A Study on the Side-Channel Analysis Trends for Application to IoT Devices

Bo-Yeon Sim and Dong-Guk Han*
Kookmin University, 77 Jeongneung-ro, Seongbuk-gu, Seoul, 02707, Korea
{qjdusls, christa}@kookmin.ac.kr

Abstract

Over the past 20 years, side-channel analysis (SCA) on IC Chip has mainly taken place. However, recently, there has been an increasing number of issues of SCAs on the Internet of things (IoT) devices. As the IoT is directly applied to all things in our real life and post-security measures are impossible or involve high costs after the introduction, it is important to create a safe IoT environment. Thus, in this paper, we introduce the trends of SCAs on IoT devices. In particular, single-trace attacks that only use side-channel information are actively studied; it eliminates the need for information about the input and output values of cryptographic algorithms. This is a very powerful attack using only a single-trace and can be applied not only to public-key cryptography but also to post-quantum cryptography, which is actively being studied to counter the quantum computing era. Therefore, this paper suggests the urgency of developing countermeasures to this.

Keywords: Public-Key Cryptography, Post-Quantum Cryptography, Side-Channel Analysis, Single-Trace, Clustering

1 Introduction

Internet of things (IoT) is a network infrastructure designed to create new services utilizing various network technologies by connecting people, objects, and data to the Internet. By providing services in various environments, it can lead to user-centered convenient and pleasant lives, improve the productivity and efficiency of the industry, and create new value-added. In particular, market revitalization is underway in the home, home appliances, medical and transportation sectors, which are closely related to our lives. As the IoT is directly applied to all things in our real life, the danger of the existing cyber world is expanding and spreading to the real world. Unlike the existing cyber environment centered on personal computers (PCs) and mobile devices, however, the IoT environment requires access to a new paradigm of information protection is needed in terms of scope, target characteristics, the person in charge of security, and protection methods.

Especially, it is urgent to create a safe IoT environment where information protection is guaranteed because it is fatal enough to threaten a person's life and post-security measures are impossible or involve high costs after the introduction. Thus, in this paper, we intend to analyze cryptographic technology to establish indicators for providing safety and reliability improvements in the IoT service. Among them, it suggests the need for research on public-key cryptographic technology to protect information on the IoT. However, even cryptographic algorithms that have been proven to be secure against theoretical cryptanalysis cannot guarantee safety against side-channel analysis (SCA). In order to use secure cryptographic algorithms, research on whether there are no remaining vulnerabilities against SCA should be conducted continuously.

Journal of Internet Services and Information Security (JISIS), volume: 10, number: 1 (February, 2020), pp. 2-21
DOI: *10.22667/JISIS.2020.02.29.002*
*Corresponding author: Dong-Guk Han, Department of Mathematics, Kookmin University, Tel: +82-(0)2-910-5898

In this paper, we focus on side-channel analysis on public-key cryptography as well as post-quantum cryptography, that is actively being studied to resist for the quantum computing era. In particular, single-trace attacks that only use side-channel information are actively studied. The investigated single-trace attacks only use side-channel information. Thus, it eliminates the need for information about the input and output values of cryptographic algorithms. The attacks are based on the following assumption: two or more differentiated operations are performed depending on sensitive variables, and it is possible to cluster side-channel information that is generated by sensitive variables.

Organization. The rest of this paper is organized as follows. In Section 2, we briefly describe SCAs on public-key and post-quantum cryptography. In Section 3 and Section 4, we then explain the latest trends of SCAs on public-key and post-quantum cryptography, respectively. We finally give a conclusion in Section 6.

2 Side-Channel Analysis

In 1996, Kocher was first presented SCA which uses physical information that occurs when cryptographic algorithms are running on embedded systems [29]. It enable to recover secret credentials, i.e. cryptographic keys, by analyzing side-channel information such as execution time, power consumption, electromagnetic emission, and photonic emission, when cryptographic algorithms are running on devices. Such SCAs include timing attack (TA), simple power analysis (SPA), differential power analysis (DPA), template attack (TA^p), and collision attack (CA) [33]. For TA^p and CA, horizontal modes can be categorized as a sort of SPA, while vertical modes can be categorized as a sort of DPA. However, in this paper, we classify TA^p and CA as different attack types from SPA and DPA. Only visual inspections of power traces are considered as a type of SPA. Furthermore, only analyses that calculate hypothetical intermediate values for multiple-trace and apply statistical methods are considered as a type of DPA.

2.1 Side-Channel Analysis on Public-Key Cryptography

There is a digital signature system that can provide authentication, non-repudiation, and integrity to ensure the reliability of digital messages or documents. Rivest-Shamir-Adleman (RSA) [55], digital signature algorithm (DSA) [44] and elliptic curve DSA (ECDSA) [36, 28] are generally used. RSA and DSA have been widely used in various applications, such as online shopping, banking, and billing, as well as a virtual private network (VPN) and anti-cloning and firmware over the air (FOTA) applications. As security and performance requirements change today, more mobile devices and web services demand smaller and faster signatures. Compare to the RSA and DSA with the same security level, the ECDSA can be a feasible alternative with a smaller key size and a more efficient implementation. The ECDSA has been used in web-based TLS, various browsers and OSs, and crypto libraries. In particular, there are many widely used ECDSA implementations associated with the blockchain and Fast IDentity Online (FIDO) running on a variety of mobile devices [25, 22].

However, the core operation of ECDSA, scalar multiplication, is vulnerable to SCAs. Various SCAs on elliptic curve cryptography (ECC) were studied. This is not just about ECDSA. RSA and DSA are also vulnerable to SCAs. Since modular exponentiation has a similar operation structure to elliptic curve cryptography (ECC) scalar multiplication, the SCAs on ECC can be applied to RSA and DSA. In other words, if the point addition operation of ECC is converted to multiplication and point doubling operation is replaced with squaring, it is RSA (or DSA). In this section, we describe SCA by focusing on ECC scalar multiplication.

2.1.1 Simple Power Analyses on ECC Scalar Multiplication

SPA is a direct method of analyzing a secret scalar using only one trace or a few traces collected during cryptographic operations [30]. Cryptographic algorithms have different power consumption patterns depending on the commands of the processor, thus, the secret scalar or the instantaneous commands can be analyzed in these patterns.

Algorithm 1 Scalar Multiplication : Binary Method (Left to Right)

Input : $P = (x, y)$ a point on $E(\mathbb{F}_q)$, a λ-bit scalar $d = (d_{\lambda-1}, d_{\lambda-2}, \cdots, d_0)_2$
Output : $Q = d \cdot P$

1: $R \leftarrow \infty$
2: **for** $i = \lambda - 1$ down to 0 **do**
3: $\quad R \leftarrow 2R$
4: $\quad$ **if** $d_i = 1$ **then**
5: $\quad\quad R \leftarrow R + P$
6: $\quad$ **end if**
7: **end for**
8: **Return** R

Algorithm 2 Scalar Multiplication : Binary Method (Right to Left)

Input : $P = (x, y)$ a point on $E(\mathbb{F}_q)$, a λ-bit scalar $d = (d_{\lambda-1}, d_{\lambda-2}, \cdots, d_0)_2$
Output : $Q = d \cdot P$

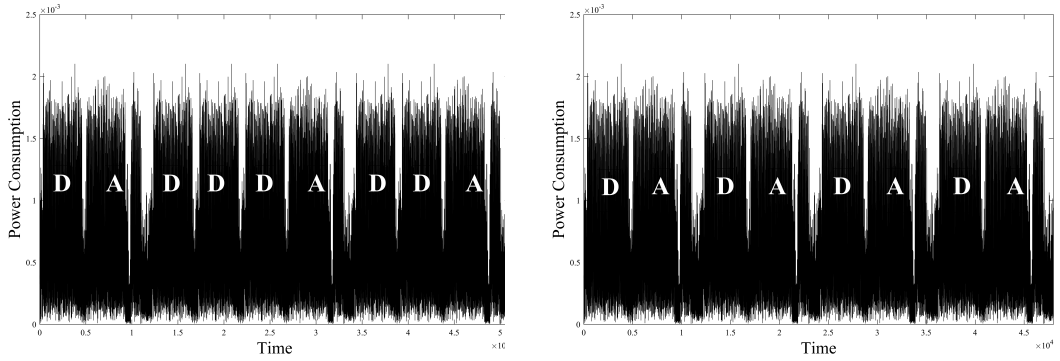
1: $R_0 \leftarrow \infty$, $R_1 \leftarrow P$
2: **for** $i = 0$ up to $\lambda - 1$ **do**
3: $\quad$ **if** $d_i = 1$ **then**
4: $\quad\quad R_0 \leftarrow R_0 + R_1$
5: $\quad$ **end if**
6: $\quad R_1 \leftarrow 2R_1$
7: **end for**
8: **Return** R_0

For example, Algorithm 1 and 2 always perform point doubling operation and only perform point addition operation when the secret key bit value is 1; thus, the secret key can be found when the power consumption patterns of point doubling and point addition operations differ. That is, this irregular sequence of instructions resulting from the secret scalar bit (i.e., conditional branches that rely on data) as shown in Figure 1(a) results in a serious security problem.

