Applying Thermal Side-Channel Attacks on Asymmetric Cryptography

Aljuffri, Abdullah; Zwalua, Marc ; Reinbrecht, Cezar Rodolfo Wedig; Hamdioui, Said; Taouil, Mottaqiallah

DOI
10.1109/TVLSI.2021.3111407

Publication date
2021

Document Version
Final published version

Published in
IEEE Transactions on Very Large Scale Integration (VLSI) Systems

Applying Thermal Side-Channel Attacks on Asymmetric Cryptography

Abdullah Aljuffri, *Member, IEEE*, Marc Zwalua, Cezar Rodolfo Wedig Reinbrecht, Said Hamdioui, *Senior Member, IEEE*, and Mottaqiallah Taouil, *Member, IEEE*

***Abstract*— Side-channel attacks (SCAs) are powerful attacks that could be used to retrieve keys from electronic devices. Several physical leakage sources can be exploited in SCAs, such as power, time, heat, and so on. Heat is one of the side-channels that is not frequently analyzed by attackers in the literature due to the high noise associated with thermal traces. This article investigates the practicality of adapting power-based SCAs [i.e., correlation power analysis (CPA) and deep-learning-based power attacks (DL-based PA)] for thermal attacks and refer to them as correlation thermal attack (CTA) and DL-based thermal attack (DL-based TA). In addition, we introduce a new attack called progressive CTA (PCTA). We evaluate the different thermal SCAs against an unprotected and protected software implementation of Rivest–Shamir–Adleman (RSA). Our results show the practicality of the three attacks (i.e. CTA, DL-based TA, and PCTA) as a 100% key recovery is realized.**

***Index Terms*— Power attack, progressive correlation, Rivest–Shamir–Adleman (RSA), side-channel attack (SCA), thermal attack.**

I. Introduction

WITHOUT proper security and protection against cyber-attacks, the fast-booming IoT market will not only put our private data at risk [1] but eventually also threaten the economy [2]. Among the cybersecurity attacks, hardware attacks have been gaining huge attention recently. Examples are fault injection [3], side-channel attack (SCA) [4], and hardware trojans [5]. Side-channel analysis is considered as one of the highest threats [4]. The reasons for that are advances in the analysis tools [e.g., deep learning (DL)] [6], the reduction of measuring equipment costs [7], and the easy access to IoT devices as they are placed in the field. In the past 20 years, several SCAs have been investigated in the literature, mostly focusing on power and timing attacks. However, in the context of IoT, power or time measurements might be very challenging. IoT devices typically use proprietary software bound to their hardware platform, which makes it very complex to inject code to perform time-based attacks [8]. In addition, electronic devices typically have onboard power regulators for efficient energy management [9]. This poses a critical limitation to perform power attacks, as monitoring the power consumption before a regulator is not suitable for attacks [10]; hence, tampering with the devices is required. A promising alternative to attacking such IoT devices is using heat as a side-channel. Capturing thermal traces is feasible since a sensor can be attached on top of the package, on top of the die (i.e., semi-invasive approach), or could be already present inside the system/chip [11], [12].

Only limited work has focused on the thermal side-channel due to the complexity of the attack (i.e., require expert skills) and noisy traces [11]. Brouchier *et al.* [11], Masti *et al.* [12], and Bartolini *et al.* [13] explored heat as covert channel and means to send hidden messages outside a system. Although it uses the same theory, covert channels are not considered SCAs. To the best of our knowledge, only two publications used heat in a SCA [14], [15]. Hutter and Schmidt [14] evaluated the possibility of a thermal SCA. Although they did not perform a direct thermal SCA, they were able to design a fault injection attack assisted by thermal leakages. Despite their success, it is a very complex semi-invasive attack whose threat model is inapplicable for most IoT devices due to fault injection requirements. More recently, Dey *et al.* [15] proposed a thermal SCA using DL. The attack focused on symmetric encryption advanced encryption standard (AES) running on multiprocessed systems. To the best of our knowledge, the latter represents the first practical demonstration of a pure thermal SCA. Nevertheless, it is still unclear if other SCA methodologies can be adapted for thermal attacks. On top of that, novel methodologies might be proposed if thermal leakage is better studied.

In this article, we perform a comprehensive investigation on the practicality of performing thermal SCAs. We target asymmetric cryptographic algorithms as they typically are more computationally intensive than symmetric ones, which, in theory, mean that they generate more heat. Since the temperature has a slow response compared with power, intensive computational tasks provide better quality traces, as shown in [12]. Hence, we show how known SCA techniques, such as simple power analysis (SPA), correlation power analysis (CPA), and deep-learning power analysis (DL-SCA) can be

Manuscript received March 27, 2021; revised July 6, 2021 and August 9, 2021; accepted August 29, 2021. Date of publication September 17, 2021; date of current version November 3, 2021. *(Corresponding author: Abdullah Aljuffri.)*

Abdullah Aljuffri is with the Department of Computer Engineering, EEMCS Faculty, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with King Abdulaziz City for Science and Technology, Riyadh 12354, Saudi Arabia (e-mail: a.a.m.aljuffri@tudelft.nl; aaljuffri@kacst.edu.sa).

Marc Zwalua, Cezar Rodolfo Wedig Reinbrecht, Said Hamdioui, and Mottaqiallah Taouil are with the Department of Computer Engineering, EEMCS Faculty, Delft University of Technology, 2628 CD Delft, The Netherlands.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2021.3111407.

Digital Object Identifier 10.1109/TVLSI.2021.3111407

adapted for thermal attacks. In addition, we propose a novel thermal attack by modifying the CPA attack; it achieves a successful and complete key recovery. We refer to this attack as progressive CTA (PCTA). All these attacks have been evaluated on unprotected and protected versions of an Rivest–Shamir–Adleman (RSA) software implementation. Finally, we present a comparison of all these SCAs techniques using both thermal and power leakage to clarify how powerful temperature-based attacks can be. The main contributions of this article can be summarized as follows.

1) Evaluation and adaptation of SPA, CPA, and DL-SCA for thermal analysis.
2) Proposal of a novel attack called PCTA and its variant for power (PCPA) as a side-channel.
3) Evaluation and comparison of thermal SCAs on unprotected and protected RSA cryptosystem implementations.
4) Comparison between the attack accuracy of thermal and power attacks.

The remainder of this article is organized as follows. Section II explains the related work in thermal side channels. Section III provides a background on RSA and SCAs. Section IV describes the thermal SCAs that are derived from classical power-based SCAs. SectionV introduces our proposed progressive CTA. Section VI presents the experimental setup and results. Section VII discusses the limitations and future work of the explored attacks. Finally, Section VIII concludes this article.

II. Related Work

Several articles addressed the topic of temperature and thermal side-channel. However, they typically focus on the manipulation of the system by using heat as means to send hidden messages to the outside [11]–[13]. Such an approach is known as a covert channel and is very useful today in many dangerous threats, like specter [16] and meltdown [17]. Only two articles presented practical attacks using thermal leakage, in which the only one of the two can be considered purely a SCA.

Brouchier *et al.* [11] used specific programs to heat the processor and encode information through the cooling fan behavior; no fan activity represents a binary zero, while the opposite a binary 1. They successfully demonstrate that the heat can be manipulated by an attacker and exploited externally by simply observing the system's behavior, i.e., the cooling system. Masti *et al.* [12] showed that processors inside a multicore system can leak or exchange information through the temperature side channel. They focused in this article mostly on a methodology to calibrate and perform interprocessor communication using temperature as a covert channel. They even showed that isolated cores are vulnerable to such a covert channel. In the last part of this article, we also presented an attack to identify running applications on the processor by correlating thermal traces. They use a profiled-based methodology, where the attacker has access to previous application traces. Despite their success, this attack is considered a coarse-grain side channel, which is not as critical as a fine-grain side-channel attack (e.g., the ones that are needed to target cryptographic keys). We mentioned that they were not able to perform a fine-grain SCA due to a low resolution and a low available sampling rate in most temperature sensors, and conclude that fine-grained attacks might not be practical. In [13], the authors also proposed a methodology to create covert channels through thermal leakage on multicore systems. Their threat model considers a spy process inside an isolated core leaking sensitive data. One of their most interesting contributions is the design of a transfer function that models the thermal leakage behavior. The objective was to estimate the maximum data capacity of the channel; this defines an upper bound on the amount of data that can be leaked through the covert channel. As a result, the authors improved the transmission rate by a factor of 20 as compared to previous work, achieving a rate of 27 bps with an 11% error rate. In summary, all these articles have presented interesting contributions on how thermal leakage can be exploited to transmit hidden messages. Their successful results indicate that meaningful data can be leaked through such channels, which motivates the scientific community to do more research on thermal SCAs.

