Efficient Noninvasive Fault Injection Method Utilizing Intentional Electromagnetic Interference

Hikaru Nishiyama, Daisuke Fujimoto, *Senior Member, IEEE*, Hideaki Sone, *Life Member, IEEE*, and Yuichi Hayashi, *Senior Member, IEEE*

***Abstract*—In the fault injection method, an electromagnetic (EM) wave is injected to temporarily cause a fault at a specific time of the encryption process, the faulty outputs are obtained from the cryptographic device, and the secret key is extracted by analyzing the faulty outputs. In the conventional method, the intentional electromagnetic interference (IEMI) wave is injected at a random time because it is difficult to obtain information on the start time of the encryption process. Thus, a cryptographic module must execute a large number of encryption trials before the occurrence of a fault that enables the secret key to be extracted. In this article, we propose a fault injection method that can generate the faults at a specific time with high probability, which is like the method of injecting an IEMI wave synchronized with the start time of the encryption process. The proposed method inserts glitches into the encryption process at fixed times by injecting a continuous sinusoidal wave of a specific frequency while controlling the amplitude and phase. This generates faults required for the secret key analysis method with a high probability even when the start time of the encryption process cannot be obtained. We experimentally demonstrate the impact of the aforementioned IEMI using the advanced encryption standard, which is an ISO/IEC 18033 block cipher, implemented as a module on a standard evaluation board. The conventional method requires more than 30 000 encryption processes to obtain the secret key. In contrast, the results indicate that we can obtain the secret key with approximately 22 encryption processes which is almost three orders of magnitude less than that with the conventional method. This confirms that secret keys can be extracted in a brief period of time. Moreover, devices previously excluded from IEMI-based fault injection because they can only be accessed for a brief period because their physical access was surveilled, may now be the target of the threat.**

***Index Terms*—Cryptographic devices, electromagnetic information security, fault injection method, intentional electromagnetic interference (IEMI).**

I. Introduction

INTENTIONAL electromagnetic interference (IEMI) poses a threat to cryptographic devices. This threat has been studied as a fault injection attack wherein hardware-level vulnerabilities in the cryptographic device can be exploited to steal information about the secret key [1], [2], [3], [4], [5], [6], [7], [8]. The attack requires an injection method that induces a temporary computational fault in the operation of the cryptographic module, and an analysis method that can extract the secret key from the faulty output obtained. Based on these requirements, existing research include theoretical methods that hypothesize the occurrence of a fault in a specific intermediate process of the cryptographic algorithm, they then use the faulty output to estimate the secret key [9], [10], [11], [12], [13], [14], [15].

Research focused on injection methods has investigated specific approaches for generating faults required for secret key estimation in real cryptographic modules. For example, IEMI-based fault injection methods propagate an IEMI wave to the clock supply circuit [2], [3] or the power supply circuit [4], [5] to cause momentary fluctuations (glitches) that generate setup-time violation faults. These methods require physical access to the device to obtain information on the start time of the encryption process so that the fault injection time can be precisely synchronized with the former. A fault can occur only at a specific time. However, these fault injection methods may not work under conditions where physical access is difficult, such as when the cryptographic module has a protective mechanism, such as tamper detection or enclosure physical protection.

In contrast, noninvasive IEMI-based fault injection methods from outside the cryptographic device has been investigated by generating clock glitches with a continuous sinusoidal wave injected via the power cable of the device [6], [7], [8]. In [6], the attacker does not have direct access to the device and cannot obtain the start time of the encryption process. Therefore, a continuous sinusoidal wave is injected without synchronization with the start time of the encryption process. In this case, clock glitches occur at random times; therefore, the secret key may be obtained after an enormous number of encryption trials.

This article proposes a fault injection method that can obtain the secret key after a small number of encryption trials, which is similar to when an attacker with access to the device injects an IEMI wave synchronized with the start time of the encryption process. The proposed method focuses on the relationship between the clock frequency of the cryptographic module and the frequency of the injected continuous sinusoidal wave. The times at which clock glitches occur during the encryption process can be kept constant by setting the frequency of the injected continuous sinusoidal wave to an integer multiple of the clock

Manuscript received 25 October 2022; revised 10 February 2023 and 23 March 2023; accepted 24 March 2023. Date of publication 18 April 2023; date of current version 15 August 2023. This work was supported in part by JSPS KAKENHI under Grant JP19H01104, Grant JP21K19772, and Grant JP21K03995, and in part by JST FOREST Program under Grant JPMJFR206L. *(Corresponding author: Yuichi Hayashi.)*

Hikaru Nishiyama, Daisuke Fujimoto, and Yuichi Hayashi are with the Division of Information Science, Nara Institute of Science and Technology, Ikoma 630-0192, Japan (e-mail: nishiyama.hikaru.na1@is.naist.jp; fujimoto@is.naist.jp; yu-ichi@is.naist.jp).

Hideaki Sone is with Tohoku University, Sendai 980-8577, Japan (e-mail: sone@tohoku.ac.jp).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TEMC.2023.3264586.

Digital Object Identifier 10.1109/TEMC.2023.3264586

frequency. Under these conditions, a fault can be generated during a particular encryption operation by varying the phase and amplitude of the continuous sinusoidal wave to control the occurrence of clock glitches. Consequently, the secret key can be obtained after a small number of encryption trials.

