TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 03/15 pp244–257
DOI: 10.26599/TST.2021.9010094
Volume 28, Number 2, April 2023

Arm PSA-Certified IoT Chip Security: A Case Study

Fei Chen, Duming Luo, Jianqiang Li*, Victor C. M. Leung*, Shiqi Li, and Junfeng Fan

Abstract: With the large scale adoption of Internet of Things (IoT) applications in people's lives and industrial manufacturing processes, IoT security has become an important problem today. IoT security significantly relies on the security of the underlying hardware chip, which often contains critical information, such as encryption key. To understand existing IoT chip security, this study analyzes the security of an IoT security chip that has obtained an Arm Platform Security Architecture (PSA) Level 2 certification. Our analysis shows that the chip leaks part of the encryption key and presents a considerable security risk. Specifically, we use commodity equipment to collect electromagnetic traces of the chip. Using a statistical T-test, we find that the target chip has physical leakage during the AES encryption process. We further use correlation analysis to locate the detailed encryption interval in the collected electromagnetic trace for the Advanced Encryption Standard (AES) encryption operation. On the basis of the intermediate value correlation analysis, we recover half of the 16-byte AES encryption key. We repeat the process for three different tests; in all the tests, we obtain the same result, and we recover around 8 bytes of the 16-byte AES encryption key. Therefore, experimental results indicate that despite the Arm PSA Level 2 certification, the target security chip still suffers from physical leakage. Upper layer application developers should impose strong security mechanisms in addition to those of the chip itself to ensure IoT application security.

Key words: Internet of Things (IoT) security chip; Arm Platform Security Architecture (PSA) certification; electromagnetic side-channel attack; Advanced Encryption Standard (AES) encryption; key leakage

1 Introduction

At present, IoT devices are being widely adopted in the form of smart homes while playing an important role in industrial manufacturing and smart cities. Common IoT applications include cameras in smart homes, wearable devices for personal use, and industrial IoT devices that replace the labor force. Predictions indicate that the usage scenarios of IoT devices will become increasingly extensive with the development of 5G networks. However, IoT devices may capture sensitive personal data and disclose trade secrets. Once IoT devices are hacked, individuals and businesses may face huge losses. Therefore, IoT security has become an important focus in academia and the industry[1–5].

IoT security includes hardware security, network security, and software security. The chips in the hardware contain critical information, such as the encryption key. Upper layer applications often rely on the secret keys embedded in these chips. Thus, chip security is one of the most important aspects of IoT security. To ensure IoT chip security, Arm and other chip security research organizations jointly launched the Platform Security Architecture (PSA) certification program in 2019[6]. This program aims to foster better IoT security

• Fei Chen, Duming Luo, and Jianqiang Li are with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China. E-mail: fchen@szu.edu.cn; 1800271043@email.szu.edu.cn; lijq@szu.edu.cn.
• Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, BC V6T 1Z4, Canada. E-mail: vleung@ieee.org.
• Shiqi Li and Junfeng Fan are with Open Security Research, Inc., Shenzhen 518000, China. E-mail: shiqi.li@osr-tech.com; fan@osr-tech.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2021-07-28; revised: 2021-12-12; accepted: 2021-12-13

© The author(s) 2023. The articles published in this open access journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/).

practice by certifying whether an IoT security chip has reached industry security standards. The PSA certification program has three levels[7], with Level 1 being a basic security requirement and with Level 3 indicating the highest security. A few chips have obtained the PSA Levels 1/2 certification, while only two chips have obtained Level 3 certification, and they did so recently in 2021. To the best of our knowledge, no research work has performed a physical leak analysis of such security chips that have already passed PSA Level 2 certification. This topic is the focus of this work.

We use an intuitive example to explain the importance of IoT chip security. Consider the application of a smart lock, which is widely used in many residents and hotels. As an attacker loiters near doors to investigate smart locks with insecure chips, the attacker could obtain the secret keys of these locks by using side-channel attacks, e.g., the one shown later in the main text of the paper. Using such secret keys, the attacker could forge a smart card to unlock the doors. This type of attack is harmful in practice. Therefore, investigating IoT chip security is of great importance.

1.1 Our work

We analyze one security chip that has obtained PSA Level 2 certification and has been used in many IoT applications. The chip is one of the products that have been certified publicly[7]; however, we do not specify it further here for privacy purposes. The chip supports AES encryption that is implemented by its built-in hardware circuit. We use the electromagnetic (EM) side-channel attack to analyze the chip. The goal is to recover the AES encryption key that is stored inside the chip. The main idea of the attack is to leverage physical EM signal leakage to derive the encryption key.

We conduct a series of experiments to analyze the chip. These experiments have the following properties:

- They do not damage the circuit of the IoT development platform where the chip is located;
- They are nonintrusive attacks that do not damage the protective layer of the chip;
- They are able to control the input of plaintext and read the ciphertext from the data output port of the chip by using its development platform;
- They are able to establish triggers on the chip development platform to locate an Advanced Encryption Standard (AES) encryption operation.

Specifically, we first use an EM probe to collect the EM traces of the chip as it executes the AES encryption algorithm. Then, we use an open-source side-channel attack framework to analyze the collected EM traces. The analysis is based on correlation computation, which helps to narrow down the detailed interval of the EM trace where the AES encryption is located. The analysis is a divide-and-conquer analysis that derives the AES encryption key byte by byte. Using the first-round 1-byte S-box output of the AES encryption, we collect the EM traces as templates corresponding to different S box outputs. To recover the secret AES encryption, we collect its EM traces and compute the traces' correlation with the assumed intermediate value traces. The correlation values are used to rank all 256 1-byte candidate keys. This process is repeated 16 times to recover all 16 bytes of the encryption key.

1.2 Challenges

The first challenge is to determine whether the target chip has physical leakage. We conduct a T-test to check for significantly different EM leakage. Specifically, we conduct two experiments. For the first experiment, we collect environmental EM signals outside the chip on random plaintexts and fixed plaintexts. For the second experiment, we repeat the same process, but we place the EM probe on the chip to collect chip information. We find that the signal for random plaintexts and fixed ones in the first experiment are not significantly different. By contrast, the signals are different in the second experiment. This result shows that the target chip leaks information during AES encryption.

The second challenge is the noise in the collected EM traces. We simulate the scenario of an EM side-channel attack as realistically as possible. In practice, the IoT development platform where the security chip is located does not use a stable power supply and is directly exposed to public places. Thus, power supply noise and environmental noise impact the recovery of the AES key from the collected EM traces. In addition, the AES algorithm is implemented in parallel by the chip's hardware circuit and incurs noise. Therefore, the EM trace values mapped simultaneously may include not only the signal values and noise generated by a byte in one step in a round of AES encryption, but also signal leakage values of multiple byte encryption. These issue make the recovery of the AES key difficult.

To solve this challenge, we mainly take two steps. In the first step, we narrow down the detailed AES encryption interval in the EM trace by using two-round correlation analysis experiments. We start by using

plaintext/ciphertext correlation analysis to determine a rough interval. Then, we use an intermediate value (i.e., the output of S-box in the AES's first round of encryption) correlation analysis to further narrow down the encryption interval. In the second step, we control the content of the plaintext by randomizing only 1 byte and keeping other bytes to a fixed value. Then, we collect different EM traces separately. For example, when the EM trace of the first byte is collected, the content of the first byte of the plaintext is a random value in [0, 255], and the content of the other bytes is set to a fixed value, such as zero. Then, we use intermediate value correlation analysis to recover the AES encryption key byte by byte.