Two articles in the literature focused on temperature SCAs for cryptographic algorithms. Hutter and Schmidt [14] investigated thermal leakage as a means to create new attacks. They first evaluate if Hamming weight (HW) can be correlated with thermal traces. The HW is one of the most used metrics to correlate keys with power traces in classical power SCAs. They showed that even under low resolution and low sampling rate, the HW could be correlated correctly. However, their study on pure side-channel analysis stopped with that. Thereafter, the authors used this information to create a fault injection attack on the RSA cryptosystem. In [15], the authors successfully developed a complete thermal SCA and successfully retrieved a cryptographic key. They targeted the AES cipher and used a convolutional neural network (CNN) to interpret the thermal leakage. Different from previous works, this article focused on mobile edge devices that use multiprocessor system-on-chips (MPSoCs). In their attack, the CNN was trained with thermal traces of AES encryptions using known keys. The chosen keys were based on the most common passwords that were used in the years 2017 and 2018. The authors provided a methodology to perform a thermal DL SCA on the AES cipher. This work has shown that temperature side channels can be successfully exploited. However, the possibility to break asymmetric cryptographic algorithms, such as RSA and elliptic curve cryptography (ECC), remains uncovered, which is the main objective of our work.

III. Background

This section presents background information on RSA cryptography and SCAs.

A. RSA Cryptography

RSA is a very popular algorithm that is used in various security-sensitive applications, such as bank transactions

Algorithm 1 Square–Multiply—Modular Exponentiation

1: **INPUT** (M, k, n); **where** M presents the text, k the key which can be represented by its binary representation as $k_0 \cdots k_{l-1}$ where $k_j \epsilon \{0, 1\}$, and n the modulus.
2: **OUTPUT**(R_0); **where** $R_0 = M^k mod(n)$
3: $R_0 = 1$
4: $R_1 = M$
5: **for** $j = 0 \leq l - 1$ **do**
6: $\quad R_0 = R_0^2 mod(n)$
7: $\quad$ **if** $k_j = 1$ **then**
8: $\quad\quad R_0 = R_0 R_1 mod(n)$
9: $\quad$ **end if**
10: **end for**
11: return R_0

Algorithm 2 Montgomery Ladder—Modular Exponentiation

1: **INPUT** (M, k, n); **where** M presents the text, k the key which can be represented by its binary representation as $k_0 \cdots k_{l-1}$ where $k_j \epsilon \{0, 1\}$, and n the modulus.
2: **OUTPUT**(R_0); **where** $R_0 = M^k mod(n)$
3: $R_0 = 1$
4: $R_1 = M$
5: **for** $j = 0 \leq l - 1$ **do**
6: $\quad$ **if** $k_j = 1$ **then**
7: $\quad\quad R_0 = Mont(R_0, R_1)$
8: $\quad\quad R_1 = Mont(R_1, R_1)$
9: $\quad$ **else if** $k_j = 0$ **then**
10: $\quad\quad R_1 = Mont(R_0, R_1)$
11: $\quad\quad R_0 = Mont(R_0, R_0)$
12: $\quad$ **end if**
13: **end for**
14: return R_0

and cloud authentication, and as key exchange protocols for symmetric cryptosystems [18], [19]. Although RSA is more efficient in hardware, many systems still perform RSA in software to prevent area overhead [20].

RSA, devised by Ron Rivest, Adi Shamir, and Leonard Adleman, in 1978, is one of the most famous public-key cryptography algorithms. RSA works as follows: to encrypt a message m the following modular exponentiation operation $c = m^e \text{mod}(n)$ is performed, where e, n, and c represent the public key, the modulus, and the ciphertext, respectively. The message can be retrieved from the ciphertext using a similar operation, i.e., $m = c^d \text{mod}(n)$, where d is the private key. The modular exponentiation is computed using square and multiply operations (see Algorithm 1). The algorithm scans the exponent bits one by one from left to right (line 5 of Algorithm 1) and based on its value either performs only a square operation when the exponent bit is 0 (see line 6) or both a square and multiplication (lines 6 and 8) when the exponent bit is 1 (line 7). Hence, there is a direct correlation between such operations and the key.

From an algorithmic point of view, RSA is considered secure. Its security is based on the hardness of the integer factorization problem [21]. In other words, to break RSA the modulus n must be factorized to its prime numbers, which is mathematically impossible when a large key is used. However, the implementation (such as the square–multiply algorithm) is vulnerable to SCAs and the key can be retrieved once the square and multiply operations have been identified. Many countermeasures have been proposed to defend against such attacks. One of the well-known methods is using Montgomery ladder [22], where both square and multiply operations are executed in the Montgomery domain. In the Montgomery ladder, both square and multiply have the same thermal behavior, making it difficult for an attacker to distinguish between them. This article focuses on a protected RSA implementation based on the Montgomery ladder, as shown in Algorithm 2.

B. Side-Channel Attacks

SCAs can be classified into nonprofiled and profiled attacks. Nonprofiled attacks only use traces measured directly from the target device. Profiled attacks construct a profile of the target device and use this profile to attack the target device. The profile is created from a similar device that is in the control of the attacker.

In this article, we describe the nonprofiled attacks based on their chronological order of appearance and evaluate how suited they are for thermal SCAs. For the nonprofiled attacks, we consider SPA [23], differential power analysis (DPA)/CPA [24], and machine learning (ML) [25]. For the profiled attacks, we only focus on DL-based SCA. Template-based attack (TBA) seems at first sight not worth investigating as its mathematical model uses the multivariate Gaussian distribution to build the profile. This mathematical function might not be appropriate for thermal leakage. Creating a suitable mathematical model for thermal is outside the scope of this article. In contrast, the DL approach has the ability to figure out a suitable model for the characterization and assessment of leakage behavior. Next, we describe the different attacks.

1) Nonprofiled—Simple Power Analysis: An SPA attack can be performed by simply observing the differences in power consumption of the targeted operation during its execution. Note that there are no special mathematical calculations required to apply this type of attack. One of the famous examples is the attack on the naive RSA implementation based on the multiply–square algorithm, as described in Section III-A. The attacker is able to recognize the key by observing the peak power values of the square and multiply operations during encryption/decryption [23].

2) Nonprofiled—Differential Power Analysis: One of the most powerful DPA attacks is CPA [24]. The correlation SCAs work as follows for RSA: given a number of recorded traces N of an RSA implementation, the used key can be retrieved using the Pearson coefficient correlation, which is calculated in (1). Note that the goal here is to identify whether the executed operation is a square or multiply. In (1), $h_{k,i}$ is the HW of the ith operation (i.e., square and multiply), while k is the bit location of the encryption/decryption execution. The $t_{k,j}$

denotes the sample point j within the subtrace k

$$r_{i,j} = \frac{\sum^{n}(h_{k,i} - \mu_{h_i})(\dot{t}_{k,j} - \mu_{t_j})}{\sqrt{\sum^{n}(h_{k,i} - \mu_{h_i})^2 \dot{\sum}^{k}(t_{k,j} - \mu_{t_j})^2}}. \quad (1)$$

3) Profiled SCA—Deep Learning Power Attack: Profiled DL SCA consists of two phases, namely, the *profile phase* and *extract phase* [26].

In the *profile phase*, the attacker first selects a sample device that is similar to the target device. Then, the attacker chooses and locates an intermediate point of attack (e.g., exponent operations in RSA). Next, the attacker records multiple power traces of the selected target operation and subsequently aligns them. Thereafter, a DL neural network is designed and trained to characterize the traces. Considering RSA, the neural network is trained to distinguish the main operations which are directly related to the key bit (i.e., multiply or square). Therefore, these two operations are the labels that can be used to determine whether a key bit is "0" or "1." To train the neural network, the attacker divides the dataset (i.e., traces and their labels) into a training set (normally 80%–90% of the complete dataset) and a validation set. The training and validation phases are completed when the attacker achieves an acceptable accuracy level.

In the *extraction phase*, the attacker aims to recover the secret information from the target device. First, the attacker locates the intermediate operation in the target device, i.e., the operation that was used to train the neural network during the profiling phase. After that, he/she generates a new set of traces on the target device for the intermediate operation. The traces are aligned and used in the inference phase. Finally, the attacker reveals the secret key information by predicting the key of the newly generated traces using the previously trained neural network. In RSA, the key is recovered bit-by-bit based on the predicted operations.

IV. Thermal Side-Channel Attacks

This section shows how classical power attacks can be adapted and used for thermal attacks. First, we describe the challenges associated with thermal SCAs. Thereafter, we present our threat model considered for all the implemented attacks. Finally, we describe the methodology used in each thermal attack.