The contributions of this article to the study of the above threats are listed as follows.

1) This article proposes and verifies the feasibility of a method to estimate the clock frequency from the device's conducted emissions and determine the frequency of the sinusoidal wave injected by the attacker corresponding to the estimated clock frequency.

2) The proposed method enables fault injection only at a specific time synchronized with the start time of the encryption process.

3) Focusing on the occurrences of faulty ciphertext after fault injection into the device, we propose a method to obtain ciphertext that can be applicable to secret key analysis. This allows the secret key to be obtained faster than in conventional attacks, suggesting the possibility of a noninvasive IEMI-based fault injection method against cryptographic devices that can be accessed only for a brief period because their physical access is limited or surveilled.

4) Fault injection, which is noninvasive to the device and performed at a specific time synchronized with the start time of the encryption process, is highly dependent on the clock rise time. Therefore, we proposed a concept for a countermeasure method that focuses on the clock rise time and considers both security and EMC.

The rest of this article is organized as follows. In Section II, we describe the mechanism of fault generation by continuous sinusoidal wave injection. Subsequently, we propose a noninvasive IEMI-based fault injection method that enables the attacker to generate a fault at a specific time with a high probability of success, even when the start time of the encryption process is not synchronized with the start time of fault injection. In Section III, we provide the experimental validation that demonstrates the effectiveness of the proposed method. Moreover, we describe a concept of effective countermeasures against the proposed method. Finally, Section IV concludes the article.

II. Proposed Method

Fig. 1(a) shows the conceptual diagram of noninvasive IEMI-based fault injection method from outside the device. This method considers the transfer efficiencies of EM waves from the injection point to each module mounted on the device, and injects a continuous sinusoidal wave with a frequency that maximizes the transfer efficiency to the cryptographic module. This can generate a fault only in the cryptographic module without affecting other modules. However, if the continuous sinusoidal wave is injected without synchronization with the start time of the encryption process, clock glitches occur at random times. Consequently, faults only at a specific time necessary for key extraction are rarely generated.

In this article, the frequency of the continuous sinusoidal wave to be injected is selected by considering not only the transfer efficiency but also the clock frequency, which determines the start time of the encryption process, as shown in Fig. 1(b). The times at which clock glitches occur can be controlled and faults can be generated at regular intervals by varying the phase of the continuous sinusoidal wave under these conditions, this enables us to generate the faults necessary to obtain the secret key with a high probability in a noninvasive manner.

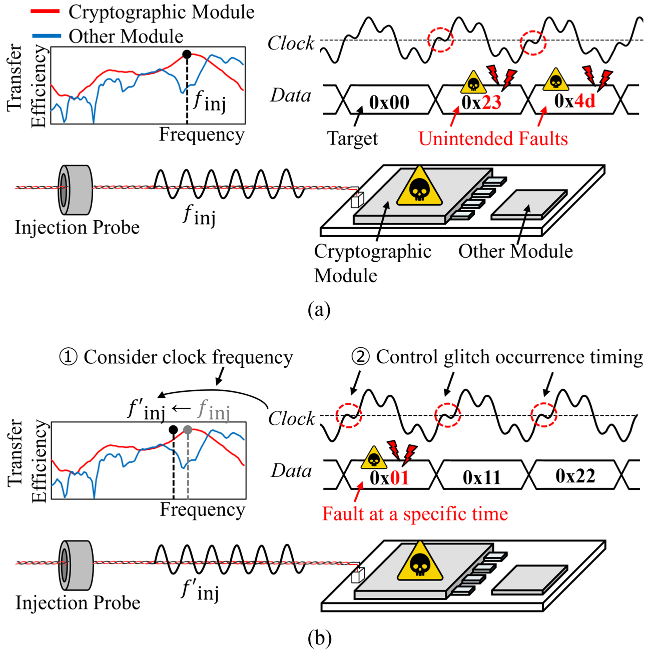


Fig. 1. Noninvasive IEMI-based fault injection method from outside the device. (a) Conventional method presented in [6]. (b) Method presented in this article. By applying the proposed method, the attacker can generate faulty ciphertexts necessary for the secret key analysis with high probability.

A. Fault Generation by Continuous Sinusoidal Wave Injection

Like in [6], the evaluation platform and target for injecting faults in this work entails a circuit with a loop architecture, which is a common implementation scheme for symmetric key cryptography. In the loop architecture, a unit process for block ciphers called round is implemented in a combinational circuit that operates repeatedly in synchronization with the rising clock edge.

Fig. 2(a) shows a conceptual diagram of the relationship between the clock signal and the time to complete the process called the data path delay time. Round R_i begins when the rising clock edge exceeds the threshold voltage V_{TH}. At this time, the delay time for each data path depends on the input value to the combinational circuit, i.e., intermediate value to the round.

To successfully execute a round, the longest data path delay (the critical path delay, t_{cp}) must be less than or equal to the clock cycle (t_{clk}). If the condition is not satisfied, a fault will be generated due to a setup time violation [16]. The device is designed such that t_{clk} is given a sufficient margin with respect to t_{cp} to account for the increase in the delay time attributed to temperature and other factors.