1.3 Results, implications, and contributions

After the analysis, we confirm that the target security chip has a physical leak. We also recover around 50% percent of the 16 bytes AES encryption key. To verify that the individual keys are not accidentally obtained, we repeat the test on two more different keys. The same result is obtained; that is, we are still able to recover half of the encryption key.

The analysis results imply the following: Existing IoT chips that have passed the basic PSA Level 2 certification may not be able to resist side-channel attacks. Indeed, PSA certification Levels 1 and 2 do not require such resistance, but such is necessary for Level 3 certification[7]. The problem is that only two chips have obtained Level 3 certification, and they did so very recently in 2021; hence, most existing chips have not reached this level. For average applications, balancing cost and security when choosing IoT chips may be reasonable. For critical applications, chip designers may offer PSA Level 3 products and IoT application designers may impose additional upper layer security mechanisms to enhance the security of emerging smart IoT applications.

In sum, the work makes the following contributions:

- We show that a commodity IoT chip with Arm PSA Level 2 certification leaks secret encryption key under an EM side-channel attack;
- We alert smart IoT application developers to impose high-level security mechanisms, in addition to chip-level ones, so as to establish another security layer for IoT applications.

1.4 Paper organization

The paper proceeds as follows: Section 2 reviews chip security and side-channel attacks. Section 3 introduces the basic information and threat model of the attacked IoT security chip. Section 4 presents the detailed analysis, including the experiment preparation, EM trace collection, EM trace analysis, and AES encryption key recovery. Section 5 concludes the study.

2 Related Work

Our work is related to Integrated Circuit (IC) privacy, side-channel attacks, and IoT security. To improve IC privacy, researchers have proposed several solutions. For side-channel attacks, researchers have proposed different attack methods to attack cryptographic chips. For general IoT security, researchers have studied security attacks and defenses. We review them as herein.

2.1 IC privacy

The IC in a chip is the intellectual property of the enterprise that designs the chip. To protect the intellectual property of the IC from being attacked and leaked, researchers have proposed two approaches[8], i.e., split manufacturing and layout camouflaging.

Zhang[8] proposed a practical logic obfuscation technique to thwart piracy, overbuilding, and reverse engineering. The scheme also protects third-party IP cores. Bi et al.[9] and Qu et al.[10] used emerging transistor technology and digital fingerprinting to protect IC intellectual property. Alasad et al.[11] introduced spintronic devices to help protect ICs with a small performance overhead. Chen et al.[12] used a logic locking test point to protect the hardware. Patnaik et al.[13] combined split manufacturing and layout camouflaging techniques to protect hardware and thereby improve IC intellectual property protection.

2.2 Side channel analysis

A side-channel attack is a type of attack which exploits the physical properties of the chip and its leaked information during execution. Researchers have found various types of side-channel attacks that are able to use the leaked information to recover a secret key in a chip. These attacks include timing analysis[14], hardware fault analysis[15], power analysis[16], EM emissions analysis[17], acoustic cryptanalysis[18], etc.

Researchers have also refined these attacks by processing the leaked information smartly. The basic idea is to look deeply into the data to find other patterns. The refined attacks include differential power analysis[16, 19], correlation power analysis[20], simple

power analysis[16, 21], template analysis[22], algebraic side-channel analysis[23], side-channel cube analysis[24], mutual information analysis[25, 26], normalized inter-class variance analysis[27], signal-to-noise ratio analysis[28, 29], unsupervised learning analysis[30], and machine learning analysis using convolutional neural network and deep learning[31–33].

Normally, side-channel attacks leverage a leakage model that simplifies the attacks. The leakage model groups data into different small-sized sets to form leakage features. The leakage features are later used to match with the leakage of the attacked security chip. Notable leakage models are the Hamming weight model[20, 34, 35], Hamming distance model[34–36], and mono-bit model[37, 38].

2.3 IoT security

IoT applications have been adopted in various applications, e.g., smart voice assistant[39, 40], smart plug[1], and smart data logger[41]. Although these applications bring convenience to people's lives, they also come with security concerns. For instance, Diao et al.[40] showed that voice assistant applications could leak user's private information. Ling et al.[1] found that a user's smart plug could be made unusable due to attacks.

To enhance IoT security, researchers have proposed various approaches. Azrour et al.[42] proposed an authentication scheme to secure IoT applications. The proposed scheme employs hashing and efficient exclusion-or operations to achieve mutual authentication. Alladi et al.[43] also proposed using encryption and integrity checks to enhance consumer IoT application security. IoT security is obviously being intensively studied; interested readers may refer to some recent surveys (e.g., Ref. [44]) for a more broad and detailed review.

3 Target Security Chip and Threat Model

3.1 Target security chip

We studied one security chip that is widely used for IoT applications, e.g., smart homes, smart city facilities, fingerprint cards, fingerprint locks, digital currency authentication devices, and wireless sensor node devices. The chip has passed the Arm PSA Level 2 certification[7]. The security chip uses Armv8-M architecture and TrustZone technology, including hardware encryption accelerators and real random number generators with different encryption algorithms. This chip supports "secure" AES-128 encryption using the hardware encryption accelerator in the trust zone. We introduce the basis as follows.

Arm PSA improves IoT hardware security and reduces development costs. The architecture consists of four key phases: analysis, design, implementation, and certification. PSA certification, in particular, is divided into three levels. As of July 27, 2021, a few products have passed PSA Level 1 certification; nine products have passed PSA Level 2 certification; and two products have passed Level 3 certification, and both of them were certified very recently in 2021[7]. After passing the PSA Levels 1/2 certification, the chip is recognized as able to meet the universal security standards of the industry[6]. PSA Levels 1/2 certification does not require resistance to side-channel attacks, whereas Level 3 certification does require such resistance. Nevertheless, given that most existing chips lack Level 3 certification and that the target chip is used in common emerging smart IoT applications, we should still analyze the target chip using side-channel analysis to comprehensively understand the security of emerging smart IoT applications.

Armv8-M architecture[45] is Arm's latest instruction set architecture. It is compatible with the 32-bit instruction set in the Armv7 architecture, and it adds a 64-bit instruction set. The Armv8-M architecture also supports hardware virtualization. Hence, the computing power of the devices using this architecture is greatly enhanced.

TrustZone technology[46] is Arm's solution that combines software and hardware to improve chip security. Its main essence is to divide hardware and software into a secure world and a non-secure world. The secure world can access the resources of the non-secure world, but the non-secure world cannot access the resources of the secure world. Therefore, the secure and non-secure worlds are isolated. Independent security systems are used in the secure world. If developers need to use the secure world, then they need to access it through the provided API.

3.2 Threat model

Our work focuses on the use of EM side-channel attacks to recover the secret key of the AES-128 algorithm running in the target security chip. We explain the threat model by describing the attacker's target, knowledge, ability, and attack strategy in detail.