A. Challenges of Thermal SCA

Similarly, as is the case for power attacks, the HW and Hamming distance (HD) can be used to model the thermal activity of a processor [14]. If an operand with a large HW value is used in a serial multiplier, it will produce more heat with respect to the case where the HW is low. Consequently, the temperature will rise. However, there are a few differences compared with the power consumption that cannot be modeled by HW or HD, such as the accumulative effect of temperature over time. Therefore, a model closer to the physical behavior is needed to help us to understand the thermal leakage.

One way of modeling the physical behavior of the thermal leakage is by analyzing the system as an RC-network [27]. This network behaves like a low-pass filter with a cutoff frequency somewhere in the kilohertz range. This frequency response could pose a problem since most computers tend to run in the gigahertz range. With a low-pass filter (even if it is the first order), it is very hard to measure any useful data when a system is running at 800 MHz. Luckily, it is not required to capture every clock cycle for side-channel analysis, as long as multiply and square operations can be differentiated. If these operations take long enough, differences should be visible in the thermal traces.

Another issue with thermal leakage is that they have an integrative behavior, i.e., the temperature accumulates when a task is executed (e.g., RSA decryption). The contribution of thermal leakage from a previous operation is still present in its following operation. Hence, it is likely that the same operation (e.g., a multiplication) at a different time moment will have a different starting temperature. To overcome this, two approaches are possible. In the first approach, pauses can be inserted between the operations. This can be achieved by periodically stopping the clock or by introducing pauses after each operation (e.g., periodically forcing an interruption in the processor). The second approach consists of rapidly cooling the system after each operation. This can be realized by adding external cooling, such as a high-speed fan.

The last issue thermal traces suffer from is the variation of temperature offset with time. While the power is regulated by a voltage controller, the temperature of the environment and the processor are not directly controlled. As a result, the offset varies multiple times during execution. Today, most systems employ dynamic clock and voltage scaling to keep the temperature in a certain safe range [28]. Consequently, the thermal leakage also presents drifts in the offset during execution, but in a much regular behavior. Therefore, thermal side-channel analysis requires a mechanism to filter out such drifts, as they most likely will not be able to work on traces with varying offsets.

B. Threat Model

Our threat model for thermal SCAs consists of the following assumptions.

1) The attacker has direct access to the target device in order to record the thermal traces of the executed decryptions.
2) The attacker has access to the ciphertext.
3) The attacker can acquire a similar device. We target devices using off-the-shelf parts. such as ARM, AVR, and so on.
4) The attacker has the ability to slow down the target operation (i.e., RSA encryption/decryption). It can be achieved by manipulating the external crystal, by forcing the target device to compute many tasks in parallel, or by provoking interruptions calls.

C. Simple Thermal Attack

Similarly, as for SPA, STA aims to distinguish the operations by visually inspecting the thermal traces. In order to investigate how difficult it is to make visual inspections on

Algorithm 3 Simple Thermal Side-Channel Analysis Test

```
for i := 0 ⇒ N do
   i++
end for
pause
for i := 0 ⇒ 2N do
   i++
end for
pause
```

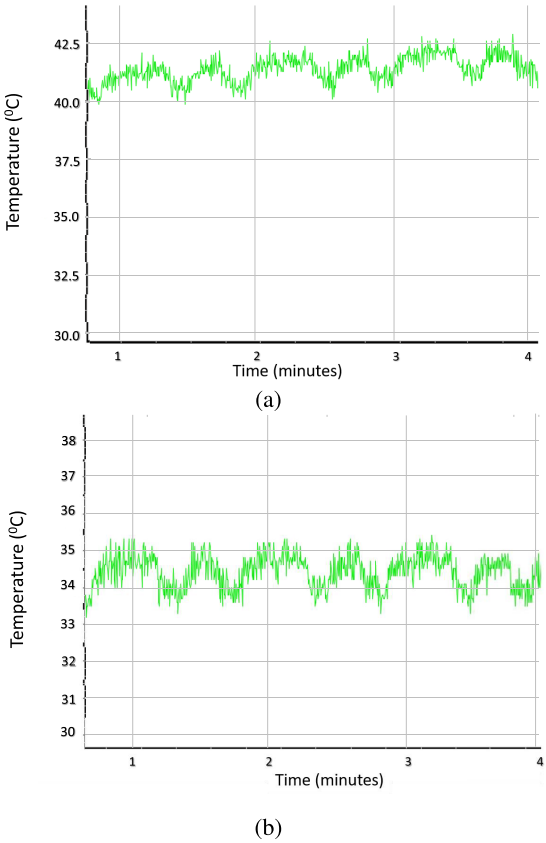

Fig. 1. Simple thermal trace analysis. (a) Without cooling. (b) With cooling.

thermal traces, we performed a small experiment. Under an ARM-based system-on-chip inside the PYNQ-Z1 board [29], we run a test application and record its thermal traces. The temperature was measured by the internal analog-to-digital converter (ADC) connected to an embedded temperature sensor which is integrated into the target board [29]. The application consisted of two loops without any code inside, as shown in Algorithm 3. After each loop, a pause was inserted. Such a pause makes the visual analysis better as both operations will show isolated behavior in the traces.

The differences in temperature from executing different operations can clearly be identified (as seen in Fig. 1). Fig. 1(a) shows the thermal trace when no cooling is applied. As can be seen from Fig. 1, after some time, the temperature starts to increase. This impacts the quality of the traces negatively. In order to minimize this, a cooling fan was installed on top of the processor chip. Consequently, the overall temperature

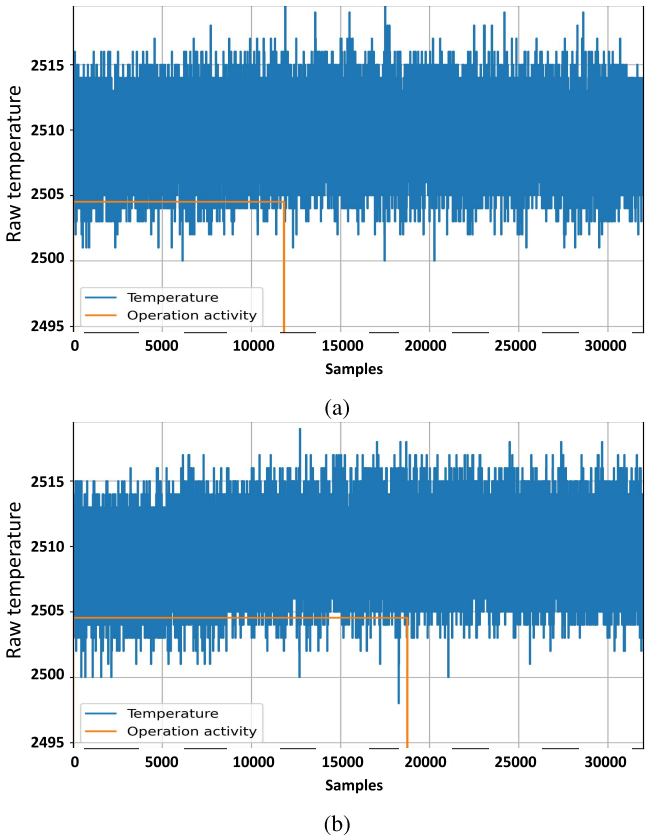

Fig. 2. Comparison of square and multiply operations of RSA. (a) Multiply operation. (b) Square operation.

reduced by almost 10 °C with no temperature drifts, as shown in Fig. 1(b); this clearly improves the quality of the traces. In Fig. 1(a) and (b), there are two distinguishable shapes visible, a shorter and a longer one corresponding to the different loop sizes. Another interesting observation is that the peak temperature differs slightly for both loops; the system reaches a temperature that is approximately 0.5° higher when the longer loop is executed.

1) Methodology: STA can be described by the following sequence of steps:

1) Setup target device with a cooling system;
2) Collect thermal traces when running the target operation;
3) Find points-of-interest (POIs) (optional);
4) Use visual inspection to retrieve the key.