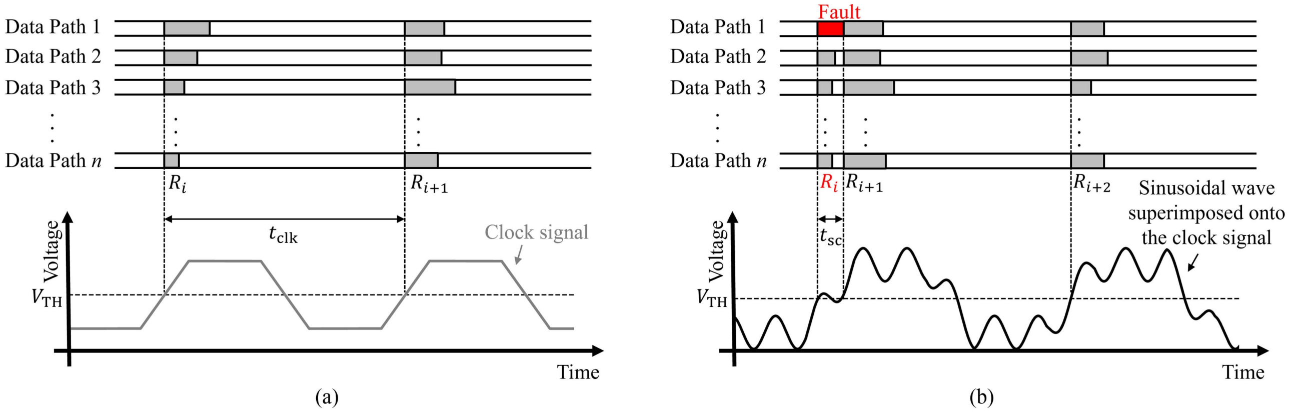


Fig. 2. (a) Relationship between the rise time of clock signal and processing time of a single-round. (b) One-byte fault occurrence generated by executing irregular round operation.

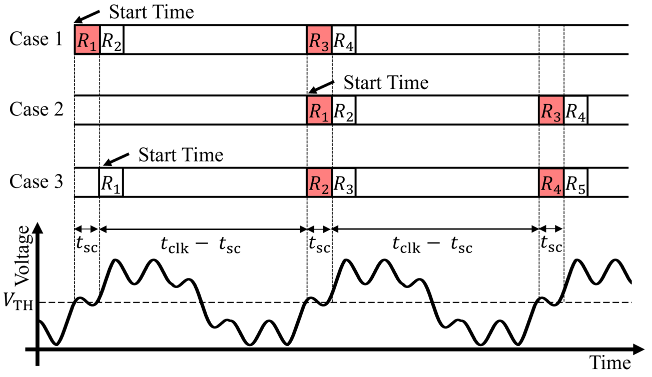


Fig. 3. Relationships between the start time of the encryption process and overclock occurrence time. As shown in Cases 1 and 2, fault occurrence rounds become independent of the start time of the encryption process when the injected frequency is set to an integer multiple of the clock frequency.

Fig. 2(b) shows that a clock glitch causes an irregular rising edge in addition to the original rising clock edge when the clock glitch is superimposed near V_{TH}. This causes the irregular execution of the next round R_{i+1} in addition to the current round R_i. At this time, if t_{cp} in R_i is longer than the clock cycle shortened by the clock glitch t_{sc}, the process of R_{i+1} can be executed before the process of R_i completes, which generates a setup-time violation fault.

B. Proposed Noninvasive Fault Injection Method

In this section, we describe the proposed method in detail. Most of the existing research on the analysis methods assume that a fault is generated only in a specific single-round of the encryption process. In this article, we focus on the single-round fault as the fault required for secret key estimation.

We injected a continuous sinusoidal wave with a frequency that is an integer multiple of the clock frequency. This can keep the occurrence time of the clock glitches at each rising edge constant. Therefore, as illustrated in Fig. 3, the length of t_{sc} becomes constant. As shown in Cases 1 and 2 in Fig. 3, the round in which the fault occurs is the same regardless of which clock is synchronized with the encryption process.

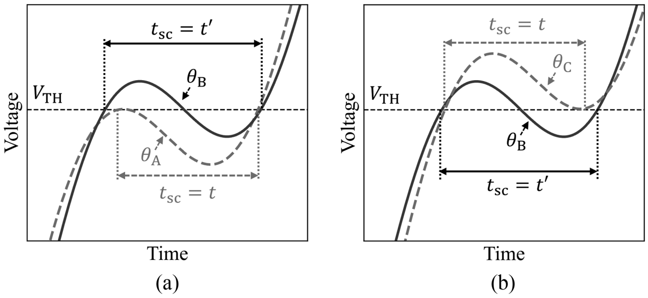


Fig. 4. Sinusoidal waves superimposed onto the clock signals with three different phases. (a) Length of t_{sc} increases with the controlling the phase. (b) Length of t_{sc} decreases with the controlling the phase.

In this method, the round at which the fault occurs will vary depending on which rising clock edge is synchronized with the start time of the encryption process, the original one or the false one induced by the glitch. For example, if the start time of the encryption process is synchronized with the first rising edge, overclocking occurs in an odd-numbered round, as indicated in Cases 1 and 2. The overclocking occurs in an even-numbered round if the start time of the encryption process is synchronized with the second rising edge, as shown in Case 3. Furthermore, the length of t_{sc} is significantly shorter than the time interval from the second rising edge to the first rising edge of the next clock ($t_{\mathrm{clk}} - t_{\mathrm{sc}}$) because the device is designed so that t_{clk} is given a sufficient margin with respect to t_{cp}. No fault is expected to occur in the round processing executed in synchronization with the second rising edge.