The attacker's target is to recover all/part of the key of the encryption algorithm from the encryption operations of the target security chip which has PSA

Level 2 certification. The result should highlight that the security chip has the risk of leaking private information physically.

The attacker's knowledge includes plaintext and ciphertext pairs for a training key and the attacked unknown key. The attacker is able to call the AES encryption API by referring to the user manual of the target chip. Note that the attacker does not know the structure of the encryption circuit inside the target security chip. The attacker also does not know the protection strategy that may be used by the target chip. Specifically, the attack is a type of gray box test.

The attacker is able to do the following:

• Establish trigger signals on the IoT development platform;

• Invoke the device API multiple times and obtain the plaintext and ciphertext pairs;

• Use EM probes and oscilloscopes to collect EM traces;

• Analyze EM traces;

• Meanwhile, that attacker is not able to directly access the protected encryption key.

Attackers employ different strategies. In this work, we roughly employ the following approach: First, after collecting the EM traces and obtaining the corresponding plaintext and ciphertext pairs, the attacker preprocesses the EM trace. Preprocessing may include filtering, frequency domain transformation, horizontal movement operations, identification of the pattern and peak, and extraction of the index for splicing. Second, the attacker needs to conduct statistical tests on the preprocessed EM traces. If the test indicates a leakage, the attacker then conducts correlation analysis of the plaintext and ciphertext on the preprocessed EM traces. Third, if strong correlation exists in the EM traces, then the attacker conducts intermediate value correlation analysis. The intermediate values are normally related to the secret encryption key. After determining a clear correlation between the intermediate values, the attacks can then be determined on the basis of these values.

4 Detailed Analysis

4.1 Methodology

We use EM side channel attack to analyze the target chip. At a high level, Fig. 1 shows the methodology. We summarize it into three parts: experiment preparation, EM trace processing, and key recovery.

We collect the EM signal when the chip performs

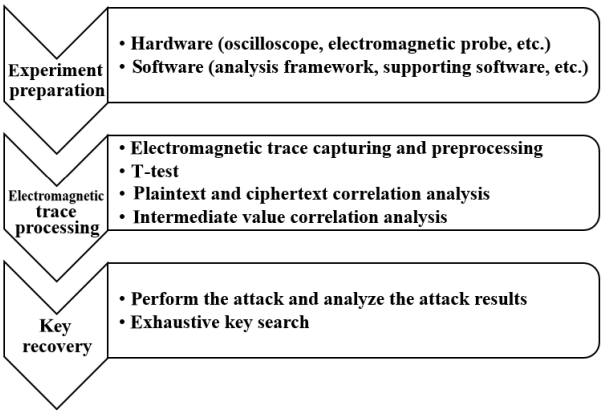

Fig. 1 Analysis methodology.

cryptographic operations. We then analyze the EM trace of the captured EM signal. On the basis of the EM trace, we finally try to recover the key using correlation analysis. To perform the analysis, we need to set up a hardware environment, capture data, and analyze the data using certain analysis methodology and key recovery strategies.

Experiment preparation is mainly divided into hardware and software preparation. The hardware needs to prepare instruments for collecting and displaying EM traces, including oscilloscopes and EM probes. On the software side, the programming environment, EM trace analysis framework, and supporting software that communicates with the hardware need to be prepared.

EM trace processing is mainly divided into preprocessing, statistical T-test, and correlation analysis. Preprocessing is to reduce noise in the EM trace and facilitate subsequent analysis. Statistical T-test is another physical leak test of the EM trace after the preprocessing step. Correlation analysis is divided into plaintext correlation analysis, ciphertext correlation analysis, and intermediate value correlation analysis; it is mainly used to determine attack parameters, such as attack range in the captured EM trace.

Key recovery mainly consists of executing an attack, analyzing the results of the attack, and formulating an exhaustive key recovery strategy. Executing an attack and analyzing the results of the attack require the use of programs to record the plaintext-ciphertext pairs and the EM power values during the attack. Exhaustive key recovery is based on a divide-and-conquer method to restore the key byte by byte.

4.2 Experiment preparation

To analyze the security of the target chip, we need to set up an experimental environment. The setup includes

hardware and software preparations. We detail them in this subsection.

Figure 2 shows the hardware setup. We used an EM probe with a measurement bandwidth of 30 MHz–3 GHz, a Pico3000 oscilloscope[47], and a low-pass filter to collect and observe EM traces. As side-channel attacks could be easily affected by the environment, we should keep the experimental environment as stable as possible. In our analysis, we maintained a constant temperature. When collecting EM traces, we placed the acquisition position of the probe in a fixed state and at a sufficiently close distance to the chip.

For the software preparation, we used the SSCOM serial port assistant[48] to debug the device. We used the Pico3000 oscilloscope supporting software to observe the EM trace. To further analyze the EM trace, we also used other software frameworks based on Python 3.6. Specifically, we used Scared[49], which is an open-source side-channel analysis framework. The experiment mainly uses the oscilloscope module, communication module, and verification module. The oscilloscope module is used to control the oscilloscope parameters for data acquisition. The communication module is utilized for the random transmission of plaintext on the software to the IoT chip development platform. It is also used to receive the corresponding ciphertext from the IoT chip development platform. The verification module is used to verify whether the transmitted plaintext corresponds to the ciphertext received from the platform. This framework also includes a callback module and a scheduling module for advanced EM trace collection, but it was not used in our analysis.

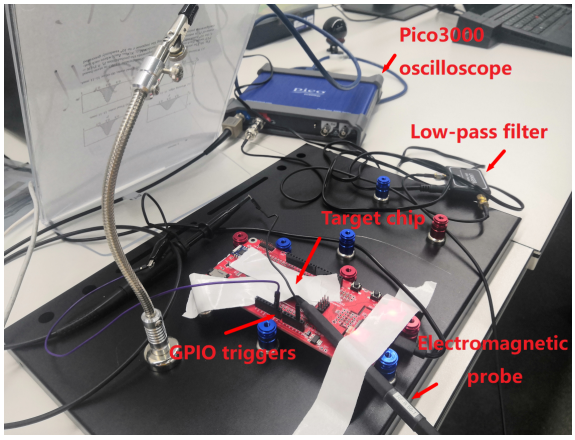

Fig. 2 Experimental environment.

After the hardware and software were prepared, we started to prepare for the collection of EM traces. By referring to the manual of the target IoT chip and its development platform, we learned to use its API to invoke AES encryption. When invoking the AES encryption API, we wrapped the API invocation with two General Purpose Input/Output (GPIO) trigger calls. In this way, we could identify the AES call by inducing two spikes in the captured EM trace. To ensure the stability of the trigger, we used the oscilloscope to validate its amplitude. If it was not stable, then we adjusted the EM probe such that it was near the point where the EM leakage of the chip was the strongest. We also adapted the parameters for the oscilloscope software manually such that the most suitable/stable parameters were used to capture EM traces.

