IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 February 2023, accepted 1 May 2023, date of publication 4 May 2023, date of current version 10 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3273150

RESEARCH ARTICLE

Stealing Keys From Hardware Wallets: A Single Trace Side-Channel Attack on Elliptic Curve Scalar Multiplication Without Profiling

DONGJUN PARK[1], MINSIG CHOI[1], GYUSANG KIM[1], DAEHYEON BAE[1], HEESEOK KIM[2], (Member, IEEE), AND SEOKHIE HONG[1], (Member, IEEE)
[1]Institute of Cyber Security and Privacy (ICSP), Korea University, Seoul 02841, South Korea
[2]Department of AI Cyber Security, College of Science and Technology, Korea University, Sejong Campus, Sejong-si 30019, South Korea

Corresponding author: Seokhie Hong (shhong@korea.ac.kr)

This work was supported in part by the Military Crypto Research Center funded by the Defense Acquisition Program Administration (DAPA) and the Agency for Defense Development (ADD) under Grant UD210027XD.

ABSTRACT Over the past decade, decentralized cryptocurrencies have received attention in industry and academia. Hardware wallets are dedicated devices that manage cryptocurrencies safely without entrusting cryptographic keys to a third party. Side-channel attacks have been widely studied in cryptanalysis and have already been proven threatening, but analysis on hardware wallets still needs to be researched. Although the previous work demonstrated several side-channel vulnerabilities, their attacks require a finely controlled environment or a learning phase of target devices' physical properties before the attacks. This paper proposes a side-channel attack on hardware wallets extracting private keys. The proposed attack needs a single power trace measured when wallets process elliptic curve scalar multiplication with private keys. Our attack is reasonable since we do not damage the device under attack and do not target a specific device but an algorithm; it is widely applicable to wallets using that algorithm or analogous ones. It also presents the attack results conducted with three datasets: simulation, ChipWhisperer, and actual dataset collected from the Trezor Model One, the first and representative hardware wallets which comply with the de facto standard of hardware wallets.

INDEX TERMS Cryptocurrency, hardware security, power analysis, side-channel attack.

I. INTRODUCTION

Decentralized cryptocurrencies utilize blockchain technology that allows every participant in a network to share a distributed ledger without any central authority [1]. Participants in a blockchain network use digital signatures enabling others to validate transactions and verify their ownership. As long as the majority has not tampered with the network, consensus mechanisms, such as proof-of-work based on hash algorithms, prevent the double-spending problem [2]. The main structure of blockchain consists of cryptographic algorithms.

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

Even if cryptographic primitives have been considered computationally infeasible to break, one should pay attention to the actual use of cryptocurrencies. Several cryptocurrency exchanges, which provide blockchain networking, transaction processing, and key management services for those unfamiliar with cryptography, have lost cryptographic keys, thereby losing cryptocurrencies [3]. The exchanges even have a contradiction in that they tend to fully control the users' secret information, while the core principle of blockchain is decentralization [4].

One of the resolutions for users who want to handle secret information themselves is to employ a hardware wallet, a dedicated device that stores keys and performs cryptographic operations [5]. Hardware wallets are secure against online hacking because they are physically disconnected from the

Internet unless they process transactions. Furthermore, some manufacturers have disclosed the implementation details for public and formal verification of hardware wallets, including cryptographic algorithms [6].

Cryptographic devices should also be secure against physical attacks, especially side-channel attacks that exploit physical leakages of implementation [7]. There are two kinds of side-channel attacks in the literature. The first is power analysis attacks that analyze the amount of electric power consumed by a device while the device is legitimately processing intermediate values related to secret information [8]. Since intermediate values affect the device's power distribution network, leading to voltage drop, it is possible to extract secret information by analyzing power traces [9]. Electromagnetic (EM) radiation from the device is also exploitable due to the fundamental laws of electromagnetism [10]. The second is fault injection attacks that analyze erroneous output intended by attackers. To cause those errors, an attacker tampers with the device's clock frequency [11] or voltage supply [12]. Also, exposing the device to an intense laser [13] or EM pulse [14] can cause exploitable errors.

A. RELATED WORKS

A recent line of papers has shown that side-channel leaks the secret values of hardware wallets, resulting in hardware wallet cloning and cryptocurrency theft. In [15], they demonstrated that voltage glitches could downgrade the device's readout protection level from 2 to 1, granting access to the static random-access memory (SRAM). After that, they could read the recovery seed through debugging ports by forcefully halting the firmware upgrade before the SRAM gets cleared since the wallet backs up recovery seeds to SRAM for an upgrade. However, the wallet owner can notice the attack because the package should be removed to use debuggers.

Another related work presented that EM fault injection made it possible to bypass a protection mechanism of request handling [16]. With the host computer and the wallet connected by a USB cable, they sent a request message to read the flash memory containing the recovery seed. The request handler would generally reject this request, but they injected faults and skipped a comparison instruction that checks whether a received request accesses flash memory. It is difficult for the wallet owner to know the attack has occurred because it has not physically damaged the wallet. For realizing this attack, the most challenging task is fine-tuning the parameters, such as location, duration, intensity, and delay, to trigger comparison skipping.

Not only fault injection attacks but power analysis attacks also reveal the secret values of hardware wallets. In 2019, the wallet manufacturer Ledger mounted two profiled attacks on the Trezor wallet, another manufacturer's product, showing that the personal identification number (PIN) and private keys are vulnerable [17]. The first attack extracted a four-digit PIN utilizing the observation that the power patterns when the input PIN coincides with the stored one differ from when they do not. Before the attack, they profiled 40 power templates from 0 to 9 for each digit with a device whose PIN is known. Then, they made PIN login attempts with another device of the same model (the correct PIN is unknown) and gathered corresponding power traces during the PIN verification. Finally, the power traces are compared with the templates to guess the correct PIN digit-by-digit. The success rate of their matching was 100%, which means the attacker would reconstruct the correct PIN within 10 attempts in the worst case (5.5 attempts on average) and unlock the wallet. This vulnerability has been mitigated by modifying the operations inside the PIN verification so that the entire PIN can be recovered only after more attempts, whereas the wallet wipes its data after 16 attempts.

The most relevant work to our study is the second attack of the Ledger research team that extracted a private key from the elliptic curve scalar multiplication (ECSM) algorithm [17]. Roughly speaking, the ECSM of the Trezor consists of 64 iterations of point addition and conditional negation, which process a 256-bit secret scalar by 4 bits (see III-B for the details). Point addition is carried out by referring to the index number of eight precomputed operands, so this operation has a 3-bit information of scalar. The remaining 1-bit information is at conditional negation, which reverses the sign of accumulated result according to a condition.

B. CONTRIBUTIONS

This paper proposes a side-channel attack on an open-source cryptographic library in commercial hardware wallets. The proposed attack extracts a private key, which is sensitive information because anyone with a key can sign the transactions and use them as a payment method. Specifically, the attack exploits a single power trace gathered when a processor executes the ECSM algorithm, a cryptographic operation necessary to generate a receiving address of cryptocurrencies. We referred to the Trezor library [18] because the Trezor complies with the de facto standard in cryptocurrency, Bitcoin Improvement Proposal (BIP) [19], [20], [21], [22], [23], and also it is the first hardware wallet for commercial use that has inspired many other hardware wallets. We mounted our attack with three datasets: simulation, ChipWhisperer, and actual dataset collected from Trezor Model One.

Compared to [17], our attack does not require prior profiling setup with a fully controlled device. Furthermore, the proposed attack needs fewer power traces, ideally just one. Unlike the fault injection attacks that require fine-tuning the attack setup [15], [16], the proposed attack only needs to set the time duration as long as possible so that the power trace contains the target algorithm. The proposed attack will be possible without damaging the package by collecting power traces from the power supply line rather than the wallet itself or by locating EM probes near the wallet's processing unit.

Our contribution is summarized as mainly twofold.

- To the best of our knowledge, our work is the first key-extracting attack without profiling phases. Since our attack is not a profiled one targeting a specific device, it is applicable whenever the device uses a targeted algorithm or analogous ones. We targeted ECSM, a frequent operation in hardware wallets that derives hierarchical deterministic keys and produces receiving addresses. An attacker with the key can steal the coin currently deposited at the corresponding address and the one that will be in the future.

- The proposed attack is under the reasonable assumption; Hence, the attacker can readily reproduce the attack, and the wallet owner can seldom notice the attack. We assume an attacker can collect just one power or EM trace. The attacker does not modify the firmware inside the wallet nor damage the package to acquire such a trace. There is no need to fine-tune the equipment used for power acquisition nor to put the specific query the attacker wants into the target under attack.