In addition, the attacker needs to estimate the clock frequency from outside the device for executing the method described above. The clock frequency can be estimated from outside the device by observing the EM radiation from the device and assuming the clock signal is the noise source because the clock is the primary source of noise emitted from the device [17].

Next, we describe the proposed method of controlling the length of t_{sc}, which involves controlling the phase of the continuous sinusoidal wave. Fig. 4 displays an image of the rising clock edge when three continuous sinusoidal waves with different

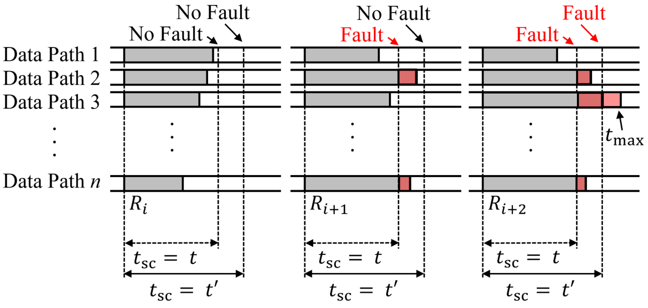


Fig. 5. Variation in the number of rounds in which a fault occurs by controlling the phase of the continuous sinusoidal wave. Single-round faults can be generated by setting t_{sc} short for only the longest data path delay (t_{max}) because the length of t_{cp} varies for each round operation.

phases are injected. In Fig. 4(a), the case of θ_A, the clock glitch exceeds the threshold and overclocking occurs. In this state, the length of t_{sc} is minimized ($t_{sc} = t$). The length of t_{sc} increases until the phase reaches θ_B when the phase is controlled from this state and when the length of t_{sc} is maximized ($t_{sc} = t'$). Then, the length of t_{sc} decreases until the phase reaches θ_C shown in Fig. 4(b), when the phase is controlled from state θ_B and when the length of t_{sc} is again minimized ($t_{sc} = t$). No fault occurs in the subsequent phases.

Fig. 5 shows an image of the variation in the number of rounds in which a fault occurs when the length of t_{sc} is increased by phase control. The length of t_{cp} varies from round to round even if the length of t_{sc} is constant for each rising edge, and therefore, whether a fault occurs depends on the round. In the example shown in Fig. 5, t_{cp} in byte 3 of R_{i+2} has the longest data path delay of all operations (assuming that this delay time is t_{max}), and therefore, it is possible to generate a fault only in a single-round by setting t_{sc} to generate faults only for operations executed at this time t_{max}. However, the round and byte in which the fault occurs will be randomized because the round where t_{max} is observed and the length of t_{max} depends on the input value. The intermediate value of the encryption process and output ciphertext exhibit uniformity because of the nature of the algorithm when the random plaintext is input. Therefore, we assume that the occurrence probability of the round with t_{max} is uniform and its length of t_{max} is normally distributed [18].

The above information cannot be obtained when executing the proposed method from outside the device, and therefore, a technique for identifying the phase at which a single-round fault occurs with a high probability without knowing this information is described below.

Fig. 6 shows the relationships between variation in the length of t_{sc}, the total number of fault occurrences, and the number of single-round fault occurrences for each phase of a continuous sinusoidal wave. Here, t_{sc}= 0 indicates that there is no glitch, and N indicates the number of trials of the encryption trials. The attacker can observe only information about whether a fault has occurred from the output ciphertext, however, the attacker cannot determine whether those faults occurred in a single-round.

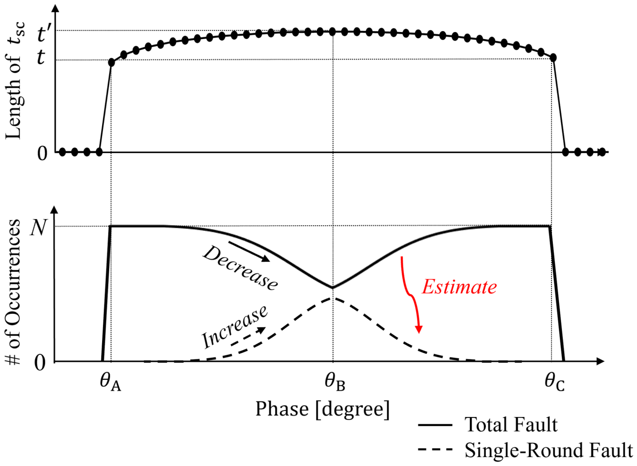


Fig. 6. Relationships between variation in the length of t_{sc}, the total number of fault occurrences, and the number of single-round fault occurrences by controlling the phase of the continuous sinusoidal wave.

When the phase of the injected continuous sinusoidal wave is θ_A or θ_C in Fig. 4, t_{sc} is shorter than t_{cp} in each round, and multiple-round faults may occur in all encryption processes. If phase is controlled from this state and t_{sc} is increased, t_{max} and t_{sc} reach the value as in the state shown in Fig. 5. At this time, the number of rounds in which faults occur in an encryption process is decreased, and the number of single-round fault occurrences will begin to increase. If t_{sc} is further increased, the number of encryption processes without fault increases by satisfying $t_{max} < t_{sc}$, and the total number of fault occurrences will begin to decrease. The attacker can estimate the length of t_{sc} that generates a single-round fault with high probability by observing this variation.