Once the preparation was finished and the captured signal was stable, we started to collect the EM traces. We collected 1000 samples before the trigger and another 1000 samples after the trigger. In this way, we could gain an overview of the EM trace and determine how many points should be used after the trigger. Figure 3 shows an example of 100 captured EM traces. The x-axis denotes the time index of the captured data; the y-axis denotes

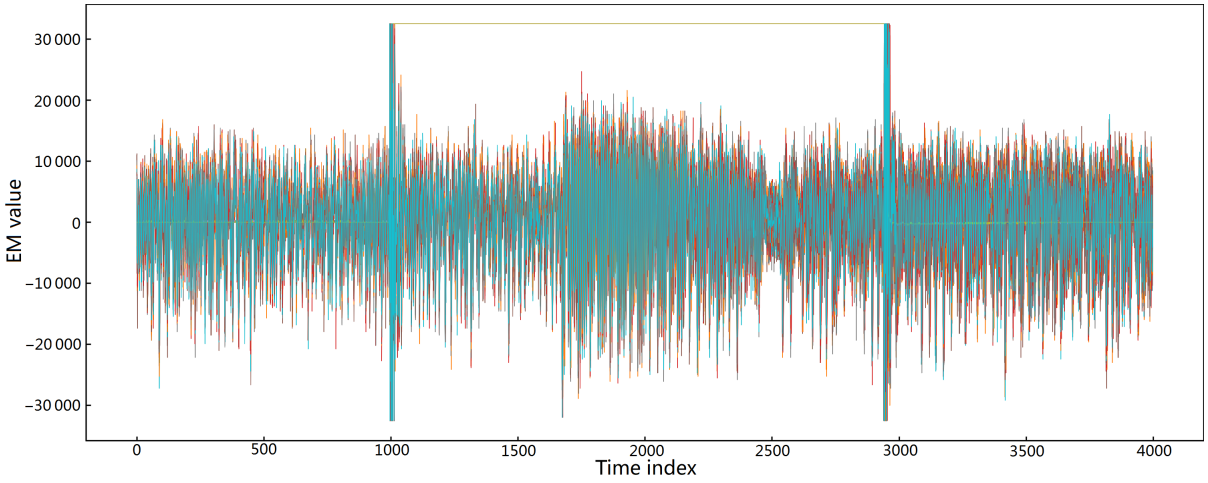

Fig. 3 100 initial overlapping electromagnetic traces.

the EM radiation intensity at that moment. When the time index is 1000, the trigger starts; when the index is 3000, the trigger ends. According to the trigger code and the setting of the oscilloscope, we actually collected 2000 points of the EM trace during the AES encryption operation. Using the trigger trick, we reduced the cost of hard disk resources to store the captured traces.

4.3 Electromagnetic trace processing

With all the preparation work done, we started collecting and processing the EM traces. The goal was to determine whether the chip leaks information and whether the leak exists in the captured EM trace. The processing of the EM traces mainly comprises four steps, i.e., preprocessing, T-test, plaintext and ciphertext correlation analysis, and intermediate value correlation analysis. In the following, we discuss these steps in detail.

4.3.1 Electromagnetic trace preprocessing

The EM traces are preprocessed as we do not know which defensive strategies the target chip may adopt to protect its security. Thus, we need to make all the collected EM traces have the same characteristic. An EM trace collected without preprocessing may not be conducive to EM trace analysis.

In the analysis, we mainly tried to align all the collected EM traces. The alignment involves four operations: horizontal movement, pattern recognition, peak and width recognition, and index extraction. For the target chip, we first averaged all the collected EM traces and then randomly extracted other traces for comparison with the average. Figure 4 shows the averaged and randomly extracted EM trace; the orange color marks the mean EM trace while the blue color is the randomly drawn EM trace.

Figure 4 shows that the averaged EM trace has a high degree of overlap with the random EM trace in the yellow-green shaded part. We repeated the experiment and found that the same holds for all EM traces. We then concluded that the collected EM traces were aligned with one another without excessive preprocessing. Thus, in our later analysis, the EM traces that we collected subsequently were not preprocessed further.

4.3.2 T-test

Next, we aim to understand whether the target chip leaks information during the encryption operation. We mainly use a T-test to distinguish the captured EM trace for random plaintexts and fixed plaintexts in two different settings. In one setting, we placed the EM probe in the environment. In the other setting, we fixed the probe on the part of the target chip where the EM signal was the strongest.

The T-test steps are as follows (see Fig. 5):

- Use Pico3000 oscilloscope to collect traces for random plaintexts and a fixed plaintext;
- Preprocess the traces to store them on the local disk;
- Group the traces into two categories, i.e., random plaintext and fixed plaintext;
- Conduct a statistical T-test to validate the differences of the traces.

We explain the steps in detail. We fixed an encryption key using the development platform of the chip. Using the fixed key, we encrypted different plaintexts. There are two types of plaintext in the collected EM trace. One is the randomly generated 16-byte plaintext. The other is the fixed 16-byte plaintext. To reduce the impact of the environment on the collection of EM traces, we used a pseudorandom sequence of 0 s and 1 s. When the random number was 0, we chose to transmit and encrypt the fixed plaintext; otherwise, we used the random plaintext. In total, we collected 40 000 EM traces. Each EM trace was collected from 1000 points before the trigger, and each trace had 3000 points. After collecting the EM traces, we grouped them into fixed plaintext EM traces and random plaintext EM traces. Finally, we conducted a T-test.

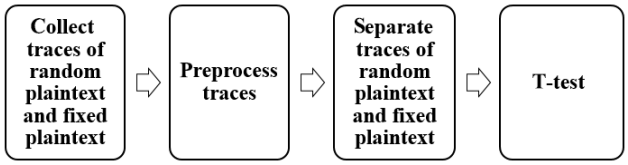

Fig. 5 T-test steps.

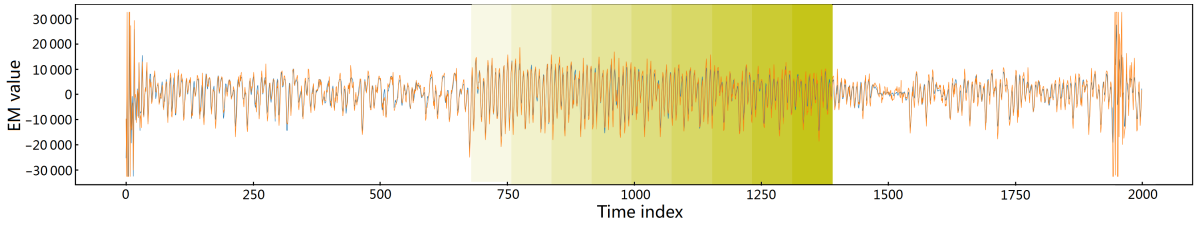

Fig. 4 Electromagnetic trace alignment analysis by comparing averaged and random traces.

Figure 6 shows the environmental T-test result. The hypothesis is as follows: first, the two types of EM traces corresponding to random and fixed plaintexts are the same; second, they are different. The T-test results of the environment-tested EM trace do not exceed [–4.5, 4.5]. Thus, we concluded that the environment exerts a certain influence on the side-channel attack experiment; however, the difference between the two experiments is not significant. No significant leakage exists for the encryption process of the target chip in this environmental setting.

Figure 7 shows the target chip's T-test result. The hypothesis is similar. We found some ranges (i.e., dark yellow shaded area) inside the trigger where the test results exceed [–4.5, 4.5]. This result indicated that the two types of captured EM traces are significantly different. That is, the EM signals between a fixed plaintext and a random plaintext are different.