Property 1. *If the point doubling operation is different from the point addition operation and an algorithm behaves irregularly according to the secret scalar bit d_i, then the algorithm is vulnerable to simple power analyses.*

2.1.2 Differential Power Analyses on ECC Scalar Multiplication

DPA is a statistical analysis method that analyzes multiple power consumption traces to find a secret scalar[30]. DPA generally based on the fact that power consumption relies on manipulated data values. To perform DPA, you need to know the input or output values of cryptographic algorithms. Similarly, address bit DPA exists on the basis of the fact that power consumption relies on the address value of the register which handles data during operation.



(a) Binary (b) Regular binary

Figure 1: Power consumption trace of scalar multiplication

Algorithm 3 Scalar Multiplication : Doubling and Addition Always (Left to Right) (refer to [14])

Input : $P = (x, y)$ a point on $E(\mathbb{F}_q)$, a λ-bit scalar $d = (d_{\lambda-1}, d_{\lambda-2}, \cdots, d_0)_2$
Output : $Q = d \cdot P$

1: $R_0 \leftarrow \infty$, $R_1 \leftarrow \infty$
2: **for** $i = \lambda - 1$ down to 0 **do**
3: $\quad R_0 \leftarrow 2R_0$
4: $\quad R_{1-d_i} \leftarrow R_0 + P$
5: **end for**
6: **Return** R_0

Algorithm 4 Scalar Multiplication : Montgomery Ladder (Left to Right) (refer to [38, 24])

Input : P is a point on an elliptic curve, a λ-bit scalar $d = (d_{\lambda-1}, \cdots, d_0)_2$
Output : $Q = d \cdot P$

1: $R_0 \leftarrow \infty$, $R_1 \leftarrow P$
2: **for** $i = \lambda - 1$ down to 0 **do**
3: $\quad R_{1-d_i} \leftarrow R_{d_i} + R_{1-d_i}$
4: $\quad R_{d_i} \leftarrow 2R_{d_i}$
5: **end for**
6: **Return** R_0

Algorithm 5 Scalar Multiplication : Joye's Add Only (Right to Left) (refer to [23])

Input : P is a point on an elliptic curve, a λ-bit scalar $d = (d_{\lambda-1}, \cdots, d_0)_2$
Output : $Q = d \cdot P$

1: $R_0 \leftarrow \infty$, $R_1 \leftarrow P$, $R_2 \leftarrow P$
2: **for** $i = 0$ up to $\lambda - 1$ **do**
3: $\quad R_{1-d_i} \leftarrow R_{1-d_i} + R_2$
4: $\quad R_2 \leftarrow R_0 + R_1$
5: **end for**
6: **Return** R_0

Thus, SPA countermeasure Algorithm 3 with a regular power consumption sequence, as shown in Figure 1(b), is vulnerable to DPA. Other SPA countermeasures Algorithm 4 and 5 are also vulnerable to

DPA. To cope with this, randomization techniques are commonly used, which eliminate any guessable relation between intermediate values and power consumption [13, 14, 34].

Property 2. *If an algorithm uses a fixed secret d, and if it is possible to calculate hypothetical intermediate states $v_{i,j} = f(p_i, d_j)$ for all $\mathscr{P}$ known non-constant values p_i and for all $\mathscr{D}$ candidates d_j of d, then the algorithm is vulnerable to differential power analyses (or correlation power analyses). At this time, $\mathscr{D}$ should be small enough so that all hypotheses $v_{i,j}$ can be exhausted.*

2.1.3 Sophisticated Power Analyses on ECC Scalar Multiplication

SPA-and DPA-resistant countermeasures can be defeated by sophisticated attacks, such as TA$^{\mathrm{p}}$ [20, 39] or CA [19, 49, 65]. A TA$^{\mathrm{p}}$ characterizes power consumption traces by a multivariate normal distribution to build templates and matches power consumption leakage to the templates to find a secret scalar value.

A CA is an attack based on the interrelationships among intermediate data (i.e. collisions of two intermediate values). For example, CA on Algorithm 3, when the $d_i = 0$, collision of input data $(d_{\lambda-1}, d_{\lambda-2}, \cdots, d_{i+1}, 0)_2 \cdot P$ occurs between the point addition operation of i iteration and the point doubling operation of $(i-1)$ iteration as follows:

- point doubling of i iteration

$$= 2((d_{\lambda-1}, d_{\lambda-2}, \cdots, d_{i+1})_2 \cdot P)$$

- point addition of i iteration

$$= (d_{\lambda-1}, d_{\lambda-2}, \cdots, d_{i+1}, 0)_2 \cdot P + P$$

- point doubling of $(i-1)$ iteration

$$= \begin{cases} 2((d_{\lambda-1}, d_{\lambda-2}, \cdots, d_{i+1}, 0)_2 \cdot P), & \text{if} \quad d_i = 0 \\ 2((d_{\lambda-1}, d_{\lambda-2}, \cdots, d_{i+1}, 1)_2 \cdot P), & \text{if} \quad d_i = 1 \end{cases}$$

but when $d_i = 1$, input data collision does not occur. Thus, depending on the occurrence of the collision, the secret scalar bits can be extracted. We can classify CA into five types, which are experimentally proven, as follows.

Property 3. *If information collision is determined according to the secret scalar bit d_i, then the algorithm is vulnerable to collision attack.*

(i) *When the collision of input data of two same operations is determined according to the secret scalar bit d_i* [17, 70, 21, 16, 74].

(ii) *When the collision of input data of two different operations is determined according to the secret scalar bit d_i* [17].

(iii) *When the collision of two operations, using the same input data, is determined according to the secret scalar bit d_i* [21].

(iv) *When the use of register for data saving (or loading) is determined according to the secret scalar bit d_i* [20, 19, 65, 49, 48].

(v) *When the collision between saved and loaded data, at the data saving and loading step, is determined according to the secret scalar bit d_i* [17].

In addition, for algorithms with operations that reference pre-computation tables, a vulnerability exists in CA when it repeats references to the same location of the table. This vulnerability exists because register addresses or input data values can conflict when loading data from the pre-computation table [45, 26].

Property 4. *If the algorithm repeats the reference to the same position of the pre-computation table, then the algorithm is vulnerable to collision attack.*

So far, no theoretically perfect countermeasures have been proposed for TA$^{\text{p}}$s and CAs. However, the drawback is that obtaining power consumption traces with a high signal-to-noise ratio requires precise pre-processing, such as decapsulation, localization, multi-probe [20, 19, 49, 65]. Decapsulation, in particular, requires physical modification of the target devices, and many traces are required to build templates.

2.2 Side-Channel Analysis on Post-Quantum Cryptography

The security of public key cryptosystems (PKCs) primarily is based on the difficulty of number theory problems, such as factoring large integers or finding discrete logarithms. Shor, however, proposed an algorithm that can solve such problems in polynomial time, given a practical large-scale quantum computer [58]. Since quantum computers become critical threats to the current PKCs, such as Rivest-Shamir-Adleman (RSA) and elliptic curve cryptography (ECC) [55, 36, 28], there are an increasing needs for post-quantum cryptography (PQC) that is secure against both quantum and classical computers.

The National Security Agency (NSA) thus announced that the list of Suite B cryptographic algorithms would be updated to PQC algorithms [1]. The National Institute of Standards and Technology (NIST) also released an internal report (NISTIR) 8105: Reports on PQC [11], giving an analysis of the current state of quantum computing and then discussing the need of PQC standardization. In December 2016, the NIST announced a call for proposals for PQC standardization [42]. In contrast to the Advanced Encryption Standard (AES) and Secure Hash Algorithm version 3 (SHA-3) competitions, which selected a single algorithm, the NIST aims to recommend several PQC algorithms [40, 41]. In the first-round submissions, sixty-nine proposals on public key encryption, key establishment, and digital signature algorithms were accepted. In the following second-round, twenty-six candidates have been survived [43].