The range of lengths that t_{sc} can assume by means of the phase control illustrated in Fig. 6 depends on the injected frequency, which can satisfy the condition of being an integer multiple of the clock frequency, and on the value of the transfer efficiency to the cryptographic module. Therefore, the injected frequency can be modified to estimate the aforementioned value t_{sc} if the range of lengths that t_{sc} can assume by means of the phase control does not include cases where a single-round fault occurs.

III. Experimental Validation

The experiment in this section, like that reported in previous studies [6], [7], [8], targets the advanced encryption standard (AES) [19], which is a widely used symmetric-key cryptographic algorithm. In the experiment, we first demonstrate that the clock frequency at which the cryptographic module implementing AES operates can be estimated from outside the device, and that a frequency that is an integer multiple of the clock frequency can be injected such that the occurrence time of clock glitches can be kept constant for each clock signal. Subsequently, we conducted a fault injection experiment using the proposed method. Using differential fault analysis (DFA) [10], which is a typical secret key analysis method for symmetric-key cryptography, we verify that the faults required for DFA can be generated with a higher probability than that reported in [6].

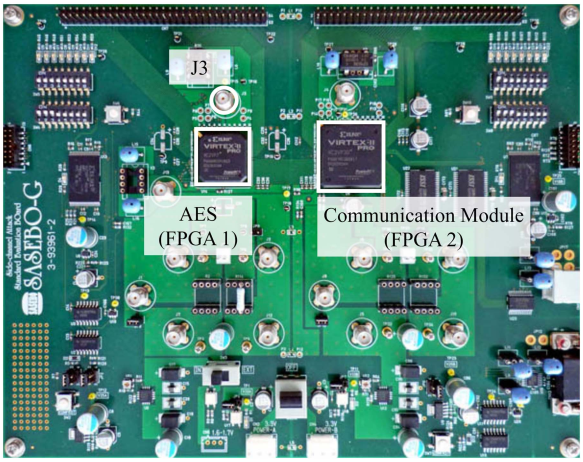


Fig. 7. Actual photo of SASEBO-G.

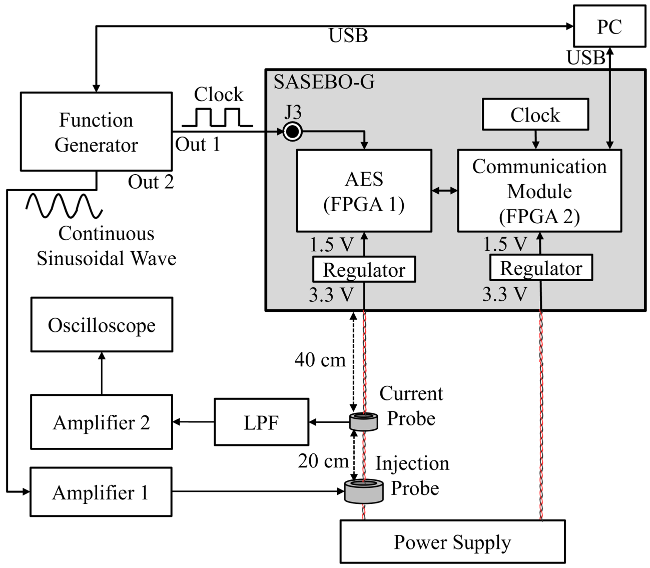


Fig. 8. Block diagram of the experimental setup.

TABLE I
EQUIPMENT USED IN THE EXPERIMENT

Function Generator	Keysight 81160A
Amplifier 1	R&K A000110-4040-R
Amplifier 2	R&K LA101-0S
Low-pass Filter	Mini-Circuits VLF-225+ (DC-225 MHz)
Oscilloscope	Keysight DSOX3054A
Injection Probe	FCC F-140 (100 kHz to 1.3 GHz)
Current Probe	FCC F-2000 (10 MHz to 3 GHz)

A. *Experimental Setup*

Fig. 7 displays a photo of the test board with the built-in cryptographic module used in this experiment called the side-channel attack standard evaluation board (SASEBO-G) [20]. The SASEBO-G is equipped with two field-programmable gate arrays (FPGAs) denoted as FPGA 1 and FPGA 2. A composite-field S-Box AES [21] cryptographic module is implemented in FPGA 1, whereas a module for communication between the PC and FPGA 1 is implemented in FPGA 2. Next, the reference value of secret key is (0x2b7e151628aed2a6abf7158809cf4f3c) as given in the algorithm specification [19].