Thus, we derived two pieces of information. First, the target chip leaked information during the encryption operation. Second, we narrowed down the spot of the leakage inside the trace as in the yellow shaded area in Fig. 7.

4.3.3 Plaintext and ciphertext correlation analysis

Next, we conducted a correlation analysis of the EM traces corresponding to different plaintexts and ciphertexts. The aim is to further narrow down the detailed position of the AES encryption operation in the captured EM trace and thereby expedite the subsequent encryption key recovery.

The steps are as follows. Using the same environment above, we recollected the EM traces using the same key by inputting random plaintexts. In total, we collected 100 000 EM traces, each of which had 2000 points from the trigger. For each byte of the plaintext, we grouped the captured EM traces according to the byte value. For each group, we computed the correlation coefficients. Among them, we chose the largest one in terms of absolute value as the correlation result for the byte. We continued this process 16 times to compute all coefficients for all 16 bytes at each sampling EM trace point.

Similarly, we computed the correlation results for the 16-byte ciphertext. Figures 8 and 9 show the correlation results of all the 16 bytes that correspond to plaintext and ciphertext correlations. The 16 curves in Figs. 8 and 9 correspond to the 16 bytes of the AES plaintext or

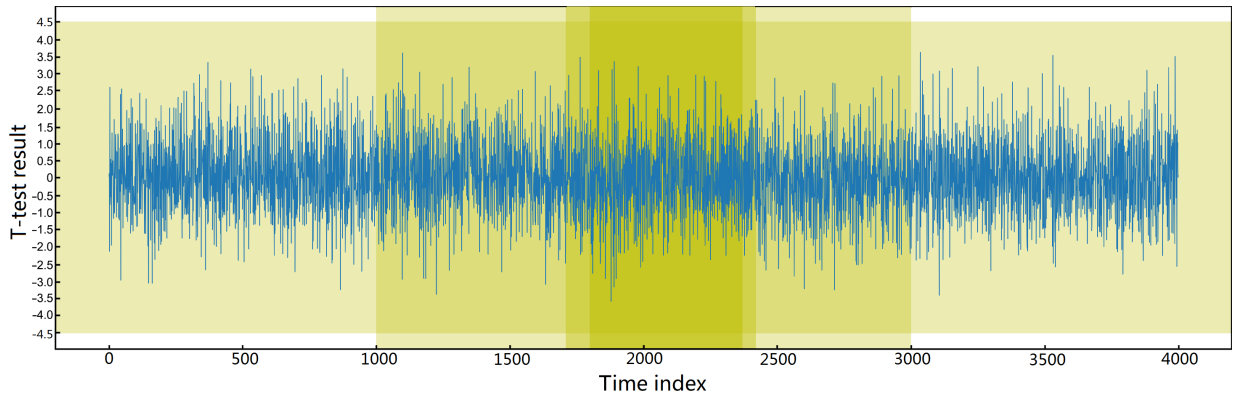

Fig. 6 T-test result of environmental electromagnetic trace.

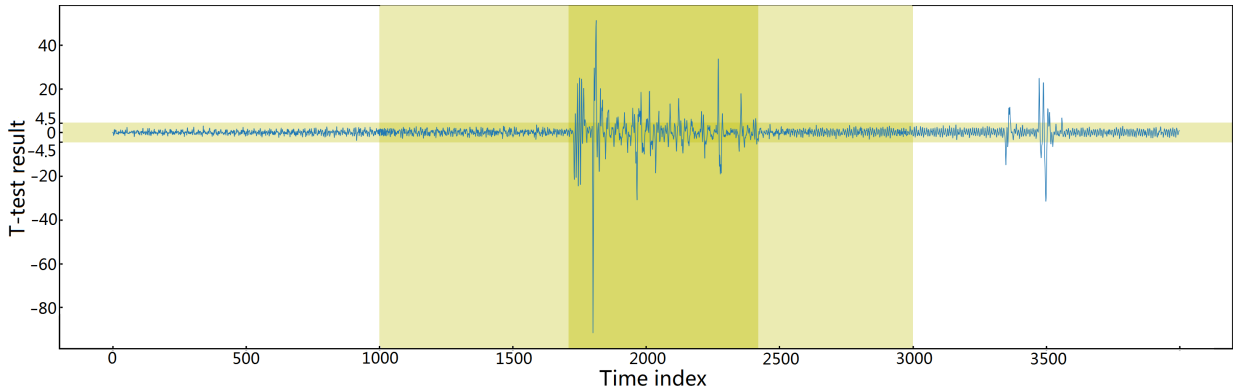

Fig. 7 T-test result of target chip electromagnetic trace.

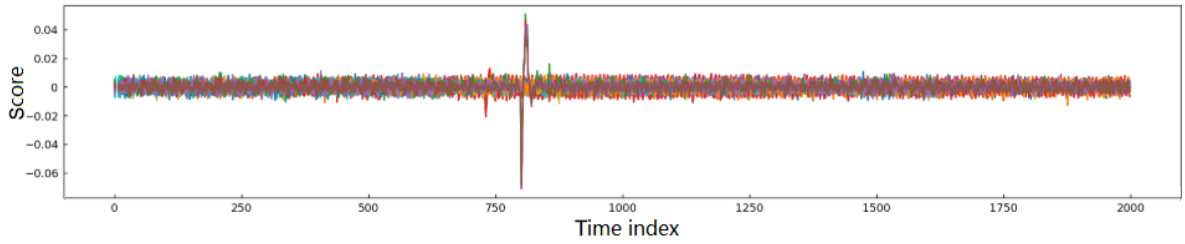

Fig. 8 Plaintext correlation analysis.

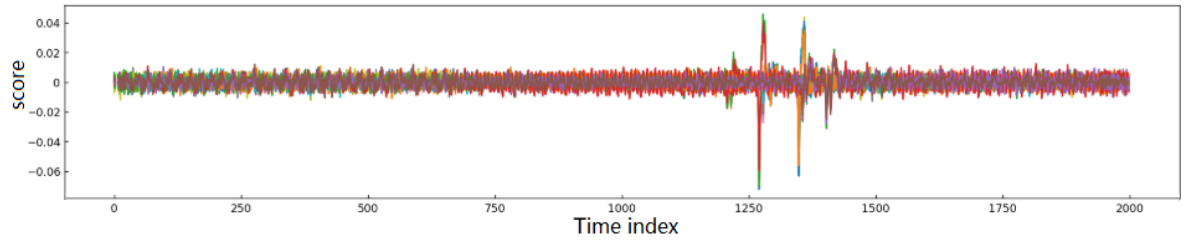

Fig. 9 Ciphertext correlation analysis.

ciphertext. We found that the correlation results of the plaintext and ciphertext have obvious peaks at certain positions. According to the position of the correlation peaks of the plaintext and ciphertext, we inferred the interval of the AES encryption operation in the EM trace. After zooming in the positions of the front and back spikes, we obtained the encryption interval among the EM trace. The encryption interval is included in the T-test leakage interval, thus further proving that the AES encryption operation of the target chip has physical leakage. After amplification and confirmation, we locked the encryption interval in the EM trace to the range [731, 1425]. This interval is the same for each EM trace.