STA requires the same setup used in the test experiment, which uses a cooling system in the target device. STA focus on an unprotected RSA decryption, where periodic interruptions are placed between square and multiply operations when collecting the traces. Hence, the attacker can evaluate each operation independently using visual inspection. For example, Fig. 2 shows the thermal trace of both operations. Note that the raw temperature on the y-axis can be transformed to degree Celsius using (2). However, for simplicity and since the relation between them is linear, we used the raw data generated by the Xilinx ADC (XADC) [30] for our security analysis

$$T = \frac{\text{Raw Temperature} \cdot 503.975}{4096} - 273.15. \tag{2}$$

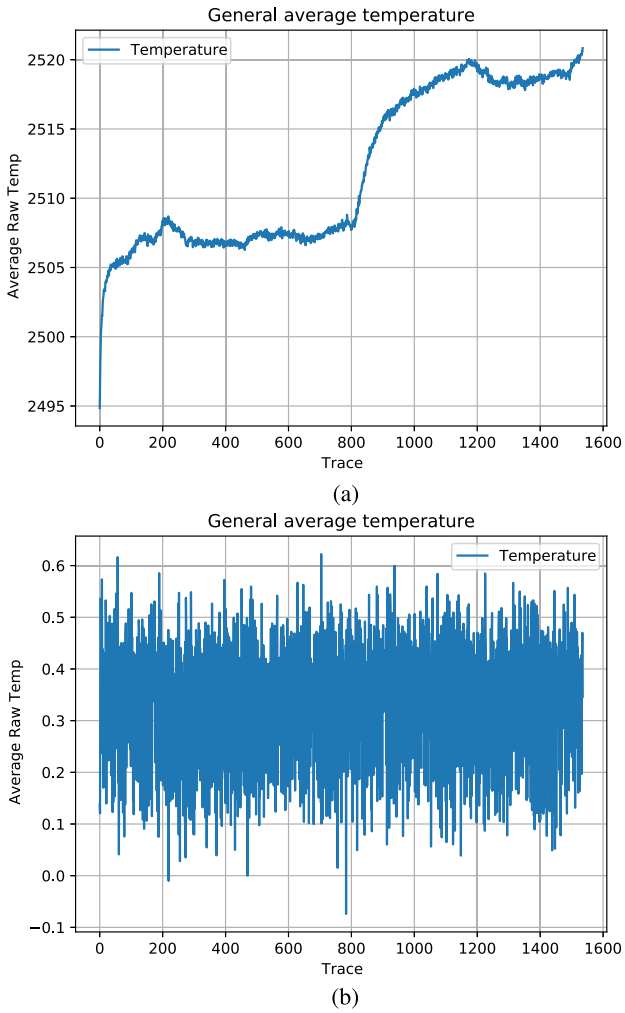

Fig. 3. Thermal traces for multiple operations. (a) Without temperature drift compensation. (b) With temperature drift compensation.

As observed in Fig. 2, each operation has different behavior. However, visual inspection is very challenging since the differences can only be identified by some specific points instead of by the entire shape of the traces. In addition, note that the temperature offset varies over time. This is known as the temperature drift effect, which is caused by the system regulating the voltage (and indirectly temperature) over time. For STA, temperature drift does not affect the analysis much. Therefore, STA is practical. In addition, STA can be improved if some extra processing steps are used to identify the most important points that can be used to distinguish between the operations. This step is defined in our methodology as *find the POI*, and several techniques can be applied such as the sum of pairwise differences or the sum of squared pairwise T-differences (SOSTs) [31].

D. Correlation Thermal Attack

As shown in STA, the thermal traces contain a temperature drift. As a result, the thermal traces have to be preprocessed before CTA can be applied. Fig. 3 shows the temperature drifting behavior again but for multiple operations. Each point represents the average temperature of 1600 thermal traces.

One way of solving this temperature drift is by subtracting the average of each operation of each trace. This is a method used to remove noise [32]. However, this is not suitable here as the average of each trace is relevant for the correlation process with the HWs. Another way of removing the drift offset is by subtracting the first value of each operation in their partial traces. However, this can be tricky due to the presence of noise. To make this more robust, it is better to take the average of the first m-samples and subtract that value from the trace. This reduces the effect of noise in calculations. This technique is based on the autozero amplifier [33] and can be observed in (3). Instead of charging a capacitor in the sampling phase, the first values are accumulated and normalized

$$T_{\text{filtered offset}} = \frac{1}{m}\sum_{i=0}^{m} t_i - T. \tag{3}$$

Fig. 3(b) shows the same information as Fig. 3(a) but compensated for the temperature drift. As can be seen, the drift is almost completely removed. To understand how this approach works, let us revisit Fig. 2(b). In Fig. 2(b), the orange line indicates when the processor is active (i.e., approximately the first 18 000 samples). The temperature offset only starts to rise somewhere between 10 000 and 15 000 samples (due to the integrator effect of the thermal behavior). This means that the first 10 000 samples can be used to reduce the drift effect. In Fig. 3(b), the drifting effect was removed by subtracting the average of the first 12 000 samples (i.e., m equals 12 000 in this case). Note that the average of each trace is unequal to zero; therefore, they can be used for correlation.

1) Methodology: The correlation thermal analysis can be described by the following sequence of steps:

1) Setup target device with a cooling system;
2) Collect thermal traces when running the target operation;
3) Remove temperature drift;
4) Estimate HW of the target operations (i.e., square and multiply);
5) Correlate thermal trace with hypothetical guesses;
6) Classify key bits as "1" or "0" according to correlation results.

After solving the temperature drift issue, the HW values have to be computed. Since there are only two possible operations (i.e., square or multiply), it makes sense to calculate the HW of the result of the square and multiply operations to use for the correlation process. Since it is not possible to compute all possible HW due to RSA key sizes, an estimation is derived from the average of a random simulation process. To make a proper estimation of these values, a simulator was created. After selecting ten different key pairs, which were generated randomly with the OpenSSL python library [34] (with key length 2048), the average HW of the results of the square operations was 922, while 461 for the multiply operations. Depending on the implementation, the actual HW might differ. However, as long as the ratio between square and multiply HW is around 2:1, the attack will work. After the HWs are created for both scenarios (i.e., square and multiply) and the thermal traces are collected and processed, it is possible to calculate the correlation matrix r. Like CPA,

Algorithm 4 Correlation Thermal Analysis for a Naive Implementation of RSA

for i := 0 ⇒ length(trace) **do**
  $r_{\text{multiply}} \leftarrow$ correlation(h_{multiply}, trace[i])
  $r_{\text{square}} \leftarrow$ correlation(h_{square}, trace[i])
  if $r_{\text{square}} \geq r_{\text{multiply}}$ **then**
    key_guess[i] ← 1
  else
    key_guess[i] ← 0
  end if
end for

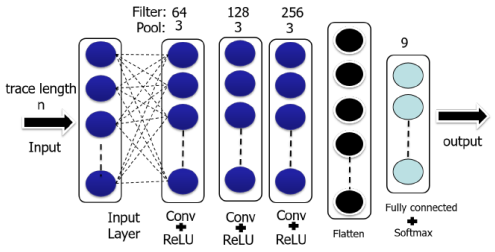

Fig. 4. CNN.

we use (1) to compute r. For each trace, there are two possible values defined as r_{square} and r_{multiply}. If $r_{\text{square}} \geq r_{\text{multiply}}$ the key-bit guess is 1, else the key-bit guess is 0. The pseudocode is provided in Algorithm 4.

E. DL-Based Thermal Attack

The goal of a SCA is to classify parts of a trace in such a way that they lead to the key. ML and DL are very suitable methodologies to realize this. The most effective solutions use supervised learning (i.e., when the attacker has a similar device to train the network) and CNNs [26]. CNN is a popular and effective way of (image) classification and recognition [35]. Although CNNs can be very complex, they have one advantage over the techniques, such as CPA/CTA, as they can learn the leakage behavior.

A representative example of our target CNN is presented in Fig. 4. The input is a 1-D thermal trace followed by convolutional layers interleaved with pooling layers. Each convolutional layer contains a rectified linear unit (ReLU) activation function [36], batch normalization, and Gaussian noise insertion. Batch normalization and Gaussian noise are used to avoid overfitting. Our first attempts used dropout; however, after many trials, the combination of batch normalization and Gaussian noise provided the best results (about 90% accuracy during the training phase). The last part of the CNN consists of the classification, which contains flattened, fully connected, and SoftMax layers. The SoftMax layer maps the output values between 0 and 1.

1) Methodology: The DL-based thermal attack can be described by the following sequence of steps:

1) Setup template device with a cooling system.
2) Collect thermal traces when running the target operation with a known key.
3) Remove temperature drift.
4) Slice and label the traces.
5) Separate part of the traces into a training and validation set.
6) Train the CNN using the training set.
7) Validate the CNN using the validation set.
8) Setup target device with a cooling system.
9) Collect thermal traces when running target operation.
10) Remove temperature drift (create evaluation set).
11) Apply the evaluation set to the trained CNN.
12) Collect more traces to perform a vertical attack.

The attack starts by recording traces from the template device, where the input and key applied are known. Subsequently, preprocessing is applied to remove the temperature drift. For this purpose, the same technique as described in Section IV-D [see (3)] is used. Thereafter, the traces are divided into two sets, i.e., a training and validation set.