Already since 10 years ago, researches of SCA on multivariate quadratic equations, code-based, and lattice-based cryptography has been conducted. Since the PQC challenge presented secure implementation against SCA as a development consideration, a study for countermeasures is required. In this section, we describe SCA by focusing on lattice-based and code-based cryptography which are promising candidates of standardization.

2.2.1 Side-Channel Analyses on Code-Based Cryptography

Simple Power Analyses on Code-Based Cryptography. Maurich *et al.* [72] presented SPAs on QC-MDPC McEliece cryptosystem, i.e. a message recovery attack and a private key recovery attack. For a private key recovery, they exploited the fact that different patterns of power consumption are observed depending on whether the conditional branch instruction is executed or not, when generating the next row of the private key in the syndrome computation. They presented experiment results based on two types of software implementations, i.e. AVR and ARM. They also proposed a constant-time implementation as a countermeasure using the ARM Thumb-2 assembly language. More specifically, they adopted the *mask* value, which is either zero or all bits are 1, and the logical AND instruction to choose which data to use. We classify the property, which includes Property 1, used in the attacks as follows.

Property 5. *If an algorithm behaves irregularly according to the secret value, then the algorithm is vulnerable to simple power analyses.*

Timing Attacks on Code-Based Cryptography. Strenzke *et al.* [69] first proposed a SCA against the McEliece cryptosystem. They presented a TA on the degree of error locator polynomial in the Patterson algorithm [47] exploiting the fact that the difference in computation time depends on the polynomial degree. Other types of TAs against the McEliece have been followed as in [59, 66, 67, 68, 5]. Strenzke *et al.* [69] also described power analysis against the parity-check matrix of McEliece key generation. Chaulet *et al.* [7] discussed that variable time decoders, such as bit flipping (BF) algorithm, may leak partial information. Since the number of iterations of the algorithm depends on the error pattern as well as the parity-check matrix, the algorithm may leak information about a private key and consequently allow a successful TA. They thus proposed minimizing the number of iterations by adapting threshold values as a function of the syndrome weight to make a constant-time decoder. This attack is based on Property 5, since, if an algorithm is vulnerable to SPA, then the algorithm is generally also vulnerable to TA.

Differential Power Analyses on Code-Based Cryptography. Chen *et al.* [8] presented a horizontal DPA on the QC-MDPC McEliece cryptosystem, which is a private key recovery attack on the asymmetric decryption algorithm using the chosen ciphertexts. They successfully recovered substantial parts of the private key by a DPA during syndrome computation and key rotation. They make use of the public key to recover the whole private key or to correct remaining errors using an algebraic step. Their attack target was the field programmable gate array (FPGA) implementation presented at DATE 2014 [71]. Since hardware implementations operate in parallel, they applied the chosen ciphertext DPA. They also suggested a threshold implementation based on boolean masking as a countermeasure [9]. The further analysis and countermeasure are also proposed [10]. The property used in the attacks is the same as Property 2. Chou suggested a constant-time implementation for QC-MDPC code-based cryptography to mitigate TAs [12]. This countermeasure was later found to become vulnerable to a DPA in private syndrome computation, as described by Rossi *et al.* [56]. The proposed attack, however, still could not completely recover accurate secret indices, requiring further solving linear equations to obtain entire secret information.

Additionally, various SPAs and DPAs against the McEliece cryptosystem can be found in [18, 37, 72, 8, 50, 51, 15], and fault injection attacks in [6, 67].

2.2.2 Side-Channel Analyses on Lattice-Based Cryptography

Simple Power Analyses on Lattice-Based Cryptography. Lee et al. proposed SPA against NTRU and they used the fact that the power consumption pattern when adding zeros is different from when non-zero values are added [31]. Roy et al. presented an attack that acquired the number of bits scanned from a ROM-word and a comparator when performing the Knuth-Yao sampling through a simple power analysis. If the number of bits scanned is known, it is possible to know the value of the terminal node, which can reduce or determine the candidate for the coefficient of error polynomial [57]. Park et al. suggested a chosen-ciphertext SPA against Ring-LWE scheme and they used the fact that the power consumption pattern when performing a modular operation is different from when a modular operation is not performed [46]. The property used in the attacks is the same as Property 5.

Timing Attacks on Lattice-Based Cryptography. Silverman et al. proposed TA against NTRU. Their attack is based on the number of hash calls used in the decoding process was proposed [60].

Differential Power Analyses on Lattice-Based Cryptography. Lee et al. also proposed CPA against NTRU and a countermeasure against their attack [31]. Wang et al. suggested DPA using chosen-ciphertexts to find the private key. Their attacks select two different ciphertexts, then collect power consumption traces for each ciphertext and obtain an average trace. It then aims to acquire confidential information using the difference between the two average traces [76]. In 2015 and 2016, Reparaz et al. proposed countermeasures to CPA in CHES and PQCrypto [54, 53]. The paper proposed in 2015 showed that the secret key of the ring-LWE scheme running on the SASEBO-G board was extracted with less than 1000 power consumption traces, and in 2016 the secret key of the ring-LWE scheme operating on ARM Cortex-M4 processor was extracted with less than 2,000 electromagnetic traces.

Additionally, various SCAs, such as CA, TA$^{\mathrm{p}}$, against lattice-based cryptography can be found in [77, 52, 4, 3].

3 Single-Trace Attacks on Public-Key Cryptography

Binary scalar multiplication, the main operation of elliptic curve cryptography, is vulnerable to side-channel analysis. In particular, it is vulnerable to side-channel analysis using power consumption and electromagnetic emission patterns. Therefore, various countermeasures were studied. However, the focus was on eliminating patterns in conditional branches, statistical characteristics based on intermediate values, or interrelationships between data. Although during the check phase, secret scalar bits are directly loaded, countermeasures for this phase were not considered. Since the secret scalar bit values are extracted and stored in the variables, thus, if the vulnerability is found to exist, the secret scalar may be exposed. In [61, 63, 2], the authors verify that this vulnerability is sufficient to find a secret key.

3.1 Attack on ECC Scalar Multiplication and RSA Modular Exponentiation

Sim et al. analyzed the power consumption (they also considered information leakage via electromagnetic emanation) characteristics of the key bit identification phase and experimentally demonstrated that attacks based on these characteristics could restore secret scalar bits [61, 63]. The proposed attacks require only a single power consumption or electromagnetic trace. They also do not need to know in–out values. This allows us to defeat any combination of existing countermeasures.

Two implementations (i.e. hardware and software) were targeted, and the authors were able to restore the secret scalar bits by applying VI-SPA (visual inspection-based SPA) and clustering algorithms. Among various scalar multiplication algorithms, they focused on binary scalar multiplication algorithms. Power consumption traces, power consumption traces passed through a low-pass filter, electromagnetic traces, and electromagnetic traces passed through a low-pass filter were used to show four experimental results. Although the ECC binary scalar multiplication algorithm was focused, the proposed attack can also be applied to the RSA binary modular exponentiation algorithms.

Binary scalar multiplication consists of iterative operations determined depending on the i-th bit d_i value of the secret scalar d (Algorithm 4 and 5). Therefore, a key bit identification phase exists, in which the i-th scalar bit value is extracted from a λ-bit scalar string $d = (d_{\lambda-1}, d_{\lambda-2}, \cdots, d_1, d_0)_2$ and stored in a d_i variable, at the beginning of each i-th iteration. This results in power consumption associated with the d_i value. Moreover, in regular algorithms, the referred register addresses $RegAddr_{d_i}$ differ according to the d_i value, and these affects power consumption. Therefore, power consumption properties can be classified according to sensitive variables.