C. ORGANIZATION

The rest of this paper is organized as follows. Section II defines notations and introduces background knowledge on side-channel attacks. Section III describes the standard structure of hardware wallets based on BIP documents. It also describes cryptographic algorithms implemented in hardware wallets, especially in the Trezor wallet. Section IV proposes a power analysis attack on hardware wallets in a non-profiled environment. This section explains how our attack extracts sensitive values from cryptographic algorithms and how to reconstruct them into a private key. Section V presents the experimental results with three datasets. Section VI discusses the impact on the real-world and several countermeasures against the attack. Lastly, Section VII concludes the paper and suggests future works.

II. PRELIMINARIES

In this section, we introduce our notations and assumptions. Then, the following subsections briefly introduce two side-channel analysis techniques necessary to understand this paper: Simple Power Analysis and Correlation Power Analysis.

Let a be an integer. The radix b representation of a is $(a_{n-1}a_{n-2}\cdots a_0)_b$ for some n. If $b = 2$ (binary notation), a_{n-1} and a_0 are called the most significant bit (MSB) and least significant bit (LSB), respectively. In hexadecimal representation, the digits are not italicized, and the symbol '0x' is prefixed, omitting some leading zeros.

We assume that the device consumes power depending on the Hamming weight of an intermediate value, which is a function of the known value p and unknown value k. We represent the intermediate value as $v = (v_{31}v_{30}\cdots v_0)_2$, assuming the 32-bit architecture, and its Hamming weight as $h = \sum_{i=0}^{31} v_i$. We also assume that the device has a white Gaussian noise ϵ with a variance σ^2. Finally, we model the device's power consumption as $l = h + \epsilon$.

Table 1 shows the meaning of specific symbols used in this paper. We refer to Standards for Efficient Cryptography 2 (SEC 2) definitions regarding the elliptic curve domain parameters [24]. Unless otherwise stated, we only consider symmetric and public key cryptosystems with a key size of 128 and 256 bits, respectively.

TABLE 1. Meaning of symbols used in this paper.

Symbol	Meaning
secp256k1	The elliptic curve parameters as defined in SEC 2
p_{256}	The prime modulus of the secp256k1
n_{256}	The group order of the secp256k1
G	The base point of the secp256k1
$\|\|$	Concatenate two expressions on either side
$\leftarrow$	Assign the expression on the right side to the left
$\neg$	Bitwise negation operator
$\gg$	Bitwise right shift operator
$\wedge$	Bitwise and operator
$\vee$	Bitwise or operator
$\oplus$	Bitwise exclusive or operator
$[i][j]$	The entry at the i-th row and j-th column
$\mathbb{E}$	Sample mean
$\mathbb{V}$	Sample variance

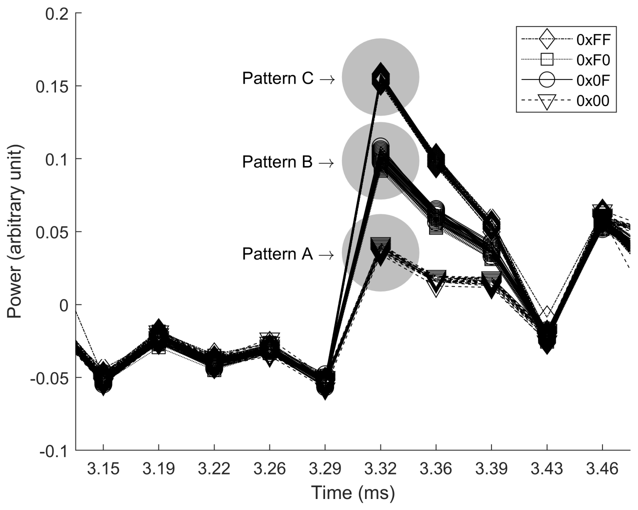


FIGURE 1. Power consumption traces processing the values 0x00, 0x0F, 0xF0, and 0xFF. It shows three power patterns according to the Hamming weight of 0, 4, and 8.

A. SIMPLE POWER ANALYSIS

Simple Power Analysis (SPA) is an intuitive method that exploits power patterns distinguished due to a difference in Hamming weights [8]. For example, the larger the Hamming weight of the data processed, the sharper the slope of the power pattern, as shown in Fig. 1. It is known to be a powerful attack, even effective to the latest cryptographic systems, such as post-quantum cryptography [25] and quantum key distribution [26].

Let $V = \{v_1, v_2, \cdots v_n\}$ be a set of n values that an intermediate variable can have according to a particular statement. It should be sorted in ascending order with respect to the Hamming weight of each value, that is, $h_1 < h_2 < \cdots < h_n$. Then, measured power traces during the statement will be clustered into n groups by distinguishing their pattern. One can conclude that the traces in i-th lowest power group correspond to the value v_i.

For the SPA attack to succeed, Hamming weights of the values in V should be unique and significantly different among them, and the number of possible Hamming weights n should be small. Otherwise, the power patterns might overlap, so it is challenging to partition them into separate groups. For example, Fig. 1 shows the one hundred power traces measured during the statement that processed the intermediate values of 0x00, 0x0F, 0xF0, and 0xFF only. The traces in groups A and C are considered to be due to the intermediate values of 0x00 and 0xFF, respectively. However, one cannot decide whether each trace in group B corresponds to which intermediate value since the Hamming weights of 0x0F and 0xF0 are the same.

B. CORRELATION POWER ANALYSIS

Correlation Power Analysis (CPA) is a statistical method that exploits the correlation between hypothetical power consumption and measured power traces [9]. The CPA attacks primarily target symmetric key cryptosystems such as Advanced Encryption Standard (AES) [27] because it requires multiple pairs of power trace and intermediate value. However, in public key cryptosystems, just one encryption occurs, and ephemeral keys are involved in each encryption, making it impossible to compute intermediate values. In this case, a variant of CPA called Horizontal Correlation Analysis (HCA) is applicable [28]. The difference is that CPA queries many times to acquire multiple traces, whereas HCA needs only one query for obtaining multiple subtraces divided from a single trace when the target algorithm has an iterative statement.

Let $T = \{t_1, t_2, \cdots t_n\}$ be a set of n power traces (or subtraces) measured during a cryptographic operation. When each trace t_i has m time points, let us represent the j-th point as $t_{i,j}$ for $1 \le j \le m$. All traces should be aligned so that the power consumption of the same instruction appears at the same point. One can compute the intermediate values v_i from known values p_i by guessing an unknown value k. By analyzing the correlation between the power model ($l_i = h_i + \epsilon$) and the actual power consumption t_i at some points of interest, it is possible to determine which guessed key is most related to the actual power consumption.

The Pearson correlation coefficient measures a linear relationship between two sets of data L and T as follows:

$$\rho_{LT} = \frac{\mathbb{E}[LT] - \mathbb{E}[L]\mathbb{E}[T]}{\sqrt{\mathbb{E}[L^2] - \mathbb{E}^2[L]}\sqrt{\mathbb{E}[T^2] - \mathbb{E}^2[T]}}. \tag{1}$$

Since we have assumed the white Gaussian noise, by eliminating zero terms, the sample correlation coefficient between n sampled power model $l_i \in L$ $(1 \le i \le n)$ and n sampled power consumption $t_i \in T$ at the point j is calculated as

$$r_j = \left| \frac{n\sum l_i t_{i,j} - \sum l_i \sum t_{i,j}}{\sqrt{n\sum l_i^2 - (\sum l_i)^2}\sqrt{n\sum t_{i,j}^2 - (\sum t_{i,j})^2}} \right| \tag{2}$$

for the guessed key k. Note that the absolute value of the correlation coefficient is taken since the negative correlation

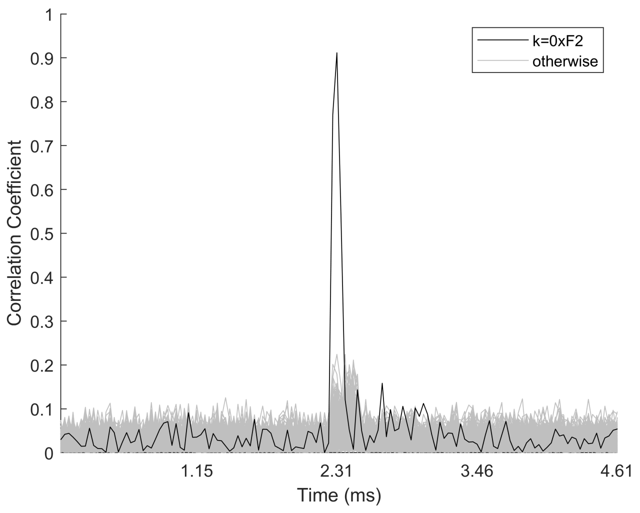


FIGURE 2. **Pearson correlation coefficients between the power models and measured traces. The model with guessed key 0xF2 shows the highest correlation at near 2.31ms.**

is just due to the reversed setup of power probes. Please also note that white noise converges to zero when the number of traces is sufficiently large. One can conclude that the correct key $k^\star$ is a key that maximizes the correlation between hypothetical and measured traces, or $k^\star = \arg\max_k(r_j(k))$.