4.3.4 Intermediate value correlation analysis

Although the plaintext/ciphertext correlation analysis has narrowed down the AES encryption interval, we further narrowed down such interval to reduce the attack time. We conducted an intermediate value correlation analysis. The idea was to investigate the EM traces corresponding to some fixed internal states of the AES encryption. Specifically, we used the output of the *S*-box in the first round of AES encryption as the intermediate value.

The detailed correlation analysis was similar to the correlation analysis of the plaintext/ciphertext, except that the EM traces were grouped according to the intermediate values. Specifically, we first used 100 000 EM traces with 16 bytes of random plaintext to perform the intermediate value correlation analysis. However, we found that the peak of the correlation of the intermediate value was not obvious. After studying the target chip and its hardware security accelerator, we conjectured the reason for the inconspicuous result. It might be because the AES encryption operation was implemented through parallel circuits. Hence, the general method could not achieve good results.

Later, we adopted another strategy. We only randomized one byte of the plaintext; the other bytes of the plaintext were all set to a fixed value. We recollected the EM traces and performed intermediate value correlation analysis. Similarly, we conducted the correlation analysis for the remaining 15 bytes of the intermediate value.

We found that although the locations of the 16-byte intermediate correlation peaks did not appear in order, they always appeared in a stable range. Therefore, we took the union of the intervals where the correlation peaks of each intermediate value were located as the attack range of [860, 915].

4.4 Key recovery

Finally, we studied whether it was possible to recover an encryption key of the target chip. We further conducted three experiments on three encryption keys, as shown in Table 1. In all three experiments, we were able to recover a part of the key, e.g., up to 10 bytes of the total 16-byte key. This result showed that the target chip is indeed leaking key information.

The attack can be described as follows. We attacked the key byte by byte using the intermediate correlation analysis. For one specific byte, there are 256 different guessing keys. Fixing a plaintext, we obtained its EM

Table 1 Attacked keys.

Version	Byte															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Key 1	0xa	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
Key 2	0x1a	0xca	0xbb	0xa6	0x4	0x15	0x86	0x7	0x28	0x10	0xa	0xb	0xc3	0xcd	0xfe	0xee
Key 3	0xbb	0x2a	0xbd	0xa1	0x74	0x1b	0x8e	0xe7	0x28	0x10	0xca	0xb0	0x3	0xcd	0xfe	0xae

trace on the intermediate value. We computed the 256 different correlations with the EM trace obtained in the intermediate analysis step. By choosing the largest correlation, we derived the correct intermediate value and thus the corresponding key byte.

After repeating this process for all 16 key bytes, we were finally able to recover the AES encryption key. In some cases, the largest correlation result does not always correspond to the correct key byte due to noise. Thus, we chose the top 10 candidate keys among 256 possible keys for one key byte. Then, an exhaustive search was performed to search for the correct key; it is much faster than brute force searching 2^{128} possible encryption keys.

4.4.1 Key recovery using intermediate correlation analysis

The experimental setup is as follows. We used only one randomized byte of the plaintext, while the other bytes of the plaintext were set as fixed. We used two attack models. One used the intermediate value $S(i)$, which is the i-th byte output of the S-box in the first round of AES encryption; the other is

$$S(i) \oplus S(i-1).$$

Corresponding to the intermediate value analysis, the attack range of the EM trace was [860, 915].

To recover the encryption key, we mainly focused on the correlation between the collected EM traces and the ones that were captured in the intermediate value analysis. The main idea is that when the plaintext is known, the attacker obtains different hypothetical intermediate values, including the true intermediate value by guessing the byte key K_i where $0 \leqslant i \leqslant 15$. Only the true intermediate value will have a strong correlation with the EM trace that corresponds to the target chip on an unknown AES encryption key; then, one byte key can be successfully recovered. The EM traces required to recover keys of different bytes in our experiment were not exactly the same. It ranged between 20 000 and 1 000 000.

Table 2 shows the key recovery result. The column "Top 10/1" denotes how many key bytes among the 16 key bytes are ranked in the first 10/1 choices sorted by the correlation analysis result. For the first experiment on attacking key 1, we randomized only 1 byte in the plaintext and set the other bytes fixed to be 0 or 1. When using $S(i)$ as the intermediate value as in the second row, 6 bytes of the 16 key bytes were ranked first in the correlation analysis. That is, these 6 bytes were easily recovered. Another 4 bytes were ranked in the top 10 candidate key bytes according to the correlation result. Hence, these 4 bytes could be searched more easily by exhausting the potential key spaces. A similar analysis applies to the other keys.

In general, about 50% of the key bytes were ranked as the top candidate keys according to the correlation analysis. About 68% of the key bytes were ranked in the top 10. This result offers strong evidence that the target chip has a considerable security risk.

4.4.2 Exhaustive key search

The results indicate that the attack experiments recover more than half of the bytes of the encryption key. However, we could not fully recover all the 16 key bytes because our experimental conditions were

Table 2 Key recovery ranking based on different intermediate values.

Version	Byte																Top 10	Top 1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Key 1; fixed byte 1; intermediate value $S(i)$	1	7	15	31	25	1	2	1	2	35	1	25	6	12	1	1	10	6
Key 1; fixed byte 1; intermediate value $S(i) \oplus S(i-1)$	1	91	5	5	75	1	1	1	1	20	1	1	1	7	1	1	13	10
Key 1; fixed byte 0; intermediate value $S(i)$	11	100	20	4	1	195	3	2	78	6	1	1	1	107	56	39	8	4
Key 1; fixed byte 0; intermediate value $S(i) \oplus S(i-1)$	11	3	206	34	1	35	3	1	5	6	1	1	1	173	8	66	10	5
Key 2; fixed byte 0; intermediate value $S(i)$	1	1	11	1	1	16	36	1	1	33	1	1	53	202	1	1	10	10
Key 2; fixed byte 1; intermediate value $S(i) \oplus S(i-1)$	1	162	6	1	1	16	143	1	1	8	161	3	6	1	1	2	12	7
Key 3; fixed byte 0; intermediate value $S(i)$	1	117	1	120	2	34	22	1	1	9	9	2	1	1	52	3	11	6
Key 3; fixed byte 1; intermediate value $S(i) \oplus S(i-1)$	1	86	1	255	164	66	32	1	1	24	25	1	1	44	145	5	7	6

relatively simple and the attack method was a general attack.

One can leverage an exhaustive key search to recover the complete key potentially. The specific exhaustion strategy is as follows. According to the correlation analysis result, the potential keys are ranked. If the correlation analysis shows a fairly obvious spike for candidate key bytes, we may set an exhaustive space as the top candidates (for example, top 10 candidate key bytes). That is, ranking is performed according to the correlation result, and the guessing key corresponding to the top key byte candidates is selected. The order of the exhaustive key search is from the higher correlation key byte to the lower ones. If the correlation analysis does not show a sharp peak, we may set the exhaustion space of the key of using all 256 potential key bytes. This strategy accelerates key searching.