Property 6. In hardware implementations, power consumption in the sensitive variable identification phase is simultaneously affected by:

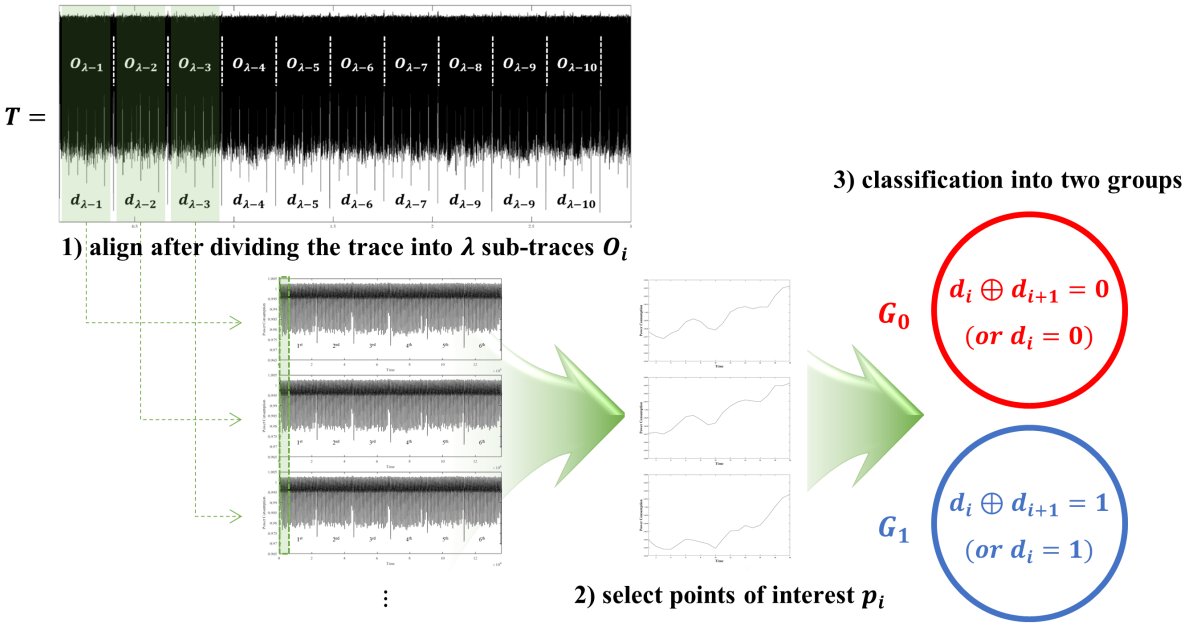


Figure 2: The attack methodology [61, 63]

1. the Hamming distance between two consecutive bits d_{i+1} and d_i, i.e. $d_{i+1} \oplus d_i$;
2. the Hamming distance between referred register addresses $RegAddr_{d_{i+1}}$ and $RegAddr_{d_i}$ determined by d_{i+1} and d_i, i.e. $RegAddr_{d_{i+1}} \oplus RegAddr_{d_i}$.

Thus, if two consecutive bits are the same, i.e. $d_{i+1} = d_i$; power consumption related to $d_{i+1} \oplus d_i = 0$ and $RegAddr_{d_{i+1}} \oplus RegAddr_{d_i} = 0$ occurs at the same time. Otherwise, power consumption related to $d_{i+1} \oplus d_i = 1$ and $RegAddr_{d_{i+1}} \oplus RegAddr_{d_i} \neq 0$ occurs at the same time $(0 \leq i < \lambda - 1)$.

Property 7. In software implementations, power consumption is affected by:

1. the Hamming weight of i-th secret bit value d_i;
2. the Hamming weight of referred register address $RegAddr_{d_i}$ determined by value of i-th secret bit d_i.

Thus, if the i-th secret bit value is 0, i.e. $d_i = 0$, then power consumption related to 0 and $RegAddr_0$ occurs. Otherwise, power consumption related to 1 and $RegAddr_1$ occurs $(0 \leq i \leq \lambda - 1)$.

The proposed attack framework is shown in Figure 2. The experiments focused on the Montgomery–López–Dahab ladder algorithm protected by scalar randomization in hardware implementations. It also shows that a single trace can be used to extract secret key bits at a 100% success rate, as shown in Figure 3. In addition, attacks did not require sophisticated pre-processing and could defeat existing countermeasures using a single trace. The software implementation focused on the key bit identification functions of mbedTLS and OpenSSL. Since the success rate was over 94%, brute-force attacks could still restore the entire secret scalar bits.

3.2 Attack on RSA Modular Exponentiation

Alam et al. announced the results of a single electromagnetic attack on constant-time blind RSA in USENIX with OpenSSL version 1.1.0g [2]. The experiment was conducted by collecting the electromag-

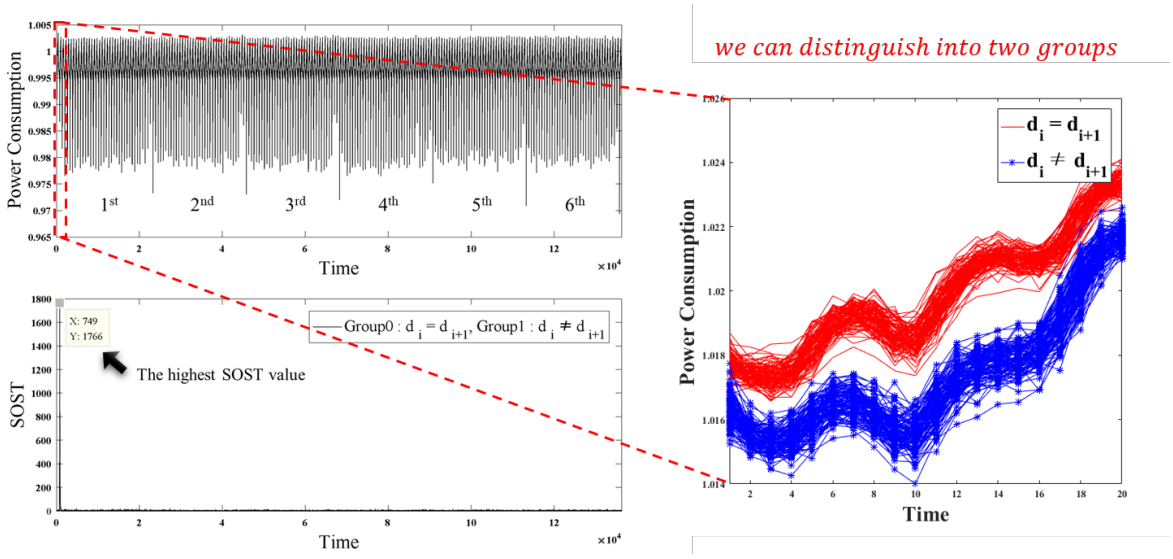
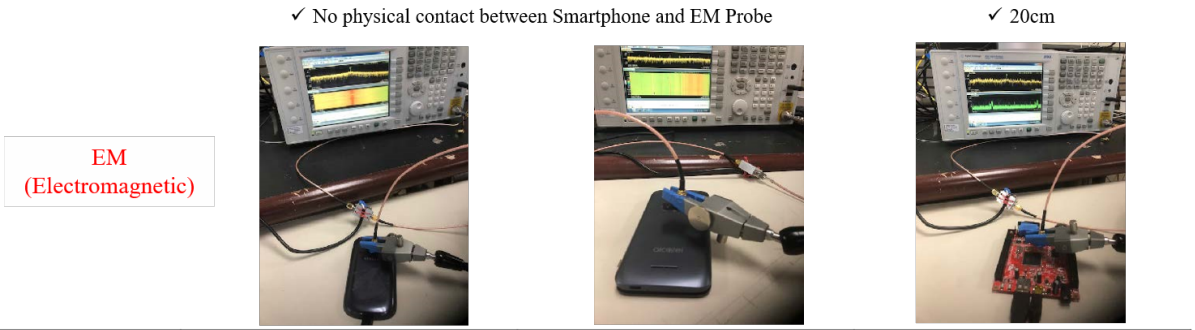


Figure 3: Classification result (hardware implementation, power consumption)[61, 63]

netic traces that occur during operation on a total of three types of smart phones, as shown in Figure 4. Smart phones are equipped with ARM processors, and the experiment showed that they can restore secret key bits at a success rate of 95.7 to 99.6 %.