The next step in the process is to determine the labels to perform the training. In the unprotected RSA implementation, it makes sense to look at two possible label structures.

1) Labels method A: *square* and *multiply*.
2) Labels method B: *square square*, *square multiply*, and *multiply square*.

Method A has as advantage that it is the simplest approach. It just requires having traces with a single-square or single-multiply operation. In case a certain operation is wrongly predicted, it might be that two multiply operations follow each other up; this is, however, not possible, and hence, should be corrected. This is not possible in method B, which automatically eliminates/ignores incoherent results. However, method B requires a larger and more complex CNN. In this work, we used method A for labeling.

The next step is to train the neural network. The training parameters are the following.

1) Initialization: Glorot [37] is used to initialize the weights and biases. Glorot is an advanced technique (as compared with, e.g., random initialization), where the initialization values are computed based on the width of its preceding and successive layers.
2) Loss Function: The loss function is defined by the categorical entropy technique to compute the error function.
3) Optimization: For optimization, Adam [38] is used; it is a special technique that uses adaptive learning rates for each parameter which typically gives good results.
4) Regularization: Both batch normalization and Gaussian noise are applied. Batch normalization originally was introduced to reduce the random effects of initialization parameters and input data [39]. However, it has been successfully applied to improve generalization [40]. The Gaussian layer adds noise to the data before it enters a neuron. As a consequence, instead of only being able to classify the used image, the neural network is also able to classify small changes on that image [41].

After the training achieves sufficient accuracy results (i.e., around 90%), we validate the trained CNN using the validation traces. Thereafter, the attacker can collect the thermal traces from the target device. The attacker has to remove the temperature drift from the collected traces. This trace set

is defined as an evaluation set. Subsequently, the attacker applies the evaluation set to the CNN. Finally, more traces of different executions are collected and fed into the CNN. As more traces are used to determine the key, the higher is the chance of a successful attack. This strategy is defined as a vertical attack [42].

V. Improving the Correlation Thermal Attack

When blinding countermeasures like the Montgomery ladder are used, the HW between the different operations will be very similar. As a result, the CTA will not succeed. To overcome this issue, we modified the CTA attack and introduced a novel PCTA. First, we present the concept behind this attack followed by the attack methodology.

A. Concept

The only reason why CTA works for the unprotected RSA implementation is that there is a simple relation between the HW and key. However, if this relationship does not exist as is the case for the Montgomery ladder implementation, it is hard to consider all potential HW values for key sizes larger than 32 bits (as this would require to compare at least 2^{32} HW values).

To limit the number of possible HW values, one could suggest splitting the key into smaller sections, such as ten bits for example. In this case, it would only require the evaluation of 2^{10} possible HW values. This is possible for the first ten bits, but the HW of the result of the 11th bit depends on the previous ten bits. In other words, to calculate the 11th bit, 2^{11} HW values would need to be evaluated. One way to overcome this issue is to apply a correlation analysis on these first ten bits. Once the first ten key bits are predicted correctly, it is possible to calculate the possible HW values of the next ten bits. This process can be repeated progressively until all key bits are retrieved. This is the strategy that is applied in the progressive correlation attack.

B. Progressive Correlation Thermal Attack

In the PCTA attack, a fixed amount of key bits are processed sequentially. As a consequence, when a subkey of ten bits is guessed incorrect, the following subkeys will also be incorrect since they depend on each other. Therefore, it is essential to come up with a proper estimation. To do this, we increase the differences in HW values by using a specific message (i.e., ciphertext since we attack RSA decryption). This specific messages is crafted in the form of $m = N-1$ [43]. An example of this can be seen in Algorithm 5.

Analyzing the values of R0 and R1 in Algorithm 5, it looks that each round computes exactly the same values. The answers are always $N-1 = 76$ and 1. However, the multiplications to compute such results are slightly different, which can be

1) M1: $1 \cdot 1 \bmod N \equiv 1 \bmod N$
2) M2: $1 \cdot (N-1) \bmod N \equiv (N-1) \cdot 1 \bmod N \equiv N-1 \bmod N$
3) M3: $(N-1) \cdot (N-1) \bmod N \equiv 1 \bmod N$.

Algorithm 5 Example of Calculating RSA With a Specific Message

Variables Declaration:
$d = 43; (b101011)$ {Private Key d}
$N = 77$; {Public Key N}
$c = 76$; {Ciphertext c}
$R_0 = 1$;
$R_1 = c = 76$;

Decryption:

$$
\begin{array}{lll}
e[0]=1 & \begin{cases} R_0 = (R_0 * R_1) & mod \quad N \Rightarrow (1*76) \quad mod \quad 77 = 76 \quad \{M2\} \\ R_1 = (R_1 * R_1) & mod \quad N \Rightarrow (76*76) \quad mod \quad 77 = 1 \quad \{M3\} \end{cases} \\
e[1]=1 & \begin{cases} R_0 = (R_0 * R_1) & mod \quad N \Rightarrow (76*1) \quad mod \quad 77 = 76 \quad \{M2\} \\ R_1 = (R_1 * R_1) & mod \quad N \Rightarrow (1*1) \quad mod \quad 77 = 1 \quad \{M1\} \end{cases} \\
e[2]=0 & \begin{cases} R_1 = (R_0 * R_1) & mod \quad N \Rightarrow (76*1) \quad mod \quad 77 = 76 \quad \{M2\} \\ R_0 = (R_0 * R_0) & mod \quad N \Rightarrow (76*76) \quad mod \quad 77 = 1 \quad \{M3\} \end{cases} \\
e[3]=1 & \begin{cases} R_0 = (R_0 * R_1) & mod \quad N \Rightarrow (1*76) \quad mod \quad 77 = 76 \quad \{M2\} \\ R_1 = (R_1 * R_1) & mod \quad N \Rightarrow (76*76) \quad mod \quad 77 = 1 \quad \{M3\} \end{cases} \\
e[4]=0 & \begin{cases} R_1 = (R_0 * R_1) & mod \quad N \Rightarrow (76*1) \quad mod \quad 77 = 76 \quad \{M2\} \\ R_0 = (R_0 * R_0) & mod \quad N \Rightarrow (76*76) \quad mod \quad 77 = 1 \quad \{M3\} \end{cases} \\
e[5]=1 & \begin{cases} R_0 = (R_0 * R_1) & mod \quad N \Rightarrow (1*76) \quad mod \quad 77 = 76 \quad \{M2\} \\ R_1 = (R_1 * R_1) & mod \quad N \Rightarrow (76*76) \quad mod \quad 77 = 1 \quad \{M3\} \end{cases}
\end{array}
$$

TABLE I
Operations According to Current Key-Bit e[i] and Previous Key-Bit e[i − 1]

		e[i-1] = 0	e[i-1] = 1
e[i]	0	M2 M1	M2 M3
	1	M2 M3	M2 M1

This means that there are only three different HW values possible when we look at the output of the multiplication (without the modulo operation). Note that depending on the previous key bit, there are only two possible combinations of operations, either M2 followed by M1 or M2 followed by M3. This behavior is shown in Table I. Consequently, when multiplications M1 and M3 are identified the current key bit can be retrieved. Luckily, M1 and M3 have very different HW values, which improve the chance of a successful key correlation.

Note that this approach can be extended by considering ten bits of the key simultaneously instead of one bit. However, this requires that 2^{10} different HW values have to be computed.

1) Methodology: The progressive correlation thermal analysis can be described by the following sequence of steps.

1) Setup template device with a cooling system.
2) Prepare the input message of the target operation.
3) Collect thermal traces when running the target operation.
4) Remove temperature drift.
5) Define subkey size.
6) Perform CTA on the subkey.
7) Use the resulting guessed subkey to perform CTA in the next subkey.
8) Repeat Step 7 until the end of the trace.
9) Collect more traces to perform a vertical attack.

The PCTA attack is very similar to the CTA attack but attacks only part of the key at a time using a specific message.

TABLE II
RESULTS OF CTA ON THE UNPROTECTED IMPLEMENTATION OF RSA

1	2	3	4	5	6	7	8	9	Vertical
93.1%	92.8%	92.8%	93.0%	94.4%	92.8%	94.3	93.6%	92.9%	100%

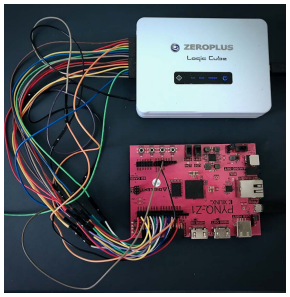

Fig. 5. Measurement setup with logic analyzer.