4.5 Result analysis

From the experimental results, we can prove that the target chip has physical leakage. Indeed, our analysis successfully recovered a significant part of the key bytes. Our experiment environment was rudimentary; we also did not use high-precision instruments and did not greatly preprocess the collected EM traces. However, we directly recovered nearly half of the keys through attacks, thereby greatly reducing the key exhaustive search space. To ensure the rigor of the experiment, we also used three different keys to avoid accidental key recovery. Combined with an exhaustive search strategy, one may obtain key information in a real security chip much faster than using brute force key search.

4.6 Discussion

The target chip has passed Arm's PSA Level 2 certification and is already more secure than most security chips in the market. Despite this certification, it is not sufficient as it still has the risk of leaking secret encryption key information. PSA Level 1/2 certification is only a basic requirement for security chips, whereas PSA Level 3 certification is more secure. However, only two chips have obtained a Level 3 certification, and most existing chips have yet to obtain such certification.

Our analysis has two implications. From the producer's perspective, a PSA Level 3 certification is better as it enables a stricter security guarantee. In addition to the general circuit safety standards, the security design of a security chip should also consider the physical leakage that may be exploited by side-channel attacks. The security chip designer could cooperate with a professional hardware security chip design company to conduct a comprehensive security inspection from design to testing. As the security chip is a core part of IoT, which is used widely, chip/system owners could shorten the time to update the embedded secret keys in the chips to ensure security for IoT applications. The system should also have additional security mechanisms in case the chip leaks encryption key information.

From the consumer's perspective, a balance should be established between chip security, performance, and price. A consumer may also read the detailed PSA certification manuals to have a deep understanding of the PSA certifications[7]. As far as we know, this work is the first publicly reported attack on a chip with Arm PSA Level 2 certification. Thus, experimental comparison with existing attacks is not yet possible.

5 Conclusion

This work mainly analyzes a security chip that has passed Arm's PSA Level 2 certification. We have successfully recovered half of the bytes of the AES encryption key in the security chip by using EM side-channel analysis. We show the detailed process to recover the key, from the preparation of the experiment environment to the final encryption key derivation using EM trace analysis. The analysis is repeated using three different keys to validate the effectiveness of the attack. Our conclusion is that PSA Level 1/2 certification, although a mainstream market chip security certification, is only a basic requirement of IoT chip security. For critical applications, chip producers and consumers should focus on PSA level 3 certification. Although only two chips have obtained such certification as of July 2021, they represent a future direction for IoT chip designs. Upper layer IoT applications should also have security mechanisms other than the existing ones in security chips.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (Nos. 61872243 and U1713212), Guangdong Basic and Applied Basic Research Foundation (No. 2020A1515011489), the Natural Science Foundation of Guangdong Province-Outstanding Youth Program (No. 2019B151502018), and Shenzhen Science and Technology Innovation Commission (No. R2020A045).

Responsible Disclosure

The analysis result found in this research has been communicated to the chip manufacturer.

References

[1] Z. Ling, J. Z. Luo, Y. L. Xu, C. Gao, K. Wu, and X. W. Fu, Security vulnerabilities of internet of things: A case study of the smart plug system, *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1899–1909, 2017.

[2] Z. N. Mohammad, F. Farha, A. O. M. Abuassba, S. K. Yang, and F. Zhou, Access control and authorization in smart homes: A survey, *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 906–917, 2021.

[3] D. W. Wei, H. S. Ning, F. F. Shi, Y. L. Wan, J. B. Xu, S. K. Yang, and L. Zhu, Dataflow management in the internet of things: Sensing, control, and security, *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 918–930, 2021.

[4] Z. Ling, C. Gao, C. Sano, C. Toe, Z. P. Li, and X. W. Fu, STIR: A smart and trustworthy IoT system interconnecting legacy IR devices, *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3958–3967, 2020.

[5] W. P. Wang, Z. R. Wang, Z. F. Zhou, H. X. Deng, W. L. Zhao, C. Y. Wang, and Y. Z. Guo, Anomaly detection of industrial control systems based on transfer learning, *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 821–832, 2021.

[6] Arm, Arm platform security architecture: Overview, https://www.design-reuse-embedded.com/displayIP/iot_2_arm_platform_security_architecture, 2021.

[7] PSA, PSA Certified Products, https://www.psacertified.org/certified-products/, 2021.

[8] J. L. Zhang, A practical logic obfuscation technique for hardware security, *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 24, no. 3, pp. 1193–1197, 2016.

[9] Y. Bi, X. S. Hu, Y. Jin, M. Niemier, K. Shamsi, and X. Z. Yin, Enhancing hardware security with emerging transistor technologies, in *Proc. 26th Edition on Great Lakes Symp. on VLSI*, Boston, MA, USA, 2016, pp. 305–310.

[10] G. Qu, C. Dunbar, X. Chen, and A. J. Cui, Digital fingerprint: A practical hardware security primitive, in *Digital Fingerprinting*, C. Wang, R. Gerdes, Y. Guan, S. Kasera, eds. New York, NY, USA: Springer, 2016, pp. 89–114.

[11] Q. Alasad, J. Yuan, and D. L. Fan, Leveraging all-spin logic to improve hardware security, in *Proc. of the on Great Lakes Symp. on VLSI 2017*, Banff, Canada, 2017, pp. 491–494.

[12] M. Chen, E. Moghaddam, N. Mukherjee, J. Rajski, J. Tyszer, and J. Zawada, Hardware protection via logic locking test points, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3020–3030, 2018.

[13] S. Patnaik, M. Ashraf, O. Sinanoglu, and J. Knechtel, A modern approach to IP protection and Trojan prevention: Split manufacturing for 3D ICs and obfuscation of vertical interconnects, *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 4, pp. 1815–1834, 2021.

[14] P. C. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, in *Proc. 16th Annu. Int. Cryptology Conf.*, Santa Barbara, CA, USA, 1996, pp. 104–113.

[15] D. Boneh, R. A. DeMillo, and R. J. Lipton, On the importance of checking cryptographic protocols for faults, in *Proc. Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Konstanz, Germany, 1997, pp. 37–51.

[16] P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, in *Proc. 19th Annu. Int. Cryptology Conf.*, Santa Barbara, CA, USA, 1999, pp. 388–397.

[17] J. J. Quisquater, and D. Samyde, A new tool for non-intrusive analysis of smart cards based on electromagnetic emissions: The SEMA and DEMA methods, presented at the EUROCRYPT 2000 Rump Session, https://link.springer.com/chapter/10.1007/3-540-45418-7_17, 2000.

[18] D. Genkin, A. Shamir, and E. Tromer, Acoustic cryptanalysis, *J. Cryptol.*, vol. 30, no. 2, pp. 392–443, 2017.

[19] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, Introduction to differential power analysis, *J. Cryptogr. Eng.*, vol. 1, no. 1, pp. 5–27, 2011.

[20] E. Brier, C. Clavier, and F. Olivier, Correlation power analysis with a leakage model, in *Proc. 6th Int. Workshop on Cryptographic Hardware and Embedded Systems*, Cambridge, MA, USA, 2004, pp. 16–29.

[21] S. Mangard, A simple power-analysis (SPA) attack on implementations of the AES key expansion, in *Proc. 5th Int. Conf. on Information Security and Cryptology*, Seoul, Republic of Korea, 2002, pp. 343–358.