	Target platform 1	Target platform 2	Target platform 3
Target specification	Samsung Galaxy Centura SCH-S738C	AT&T Alcatel Ideal	A13-OLinuXino
	Android-based mobile phone		Linux-based embedded system board
	ARM processor (Operating at high 800 MHz to 1.1 GHz)		
Measurement equip.	Ettus USRP B200-mini receiver (can receive up to 56 MHz of bandwidth around a center freq. between 70 MHz and 6GHz)		
	Keysight N9020A MXA spectrum analyzer		
Measurement setup	40 MHz band around the clock frequency (sample rate that is $<$ 5% of the processor's clock freq.)		
Preprocessing	AM demodulation, up sample it by factor of 4		
Attack type	Single-trace attack		

Figure 4: Experiment environments [2]

Constant-time blind RSA means fixed window method algorithm and sliding window method algorithm with message randomization applied. The analysis location is where the function `BN_is_bit_set` is performed to identify the i-th secret key bit in the n-bit key string d; it is the same position with the attack

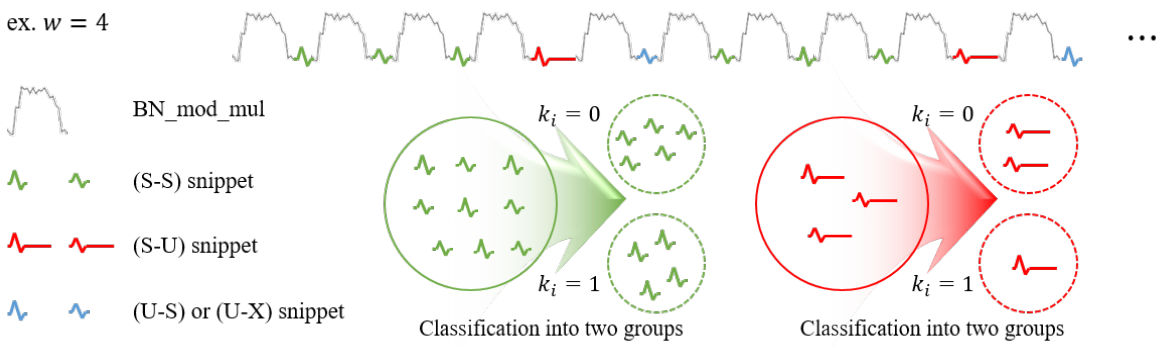


Figure 5: The attack methodology [2]

proposed in [61, 63]. The function `BN_is_bit_set` returns a value of 0 or 1 because it returns the i-th bit value of the secret key d. Thus, the location where the `BN_is_bit_set` function is to be returned is points of interest (PoI), and electromagnetic traces of PoI can be classified into two sets according to the d_i value.

Alam et al. specified the points between the `BN_mod_mul` operations as snippets, and a total of three types of snippets have been separated to PoI. The snippets have S-S snippets between square and square operations, S-U snippets between square and multiplication operations, and U-S snippets between multiplication and square operations. They clustered snippets into two groups and extracted secret key bits Figure 5.

4 Single-Trace Attacks on Post-Quantum Cryptography

To eliminate vulnerabilities against TA, various countermeasures perform in constant-time were proposed. However, they only focused on making the computational time is constant. The vulnerabilities to SPA were reported [27, 64, 62] because the difference in the amount of side-channel information that occurred by the type of operation used was not taken into account.

4.1 Attack on Constant-Time CDT Sampler

The Gaussian sampler is used for lattice-based cryptography that is based on the hardness of solving learning with errors (LWE) problem. To increase the performance of the Gaussian sampler, a cumulative distribution table (CDT) sampler was proposed and most widely used. However, the CDT sampler is vulnerable to TA, thus, a constant-time CDT sampler was proposed.

Algorithm 6 Constant-time CDT sampling

Input : CDT table ψ of length l, σ, τ
Output : Sampled value S

1: $S \leftarrow 0$
2: $rnd \leftarrow [0, \tau\sigma) \cap \mathbb{Z}$ uniformly at random
3: $sign \leftarrow [0,1] \cap \mathbb{Z}$ uniformly at random
4: **for** $i = 0$ up to $l-1$ **do**
5: $\quad S += (\psi - rnd) \gg 15$
6: **end for**
7: $S \leftarrow ((-sign) \wedge S) + sign$
8: **Return** S

Kim et al. proposed a single trace analysis on constant-time CDT sampler [27]. They targeted the steps 4 to 7 of the Algorithm 6. The attack utilized the fact that negative values will have the Hamming weight of 11.5 on average and positive values will have the Hamming weight of 4.5 on average on 16-bit processor. Additionally, the attack is also carried out based on that the sign of the *subtracted value* $(\psi - rnd)$ determines whether to add 0 or 1 to the S.

$$subtracted\ value = \begin{cases} \texttt{positive} & , \text{ if } \psi - rnd \geq 0; \\ \texttt{negative} & , \text{ if } \psi - rnd < 0. \end{cases} \quad S \ += \begin{cases} 0 & , \text{ if } \psi - rnd \geq 0; \\ 1 & , \text{ if } \psi - rnd < 0. \end{cases}$$

Property 8. On 16-bit processor, if *subtracted value* is `positive`, the power consumption is related to 4.5 on average, which is the Hamming weight of the *subtracted value*. Additionally, when $(\psi - rnd) \gg 15$ is added to S, the power consumption associated with 0 occurs. In contrast, when *subtracted value* is `negative`, the power consumption is related to 11.5 on average. Besides, when $(\psi - rnd) \gg 15$ is added to S, the power consumption associated with 1 occurs.

It is possible to cluster the power consumption traces into two groups depends on the sign of the *subtracted value*. Thus, error S can be extracted using a single-trace and secret key can be found in polynomial time using the Gaussian elimination.

4.2 Attack on Quasi-Cyclic Code-Based Cryptography

Chou suggested a constant-time implementation for QC-MDPC code-based cryptography to mitigate TAs [12]. This countermeasure was later found to become vulnerable to a DPA in private syndrome computation, as described by Rossi *et al.* [56]. The proposed attack, however, still could not completely recover accurate secret indices, requiring further solving linear equations to obtain entire secret information.

Sim et al. proposed a multiple-trace attack, that recovers entire secret indices eliminating the need for additionally solving linear equations, on the constant-time multiplication for syndrome computation [64]. Moreover, they proposed a single-trace attack that allows recovering secret indices even when using ephemeral keys, or DPA countermeasures proposed in [56, 9]. In particular, if a processor only provides single bit shift instructions, it is possible to find the whole bits of secret indices. Furthermore, even if processors do not provide single bit shift instructions, the attacker can extract substantial parts of secret indices.

The proposed attack exploits the fact that rotation is always carried out, and also that the *mask* value as determined by the value of the secret bit is used to obtain accurate results. Hence, their attack can make the latest countermeasures proposed for secure private syndrome computation obsolete. On 8-bit processor, the *mask* value is as shown below:

$$mask = \begin{cases} \texttt{0x00} & , \text{ if } d_i = 0; \\ \texttt{0xff} & , \text{ if } d_i = 1. \end{cases}$$

Since the *mask* value is divided into two groups according to secret bit d_i, it is possible to recover secret bit by clustering power consumption traces into two groups such as Figure 6. The power consumption property is defined as follows.

Property 9. Assume that *mask* is `0x00` or `0xff` on a 8-bit processor; therefore, if *mask* is `0x00`, the power consumption is related to 0. In contrast, when *mask* is `0xff`, the power consumption is related to 8, which is the Hamming weight of the *mask* value.

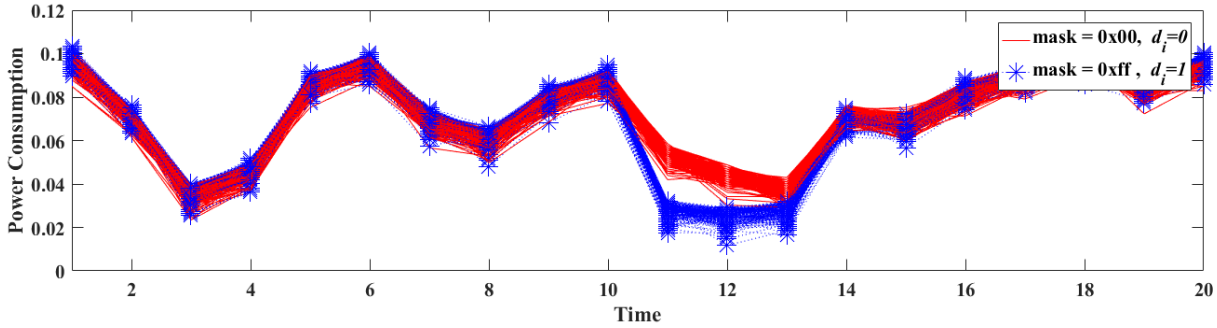


Figure 6: Classification result (8-bit processor, power consumption) [64]

The BIKE and LEDAcrypt are constructed using QC-MDPC and QC-LDPC codes, respectively, and they are the second-round candidates of the NIST PQC standardization. Since syndrome computations of these two schemes were not designed to resist SCAs, Sim et al. assume that the countermeasures [12, 56, 9] are applied to remove each of TA and DPA vulnerability. Their experiment results show that these two schemes may become vulnerable to the proposed multiple/single-trace attacks when they use long-term key pairs. These schemes may become vulnerable to their single-trace attack even when using ephemeral keys.