To successfully retrieve the key, a vertical attack is preferred, where the subkey is evaluated using multiple traces simultaneously. The subkey that is the most common among the different traces is most likely the correct subkey. Note that it is theoretically possible to compare the subkeys of multiple traces on a bit level instead of a subkey level. However, this will not improve the results as the derived subkey could end up not being a result of any of the traces. Therefore, comparing each subkey of all the gathered traces is the preferred method. Finally, the strategy that is used to craft the input message adds a new requirement to the threat model. It does not necessarily make the attack unpractical but might limit its application in the field.

VI. EXPERIMENTAL RESULTS

This section first explains the measurement setup of the performed experiments. Thereafter, it provides the results of the CTA, DL-based thermal attack, and PCTA. Finally, it compares the achieved results.

A. Measurement Setup and Performed Experiments

All experiments were performed on the PYNQ-Z1 development board [29]. The PYNQ board runs bare-metal C(++) code on one of the two available ARM-A9 cores. We used the board to run both the unprotected RSA (i.e., square and multiply) and the protected RSA (i.e., Montgomery ladder) implementations using a 1024-bit key. PYNQ-Z1 has an embedded ADC XADC [30], which is connected to power and temperature sensors. It has a resolution of 12 bits and a sampling rate of up to 1 mega sample per second (MSPS). In order to read out the data from XADC with minimal noise effects, the Zeroplus Logic Cube (LAP-C 16032) digital logic analyzer is used [44]. The collection of traces, the steps to process them, and the attacks have been coded in Python scripts. The measurement setup can be seen in Fig. 5. Note that the experiment was conducted at room temperature using a clock frequency of the system equals to 650 MHz and a supply voltage of 1 V.

Using this setup, the following experiments were conducted.

1) CTA: In this experiment, CTA is used to attack the unprotected RSA implementation. Nine traces were used.
2) DL-based thermal attack: In this experiment, the unprotected RSA implementation is attacked using DL. Two different sets of traces were used, one with a known key and one with an unknown key. The former set was used for training and validation, while the latter set was used for evaluation. This experiment uses five traces for training and validation and nine traces for evaluation.
3) Progressive correlation thermal attack: This experiment use PCTA to attack the Montgomery ladder RSA implementation. Five traces were used.
4) Comparison between power and thermal attacks: In this experiment, the results of all evaluated thermal attacks (i.e., the experiments earlier) are compared with their equivalent power attacks, and hence, we also collected power traces.

To evaluate these experiments, the following two metrics are used.

1) Performance: The performance is defined as the attack accuracy, i.e., the percentage of key bits that are guessed correctly. A successful attack requires 100% accuracy for the RSA cryptosystem. Due to the usage of very large key sizes in RSA, even when 95%–98% of the key bits have been guessed correctly, it is still hard to brute-force the wrongly predicted key bits when their locations are not known.
2) Error histogram: This histogram shows the occurrences of wrong guesses for the trace samples. It provides evidence that vertical attacks are practical, as long as the errors are more or less random (i.e., have a uniform distribution).

B. Correlation Thermal Attack Results

The results of the CTA experiment are presented in Table II. Among the nine attacked traces, the performance varies between 92% and 95%. In case the traces are attacked isolated, such results are insufficient for a successful attack. However, when analyzing the error histogram shown in Fig. 6, we observe that the errors of these traces are distributed uniformly along the key-bits. Consequently, this means that the key can be reconstructed using a vertical attack with majority voting. Note that the histogram also shows that the first two bits are wrongly guessed in most traces. This makes sense as the input of the square and multiply is not limited by the modulo yet. However, this does not prohibit the reconstruction of the key, as the location of such faulty bits is fixed; hence, they can be easily brute-forced. Therefore, our CTA attack

TABLE III
RESULTS OF CNN ON THE UNPROTECTED IMPLEMENTATION OF RSA

1	2	3	4	5	6	7	8	9	Vertical
93.7%	93.9%	93%	93.6%	95.1%	91%	94.1%	94.2%	92.7%	100%

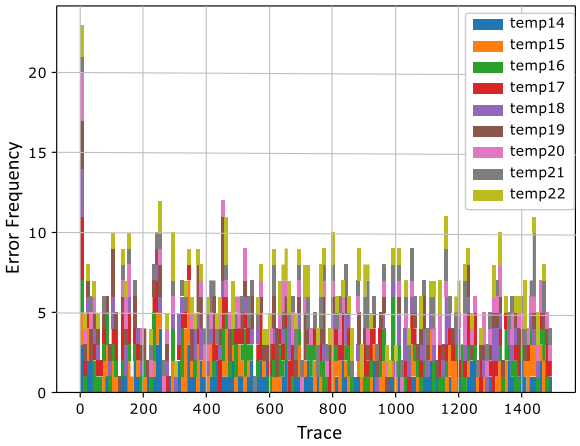

Fig. 6. Error histogram of CTA.

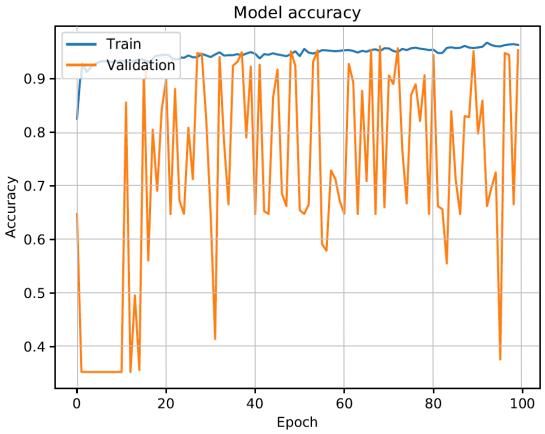

Fig. 7. Training and validation curves of the CNN.

achieves 100% accuracy with only nine traces when a vertical attack is used, as shown in the last column of Table II.

C. DL-Based Thermal Attack Results

The DL-based thermal attack has two phases. The training and attack phases.

1) Training Phase: In order to train the network, these traces had to be separated into two groups. The first group consists of 90% of the total traces for training and 10% for validation. To make sure that the validation group was uniform, all the traces were first randomly shuffled and afterward split. This makes sure that CNN is able to handle all kinds of offset. After shuffling, we calculated the percentage of ones and zeros at each dataset. The training of the CNN is considered successful when at least 90% accuracy is achieved. Fig. 7 shows the accuracy results of the training phase; it shows the results for both the training and validation sets. Both cases

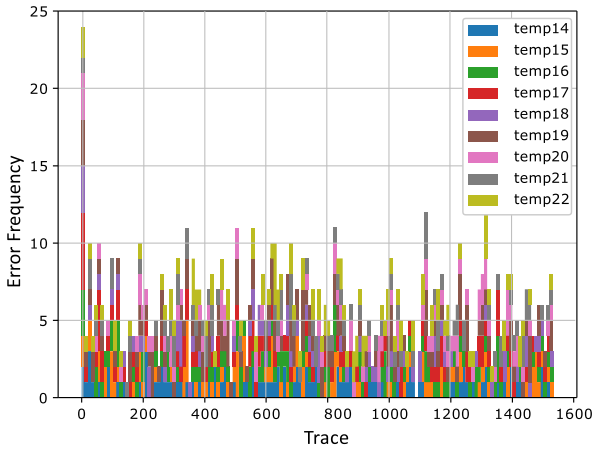

Fig. 8. Error histogram of DL-based thermal attack.

TABLE IV
RESULTS OF PCTA ON THE (MONTGOMERY IMPLEMENTATION OF RSA WITH $c = N - 1$)

1	2	3	4	5	Vertical
257	97	96	47	108	1024
25%	9.4%	9.3%	4.5%	10.5%	100%

achieved a maximum accuracy above 90%, which indicates that the attack can be successful.

2) Attack Phase: In this phase, the evaluation set consisting of nine traces is used to retrieve the key. Table III shows that the results of the individual traces have a 91%–95% accuracy. We also applied error analysis to understand the error distribution behavior. The combined error histogram of all the traces can be seen in Fig. 8. It shows again a uniform distribution of the errors except for the first bits. Hence, a vertical attack is possible. Since the key in all executions is fixed, the CNN was able to completely retrieve the key when a vertical attack was applied on these nine traces.

D. Progressive Correlation Thermal Attack Results

PCTA is highly dependent on the correct key guesses during the iterations over the subkeys. This means that one incorrect guess makes the attack unsuccessful. Table IV shows for five different traces the number of key-bits that have been predicted correctly until the first faulty prediction occurred. Note that in the best case, only 25% of the key-bits have been predicted correctly.