However, an attacker does not know the exact point of interest. The most elementary resolution to this problem is to calculate $r_j(k)$ for every time point $1 \le j \le m$. For example, Fig. 2 shows the correlation coefficient between the power model with 256 guessed round keys and the 1,000 power traces measured during the first round of the Advanced Encryption Standard (AES) algorithm [27]. In this case, the power model is the Hamming weight of $S[p_i \oplus k]$ where S refers to the substitution table of the AES and p_i stands for 1,000 plaintexts of one byte. The highest correlation coefficient for the power model with $k = $ 0xF2 at $j = 2.31$ms implies that the actual round key is 0xF2 and that the AES substitution appears at that time.

III. ANALYSIS OF HARDWARE WALLETS

This section describes the standard structure of hardware wallets based on BIP documents. It also describes elliptic curve cryptography used in hardware wallets.

A. HIERARCHICAL DETERMINISTIC WALLETS

In cryptocurrency software, private keys are needed and generated using random numbers and the Password-Based Key Derivation Function 2 (PBKDF2). However, it is inefficient to generate a new random number every time a new private key is needed. Sometimes small devices such as hardware wallets do not embed random number generators. These promote the development of a hierarchical deterministic key tree, as proposed in BIP-0032 [19], that consists of multiple keys derived from a root key as illustrated in Fig. 3.

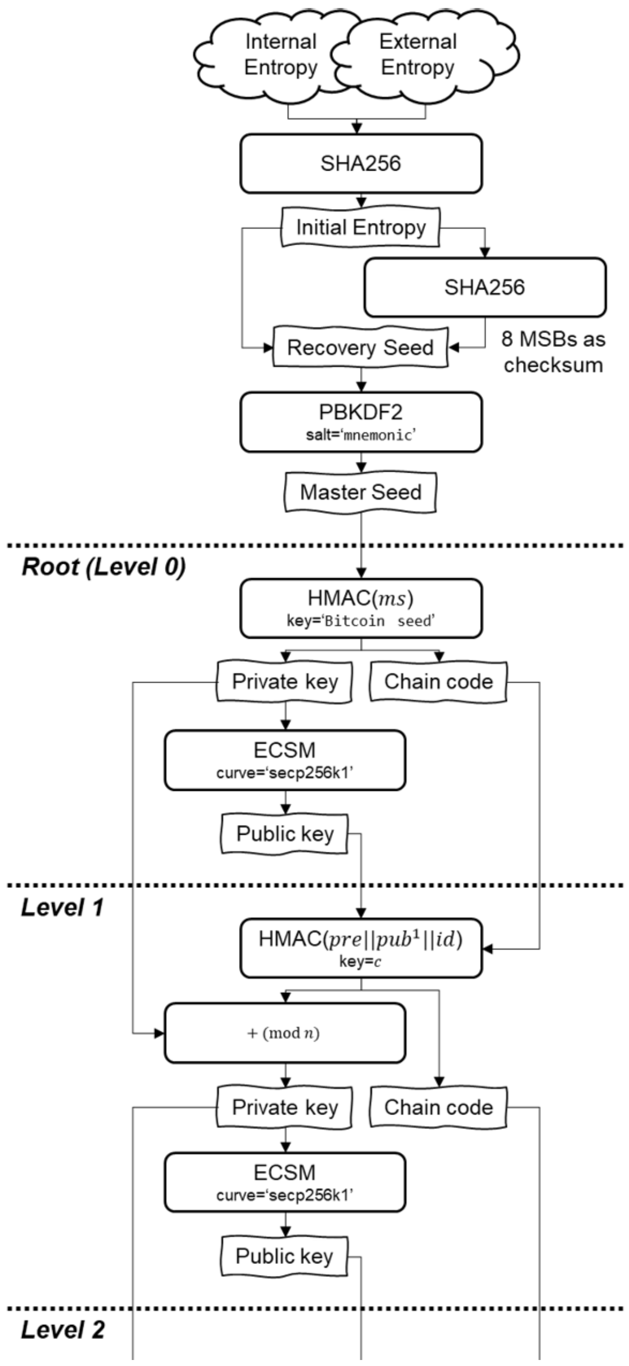


FIGURE 3. **A standard structure of hierarchical deterministic key trees.**

Creating a root node is as follows. A 256-bit initial entropy and its 8-bit checksum are concatenated and are input to the PBKDF2 with the string `mnemonic` as the salt for obtaining a 512-bit master seed. After that, the master seed is input to the Hash-based Message Authentication Code (HMAC) with the string `Bitcoin seed` as the HMAC key in the case of Bitcoin. The output's upper and lower 256 bits are used as the private key and chain code, respectively. Finally, the public key is the x-coordinate of the point calculated by ECSM with the private key and the base point G.

To derive a child node, the parent's public key[1] and an identifier of the child are input to HMAC with the parent's chain code as an HMAC key. On the one hand, the sum of the output's upper 256 bits and the parent's private key modulo n_{256} will be the child's private key. On the other hand, the output's lower 256 bits are used as the child's chain code. Again, the public key is the x-coordinate of the point calculated by ECSM with the private key and the base point.

To summarize, the private key, public key, and chain code at level d, namely prv_d, pub_d, and c_d, are derived as follows with the HMAC function H and ECSM ($\times$):

$$prv_d \,||\, c_d = \begin{cases} H(\text{Bitcoin seed}, ms) & \text{if } d = 0 \\ H(c_{d-1}, pre \,||\, pub_{d-1} \,||\, id) & \text{if } d \geq 1. \end{cases} \tag{3}$$

$$(pub_d, y_d) = prv_d \times G. \tag{4}$$

where ms stands for master seed. Additionally, one byte prefix pre is followed by pub_{d-1} to indicate the parity of y-coordinate (0x02 if y_{d-1} is even, 0x03 otherwise). This derivation method can derive 2^{32} nodes per layer by changing the 32-bit identifier id. It has no limit on the depth of the tree.

The subsequent BIPs define the roles of each layer. In BIP-0043 [20], the first layer was defined as a purpose layer such that node identifiers in the first layer represent the BIP documents defining the roles of lower layers (For example, the roles of nodes below m/44/ are defined in BIP-0044). BIP-0044 [21], 0049 [22], and 0084 [23] define a tree structure of depth five with level 2 as coin type, 3 as account, 4 as change, and 5 as address. The three proposals are the same except for encoding methods for address in the fifth layer (P2PKH, P2SH, and P2WPKH, respectively).

B. CRYPTOGRAPHIC ALGORITHMS IN WALLETS

A digital signature algorithm is for verifying the authenticity of digital messages. Digital signatures give the public a reason to believe that the transaction was sent by the claimed signer, making it possible to build a decentralized cryptocurrency. A person with a private key can generate an address and use a coin deposited at that address with the corresponding key.

Digital signature algorithms based on elliptic curves are suitable for small devices with limited computing resources like hardware wallets because they have the same security level as previous cryptosystems, even with shorter keys [29]. In elliptic curve cryptosystems, ECSM is an operation related to security since the input value is typically confidential. Plenty of works exist to evaluate the physical security of the implementations [30], [31], [32], [33] and enhance their security [34], [35], [36], [37]. ECSM is also related to efficiency because it is heavy and often called during the child key derivation. Optimizing ECSM will increase the throughput of the entire cryptosystem [38], [39], [40], [41].

[1]The hardened version of child key derivation takes the private key as an input of HMAC instead. We only consider the non-hardened version since it does not interfere with the claim of this paper.

Algorithm 1 Elliptic Curve Scalar Multiplication

Input: A scalar k, the curve parameters secp256k1, and the precomputed table $T[i][j] = (2j + 1)16^iG$ in affine coordinates for $0 \leq i \leq 63$ and $0 \leq j \leq 7$.