Fig. 8 shows a block diagram of the experimental setup, and Table I lists all the equipment used in the experiment. For FPGA 1, on which the cryptographic module is mounted, the clock is supplied from the port "Out 1" of the function generator through the SMA port to supply clock signal (J3) for improving the reproducibility of the experiment. For FPGA 2, the clock signal is supplied from a crystal oscillator asynchronously with the clock signal supplied to FPGA1. The clock frequency was set to 24 MHz for both FPGA 1 and FPGA 2, as in previous studies [6], [7], [8]. The clock signal transmitted via the power cable is observed with an oscilloscope after it is passed through a low pass filter to eliminate unintended external noise. It is then amplified by Amplifier 2. The continuous sinusoidal wave to be injected is generated the port "Out 2" of the same individual function generator that supplies the clock signal to FPGA 1. After amplification using Amplifier 1, the signal was injected into the power cable connected to FPGA 1 via an injection probe placed 600 mm away from SASEBO-G. The frequency of the continuous sinusoidal wave starts at 48 MHz, which is twice the clock frequency, and it was increased in 24 MHz increments until a frequency at which the variation in the number of occurrences of faults can be observed adequately. The amplitude was set to a starting value of 0.10 V_{pp}, at which no fault was observed in the output of the FPGA 1. It was then increased until the occurrence of a fault could be confirmed. The phase was swept in 1° increments over a 360° range and the encryption process was performed 200 times for each phase. In an actual attack scenario, a randomly generated input value will be used, and therefore the input plaintext value differs for each encryption process. However, we generated the plaintext dataset in advance and used the same dataset for each phase to prevent the change of experimental conditions from one phase trial to another.

B. *Observation of Clock Signal From Outside the Device*

Here, we first demonstrate that the clock frequency of the cryptographic module can be estimated from outside the device using a current probe to observe the clock signal transmitted via the power cable. Moreover, we show that the occurrence time of the clock glitch can be kept constant.

Fig. 9(a) shows the clock signal supplied from the function generator to J3, and Fig. 9(b) illustrates the clock signal observed from the power cable. A comparison of Fig. 9(a) and (b) confirms that the clock cycles match the clock frequency at which the cryptographic module operates can be estimated even from outside the device. Fig. 9(c) shows an example waveform when a continuous sinusoidal wave of 144 MHz, which is six times the clock frequency, is injected via the power cable. Fig. 9(c) shows that the times of glitches superimposed on each clock

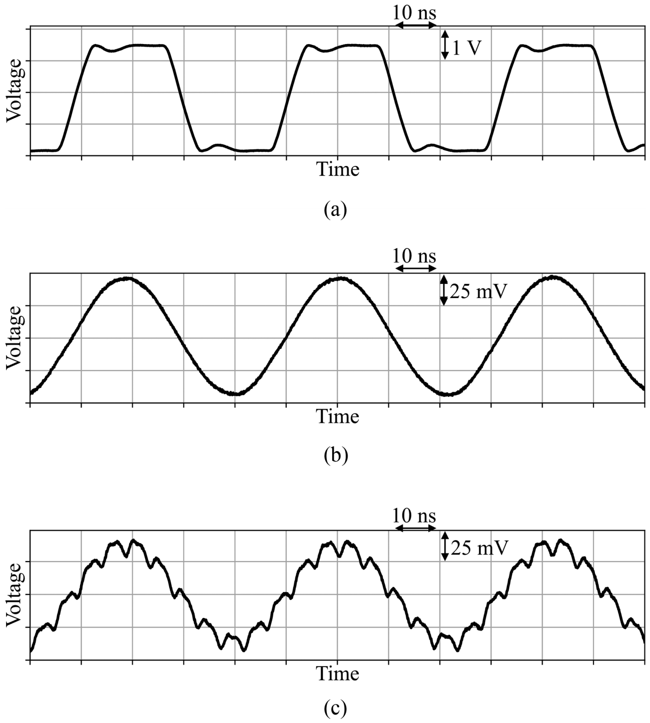


Fig. 9. (a) Original clock signal directly observed output from the function generator. (b) Clock signal observed from the power cable. (c) Clock signal superimposed with a sinusoidal wave with a frequency of 144 MHz.

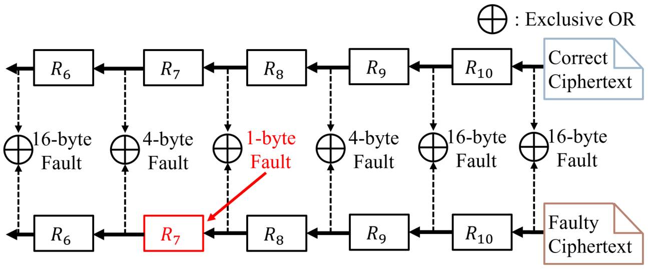


Fig. 10. Method for estimating the fault occurrence round and the number of fault bytes.

signal can be kept constant by injecting a frequency that is an integer multiple of the clock frequency.

C. Estimation of the Fault Occurrence Round for Evaluating Experimental Results

Fig. 10 shows a conceptual diagram of this method. First, the ciphertext output from the cryptographic module is obtained and the corresponding correct ciphertext is written back in a procedure equivalent to decryption. The exclusive OR of the correct and incorrect output intermediate values is then calculated for each round to obtain the number of fault bytes. The fault is expanded every four bytes by a linear transformation called MixColumns, which refers to a one element process in a round. For example, if a one-byte fault occurs in the input of MixColumns, four-byte faults occur in the output of MixColumns. Therefore, if a fault occurs in one round, the number of fault bytes increases in the subsequent rounds. Similarly, the number of fault bytes

