalytic research. On round-reduced versions of Trivium, cube attacks in a chosen IV scenario and algebraic attacks using different values for the IV are reported [32]. We stress that full-round Trivium is still not broken, meaning that there are no known attacks that can recover the key from known keystream and IV. Further, both cube attacks and algebraic attacks are not applicable to our concept as each IC uses a fixed and unknown IV. The option of introducing errors in the published ID sequence fundamentally enhances the resistance on any cryptanalytic attack.

We note, that simple power analysis (SPA) [34] on the keystream bits is assumed to be applicable by observing the processing of the leakage generator. However, as shown in Figure 7, it is not possible to determine the value of the bit that is sent out over the EM side channel with a single measurement. For noise reduction, averaging of leakage traces for several iterations is feasible. A careful leakage analysis for each keystream bit can result in the disclosure of a keystream bit sequence. The impact of SPA is, however, limited in our concept. Even if the adversary knows a high number of keystream bits, the internal state of Trivium of the genuine chip remains unknown and therefore the adversary cannot set up a Trivium circuit with the same initial state in a cloned design. The only remaining option is to store many disclosed key stream bits in memory in a cloned design which results in high costs as the adversary cannot foresee the offset of the next releases.

Further implementation attacks on a Trivium circuit include differential power analysis (DPA), e.g. [35], and differential fault analysis (DFA), e.g. [36]. DPA requires known or chosen IVs and is therefore not applicable to our design. DFA requires that consecutive keystream bits are known to the attacker, both for the error-free computation and for each erroneous computation, which is the reason why we add errors to the published parts of the keystream.

Especially for erroneous computations this either requires to successfully apply SPA with a single leakage trace or to induce the faults in a very precise and reproducible way. Both possibilities are susceptible to errors, which makes an application of DFA very difficult.

I. Cloned ICs

The purpose of cloning ICs is to save R&D cost by getting access to the netlist of the IC or by reverse-engineering it. There are two types of cloned ICs: 1) cloned ICs with forged marks that are made to look like the original markings; 2) cloned and rebranded ICs, where the purpose is to steal the design and to sell it under a different name. Our approach can be used to detect the first case, as the ICs will not have the registered keys. The second case poses a different problem of proving that an intellectual property (IP) theft has taken place. This problem can be addressed by reusing the leakage circuits in order to send a watermark that is unique in each IC design. For example, the fixed calibration sequence can be replaced by a watermark. IP ownership can be proven by computing a significant correlation between a side channel measurement and the watermark. This countermeasure increases attacker effort, who now has to find and remove the watermark.

VI. Conclusion

We have presented a novel approach for IC identification in which the original chip manufacturer writes unique key/IV pairs into each IC post-fabrication, and the user correlates the EM leakage of the IC with the publicly released identification sequences. Each key/IV pair is stored securely in antifuse one-time programmable memory, and is used to initialize a stream cipher. The keystream is leaked over the side channel by using several leakage circuits introduced in order to amplify the leakage. The identification sequences are frequently released by the chip manufacturer so that the IC owners can quickly identify their ICs at any time without tedious interaction with the chip manufacturer. The identification can be done using low-cost equipment—only a low-cost digital oscilloscope, an EM probe, and a standard PC are necessary. With our approach remarked, overproduced, and out-of-spec ICs can be detected.

Acknowledgment

The authors would like to thank Georg T. Becker, Christof Paar, and Amir Moradi for fruitful discussions, and Matthias Urfels for writing the AES core used in our experiments.

References

[1] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.

[2] M. Tehranipoor, H. Salmani, and X. Zhang, *Integrated Circuit Authentication: Hardware Trojans and Counterfeit Detection*. Springer Publishing Company, Incorporated, 2013.

[3] U.S. Department of Commerce, Bureau of Industry and Security, Office of Technology Evaluation, *Defense Industrial Base Assessment: Counterfeit Electronics*, 2010.

[4] N. Robson, J. Safran, C. Kothandaraman, A. Cestero, X. Chen, R. Rajeevakumar, A. Leslie, D. Moy, T. Kirihata, and S. Iyer, "Electrically programmable fuse (eFUSE): From memory redundancy to autonomic chips," in *2007 IEEE Custom Integrated Circuits Conference*, Sept 2007, pp. 799–804.

[5] M. Miller, J. Meraglia, and J. Hayward, "Traceability in the age of globalization: A proposal for a marking protocol to assure authenticity of electronic parts," in *SAE Technical Paper*. SAE International, Oct. 2012.

[6] R. Maes and I. Verbauwhede, *Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 3–37.

[7] G. DeJean and D. Kirovski, "RF-DNA: Radio-frequency certificates of authenticity," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, P. Paillier and I. Verbauwhede, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 346–363.

[8] C. Kuemin, L. Nowack, L. Bozano, N. D. Spencer, and H. Wolf, "Oriented assembly of gold nanorods on the single-particle level," *Advanced Functional Materials*, vol. 22, no. 4, pp. 702–708, 2011.