Output: kG.

```
1: $a \leftarrow k + 2^{256}$.
2: if $a$ is even then
3:     $a \leftarrow a - n_{256}$.
4: if $a = 0$ then
5:     return $\infty$.
6: $b \leftarrow a \wedge 0x1F$.
7: $b \leftarrow (b \oplus ((b \gg 4) - 1)) \wedge 0xF$.
8: $R \leftarrow J2A(T[0][b \gg 1])$.
9: for $i \leftarrow 1$ to 63 do
10:     $a \leftarrow a \gg 4$.
11:     $b \leftarrow a \wedge 0x1F$.
12:     $b \leftarrow (b \oplus ((b \gg 4) - 1)) \wedge 0xF$.
13:     if $\neg b \wedge 1$ then
14:         $R \leftarrow -R$.
15:     $R \leftarrow T[i][b \gg 1] + R$.
16: if $\neg(a \gg 4) \wedge 1$ then
17:     $R \leftarrow -R$.
18: return $A2J(R)$.
```

Algorithm 2 Conditional Negation

Input: A condition c, a point $R = (x, (y_8y_7 \cdots y_0)_{2^{29}})$, and the prime modulus $p_{256} = (p_8p_7 \cdots p_0)_{2^{29}}$.

Output: If $c = 1$, then $-R$, else R.

```
1: $\alpha \leftarrow -c$.
2: $\beta \leftarrow \neg\alpha$.
3: $a \leftarrow 1$.
4: $b \leftarrow 0$.
5: for $i \leftarrow 0$ to 8 do
6:     $a \leftarrow a + 0x1FFFFFFF + 2p_i - y_i$.
7:     $b \leftarrow b + p_i + y_i$.
8:     $y_i \leftarrow ((a \wedge \alpha) \vee (b \wedge \beta)) \wedge 0x1FFFFFFF$.
9:     $a \leftarrow a \gg 29$.
10:    $b \leftarrow b \gg 29$.
11: return $(x, y)$.
```

Algorithm 3 Point Addition

Input: An affine point $T = (x, y)$, a Jacobian point $R = (X : Y : Z)$, and the prime modulus p_{256}.

Output: $T + R$ in Jacobian coordinates.

```
1: $P1 \leftarrow Z \times Z \mod p_{256}$.
2: $P2 \leftarrow P1 \times Z \mod p_{256}$.
3: $P1 \leftarrow x \times P1 \mod p_{256}$.        // $x \times Z^2$
4: $P4 \leftarrow P1 - X \mod p_{256}$.
5: $P1 \leftarrow P1 + X \mod p_{256}$.
6: $P2 \leftarrow y \times P2 \mod p_{256}$.        // $y \times Z^3$
   /* More steps... */
```

Algorithm 4 Field Multiplication Modulo p_{256}

Input: Two 256-bit integers $a = (a_8a_7 \cdots a_0)_{2^{29}}$ and $b = (b_8b_7 \cdots b_0)_{2^{29}}$, and the prime modulus p_{256}.

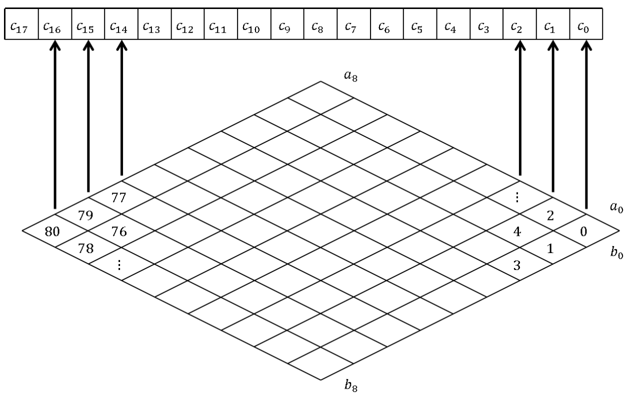
Output: $(c_8c_7 \cdots c_0)_{2^{29}} = a \times b \mod p_{256}$.

```
1: $t \leftarrow 0$.                // Double precision integer
2: for $i \leftarrow 0$ to 8 do
3:     for $j \leftarrow 0$ to $i$ do
4:         $t \leftarrow t + a_j \times b_{i-j}$.
5:     $c_i \leftarrow t \wedge 0x1FFFFFFF$.
6:     $t \leftarrow t \gg 29$.
7: for $i \leftarrow 9$ to 16 do
8:     for $j \leftarrow i - 8$ to 8 do
9:         $t \leftarrow t + a_j \times b_{i-j}$.
10:    $c_i \leftarrow t \wedge 0x1FFFFFFF$.
11:    $t \leftarrow t \gg 29$.
12: $c_{17} \leftarrow t$.
13: return $fastmod(c, p_{256})$.     // Fast modular reduction
```

Trezor Model One gives a good example of secure and efficient ECSM implementation, as Alg. 1 shows [18]. It comprises 64 iterative conditional negations (Alg. 2) and point additions (Alg. 3), which process a 256-bit secret scalar by 4 bits. Considering security, constant-time programming applies so that the execution time is independent of the input value. This algorithm gains efficiency benefits from the precomputed table of 36 MB. Typical binary algorithms require 6.49 times more field multiplications than this.

A detailed explanation of the ECSM (Alg. 1) is as follows. Steps 1-5 make a temporary scalar $a = k + 2^{256}$ neither even nor zero. Since $k \equiv a - 2^{256} \mod n_{256}$, we can compute $kG = \sum_{i=0}^{63} a_i 16^i G$ for $a = (a_{64}a_{63} \cdots a_0)_{16}$ and $a_{64} = 1$. To use the precomputed points $(2j + 1)16^iG$ for $0 \leq i \leq 7$ and $0 \leq j \leq 7$, we need a signed-digit representation of a having odd digits only. a_{64} is 1 and thus odd due to Step 1. a_0 is also odd because, in Step 3, we subtract n_{256}, which is odd, from a only if a is even. We can make the remaining digits a_i odd by adding 1 to a_i and subtracting 16 from a_{i-1}. Steps 6-8 compute $R = |a_0|G$ in Jacobian coordinates. For $1 \leq i \leq 63$, Steps 10-12 construct $|a_i|$, Steps 13-14 negate the previous result if $\text{sgn}(a_i) \neq \text{sgn}(a_{i-1})$, and Step 15 performs affine-Jacobian mixed point addition using T. The result will be $R = \text{sgn}(a_i)\sum_{t=0}^{i} a_t 16^t G$ at the end of Step 15. Steps 16-17 negate the last result with the same condition above. Finally, Step 18 brings the Jacobian point back to affine coordinates.

Alg. 2 shows the pseudo-code for conditional negating an elliptic curve point. Explicitly, this algorithm takes an input $R = (x, y)$ and returns the additive inverse $-R = (x, -y)$ only if the condition is $c = 1$. Otherwise, the output is the same as the input. For the execution time to be constant, the implementation prepares two 32-bit masks α for a in Step 1 and β for b in Step 2. If an attacker knows their exact values through side-channel analysis on Steps 1 or 2, it will leak c, 1-bit information of the temporary scalar a. The rest of Algorithm computes $-y \mod p_{256}$ with the range from p_{256} (inclusive) to $2p_{256}$ (exclusive) by 29 bits for each iteration.



c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0
a_8
a_0
b_0
b_8
80
79
78
77
76
2
1
0
4
3

FIGURE 4. Multiple precision multiplication with the product scanning method.

Alg. 3 shows the mixed-coordinate point addition. This algorithm requires an affine point T and a Jacobian point R and returns $T + R$ in Jacobian coordinates within 14 field multiplications modulo p_{256}. T is one of the precomputed points among the eight in the i-th row of the table. One has yet to learn which point contributes to computation because the table index $b \gg 1$ in Alg. 1 remains unknown. On the contrary, if an attacker knows the exact point through side-channel attacks on Steps 5 or 8, it will leak 3-bit information of the temporary scalar a.

Alg. 4 shows the product scanning method for multiplying two multiple precision integers modulo p_{256}. Product scanning takes fewer memory operations and registers than the operand scanning method (also known as the schoolbook method). Steps 2-6 compute the lower half of $c = a \times b$ by multiplying two 29-bit integers a_j and b_{i-j} and accumulating it into a double precision integer t. Consecutively, Steps 7-12 compute the remaining upper half. Fig. 4 illustrates the order in which the two operands are processed. Finally, Step 13 reduces the 512-bit c to 256 bits by using a fast modular reduction algorithm applicable when a modulus is the form of a generalized Mersenne prime [42].

In addition to Alg. 1, Trezor supports an alternative ECSM algorithm that precomputes a fresh table before performing every single ECSM. This option enhances security since every point in the table has a randomized Z-coordinate in Jacobian form. It requires 3.91 times more field multiplications than Alg. 1, excluding precomputation tasks for the table of size 1.7MB. Unfortunately, this option is difficult to activate because one should change the macro variables in the source code and program the modified firmware into the device to activate it. Even the incorrectly modified firmware causes the bootloader to warn.