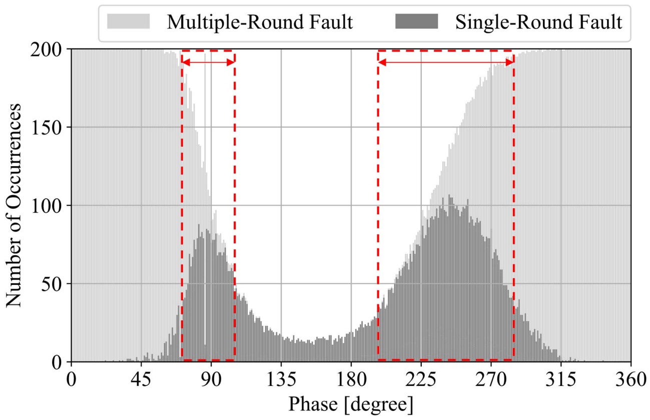


Fig. 11. Number of fault occurrences by injecting the continuous sinusoidal wave with the controlled phase. The number of single-round faults is highlighted as dark gray, and the number of multiple-round faults as light gray.

increases by the inverse MixColumns operation when decrypting from the fault occurrence round. Fig. 10 shows an example when a one-byte fault occurs in round 7. In this case, the number of fault bytes increases after round 8 and before round 6. Therefore, the fault occurrence round can be assumed to be the round with the smallest number of fault bytes. In this article, the above method is used for evaluating the fault occurrence round and the number of fault bytes in the experimental results.

This method cannot accurately isolate the round in which the fault occurs if a fault occurs in more than two rounds. Furthermore, as shown in [22], the reliability of this method decreases when the number of fault bytes is 13 or more because it includes cases in which the encryption process is not performed due to communication faults or the like [22]. In this article, as in [22], the fault is evaluated as a single-round fault if the number of fault bytes is less than or equal to 12, otherwise, the fault is evaluated as a multiple-round fault.

In addition, the MixColumns operation is only skipped in round 10 in the case that AES-128 is used. This method cannot distinguish faults that occur in these rounds because the number of fault bytes is not expanded between the input intermediate value and the output value in round 10. Moreover, t_{cp} will be shorter than in the other rounds because the MixColumns operation is skipped in round 10, and, it will be unlikely to require a delay time of $t_{\max}$. Therefore, in this article, a fault that occurs in either round 9 or round 10 is treated as a fault that occurred in round 9.

D. Experimental Results and Discussion

Fig. 11 shows the number of multiple-round faults (light gray) and number of single-round faults (dark gray) when the proposed method was applied to the cryptographic module for 200 encryption processes. The experiment confirmed that at 144 MHz (six times the clock frequency estimated in Section III-B), the total number of fault occurrences decreased, and the number of single-round fault occurrences increased in the phase range from 70° to 110° and from 200° to 280°, as shown in the red boxes in Fig. 11.

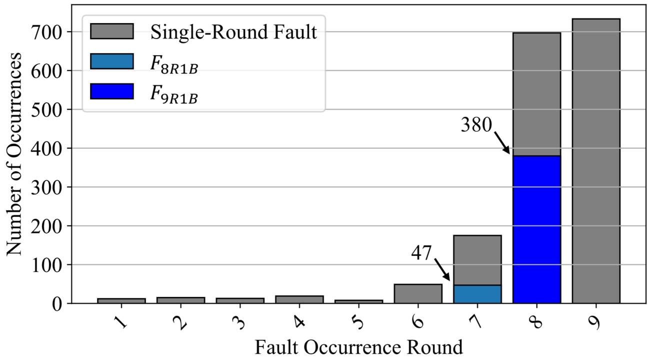


Fig. 12. Distribution of the number of single-round fault occurrences and the number of fault occurrences required for DFA in the range from 70° to 110°.

Subsequently, we focus on the number of fault occurrences required for Piret's DFA. This method assumes the occurrence of a one-byte fault in the input intermediate value of round 8 or 9 of AES-128. The difference between the faulty ciphertext that match the assumption and the corresponding correct ciphertext is calculated, and the number of candidate secret keys is narrowed down by calculation for several different pairs of the faulty and correct ciphertexts. The secret key can be uniquely extracted by repeating the process using the several pairs.

Fig. 12 shows the distribution of the number of single-round fault occurrences in each round and the number of fault occurrences required for DFA in the range from 70° and 110°. F_{8R1B} indicates the number of one-byte faults in the input intermediate value at round 8, and F_{9R1B} indicates the number of one-byte faults in the input intermediate value at round 9. As noted in Section II-B, the occurrence probability of $t_{\max}$ is uniform for all rounds, except for round 10. Therefore, the number of single-round faults in each round can be expected to be equal if a specific frequency, amplitude, and phase of the injected sinusoidal wave are selected for fault injection.

However, the results in Fig. 12 indicate that the frequency distribution of rounds in which single-round faults occur is biased with the fewest occurrences in the earlier rounds. The input intermediate value varies in subsequent rounds if the first fault occurs in any one round, the input intermediate value varies in subsequent rounds when there is no fault. Therefore, the length of the delay time in each round also varies. If any of these rounds requires a delay time of t_{cp} satisfying $t_{\max} < t_{\text{cp}}$, a second fault occurs, which results in a multiple-round fault. Thus, a possible source of the bias described above is the fact that, depending on the first fault occurrence round, there will be a different number of rounds with varying delay lengths. For example, if the first fault occurs in round 1, there are four future rounds (rounds 3, 5, 7, and 9) in which a second fault may occur; therefore, there is a high probability of a multiple-round fault. However, if the first fault occurs in round 7, there is only one future round (round 9) in which a second fault may occur, and thus in this case the probability of a multiple-round fault is low. This explains why the number of single-round faults is the lowest in the earlier rounds.