[9] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th Annual Design Automation Conference (DAC '07)*. New York, NY, USA: ACM, 2007, pp. 9–14.

[10] J. H. Anderson, "A PUF design for secure FPGA-based embedded systems," in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2010, pp. 1–6.

[11] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08)*, June 2008, pp. 67–70.

[12] L. Parrilla, E. Castillo, D. P. Morales, and A. García, "Hardware activation by means of PUFs and elliptic curve cryptography in field-programmable devices," *Electronics*, vol. 5, no. 1, 2016.

[13] M. M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*. Cham: Springer International Publishing, 2015.

[14] Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *Proceedings of 16th USENIX Security Symposium*, ser. SS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 291–306.

[15] J. Huang and J. Lach, "IC activation and user authentication for security-sensitive systems," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, June 2008, pp. 76–80.

[16] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending piracy of integrated circuits," in *Design, Automation and Test in Europe (DATE '08)*, March 2008, pp. 1069–1074.

[17] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC piracy using reconfigurable logic barriers," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 66–75, Jan. 2010.

[18] D. Ziener and J. Teich, "Power signature watermarking of IP cores for FPGAs," *Journal of Signal Processing Systems*, vol. 51, no. 1, pp. 123–136, 2008.

[19] D. Ziener, F. Baueregger, and J. Teich, "Using the power side channel of FPGAs for communication," in *18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM '10)*, May 2010, pp. 237–244.

[20] G. Becker, M. Kasper, A. Moradi, and C. Paar, "Side-channel based watermarks for integrated circuits," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST '10)*, June 2010, pp. 30–35.

[21] L. Bossuet, V. Fischer, and P. Bayon, "Contactless transmission of intellectual property data to protect FPGA designs," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Oct 2015, pp. 19–24.

[22] L. Bossuet, P. Bayon, and V. Fischer, "Electromagnetic transmission of intellectual property data to protect FPGA designs," in *VLSI-SoC: Design for Reliability, Security, and Low Power*, Y. Shin, C. Y. Tsui, J.-J. Kim, K. Choi, and R. Reis, Eds. Cham: Springer International Publishing, 2016, pp. 150–169.

[23] T. Kean, D. Mclaren, and C. Marsh, "Verifying the authenticity of chip designs with the DesignTag system," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08)*, June 2008, pp. 59–64.

[24] B. Stamme, "Anti-fuse memory provides robust, secure NVM option," July 2012. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1279746

[25] L. Hong, *Comparison of Embedded Non-Volatile Memory Technologies and Their Applications*, Kilopass, May 2009. [Online]. Available: http://www.kilopass.com/wp-content/uploads/2010/04/comparison_of_embedded_nvm.pdf

[26] L. Parrilla, E. Castillo, E. Todorovich, A. García, D. Morales, and G. Botella, "Improvements for the applicability of power-watermarking to embedded IP cores protection: e-coreIPP," *Digital Signal Processing*, vol. 44, pp. 110 – 122, 2015.

[27] D. G. Bonett and T. A. Wright, "Sample size requirements for estimating Pearson, Kendall and Spearman correlations," *Psychometrika*, vol. 65, no. 1, pp. 23–28, Mar 2000.

[28] *Spartan-3 Starter Kit Board User Guide: Version 1.1*, Xilinx, May 13 2005.

[29] *Altera DE2-70 User Manual: Version 1.08*, Terasic Technologies, 2009.

[30] *Side-channel Attack Standard Evaluation Board SASEBO-GII Specification: Version 1.01*, RISEC, AIST, Japan, November 30 2009.

[31] C. De Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles," in *9th International Conference on Information Security (ISC '06)*, S. K. Katsikas, J. López, M. Backes, S. Gritzalis, and B. Preneel, Eds. Berlin, Heidelberg: Springer, 2006, pp. 171–186.

[32] F.-M. Quedenfeld and C. Wolf, "Advanced algebraic attack on Trivium," in *Mathematical Aspects of Computer and Information Sciences*, I. S. Kotsireas, S. M. Rump, and C. K. Yap, Eds. Cham: Springer International Publishing, 2016, pp. 268–282.

[33] S. E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, and M. Tehranipoor, "A survey on chip to system reverse engineering," *Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 1, pp. 6:1–6:34, Apr. 2016.

[34] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer, 1999, pp. 388–397.

[35] W. Fischer, B. M. Gammel, O. Kniffler, and J. Velten, "Differential power analysis of stream ciphers," in *Topics in Cryptology – CT-RSA 2007*, M. Abe, Ed. Berlin, Heidelberg: Springer, 2006, pp. 257–270.

[36] M. Hojsík and B. Rudolf, "Differential fault analysis of Trivium," in *Fast Software Encryption*, K. Nyberg, Ed. Berlin, Heidelberg: Springer, 2008, pp. 158–172.