4.3 Attack on Countermeasures against Instruction-related Timing Attacks

LAC [32] and Hamming quasi-cyclic (HQC) [35], which are lattice-based and code-based cryptography, respectively, are the second-round candidates of NIST PQC standardization. They use error-correcting code, Bose–Chaudhuri–Hocquenghem (BCH), due to non-zero decryption failure rate. However, there are timing side-channel leakage of BCH, then it is possible to reduce the security of LAC and HQC. Thus, to counter timing attack, constant-time variants of BCH were proposed [75, 73].

Sim et al. show that countermeasures against instruction-related timing attack would be vulnerable to single-trace attacks [62], which are presented in [61, 63, 64]. The countermeasures use *determiner* to make operations, which leak timing side-channel information, perform in a constant-time. Since *determiner* is divided into two groups according to secret credentials, it is possible to recover secret credentials by clustering *determiner* into two groups. Three types of vulnerable operations can be categorized as below.

Loops whose bound is input-dependent For instance, if a loop iterates depending on input length for efficiency reasons or there are early-termination statements, it would be vulnerable to timing attack. Thus, to mitigate these vulnerabilities, the iteration length must be fixed and early-termination statements should not be used. Accordingly, repeating the iteration of the loop as maximum length and using a *bound determiner*, such as Listing 1, were proposed. Since *mask* is -1, which all the bits are 1, the *mask* value is `0xffffffff` when the bit length of data types is 32. The $determiner1$ is as shown below:

$$determiner1 = \begin{cases} \texttt{0x00000001} & , if\ i < secret\_length; \\ \texttt{0x00000000} & , if\ i \geq secret\_length. \end{cases}$$

Therefore, the results of loops determines depending on $determiner1$, and dummy operations are performed when $i \geq secret\_length$.

```
1 for (i = 0; i < max_length; i++)
```

```
{
// mask is generated based on data types
determiner1 = ((i - secret_length) & mask) >> 31;
}
```

Listing 1: Examples of constant-execution loops

Branches whose condition is input-dependent Branches execute differently depending on condition, thus, operating pattern is irregular. This irregularity induces the possibility of timing attack. To make it perform in constant-time, regular operations always carried out independent of secret credentials i and j, as shown in Listing 2. When the bit length of data types is 32, the *determiner*2 and *determiner*3 are as shown below:

$$determiner2 = \begin{cases} \texttt{0x00000001} & , if\ i = 0; \\ \texttt{0x00000000} & , if\ i = 1. \end{cases} \quad determiner3 = \begin{cases} \texttt{0x00000000} & , if\ i = 0; \\ \texttt{0xffffffff} & , if\ i = 1. \end{cases}$$

Therefore, the results determines depending on *determiner*2 and *determiner*3.

```
// mask is generated based on data types
determiner2 = !(((i - 1) & mask) >> 31);
a = b * determiner2;
determiner3 = -((uint32_t)- j >> 31);
v = x & determiner3 + v & !determiner3;
```

Listing 2: Examples of constant-execution branches

Input-dependent memory access Even though arrays are stored in contiguous memory blocks, accessing arrays depends on specific secret data would be vulnerable to timing attack. To make uniform array accessing, *blinded array access*, which accesses all elements such as Listing 3, was proposed. If the bit length of data types is 32, the *determiner*4 is as below:

$$determiner4 = \begin{cases} \texttt{0x00000000} & , if\ i \neq index; \\ \texttt{0xffffffff} & , if\ i = index. \end{cases}$$

Therefore, the results determines depending on *determiner*4.

```
for (i = 0; i < length; i++)
{
// mask is generated based on data types
xorVal = i ^ index;
// anyOnes = 0 if i = index, 1 otherwise
anyOnes = set_bit(xorVal)
determiner4 = (anyOnes & 1) - 1;
out = b[i] & determiner4;
}
```

Listing 3: Examples of uniform array access

5 Summary

In this section, we summary the attacks described in Section 3 and Section 4. The attacks do not need to know in-out values and just use single-trace to discover the secret values. Although the clustering algorithms are used except for the attack suggested by Kim et al., the clustering algorithms can be applied to Kim's attack.

Table 1: Single-Trace Attacks on PKC and PQC

Author	Target		# trace	In-out data	Distinguisher	# set
Sim et al. [61, 63]	PKC	Regular algorithm	1	No	Clustering	2
Alam et al. [2]	PKC	Window method	1	No	Clustering	2
Kim et al. [27]	PQC	Constant-time CDT sampler	1	No	SPA	2
Sim et al. [64]	PQC	Constant-time multiplication	1	No	Clustering	2
Sim et al. [62]	PQC	Constant-execution loops Constant-execution branches Constant-time array access	1	No	Clustering	2

6 Conclusion

To construct a secure IoT service, cryptosystems are needed. However, theoretically secure cryptosystems cannot guarantee safety against SCAs. Various attacks on the cryptographic system are being studied, threatening our real lives. In particular, for public-key and post-quantum cryptography, single-trace attacks that only use side-channel information are actively studied. To eliminate vulnerabilities against TA, various countermeasures performed in constant-time were proposed. However, they only focused on making the computational time constant. Making the amount of side-channel information and its pattern in constant were not taken into account, thus there are SPA vulnerabilities. As a temporary measure, the hiding methods, such as random noise and dummy operation, can be applied to increase attack complexity to counter single-trace attack. Therefore, it would be one of the interesting future research topics to construct theoretically-sound countermeasures against the single-trace attack described in this paper.

Acknowledgments

This work was supported as part of Military Crypto Research Center (UD170109ED) funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).

References

[1] N. S. Agency. Cryptography today. `https://www.nsa.gov/ia/programs/suiteb_cryptography/` [Online; accessed on February 2, 2020], 2015.

[2] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. L. Callan, A. G. Zajic, and M. Prvulovic. One&done: A single-decryption em-based attack on openssl's constant-time blinded RSA. In *Proc. of the 27th USENIX Security Symposium (SEC'18), Baltimore, MD, USA*, pages 585–602. USENIX Association, August 2018.

[3] S. An, S. Kim, S. Jin, H. Kim, and H. Kim. Single trace side channel analysis on NTRU implementation. *Applied Sciences*, 8(11):2014, October 2018.

[4] A. Aysu, Y. Tobah, M. Tiwari, A. Gerstlauer, and M. Orshansky. Horizontal side-channel vulnerabilities of post-quantum key exchange protocols. In *Proc. of the 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST'18), Washington, DC, USA*, pages 81–88. IEEE, April 2018.

[5] D. Bucerzan, P.-L. Cayrel, V. Dragoi, and T. Richmond. Improved timing attacks against the secret permutation in the mceliece pkc. *International Journal of Computers Communications & Control*, 12(1):7–25, February 2017.

[6] P.-L. Cayrel and P. Dusart. Mceliece/niederreiter pkc: Sensitivity to fault injection. In *Proc. of the 2010 5th International Conference on Future Information Technology (FutureTech'10), Busan, South Korea*, pages 1–6. IEEE, May 2010.

[7] J. Chaulet and N. Sendrier. Worst case QC-MDPC decoder for mceliece cryptosystem. arxiv-1608.06080, August 2016. `https://archive.org/details/arxiv-1608.06080` [Online; Accessed on February 2, 2019].

[8] C. Chen, T. Eisenbarth, I. von Maurich, and R. Steinwandt. Differential power analysis of a mceliece cryptosystem. In *Proc. of the 13th International Conference on Applied Cryptography and Network Security (ACNS'15), New York, New York, USA*, volume 9092 of *Lecture Notes in Computer Science*, pages 538–556. Springer, Cham, June 2015.

[9] C. Chen, T. Eisenbarth, I. von Maurich, and R. Steinwandt. Masking large keys in hardware: A masked implementation of mceliece. In *Proc. of the 22nd International Conference on Selected Areas in Cryptography (SAC'15), Sackville, New Brunswick, Canada*, volume 9566 of *Lecture Notes in Computer Science*, pages 293–309. Springer, Cham, August 2015.

[10] C. Chen, T. Eisenbarth, I. von Maurich, and R. Steinwandt. Horizontal and vertical side channel analysis of a mceliece cryptosystem. *IEEE Transactions on Information Forensics and Security*, 11(6):1093–1105, December 2016.

[11] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. Report on post-quantum cryptography. Technical report, US Department of Commerce, National Institute of Standards and Technology, 2016.

[12] T. Chou. Qcbits: Constant-time small-key code-based cryptography. In *Proc. of the 18th International Conference on Cryptographic Hardware and Embedded Systems (CHES'16), Santa Barbara, California, USA*, pages 280–300. IACR, August 2016.