IV. PROPOSED ATTACK

We have described the structure of the hierarchical deterministic wallets and the elliptic curve cryptographic algorithms implemented by Trezor. This Section proposes a single trace power analysis attack on the ECSM implementation without profiling. We present the SPA attack on the conditional negation (Alg. 2) for extracting 1/4 information of the secret in Subsection IV-A and the HCA attack on the point addition (Alg. 3) for 3/4 information in Subsection IV-B. Then, Subsection IV-C presents the private key reconstruction by combining the two pieces of information.

Algorithm 5 Conditional Variables Extraction

```
Input: Traces $T = \{t_{i,j} | t_{i,j} \in \mathbb{R}, 1 \le i \le n, 1 \le j \le m\}$.
Output: A list $\hat{c}$ that contains $n$ conditions.
 1: $V \leftarrow \mathbb{V}[T]$ along the $i$-axis.
 2: $E \leftarrow \mathbb{E}[T]$ along the $i$-axis.
 3: $count \leftarrow 0$.
 4: for $j \leftarrow 1$ to $m$ do
 5:     if $V_j > \mathbb{E}[V]$ then
 6:         $count \leftarrow count + 1$.
 7:         $poi_{count} \leftarrow j$.
 8:         $threshold \leftarrow threshold + E_j$.
 9: for $i \leftarrow 1$ to $n$ do
10:     $sum \leftarrow 0$.
11:     for $j \leftarrow 1$ to $count$ do
12:         $sum \leftarrow sum + t_{i,poi_j}$.
13:     $\hat{c}_i \leftarrow (sum > threshold)$ ? True : False.
14: return $\hat{c}$.
```

A. EXTRACTING CONDITIONAL VARIABLES

In the conditional negation (Alg. 2), Steps 1 and 2 compute 32-bit masks α and β, respectively. The possible values for the two masks are 0xFFFFFFFF and 0x00000000 depending on the conditional variable c. Their Hamming weights are 32 and 0, showing a very large difference, so the power patterns when $c = 1$ and $c = 0$ will be distinguishable. Consequently, one can estimate the conditions by clustering 64 subtraces the conditional negation produces.

Alg. 5 shows a procedure for estimating conditions $\hat{c}$. By visual inspection, one should cut the subtraces off only to contain m time samples of preparing the mask α for the i-th iteration. Let T be a set of subtraces of $1 \le i \le n = 64$ conditional negations (63 from Step 14 and 1 from Step 17 of Alg. 1). Steps 1 and 2 compute a variance trace V and an average trace E, respectively. Steps 3-8 identify points of interest (POI) that exceed the arithmetic mean of V over times. The POIs indicate where variations in power according to the mask α are considerable and maximize the distinguishability of each subtrace. *count* will be the number of POIs, and *threshold* will be the sum of all the values at POIs. Steps 9-13 make a Boolean list of conditions whether subtraces exceed the *threshold*.

For $1 \le i \le 64$, the output $\hat{c}_i$ implies whether the i-th conditional operation of the ECSM negates the point R. Implicitly the $\hat{c}_{64}$ refers to the last condition outside the loop of the ECSM. The Boolean data may need to be inverted depending on a measurement setup.

Algorithm 6 Table Indices Extraction

```
Input: Traces $O = \{t_{i,j} | t_{i,j} \in \mathbb{R}, 0 \le i < n, 1 \le j \le m\}$ and
    the base point $G$.
Output: A list $\hat{j}$ that contains $n$ indices.
1: $O \leftarrow reshape(O, (n, 162, m'))$    // $n \times 162 \times m'$ array
2: for $i \leftarrow 0$ to $n-1$ do
3:     $T \leftarrow O[i]$
4:     $maxc \leftarrow 0.$
5:     for $k \leftarrow 0$ to 7 do
6:         $((x_8 \cdots x_0)_{2^{29}}, (y_8 \cdots y_0)_{2^{29}}) \leftarrow (2k+1)16^i G.$
7:         $L[0 \ldots 80] \leftarrow HW(\{x_0, x_0, x_1, \cdots, x_7, x_8, x_8\}).$
8:         $L[81 \ldots 161] \leftarrow HW(\{y_0, y_0, y_1, \cdots, y_7, y_8, y_8\}).$
9:         for $j \leftarrow 1$ to $m'$ do
10:            $c \leftarrow r_j(k).$                      // Eq. (2)
11:            if $c > maxc$ then
12:                $c \leftarrow maxc.$
13:                $\hat{j}_i \leftarrow k.$
14: return $\hat{j}$.
```

B. EXTRACTING POINTS OF ADDITION

In the point addition (Alg. 3), Steps 5 and 8 perform field multiplications modulo p_{256} with a precomputed point $T = (x, y)$ given by the ECSM. The Jacobian-to-affine transformation (Step 8 of Alg. 1) also performs the same field multiplications. These 128 field multiplications take the first operand as x or y value of the base point G or its multiples, namely $(2j+1)16^i G$ for $0 \le i \le 63$ and $0 \le j \le 7$ ($J2A$ for $i = 0$ and point additions for $i \ge 1$), which are public domain parameters. Therefore, identifying i by visual inspection and guessing the column index j give intermediate values and the hypothetical power consumption as well.

Let k be a guessed column index for some i. Then, the first operands of two field multiplications (Alg. 4) will be x and y values of the point $(2k+1)16^i G$. The hypothetical power consumption will be the Hamming weights of $\{x_0, x_0, x_1, x_0, x_1, x_2, \cdots, y_6, y_7, y_8, y_7, y_8, y_8\}$ (The order is illustrated in Fig.4) due to the single precision multiplications in Step 4 and 9 of Alg. 4. Let us represent the power model as $L = \{l_0, l_1, \cdots l_{161}\}$ with the same order, whereas the subtraces $T = \{t_0, t_1, \cdots, t_{161}\}$ are attributed to the corresponding single precision multiplications. One can estimate the indices by determining the maximum correlation coefficients of the two sets L and T.

Alg. 6 shows a procedure for estimating indices $\hat{j}$. Let O be a set of subtraces of $n = 64$ point additions (1 in Step 8 and 63 in Step 15 of Alg. 1). Step 1 is a process in which the original trace O, which contains all 14 field multiplications in the point addition (Alg. 3), is selected only two multiplications related to the input, divided into 162 single precision multiplications, and then aligned. Step 3 brings the subtraces of the i-th iteration. Steps 5-13 determine the k consistent with a column index of the i-th iteration among the eight guesses. First, Steps 6-8 obtain intermediate values based on the guess and compute their Hamming weights; We get the power model L. Then, Steps 9-13 compute Pearson correlation coefficients between L and T for all k by using Eq. (2); We get the argument $\hat{j}_i$ at which the correlation is maximized.

For $0 \le i < 64$, the output $\hat{j}_i$ implies that the precomputed point $T[i][\hat{j}_i]$ contributes to the i-th point addition of the ECSM. Exceptionally, the $\hat{j}_0$ is attributed to the Jacobian-to-affine transformation in Step 8 of the ECSM.

Algorithm 7 Private Key Reconstruction

```
Input: The lists $\hat{c}$ and $\hat{j}$ given by Alg. 5 and 6, respectively,
    the group order $n_{256}$
Output: A private key $\hat{k}$
1: $a \leftarrow 2\hat{j}_0 + 1$
2: for $i \leftarrow 1$ to 63 do
3:     if $\hat{c}_i$ then
4:         $a \leftarrow -a$
5:     $a \leftarrow a + (2\hat{j}_i + 1)16^i$
6: if $\hat{c}_{64}$ then
7:     $a \leftarrow -a$
8: return $\hat{k} \leftarrow a \mod n_{256}$
```

C. RECONSTRUCTING PRIVATE KEY

We have extracted the conditions $\hat{c}$ and indices $\hat{j}$ by using Alg. 5 and 6, respectively. Finally, this Subsection presents the method for reconstructing the private key $\hat{k}$ from them, as shown in Alg. 7.

The ECSM Algorithm gives us the idea to recover the temporary scalar a and input scalar k. In Alg. 1, the point addition is done by calling one of the precomputed points according to the table indices i and j. It is possible to deduce that the temporary scalar should be $a = (2j+1)16^i$. Furthermore, the accumulated result will be negated only if the condition is true. This information is reflected in our deduction by inverting the sign of the current scalar according to the conditions.

V. EXPERIMENTS

This section demonstrates our attack with three datasets: simulation, ChipWhisperer, and actual dataset collected from Trezor Model One (hereafter Sim, CW, and Real, respectively). The following subsections describe each dataset's acquisition setup and present experimental results.