[22] S. Chari, J. R. Rao, and P. Rohatgi, Template attacks, in *Proc. 4th Int. Workshop on Cryptographic Hardware and Embedded Systems*, Redwood Shores, CA, USA, 2002, pp. 13–28.

[23] M. Renauld and F. X. Standaert, Algebraic side-channel attacks, in *Proc. 5th Int. Conf. on Information Security and Cryptology*, Beijing, China, 2009, pp. 393–410.

[24] I. Dinur and A. Shamir, Side channel cube attacks on block ciphers, https://eprint.iacr.org/2009/127.pdf, 2021.

[25] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, Mutual information analysis, in *Proc. 10th Int. Workshop on Cryptographic Hardware and Embedded Systems*, Washington, DC, USA, 2008, pp. 426–442.

[26] L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F. X. Standaert, and N. Veyrat-Charvillon, Mutual information analysis: A comprehensive study, *J. Cryptol.*, vol. 24, no. 2, pp. 269–291, 2011.

[27] S. Bhasin, J. L. Danger, S. Guilley, and Z. Najm, NICV: Normalized inter-class variance for detection of side-channel leakage, in *Proc. 2014 Int. Symp. on Electromagnetic Compatibility*, Tokyo, Japan, 2014, pp. 310–313.

[28] S. M. Del Pozo, F. X. Standaert, D. Kamel, and A. Moradi, Side-channel attacks from static power: When should we care? in *Proc. 2015 Design, Automation & Test in Europe*

Conf. & Exhibition, Grenoble, France, 2015, pp. 145–150.
[29] Y. S. Fei, A. A. Ding, J. Lao, and L. W. Zhang, A statistics-based fundamental model for side-channel attack analysis, https://eprint.iacr.org/2014/152.pdf, 2014.
[30] J. W. Chou, M. H. Chu, Y. L. Tsai, Y. Jin, C. M. Cheng, and S. D. Lin, An unsupervised learning model to perform side channel attack, in *Proc. 17th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, Gold Coast, Australia, 2013, pp. 414–425.
[31] S. Picek, A. Heuser, A. Jovic, S. A. Ludwig, S. Guilley, D. Jakobovic, and N. Mentens, Side-channel analysis and machine learning: A practical perspective, in *Proc. 2017 Int. Joint Conf. on Neural Networks*, Anchorage, AK, USA, 2017, pp. 4095–4102.
[32] L. X. Wei, B. Luo, Y. Li, Y. N. Liu, and Q. Xu, I know what you see: Power side-channel attack on convolutional neural network accelerators, in *Proc. 34th Annu. Comput. Security Applications Conf.*, San Juan, UT, USA, 2018, pp. 393–406.
[33] W. Yu and J. Chen, Deep learning-assisted and combined attack: A novel side-channel attack, *Electron. Lett.*, vol. 54, no. 19, pp. 1114–1116, 2018.
[34] O. Lo, W. J. Buchanan, and D. Carson, Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA), *J. Cyber Secur. Technol.*, vol. 1, no. 2, pp. 88–107, 2017.
[35] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Boston, MA, USA: Springer, 2007.
[36] J. Li, W. W. Shan, and C. X. Tian, Hamming distance model based power analysis for cryptographic algorithms, *Appl. Mech. Mater.*, vols. 121–126, pp. 867–871, 2011.
[37] E. de Chérisey, S. Guilley, O. Rioul, and P. Piantanida, Best Information is most successful: Mutual information and success rate in side-channel analysis, *IACR Transactions on Cryptographic Hardware and Embedded Systems*, doi: https://doi.org/10.13154/tches.v2019.i2.49-79.
[38] J. Doget, E. Prouff, M. Rivain, and F. X. Standaert, Univariate side channel attacks and leakage modeling, *J. Cryptogr. Eng.*, vol. 1, no. 2, pp. 123–144, 2011.
[39] V. Kepuska and G. Bohouta, Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home), in *Proc. 2018 IEEE 8th Annu. Computing and Communication Workshop and Conf. (CCWC)*, Las Vegas, NV, USA, 2018, pp. 99–103.
[40] W. R. Diao, X. Y. Liu, Z. Zhou, and K. H. Zhang, Your voice assistant is mine: How to abuse speakers to steal information and control your phone, in *Proc. 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, Scottsdale, AZ, USA, 2014, pp. 63–74.
[41] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. El Hajjaji, IoT-based data logger for weather monitoring using Arduino-based wireless sensor networks with remote graphical application and alerts, *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.
[42] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, New enhanced authentication protocol for internet of things, *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.
[43] T. Alladi, V. Chamola, B. Sikdar, and K. K. R. Choo, Consumer IoT: Security vulnerability case studies and solutions, *IEEE Consum. Electron. Mag.*, vol. 9, no. 2, pp. 17–25, 2020.
[44] F. Chen, D. M. Luo, T. Xiang, P. Chen, J. F. Fan, and H. L. Truong, IoT cloud security review: A case study approach using emerging consumer-oriented applications, *ACM Comput. Surv.*, vol. 54, no. 4, pp. 75, 2022.
[45] ARM, ARMv8-A architecture overview, https://armkeil.blob.core.windows.net/developer/Files/pdf/graphics-and-multimedia/ARMv8_Overview.pdf, 2015.
[46] Arm trustzone technology for the armv8-m architecture, https://developer.arm.com/documentation/100690/latest/, 2021.
[47] PicoScope 3000 series oscilloscope software, Pico Technology, https://www.picotech.com/oscilloscope/3000/picoscope-3000-software, 2021.
[48] Daxia, SSCOM, http://www.daxia.com/sscom/sscom5.13.1.rar, 2021.
[49] eShard, Scared, https://gitlab.com/eshard/scared, 2021.

Fei Chen received the PhD degree in computer science and engineering from The Chinese University of Hong Kong, China in 2014. He is currently an associate professor at College of Computer Science and Software Engineering, Shenzhen University, China. His research interests include data protection and privacy, as well as distributed systems and applications.

Dumming Luo received the BEng degree in computer science from Wuyi University, China in 2018. She is a master student at the College of Computer Science and Software Engineering, Shenzhen University, China. Her research interests include data protection and privacy, and IoT security.

Victor C. M. Leung received the PhD degree in electrical engineering from the University of British Columbia, Canada in 1982. He is currently a distinguished professor of computer science and software engineering at Shenzhen University, Shenzhen, China. He is also an Emeritus professor of electrical and computer engineering and the director of the Laboratory for Wireless Networks and Mobile Systems, University of British Columbia (UBC), Vancouver, Canada. His current research is in the broad areas of wireless networks and mobile systems, and he has authored widely in these areas.

Junfeng Fan received the PhD degree in electrical engineering from Katholieke Universiteit Leuven, Belgium in 2012. He is now a staff at Open Security Research, Inc. China. His current research interests include data security and IoT security.

Jianqiang Li received the PhD degree in automatic control from South China University of Technology, China in 2008. He is currently a professor at the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He has presided over three projects of the National Natural Science Foundation of China and three projects of the Natural Science Foundation of Guangdong Province, China. His current research interests include data analysis, embedded systems, and the IoTs.

Shiqi Li received the MEng degree in electrical engineering from Katholieke Universiteit Leuven, Belgium in 2009. He is a staff at Open Security Research, Inc., China. His current research interests include data security and IoT security.