[13] M. Ciet and M. Joye. (Virtually) free randomization techniques for elliptic curve cryptography. In *Proc. of the 5th International Conference on Information and Communications Security (ICICS'03), Huhehaote, China*, volume 2836 of *Lecture Notes in Computer Science*, pages 348–359. Springer, Berlin, Heidelberg, October 2003.

[14] J. Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *Proc. of the First International Workshop on Cryptographic Hardware and Embedded Systems (CHES'99), Worcester, Massachusetts, USA*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, Berlin, Heidelberg, August 1999.

[15] T. Fabšič, O. Gallo, and V. Hromada. Simple power analysis attack on the QC-LDPC McEliece cryptosystem. *Tatra Mountains Mathematical Publications*, 67(1):85–92, sep 2016.

[16] P. Fouque and F. Valette. The doubling attack - *Why Upwards Is Better than Downwards*. In *Proc. of the 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'03), Cologne, Germany*, volume 2779 of *Lecture Notes in Computer Science*, pages 269–280. Springer, Berlin, Heidelberg, September 2003.

[17] N. Hanley, H. Kim, and M. Tunstall. Exploiting collisions in addition chain-based exponentiation algorithms using a single trace. In *Proc. of the Cryptographers' Track at the RSA Conference (CT-RSA'15), San Francisco, California, USA*, volume 9048 of *Lecture Notes in Computer Science*, pages 431–448. Springer, Cham, April 2015.

[18] S. Heyse, A. Moradi, and C. Paar. Power analysis attacks on software implementations of mceliece. In *Proc. of the 3rd International Workshop on Post-Quantum Cryptography (PQCrypto'10), Darmstadt, Germany*, volume 6061 of *Lecture Notes in Computer Science*, pages 108–125. Springer, Berlin, Heidelberg, May 2010.

[19] J. Heyszl, A. Ibing, S. Mangard, F. D. Santis, and G. Sigl. Clustering algorithms for non-profiled single-

execution attacks on exponentiations. In *Proc. of the 12th International Conference on Smart Card Research and Advanced Applications (CARDIS'13), Berlin, Germany*, volume 8419 of *Lecture Notes in Computer Science*, pages 79–93. Springer, Cham, November 2013.

[20] J. Heyszl, S. Mangard, B. Heinz, F. Stumpf, and G. Sigl. Localized electromagnetic analysis of cryptographic implementations. In *Proc. of the 2012 The Cryptographers' Track at the RSA Conference (CT-RSA'12), San Francisco, California, USA*, volume 7178 of *Lecture Notes in Computer Science*, pages 231–244. Springer, Berlin, Heidelberg, February 2012.

[21] N. Homma, A. Miyamoto, T. Aoki, A. Satoh, and A. Shamir. Comparative power analysis of modular exponentiation algorithms. *IEEE Transactions on Computers*, 59(6):795–807, December 2010.

[22] N. T. Inc. FIDO ECDAA Algorithm. `https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-ecdaa-algorithm-v1.1-id-20170202.html` [Online; accessed on February 2, 2020], February 2017.

[23] M. Joye. Highly regular right-to-left algorithms for scalar multiplication. In *Proc. of the 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'07), Vienna, Austria*, volume 4727 of *Lecture Notes in Computer Science*, pages 135–147. Springer, Berlin, Heidelberg, September 2007.

[24] M. Joye and S. Yen. The montgomery powering ladder. In *Proc. of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'02), Redwood Shores, California, USA*, volume 2523 of *Lecture Notes in Computer Science*, pages 291–302. Springer, Berlin, Heidelberg, August 2002.

[25] G. Karame and E. Audroulaki. *Bitcoin and Blockchain Security*. Artech House, Inc., 2016.

[26] H. Kim, T. H. Kim, J. C. Yoon, and S. Hong. Practical second-order correlation power analysis on the message blinding method and its novel countermeasure for RSA. *ETRI Journal*, 32(1):102–111, February 2010.

[27] S. Kim and S. Hong. Single trace analysis on constant time CDT sampler and its countermeasure. *Applied Sciences*, 8(10):1809:1–1809:16, October 2018.

[28] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[29] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Proc. of the 16th Annual International Cryptology Conference (CRYPTO'96), Santa Barbara, California, USA*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, Berlin, Heidelberg, August 1996.

[30] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Proc. of the 19th Annual International Cryptology Conference (CRYPTO'99), Santa Barbara, California, USA*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, Berlin, Heidelberg, August 1999.

[31] M. Lee, J. E. Song, D. Choi, and D. Han. Countermeasures against power analysis attacks for the NTRU public key cryptosystem. *IEICE Transactions*, 93-A(1):153–163, 2010.

[32] X. Lu, Y. Liu, Z. Zhang, D. Jia, H. Xue, J. He, and B. Li. LAC: practical ring-lwe based public-key encryption with byte-level modulus. *IACR Cryptology ePrint Archive*, 2018:1009, December 2019.

[33] S. Mangard, E. Oswald, and T. Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[34] D. May, H. L. Muller, and N. P. Smart. Random register renaming to foil DPA. In *Proc. of the 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES'01), Paris, France*, volume 2162 of *Lecture Notes in Computer Science*, pages 28–38. Springer, Berlin, Heidelberg, May 2001.

[35] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor. HQC (Hamming Quasi-Cyclic). `http://pqc-hqc.org/` [Online; accessed on February 2, 2020].

[36] V. S. Miller. Use of elliptic curves in cryptography. In *Proc. of the 1985 Conference on the Theory and Application of Cryptographic Techniques (CRYPTO'85), Santa Barbara, California, USA*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, Berlin, Heidelberg, August 1985.

[37] H. G. Molter, M. Stöttinger, A. Shoufan, and F. Strenzke. A simple power analysis attack on a mceliece cryptoprocessor. *Journal of Cryptographic Engineering*, 1(1):29–36, February 2011.

[38] P. L. Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48(177):243–264, January 1987.

[39] E. Nascimento, L. Chmielewski, D. Oswald, and P. Schwabe. Attacking embedded ECC implementations through cmov side channels. In *Proc. of the 23rd International Conference on Selected Areas in Cryptography*

(SAC'16), St. John's, Newfoundland and Labrador, Canada, volume 10532 of *Lecture Notes in Computer Science*, pages 99–119. Springer, Cham, August 2016.

[40] NIST. AES competition. `http://csrc.nist.gov/archive/aes/` [Online; accessed on February 2, 2020], 1997.

[41] NIST. SHA-3 competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/` [Online; accessed on February 2, 2020], 2007.

[42] NIST. Post-quantum cryptography. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography` [Online; accessed on February 2, 2020], 2016.

[43] NIST. Post-Quantum Cryptography, Round 2 Submissions, NIST Computer Security Resource Center. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions` [Online; accessed on February 2, 2020], 2019.

[44] N. I. of Standards and Technology. Digital signature standard. `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf` [Online; accessed on February 2, 2020], 1994.

[45] K. Okeya and K. Sakurai. A second-order DPA attack breaks a window-method based countermeasure against side channel attacks. In *Proc. of the 5th International Conference on Information Security (ISC'02), Sao Paulo, Brazil*, volume 2433 of *Lecture Notes in Computer Science*, pages 389–401. Springer, Berlin, Heidelberg, September 2002.

[46] A. Park and D. Han. Chosen ciphertext simple power analysis on software 8-bit implementation of ring-lwe encryption. In *Proc. of the 2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST'16), Yilan, Taiwan*, pages 1–6. IEEE, December 2016.

[47] N. J. Patterson. The algebraic decoding of goppa codes. *IEEE Transactions on Information Theory*, 21(2):203–207, March 1975.

[48] G. Perin and L. Chmielewski. A semi-parametric approach for side-channel attacks on protected RSA implementations. In *Proc. of the 14th International Conference on Smart Card Research and Advanced Applications (CARDIS'15), Bochum, Germany*, volume 9514 of *Lecture Notes in Computer Science*, pages 34–53. Springer, Cham, November 2015.

[49] G. Perin, L. Imbert, L. Torres, and P. Maurine. Attacking randomized exponentiations using unsupervised learning. In *Proc. of the 5th International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'14), Paris, France*, volume 8622 of *Lecture Notes in Computer Science*, pages 144–160. Springer, Cham, April 2014.

[50] M. Petrvalsky, T. Richmond, M. Drutarovsky, P.-L. Cayrel, and V. Fischer. Countermeasure against the spa attack on an embedded mceliece cryptosystem. In *Proc. of the 2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA'15), Pardubice, Czech Republic*, pages 462–466. IEEE, April 2015.