A. EXPERIMENTAL DATASET

The first dataset, Sim, is for the simulation study obtained by assuming a specific power model with various noises. We use the linear power model $l = h + \epsilon$ where h is the Hamming weights of intermediate values. To investigate the effect of noise on the attack success rate, we added a white Gaussian noise ϵ of variance σ^2.

The second dataset, CW, is for the proof-of-concept study obtained from the side-channel evaluation system ChipWhisperer [16], as shown in Fig. 5. The device under test is equipped with an STM32F415RGT6 evaluation board with a

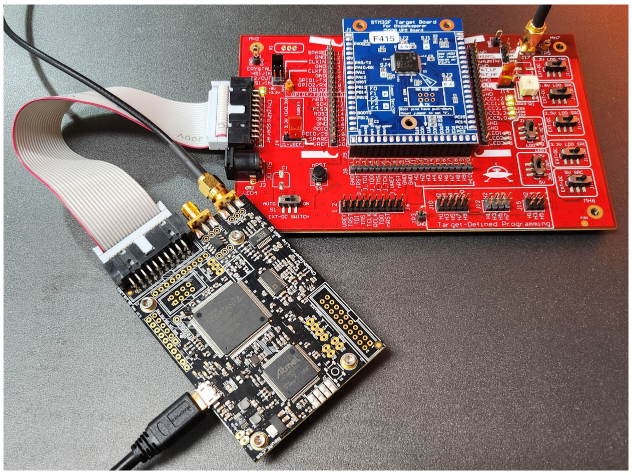


FIGURE 5. EM acquisition setup for the Sim dataset.

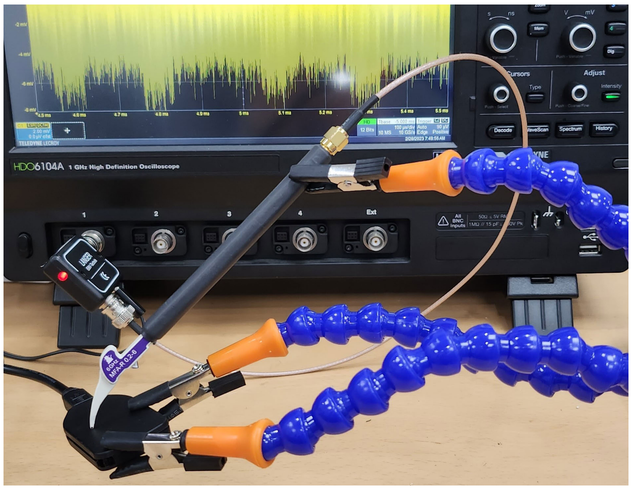


FIGURE 6. Power acquisition setup for the Real dataset.

32-bit ARM Cortex-M4 processor running at 7.37 MHz clock frequency. We acquired power traces using a ChipWhisperer-Lite oscilloscope with a sampling rate of 29.5 MHz, four times the clock frequency.

The third dataset, Real, is obtained from the Trezor Model One hardware wallet, as shown in Fig. 6. The device consists of an STM32F205RGT6 board with a 32-bit ARM Cortex-M3 processor running at 120 MHz, a USB type A port, two physical buttons, and an internal display. We acquired EM traces from the wallet's backside using a LeCroy oscilloscope HDO6104 at 10 GHz and a Langer near-field probe MFA-R 0.2-6 suitable for measuring magnetic fields. We used a hardware low-noise amplifier and a software low-pass filter for noise reduction. We queried 10 times with the same input and averaged the power traces for the same reason. Before averaging them, we applied the elastic alignment based on the dynamic time warping [43] to resolve misalignment due to unstable clocks and random interrupts, as shown in Fig. Furthermore, we utilized a level 2 discrete Haar wavelet transform to emphasize low-frequency components and down-sample the traces.

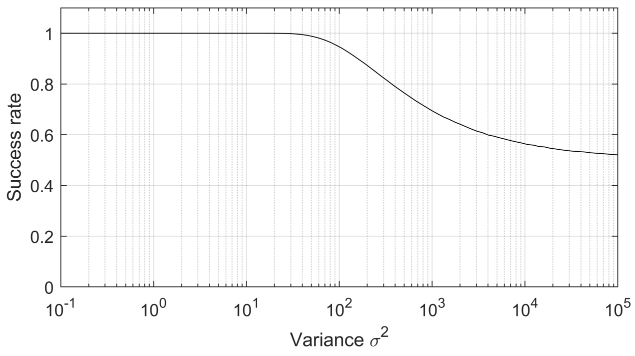


FIGURE 7. Success rate of the conditional variables extraction using the Sim dataset. The attack fails for the first time when $\sigma^2 = 1.52 \times 10^1$.

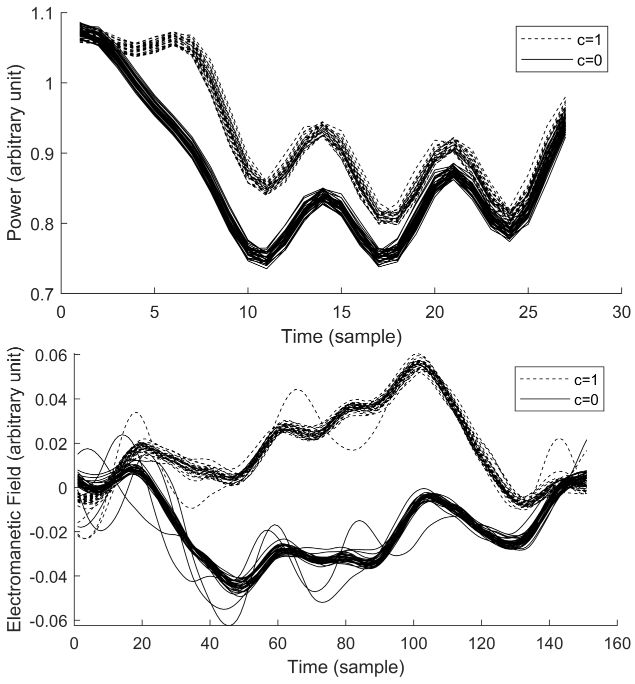


FIGURE 8. Subtraces of the CW (top) and Real (bottom) datasets zoomed in conditional negations.

B. EXPERIMENTAL RESULTS

First, we present the experimental results of the SPA attacks on conditional negation. Figure 7 shows the success rate with the Sim dataset for the various noise variances. The success rate is 100% until the noise variance reaches $\sigma^2 = 15.2$. This is because the Hamming weights of the mask differ by 32, but if the power consumption fluctuates by ± 16 due to noise, there would be a possibility of misclassification. The success rate gradually decreases and converges to 50% because the attack becomes random guesses if the noise worsens.

Figure 8 shows the power and EM subtraces measured when the ChipWhisperer and Trezor Model One perform the conditional negations, respectively. We could extract the

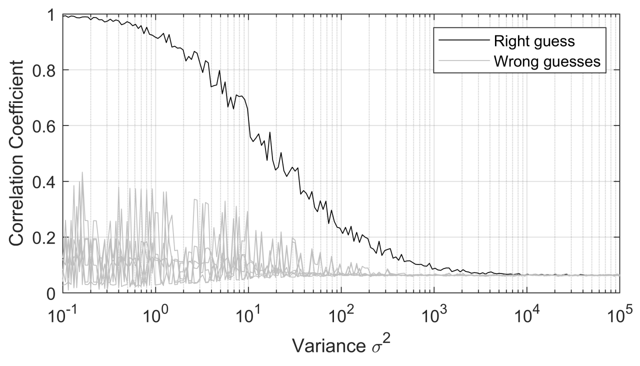


FIGURE 9. Maximum correlation coefficients of the HCA attack on point additions using the Sim dataset. The crossing occurs at $\sigma^2 = 8.8 \times 10^3$.