To increase the accuracy, also here, information from multiple traces could be combined. The first method would consist of simply adding traces together [45]. This increases the differences between low-intensity and high-intensity operations. However, it requires all traces to be perfectly aligned in time, and more importantly, they must have the same offset. The last

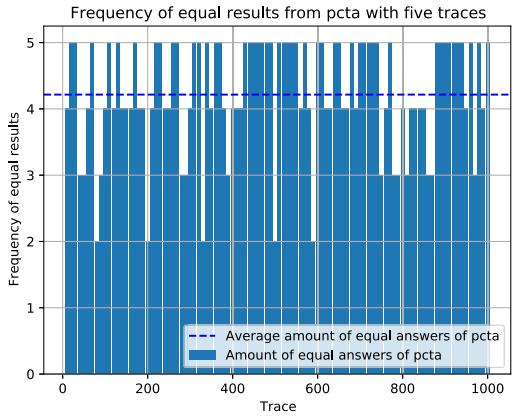

Fig. 9. Frequency of intermediate results with five traces.

TABLE V
RESULTS OF CPA AND DL-BASED POWER ATTACK ON UNPROTECTED RSA AND PCPA ON THE MONTGOMERY RSA

Attack	1	2	3	Vertical
CPA	100%	100%	100%	N/A
DL-based PA	100%	100%	100%	N/A
PCPA	100%	100%	100%	N/A

requirement is a problem in the case of temperature because our preprocessing does not completely remove the temperature drift effect. Therefore, instead of adding all the traces together, we applied a vertical attack with majority voting during every intermediate step (i.e., at each subkey correlation).

In this setting, it was possible to retrieve the full 1024-bit key with five traces only. The frequency of correctly predicted intermediate subkey values can be seen in Fig. 9. Due to the usage of five traces, the maximum frequency is five. Fig. 9 shows that the confidence of the attack was very high in most cases and that with only five traces the whole key could be recovered. We recommend using PCTA with more traces to increase the predicted key's confidence level.

E. Power Versus Thermal SCA

In this experiment, we collected power traces from our target PYNQ-Z1 board and applied a CPA, DL-based power attack, and PCPA (the variant of our proposed attack for power). The first two targeted the unprotected RSA, while the last one the Montgomery ladder implementation. All attacks used the same methodology as the thermal-based attacks. Table V presents the results for three different traces. For all power attacks, no majority voting was needed. All power attacks were able to retrieve all 1024-bits of the key within a single trace. Three different traces were used in each attack to make sure that a horizontal attack is sufficient.

A direct comparison between the results shows that power-based SCAs are more efficient in breaking RSA. Although such behavior was expected, the thermal-based attacks nevertheless also had a good efficiency. Therefore, although less dangerous than power, thermal SCAs should not be neglected when designing a secure system. Some very secure systems use several countermeasures against power leakage but ignore thermal leakage. Our results show that academia and industry should also start focusing on countermeasures for thermal leakage.

VII. DISCUSSION

This work explored thermal side-channel analysis in-depth and proposed a novel thermal attack based on progressive correlation. Next, we discuss the limitations and future work based on the experimental results.

A. Practicality

We assumed a threat model in our attack where the attacker has access to the internal ADC and can lower the clock frequency. These assumptions are reasonable. An attacker can slow down the system or get access to its sensors in a variety of methods. For example, many microcontrollers use a clock that is generated outside the chip; examples are ceramic resonators, resistor–capacitors, and external crystals. The attacker has the potential to manipulate these. Second, many CPUs allow users to overclock or underclock the machine [46]. Today's chip also typically contains build-in sensors to monitor the processor's temperature. Although these features were introduced as a safety measure to keep the CPU from overheating, malicious programs can utilize them to monitor the temperature during an attack [47]. Currently, the temperature is not seen as a security sensor that can only be accessed with special rights.

B. Comparing Thermal and Power

Experiments have shown that both thermal and power attacks perform well in an unprotected scenario, requiring only a few traces to succeed. However, when countermeasures exist, thermal attacks face more challenges. On the other hand, power traces seem to have more representative points within the trace making them more efficient for applying an attack. However, profiled-based thermal attacks are promising and more research is needed to create suitable mathematical models that are able to describe the thermal leakage better.

C. Other Algorithms

In our experiment, we focused on the study of the thermal analysis of RSA cryptography. The results showed successful attempts in extracting the secret key using both profiled and nonprofiled attack techniques. In the same manner, thermal SCAs can be carried on other asymmetric algorithms implementations. One of the most recently widely adopted algorithms is ECC. In addition, the attacks can be explored on symmetric algorithms as well such as AES.

D. Advancing the Attack

Although progressive correlation thermal showed very promising results on a protected RSA software implementation using the Montgomery ladder, the attack depends heavily on the intermediate subkey values. In other words, when a single guess of the key is wrong it will affect the guesses of all the subkeys afterward. To overcome this issue, we applied

a vertical attack where multiple executions with the same fixed key are used. This scenario is not applicable to many current RSA implementations as they use a randomization technique to randomize the key in every execution. One way to improve the progressive correlation attack is to combine it with other attacks, such as correlation-based or clustering, where every bit is treated individually.

VIII. Conclusion

This article presented a study on thermal side-channel analysis. Three different power-based SCAs were modified for thermal attacks and applied to a naive RSA implementation. These attacks are simple thermal attacks, CTA, and DL-based thermal attacks. All experimented attacks were able to retrieve the secret key with 100% accuracy. In addition, a novel progressive CTA was proposed to attack a protected version of an RSA implementation and showed very interesting results. The attack was able to extract the whole key with only five traces. This study shows that thermal attacks are feasible, and hence, appropriate countermeasures should be designed against them.

References

[1] J. Davis. (2020). *CUPDATE: The 10 Biggest Healthcare Data Breaches of 2020, So Far*. Accessed: Nov. 13, 2020. [Online]. Available: https://healthitsecurity.com/news/the-10-biggest-healthcare-data-breaches-of-2020-so-far

[2] P. Muncaster. (2020). *IoT Attacks Cost UK Firms Over $1bn*. Accessed: Oct. 21, 2020. [Online]. Available: https://www.infosecurity-magazine.com/news/iot-attacks-cost-uk-firms-over-1bn-1/

[3] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proc. IEEE*, vol. 100, no. 11, pp. 3056–3076, Nov. 2012.

[4] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," in *Proc. Phys. Secur. Test. Workshop*, Honolulu, HI, USA, 2005.

[5] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: Threat analysis and countermeasures," *Proc. IEEE*, vol. 102, no. 8, pp. 1229–1247, Aug. 2014.

[6] S. Dara and P. Tumma, "Feature extraction by using deep learning: A survey," in *Proc. 2nd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Mar. 2018, pp. 1795–1801.

[7] J. Danial, D. Das, S. Ghosh, A. Raychowdhury, and S. Sen, "SCNIFFER: Low-cost, automated, efficient electromagnetic side-channel sniffing," *IEEE Access*, vol. 8, pp. 173414–173427, 2020.

[8] J. Chen, M. Sun, and K. Zhang, "Security analysis of device binding for IP-based IoT devices," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2019, pp. 900–905.

[9] R. Krishnamoorthy, S. Priya L., S. Aswini, and C. Guna, "Design and implementation of IoT based energy management system with data acquisition," in *Proc. 7th Int. Conf. Smart Struct. Syst. (ICSSS)*, Jul. 2020, pp. 1–5.

[10] W. Yu, "Exploiting on-chip voltage regulators as a countermeasure against power analysis attacks," Ph.D. dissertation, 2017. [Online]. Available: https://digitalcommons.usf.edu/etd/6986

[11] J. Brouchier, T. Kean, C. Marsh, and D. Naccache, "Temperature attacks," *IEEE Secur. Privacy Mag.*, vol. 7, no. 2, pp. 79–82, Mar. 2009.

[12] R. J. Masti, D. Rai, A. Ranganathan, C. Müsller, L. Thiele, and S. Capkun, "Thermal covert channels on multi-core platforms," in *Proc. USENIX Secur.*, 2015, pp. 865–880.

[13] D. B. Bartolini, P. Miedl, and L. Thiele, "On the capacity of thermal covert channels in multicores," in *Proc. 11th Eur. Conf. Comput. Syst.*, Apr. 2016, pp. 1–16.

[14] M. Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.*, vol. 8419, 2013, pp. 219–235.

[15] S. Dey, A. Kumar, and K. D. Mcdonald-Maier, "ThermalAttackNet: Are CNNs making it easy to perform temperature side-channel attack in mobile edge devices?" *Future Internet*, vol. 13, no. 6, p. 146, 2021.