[51] M. Petrvalsky, T. Richmond, M. Drutarovsky, P.-L. Cayrel, and V. Fischer. Differential power analysis attack on the secure bit permutation in the mceliece cryptosystem. In *Proc. of the 2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA'16), Kosice, Slovaki*, pages 132–137. IEEE, April 2016.

[52] R. Primas, P. Pessl, and S. Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In *Proc. of the 19th International Conference on Cryptographic Hardware and Embedded Systems (CHES'17), Taipei, Taiwan*, volume 10529 of *Lecture Notes in Computer Science*, pages 513–533. Springer, Cham, September 2017.

[53] O. Reparaz, R. de Clercq, S. S. Roy, F. Vercauteren, and I. Verbauwhede. Additively homomorphic Ring-LWE masking. In *Proc. of the 7th International Workshop on Post-Quantum Cryptography (PQCrypto'16), Fukuoka, Japan*, volume 9606 of *Lecture Notes in Computer Science*, pages 233–244. Springer, Cham, February 2016.

[54] O. Reparaz, S. S. Roy, F. Vercauteren, and I. Verbauwhede. A masked Ring-LWE implementation. In *Proc. of the 17th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'15), Saint-Malo, France*, volume 9293 of *Lecture Notes in Computer Science*, pages 683–702. Springer, Berlin, Heidelberg, September 2015.

[55] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[56] M. Rossi, M. Hamburg, M. Hutter, and M. E. Marson. A side-channel assisted cryptanalytic attack against qcbits. In *Proc. of the 19th International Conference on Cryptographic Hardware and Embedded Systems*

(CHES'17), Taipei, Taiwan, volume 10529 of *Lecture Notes in Computer Science*, pages 3–23. Springer, Cham, September 2017.

[57] S. S. Roy, O. Reparaz, F. Vercauteren, and I. Verbauwhede. Compact and side channel secure discrete gaussian sampling. *IACR Cryptology ePrint Archive*, 2014:591, August 2014.

[58] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc. of the 35th IEEE Annual Symposium on Foundations of Computer Science (SFCS'94), Santa Fe, New Mexico, USA*, pages 124–134. IEEE, November 1994.

[59] A. Shoufan, F. Strenzke, H. G. Molter, and M. Stöttinger. A timing attack against patterson algorithm in the mceliece PKC. In *Proc. of the 12th International Conference on Information, Security and Cryptology (ICISC'09), Seoul, Korea*, volume 5984 of *Lecture Notes in Computer Science*, pages 161–175. Springer, Berlin, Heidelberg, December 2009.

[60] J. H. Silverman and W. Whyte. Timing attacks on ntruencrypt via variation in the number of hash calls. In *Proc. of the Cryptographers' Track at the RSA Conference (CT-RSA'07), San Francisco, California, USA*, volume 4377 of *Lecture Notes in Computer Science*, pages 208–224. Springer, Berlin, Heidelberg, February 2007.

[61] B. Sim and D. Han. Key bit-dependent attack on protected PKC using a single trace. In *Proc. of the 13th International Conference on Information Security Practice and Experience (ISPEC'17), Melbourne, Victoria, Australia*, volume 10701 of *Lecture Notes in Computer Science*, pages 168–185. Springer, Cham, December 2017.

[62] B. Sim and D. Han. Single-trace vulnerability of countermeasures against instruction-related timing attack. Cryptology ePrint Archive, Report 2019/1236, October 2019. `https://eprint.iacr.org/2019/1236` [Online; accessed on February 2, 2020].

[63] B. Sim, J. Kang, and D. Han. Key bit-dependent side-channel attacks on protected binary scalar multiplication. *Applied Sciences*, 8(11):2168, November 2018.

[64] B. Sim, J. Kwon, K. Y. Choi, J. Cho, A. Park, and D. Han. Novel side-channel attacks on quasi-cyclic code-based cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(4):180–212, August 2019.

[65] R. Specht, J. Heyszl, M. Kleinsteuber, and G. Sigl. Improving non-profiled attacks on exponentiations based on clustering and extracting leakage from multi-channel high-resolution EM measurements. In *Proc. of the 6th International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'15), Berlin, Germany*, volume 9064 of *Lecture Notes in Computer Science*, pages 3–19. Springer, Cham, April 2015.

[66] F. Strenzke. A timing attack against the secret permutation in the mceliece PKC. In *Proc. of the 3rd International Workshop on Post-Quantum Cryptography (PQCrypto'10), Darmstadt, Germany*, volume 6061 of *Lecture Notes in Computer Science*, pages 95–107. Springer, Berlin, Heidelberg, May 2010.

[67] F. Strenzke. Message-aimed side channel and fault attacks against public key cryptosystems with homomorphic properties. *Journal of Cryptographic Engineering*, 1(4):283–292, October 2011.

[68] F. Strenzke. Timing attacks against the syndrome inversion in code-based cryptosystems. In *Proc. of the 5th International Workshop on Post-Quantum Cryptography (PQCrypto'13), Limoges, France*, volume 7932 of *Lecture Notes in Computer Science*, pages 217–230. Springer, Berlin, Heidelberg, June 2013.

[69] F. Strenzke, E. Tews, H. G. Molter, R. Overbeck, and A. Shoufan. Side channels in the mceliece PKC. In *Proc. of the 2nd International Workshop on Post-Quantum Cryptography (PQCrypto'08), Cincinnati, Ohio, USA*, volume 5299 of *Lecture Notes in Computer Science*, pages 216–229. Springer, Berlin, Heidelberg, October 2008.

[70] T. Sugawara, D. Suzuki, and M. Saeki. Two operands of multipliers in side-channel attack. In *Proc. of the 6th International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'15), Berlin, Germany*, volume 9064 of *Lecture Notes in Computer Science*, pages 64–78. Springer, Cham, April 2015.

[71] I. von Maurich and T. Güneysu. Lightweight code-based cryptography: QC-MDPC mceliece encryption on reconfigurable devices. In *Proc. of the 2014 IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE'14), Dresden, Germany*, pages 1–6. IEEE, March 2014.

[72] I. von Maurich and T. Güneysu. Towards side-channel resistant implementations of QC-MDPC mceliece encryption on constrained devices. In *Proc. of the 6th International Workshop on Post-Quantum Cryptography*

(PQCrypto'14), Waterloo, Ontario, Canada, volume 8772 of *Lecture Notes in Computer Science*, pages 266–282. Springer, Cham, October 2014.

[73] G. Wafo-Tapa, S. Bettaieb, L. Bidoux, and P. Gaborit. A practicable timing attack against HQC and its countermeasure. *IACR Cryptology ePrint Archive*, 2019:909, September 2019.

[74] C. D. Walter. Sliding windows succumbs to big mac attack. In *Proc. of the 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES'01), Paris, France*, volume 2162 of *Lecture Notes in Computer Science*, pages 286–299. Springer, Berlin, Heidelberg, May 2001.

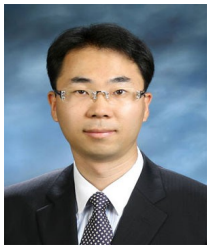
[75] M. Walters and S. S. Roy. Constant-time BCH error-correcting code. *IACR Cryptology ePrint Archive*, 2019:155, April 2019.

[76] A. Wang, X. Zheng, and Z. Wang. Power analysis attacks and countermeasures on ntru-based wireless body area networks. *TIIS*, 7(5):1094–1107, May 2013.

[77] X. Zheng, A. Wang, and W. Wei. First-order collision attack on protected NTRU cryptosystem. *Microprocess. Microsystems*, 37(6-7):601–609, August 2013.

Author Biography

Do-Yeon Sim received her BS degree in mathematics, and MS and PhD degrees in information security from Kookmin University, Seoul, Rep. of Korea, in 2013, 2015, 2020, respectively. She is currently working as a research fellow in Kookmin University.



Dong-Guk Han received his BS and MS degrees in mathematics from Korea University, Seoul, Rep. of Korea, in 1999 and 2002, respectively. He received his PhD degree in engineering in information security from Korea University, in 2005. He was a postdoctoral researcher at Future University Hakodate, Hokkaido, Japan. After finishing his doctoral course, he was then an exchange student with the Department of Computer Science and Communication Engineering, Kyushu University, Japan, from April 2004 to March 2005. From 2006 to 2009, he was a senior researcher at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. He is currently working as a professor with the Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul, Rep. of Korea.