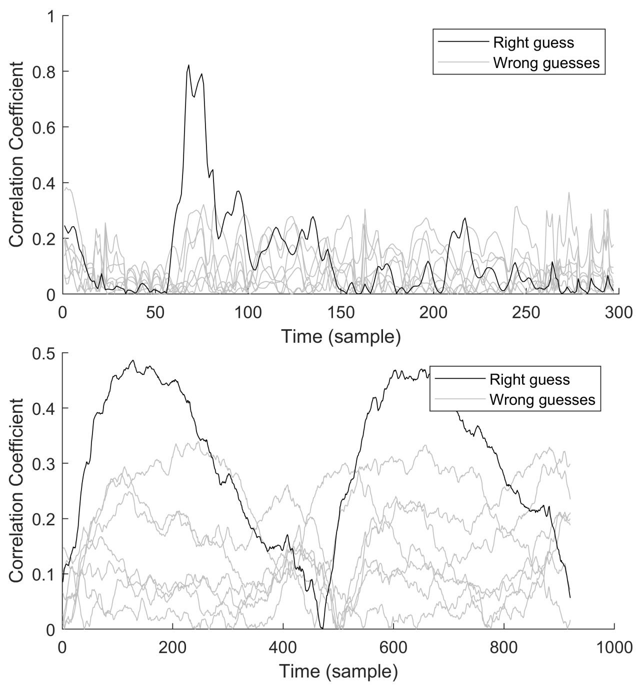


FIGURE 10. Correlation coefficients between power models and traces measured during point additions. Top: the CW dataset. Bottom: the Real dataset.

entire conditional variables c_i with the CW and Real datasets. However, the experiment with the Real dataset shows that the subtraces have unstable patterns, and the difference between the two groups is slight compared with the CW dataset, although various signal processing techniques are applied, and 10 EM traces are averaged.

Next, we present the experimental results of the HCA attacks on point addition. Figure 9 shows the maximum correlation coefficient between the Sim dataset and eight hypothetical power consumption for the various noises. Even at high noise, the correlation coefficient by the right guess (black line) is greater than that by the seven wrong guesses (gray lines). This can be attributed to HCA determining the value with 162 subtraces, whereas SPA determines the value with only one subtrace.

Figure 10 shows the HCA attack results from power and EM subtraces measured when the ChipWhisperer and Trezor Model One perform the point additions, respectively. We could extract the entire table indices j_i with the CW and Real datasets. There are two peaks for the EM subtraces of the Real dataset due to the electromagnetic properties. The first peak is the positive correlation, and the second is the absolute value of the negative correlation. Note that the Real dataset is acquired by averaging 10 EM traces.

VI. DISCUSSIONS

We have shown that one can extract partial information about the temporary scalar a through the proposed non-profiling single trace attack and have also explained how this information can be reconstructed into the input scalar k. We now discuss the impact of the proposed attack on the real world in Subsection VI-A and possible countermeasures against the proposed attack in Subsection VI-B.

A. DAMAGE EXPECTED

If the targeted ECSM was an operation on the root or non-terminal nodes, obtained private key k seems useless from an attacker's point of view since it would not participate in transactions, nor could the private key derive child nodes alone. However, if a chain code of some node is exposed, an attacker can derive all the sub-nodes belonging to that node by using the private key and chain code pair. In practice, some Bitcoin clients provide a so-called extended public key (XPUB) in the form of QR codes containing a chain code as well as BIP version, level, parent's hash, identifier, and public key. Although XPUBs are easy to access and contain the word 'public' in the name, they should be kept secret to prevent attackers from deriving child nodes.

On the other hand, if the targeted ECSM was an operation on the terminal nodes, obtained private key k would directly participate in the transactions. An attacker can claim ownership of the transactions related to the corresponding key. In this case, cryptocurrency theft occurs, so the key must be revoked immediately after the attack.

We emphasize that the proposed attack is more reasonable than previous attacks. The proposed attack neither damages the package nor takes a long time to set up an environment. The only thing required for the attack is a single power trace measured during the ECSM with private keys. To acquire such a power trace, attackers may tempt users to connect to a host device (PC or smartphone) whose measuring equipment is inbuilt in advance. Measuring power via a USB cable that looks completely normal but embeds a measurement circuit is also possible. If attackers consider EM radiation, they can measure the trace sneakingly at a distance.

B. COUNTERMEASURES

The following methods can be considered to prevent our attack. Most fundamentally, wallets should be used on a trusted system to make power traces unobtainable. Conventional methods such as masking and hiding can also be effective as a software or hardware countermeasure.

Masking methods eliminate the relationship between intermediate values and power consumption. Hiding methods increase the attack complexity by increasing the noise level. Finally, the alternative ECSM algorithm provided by the Trezor wallet (see the last paragraph of Subsection III-B) makes guessing intermediate values infeasible.

VII. CONCLUSION

This paper presented a non-profiled single trace attack on hardware wallets that are designed following the standard structure as described in BIP documents. The proposed attack extracts private keys used in elliptic curve scalar multiplication, one of the most critical secrets in hardware wallets. Countermeasures against our attack are necessary since the user's private key may fall into the attacker's hands, resulting in cryptocurrency theft.

It is worth noting that our attack can be improved by replacing signal processing or clustering algorithms with advanced ones. One noticeable example is a machine learning-based algorithm. In [44], they introduced a non-profiled autoencoder model for noise reduction and trace alignment. Discovering a novel side channel for acquiring traces is another interesting future work. Finally, suggesting a method to extract chain codes will synergize our attack, disclosing more severe vulnerabilities of wallet cloning.

REFERENCES

[1] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Nov. 16, 2022. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] F. A. Aponte-Novoa, A. L. S. Orozco, R. Villanueva-Polanco, and P. Wightman, "The 51% attack on blockchains: A mining behavior study," *IEEE Access*, vol. 9, pp. 140549–140564, 2021.

[3] A. Lazarenko and S. Avdoshin, "Financial risks of the blockchain industry: A survey of cyberattacks," in *Proc. Future Technol. Conf. (FTC)*, K. Arai, R. Bhatia, and S. Kapoor, Eds. Cham, Switzerland: Springer, 2019, pp. 368–384.

[4] M. Bartoletti, S. Lande, A. Loddo, L. Pompianu, and S. Serusi, "Cryptocurrency scams: Analysis and perspectives," *IEEE Access*, vol. 9, pp. 148353–148373, 2021.

[5] S. Suratkar, M. Shirole, and S. Bhirud, "Cryptocurrency wallet: A review," in *Proc. 4th Int. Conf. Comput., Commun. Signal Process. (ICCCSP)*, Sep. 2020, pp. 1–7.

[6] M. Arapinis, A. Gkaniatsou, D. Karakostas, and A. Kiayias, "A formal treatment of hardware wallets," in *Financial Cryptography and Data Security*, I. Goldberg and T. Moore, Eds. Cham, Switzerland: Springer, 2019, pp. 426–445.

[7] M. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Cryptography*, vol. 4, no. 2, p. 15, May 2020. [Online]. Available: https://www.mdpi.com/2410-387X/4/2/15

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

[9] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems—CHES 2004*, M. Joye and J.-J. Quisquater, Eds. Berlin, Germany: Springer, 2004, pp. 16–29.

[10] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems—CHES 2001*, Ç. K. Koç, D. Naccache, and C. Paar, Eds. Berlin, Germany: Springer, 2001, pp. 251–261.

[11] Z. Kazemi, A. Papadimitriou, I. Souvatzoglou, E. Aerabi, M. M. Ahmed, D. Hely, and V. Beroulle, "On a low cost fault injection framework for security assessment of cyber-physical systems: Clock glitch attacks," in *Proc. IEEE 4th Int. Verification Secur. Workshop (IVSW)*, Jul. 2019, pp. 7–12.

[12] C. Bozzato, R. Focardi, and F. Palmarini, "Shaping the glitch: Optimizing voltage fault injection attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 2, pp. 199–224, Feb. 2019. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/7390

[13] M. S. Kelly and K. Mayes, "High precision laser fault injection using low-cost components," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Dec. 2020, pp. 219–228.

[14] M. A. Elmohr, H. Liao, and C. H. Gebotys, "EM fault injection on ARM and RISC-V," in *Proc. 21st Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2020, pp. 206–212.