[16] P. Kocher *et al.*, "Spectre attacks: Exploiting speculative execution," 2018, *arXiv:1801.01203*. [Online]. Available: http://arxiv.org/abs/1801.01203

[17] M. Lipp *et al.*, "Meltdown," 2018, *arXiv:1801.01207*. [Online]. Available: http://arxiv.org/abs/1801.01207

[18] S. Burnett and S. Paine, *The RSA Security's Official Guide to Cryptography*. New York, NY, USA: McGraw-Hill, 2001.

[19] P. Thorsteinson and A. Ganesh, *NET Security and Cryptography*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.

[20] A. Said Alkalbani, T. Mantoro, and A. O. M. Tap, "Comparison between RSA hardware and software implementation for WSNs security schemes," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. Moslem World (ICT4M)*, Dec. 2010, pp. E84–E89.

[21] P. Gaudry, "Integer factorization and discrete logarithm problems," *Les Cours du CIRM*, vol. 4, no. 1, pp. 1–20, 2014.

[22] M. Joye and S.-M. Yen, "The Montgomery powering ladder," in *Cryptographic Hardware and Embedded Systems–(CHES)*. Berlin, Germany: Springer, 2002, pp. 291–302.

[23] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology*. Berlin, Germany: Springer, 1999, pp. 388–397.

[24] E. Brier *et al.*, "Correlation power analysis with a leakage model," in *Proc. CHES*, 2004, pp. 16–29.

[25] G. Perin and Ł. Chmielewski, "A semi-parametric approach for side-channel attacks on protected RSA implementations," in *Proc. CARDIS*, 2015, pp. 34–53.

[26] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas. (2018). *Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to Ascad Database*. ANSSI, France & CEA, LETI, MINATEC Campus, France. Verfügbar Unter. Zuletzt Geprüftam. [Online]. Available: https://eprint.iacr.org/2018/053.pdf

[27] M. Happe, A. Agne, and C. Plessl, "Measuring and predicting temperature distributions on FPGAs at run-time," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs*, Nov. 2011, pp. 55–60.

[28] J. S. Lee, K. Skadron, and S. W. Chung, "Predictive temperature-aware DVFS," *IEEE Trans. Comput.*, vol. 59, no. 1, pp. 127–133, Jan. 2010.

[29] Digilent. (2020). *PYNQ-Z1: Python Productivity for Zynq-7000 ARM/FPGA SoC*. Accessed: Nov. 13, 2020. [Online]. Available: https://store.digilentinc.com/pynq-z1-python-productivity-for-zynq-7000-arm-fpga-soc/

[30] *7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter*, Xilinx, San Jose, CA, USA, 2018.

[31] B. Gierlichs, K. Lemke-Rust, and C. Paar, "Templates vs. stochastic methods," in *Cryptographic Hardware and Embedded Systems–(CHES)*, L. Goubin and M. Matsui, Eds. Berlin, Germany: Springer, 2006, pp. 15–29.

[32] C. R. Trimble, "What is signal averaging," *Hewlett-Packard J.*, vol. 19, no. 8, pp. 2–7, 1968.

[33] C. C. Enz and G. C. Temes, "Circuit techniques for reducing the effects of op-amp imperfections: Autozeroing, correlated double sampling, and chopper stabilization," *Proc. IEEE*, vol. 84, no. 11, pp. 1584–1614, Nov. 1996.

[34] pyca. (2020). *Cryptography package for Python Developer*. Accessed: Sep. 21, 2020. [Online]. Available: https://cryptography.io/en/latest/hazmat/primitives/asymmetric/rsa/

[35] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016

[36] A. Fred Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*. [Online]. Available: http://arxiv.org/abs/1803.08375

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[38] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2014.

[39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[40] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 730–734.

[41] A. Camuto, M. Willetts, U. Şimşekli, S. Roberts, and C. Holmes, "Explicit regularisation in Gaussian noise injections," in *Proc. 34th Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2021.

[42] A. Bauer, É. Jaulmes, E. Prouff, and J. Wild, "Horizontal and vertical side-channel attacks against secure RSA implementations," in *Proc. Cryptographers' Track RSA Conf.* Berlin, Germany: Springer, 2013, pp. 1–17.

[43] Z. Ding, W. Guo, L. Su, J. Wei, and H. Gu, "Further research on N-1 attack against exponentiation algorithms," in *Information Security and Privacy*, W. Susilo and Y. Mu, Eds. Berlin, Germany: Springer, 2014, pp. 162–175.
[44] Zeroplus. (2021). *Zeroplus–Logic Analyzer LAP-C 16032*. [Online]. Available: http://www.zeroplus.com.tw/logic-analyzer_en/products.php?pdn=1&product_id=253
[45] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, and V. Verneuil, "Improved collision-correlation power analysis on first order protected AES," in *Proc. CHES*, B. Preneel and T. Takagi, Eds. Berlin, Germany: Springer, 2011, pp. 49–62.
[46] A. K. Uht and R. J. Vaccaro, "TEAPC: Adaptive computing and underclocking in a real PC," in *Proc. 1st IBM P= ac2 Conf.*, 2004, pp. 45–54.
[47] Smith, Matt. (1999). *Measuring Temperatures on Computer Chips with Speed and Accuracy-A New Approach Using Silicon Sensors and Off-Chip Processing*. Accessed: Jul. 1, 2021. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/measuring-computer-chip-temps-with-speed-and-accuracy.html

Abdullah Aljuffri (Member, IEEE) received the B.S. degree in electrical engineering from King Abdulaziz University, Jeddah, Saudi Arabia, in 2009, and the M.S. degree in computer engineering from Delft University of Technology, Delft, The Netherlands, in 2017, where he is currently working toward the Ph.D. degree with the Department of Electrical Engineering.

His research interests are in the areas of application specific integrated circuit (ASIC)/FPGA design, hardware security, and side-channel analysis, with emphasis on power attacks and countermeasures.

Marc Zwalua received the B.S. degree in electrical engineering and the M.S. degree in microelectronics from Delft University of Technology, Delft, The Netherlands, in 2017 and 2020, respectively. Part of his M.S. degree was performed at the Ecole polytechnique fédérale de Lausanne, Lausanne, Switzerland.

His research interests include design for tests, reliability and security, side-channel analysis, and battery management systems.

Cezar Rodolfo Wedig Reinbrecht received the B.Sc. degree in computer engineering and the M.Sc. in computer science (MPSoC Architectures) from the Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil, in 2009 and 2012, respectively, and the Ph.D. degree in computer science (hardware security) from the Federal University of Rio Grande do Sul, Porto Alegre, in 2017.

He worked at NSCAD Microelectronics, Porto Alegre, from 2010 to 2014 and from 2016 to 2018, where he was an Instructor/IC Designer from 2010 to 2012 and from 2017 to 2018 and a Project Manager from 2013 to 2014. In 2015, he worked with the Laboratoire Hubert Curien, Saint-Étienne, France, as a Researcher in the field of microarchitectural security. He has been a Postdoctoral Researcher with the Quantum and Computer Engineering Department, Delft University of Technology, Delft, The Netherlands, since 2018. His main integrated circuits projects are ARM-M3 SoC with support to IEEE 802.15.4, a transponder for the Brazilian Cubesat (satellite) targeted for Brazilian Environmental Data Collection System (SBCDA) [Instituto Nacional de Pesquisas Espaciais (INPE)], and a fault-tolerant SoC, containing 2 Million instructions/s with the partnership with Brazilian Space Agency, Brasilia, Brazil. His research interests include hardware security fields and targeting microarchitectural attacks and defenses.

Said Hamdioui (Senior Member, IEEE) received the M.S.E.E. and Ph.D. degrees (Hons.) from Delft University of Technology (TUDelft), Delft, The Netherlands, in 1997 and 2001, respectively.

He worked at Intel Corporation, Santa Clara, CA, USA, Philips Semiconductors Research and Development, Crolles, France, and Philips/NXP Semiconductors, Nijmegen, The Netherlands. He is currently the Chair Professor in dependable and emerging computer technologies and the Head of the Computer Engineering Laboratory, TU Delft, where he is also serving as the Head of the Quantum and Computer Engineering Department. He is also a Co-Founder and the CEO of Cognitive-IC, Delft, a startup focusing on hardware dependability solutions. His research interests include dependable CMOS nanocomputing, including testability, reliability, and hardware security, and emerging technologies and computing paradigms, including memristors for logic and storage, in-memory computing, and neuromorphic computing.

Mottaqiallah Taouil (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from the Delft University of Technology, Delft, The Netherlands, in 2009 and 2014, respectively.

He is currently an Assistant Professor with the Dependable Nano-Computing Group, Delft University of Technology. His current research interests include design for tests, reliability, and security. He has authored or coauthored more than 100 papers on these topics.