[15] D. Nedospasov, J. Datko, and T. Roth. (2018). *Wallet.Fail*. Accessed: Nov. 16, 2022. [Online]. Available: https://wallet.fail

[16] C. O'Flynn, "MIN()imum failure: EMFI attacks against USB stacks," in *Proc. 13th USENIX Workshop Offensive Technol. (WOOT)*. Santa Clara, CA, USA: USENIX Association, Aug. 2019, pp. 1–10. [Online]. Available: https://www.usenix.org/conference/woot19/presentation/oflynn

[17] M. S. Pedro, V. Servant, and C. Guillemet, "Side-channel assessment of open source hardware wallets," Cryptol. ePrint Arch., Tech. Rep. 2019/401, 2019. [Online]. Available: https://eprint.iacr.org/2019/401

[18] T. Dzetkulic, P. Rusnak, and J. Hoenicke. *Trezor Firmware*. Accessed: Nov. 16, 2022. [Online]. Available: https://github.com/trezor/trezor-firmware/blob/master/crypto/ecdsa.c#L537

[19] P. Wuille et al. (Feb. 2012). *Hierarchical Deterministic Wallets*. Accessed: Nov. 16, 2022. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

[20] M. Palatinus et al. (Apr. 2014). *Purpose Field for Deterministic Wallets*. Accessed: Nov. 16, 2022. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0043.mediawiki

[21] (Apr. 2014). *Multi-Account Hierarchy for Deterministic Wallets*. Accessed: Nov. 16, 2022. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki

[22] D. Weigl et al. (May 2016). *Derivation Scheme for P2WPKH-Nested-in-P2SH Based Accounts*. Accessed: Nov. 16, 2022. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0049.mediawiki

[23] P. Rusnak et al. (Dec. 2017) *Derivation Scheme for P2WPKH Based Accounts*. Accessed: Nov. 16, 2022. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0084.mediawiki

[24] D. R. Brown. (Jan. 2010). Sec 2: Recommended elliptic curve domain parameters. Standards for Efficient Cryptography (SEC). Accessed: Nov. 16, 2022. [Online]. Available: https://www.secg.org/sec2-v2.pdf

[25] B.-Y. Sim and D.-G. Han, "Key bit-dependent attack on protected PKC using a single trace," in *Information Security Practice and Experience*, J. K. Liu and P. Samarati, Eds. Cham, Switzerland: Springer, 2017, pp. 168–185.

[26] D. Park, G. Kim, D. Heo, S. Kim, H. Kim, and S. Hong, "Single trace side-channel attack on key reconciliation in quantum key distribution system and its efficient countermeasures," *ICT Exp.*, vol. 7, no. 1, pp. 36–40, Mar. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959521000138

[27] M. J. Dworkin, E. B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, and J. F. Dray Jr., "Advanced encryption standards (AES)," Federal Inf. Process. Standards (FIPS), Gaithersburg, MD, USA, Tech. Rep., Nov. 2001. Accessed: Nov. 16, 2022, doi: 10.6028/NIST.FIPS.197.pdf.

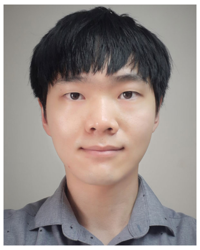
[28] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, and V. Verneuil, "Horizontal correlation analysis on exponentiation," in *Information and Communications Security*, M. Soriano, S. Qing, and J. López, Eds. Berlin, Germany: Springer, 2010, pp. 46–61.

[29] A. Regenschei and D. Moody, "Digital signature standard (DSS)," Federal Inf. Process. Standards (FIPS), Gaithersburg, MD, USA, Tech. Rep., Feb. 2023. Accessed: Nov. 16, 2022, doi: 10.6028/NIST.FIPS.186-5.pdf.

[30] D. Park, S. Lee, S. Cho, H. Kim, and S. Hong, "An improved horizontal correlation analysis using collision characteristics on lookup table based scalar multiplication algorithms," *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 30, no. 2, pp. 179–187, 2020.

[31] S. Jin, S. Lee, S. M. Cho, H. Kim, and S. Hong, "Novel key recovery attack on secure ECDSA implementation by exploiting collisions between unknown entries," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, pp. 1–26, Aug. 2021. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/9058

[32] S. Jin, S. M. Cho, H. Kim, and S. Hong, "Enhanced side-channel analysis on ECDSA employing fixed-base comb method," *IEEE Trans. Comput.*, vol. 71, no. 9, pp. 2341–2350, Sep. 2022.
[33] N. Lee, S. Hong, and H. Kim, "Single-trace attack using one-shot learning with Siamese network in non-profiled setting," *IEEE Access*, vol. 10, pp. 60778–60789, 2022.
[34] A. P. Fournaris, L. Papachristodoulou, L. Batina, and N. Sklavos, "Residue number system as a side channel and fault injection attack countermeasure in elliptic curve cryptography," in *Proc. Int. Conf. Design Technol. Integr. Syst. Nanosc. Era (DTIS)*, Apr. 2016, pp. 1–4.
[35] J. Dubeuf, D. Hely, and V. Beroulle, "Enhanced elliptic curve scalar multiplication secure against side channel attacks and safe errors," in *Constructive Side-Channel Analysis and Secure Design*, S. Guilley, Ed. Cham, Switzerland: Springer, 2017, pp. 65–82.
[36] L. Weissbart, S. Picek, and L. Batina, "One trace is all it takes: Machine learning-based side-channel attack on EdDSA," in *Security, Privacy, and Applied Cryptography Engineering*, S. Bhasin, A. Mendelson, and M. Nandi, Eds. Cham, Switzerland: Springer, 2019, pp. 86–105.
[37] S. Belaïd and M. Rivain, "High order side-channel security for elliptic-curve implementations," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2023, no. 1, pp. 238–276, Nov. 2022. Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/9952
[38] K. Okeya and T. Takagi, "The width-*w* NAF method provides small memory and fast elliptic scalar multiplications secure against side channel attacks," in *Topics in Cryptology—CT-RSA 2003*, M. Joye, Ed. Berlin, Germany: Springer, 2003, pp. 328–343.
[39] R. R. Goundar, M. Joye, A. Miyaji, M. Rivain, and A. Venelli, "Scalar multiplication on Weierstraß elliptic curves from Co-Z arithmetic," *J. Cryptograph. Eng.*, vol. 1, no. 2, pp. 161–176, Aug. 2011.
[40] D. Basu Roy and D. Mukhopadhyay, "High-speed implementation of ECC scalar multiplication in GF(p) for generic Montgomery curves," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1587–1600, Jul. 2019.
[41] P. Choi, M.-K. Lee, and D. K. Kim, "ECC coprocessor over a NIST prime field using fast partial Montgomery reduction," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 3, pp. 1206–1216, Mar. 2021.
[42] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer, 2006.
[43] J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker, "Improving differential power analysis by elastic alignment," in *Topics in Cryptology—CT-RSA 2011*, A. Kiayias, Ed. Berlin, Germany: Springer, 2011, pp. 104–119.
[44] D. Kwon, H. Kim, and S. Hong, "Non-profiled deep learning-based side-channel preprocessing with autoencoders," *IEEE Access*, vol. 9, pp. 57692–57703, 2021.



DONGJUN PARK received the B.S. degree in information security from Sejong University, Seoul, South Korea, in 2018, and the M.S. degree in information security from the Korea University, Seoul, in 2020, where he is currently pursuing the Ph.D. degree. Since 2018, he has been a Research Assistant with the Institute of Cyber Security and Privacy (ICSP), School of Cyber Security (SCS), Korea University. His research interests include cryptography, hardware security, and side-channel attacks.

MINSIG CHOI received the B.S. degree in information security from Hongik University, Seoul, South Korea, in 2023. He is currently pursuing the M.S. degree in information security with Korea University, Seoul. Since 2023, he has been a Research Assistant with the Institute of Cyber Security and Privacy (ICSP), School of Cyber Security (SCS), Korea University. His research interests include cryptography and side-channel attacks.

GYUSANG KIM received the B.S. degree in mathematics from Yonsei University, Seoul, South Korea, in 2020. He is currently pursuing the joint M.S. and Ph.D. degrees in information security with Korea University, Seoul. His research interests include cryptography, post-quantum cryptography, and side-channel attacks.

DAEHYEON BAE received the B.S. and M.S. degrees in information security engineering from Hoseo University, South Korea, in 2021 and 2022, respectively. He is currently pursuing the Ph.D. degree with the School of Cybersecurity (SCS), Korea University, Seoul, South Korea. Since 2022, he has been a Research Assistant with the Institute of Cyber Security and Privacy (ICSP), School of Cyber Security (SCS), Korea University. His research interests include side-channel attacks, hardware security, and machine learning-based cryptanalysis.

HEESEOK KIM (Member, IEEE) received the B.S. degree in mathematics from Yonsei University, Seoul, South Korea, in 2006, and the M.S. and Ph.D. degrees in engineering and information security from Korea University, Seoul, in 2008 and 2011, respectively. He was a Postdoctoral Researcher with the University of Bristol, U.K., from 2011 to 2012. From 2013 to 2016, he was a Senior Researcher with the Korea Institute of Science and Technology Information (KISTI). Since 2016, he has been with Korea University. His research interests include side-channel attacks, cryptography, and network security.

SEOKHIE HONG (Member, IEEE) received the M.S. and Ph.D. degrees in mathematics from Korea University, in 1997 and 2001, respectively. From 2000 to 2004, he was with Security Technologies Inc. From 2004 to 2005, he was a Postdoctoral Researcher with the COSIC, KU Leuven, Belgium. He joined the Graduate School of Cyber Security, Korea University. His research interests include cryptography, public and symmetric key cryptosystems, hash functions, and message authentication codes.