TABLE II
COMPARISON OF THE EXPERIMENTAL RESULTS BETWEEN PREVIOUS METHOD AND PROPOSED METHOD REGARDING FAULTS REQUIRED FOR DFA

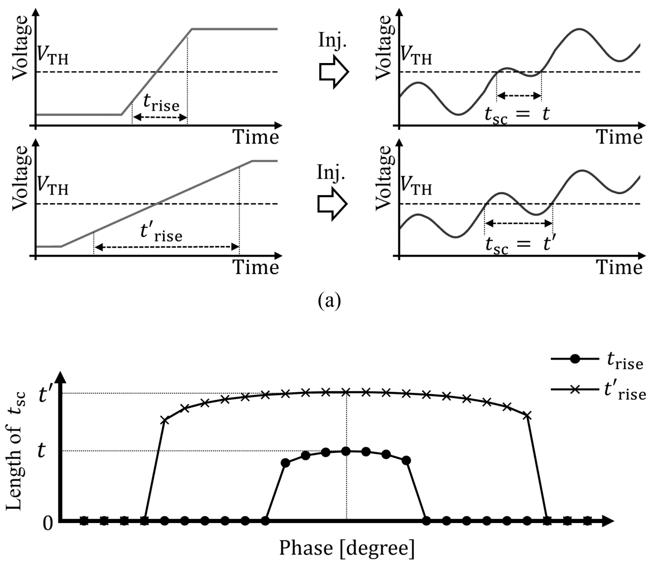
	METHOD [6]	OURS
NUMBER OF ENCRYPTION TRIALS	340 000	8200
NUMBER OF F_{8R1B}	1	47
NUMBER OF F_{9R1B}	10	380

Table II presents a comparison between the number of encryption trials and number of faults required for DFA using the method in [6] and the method proposed in this study. In [6], there is one occurrence of F_{8R1B} and ten occurrences of F_{9R1B} after 340 000 encryption trials. From this, it follows that the cryptographic module had to perform 340 000 and 30 000 operations, respectively, to generate the fault required for DFA. In contrast, the present experimental results show that, after 8200 encryption trials performed in the range between 70° to 110°, there are 47 occurrences of F_{8R1B} and 380 occurrences of F_{9R1B}. This means the cryptographic module had to perform approximately 174 ($=$ 8200/47) and 22 ($=$ 8200/380) operations, respectively, to generate the fault required for DFA. The above results demonstrate that the method proposed in this study can generate the faults necessary for the estimation of the secret key with a high probability. Furthermore, the fault required for DFA can be obtained after three orders of magnitude less encryption trials compared to that in [6].

E. Countermeasure Against the Proposed Method

We consider an approach to counteract our proposed fault injection method. One potentially effective approach is to suppress the variation in the total number of fault occurrences when the attacker controls the phase of the continuous sinusoidal wave. As a method for implementing this, we describe a countermeasure focusing on the clock rise time.

The number of fault occurrences varies depending not only on the parameters of the continuous sinusoidal wave, but also on the clock rise time of the cryptographic module [23], which is difficult for the attacker to control. This is because, as shown in Fig. 13(a), the lengths of t_{sc} will vary if the clock rise times are different even when injecting continuous sinusoidal waves of the same phase. Thus, as shown in Fig. 13(b), the range of lengths that t_{sc} can assume by means of phase control is believed to depend on the clock rise time. Therefore, it is possible that our proposed fault injection method can be counteracted by evaluating the presence or absence of variations in the total number of fault occurrences attributed to the phase control of the continuous sinusoidal wave at various clock rise times and by setting the rise time that does not cause any variation in the total number of fault occurrences. Both information security and EMC must be considered when dealing with fault injection methods, a clock signal rise time may be designed to minimize fault occurrences; however, it should not be so steep that it increases spurious EM emissions due to an increase in high-frequency signal components.



(a)

(b)

Fig. 13. (a) Difference of the length of t_{sc} according to the length of clock rise time by injecting the continuous sinusoidal wave with same phase. (b) Difference of the range of possible values of t_{sc} according to the clock rise time by controlling the phase of the continuous sinusoidal wave.

IV. Conclusion

In this article, we proposed a method that generates processor timing faults required for secret key extraction that uses approximately three orders of magnitude less encryption trials than that in [6], even for noninvasive fault injection from outside the device.

The proposed method injects a continuous sinusoidal wave which is an integer multiple of the clock frequency and controls the frequency, phase, and amplitude of the injected wave. This enables the attacker to generate glitches at specific times and causes faults at a specific time with a high probability of success even when executing fault injection from outside the device.

The above results indicate that cryptographic devices that can be physically accessed for only a brief period of time and were not subject to fault injection methods in the past, may now be compromised. Moreover, in the case of fault injection under conditions where the cryptographic equipment is not directly accessible, the secret key extraction methods that can be used are limited because the faults occur randomly [6]. However, since the proposed method can cause faults at arbitrary times, various analysis methods proposed in past studies [12], [13], [14], [15] may be applicable, and secret keys may be analyzed in a shorter time.

The feasibility of our proposed fault injection method is highly dependent on the rise time of the clock that operates the cryptographic module. Therefore, as a countermeasure, it is necessary to consider selecting a clock signal that improves security while also considering EMC due to spurious emission. The proposed noninvasive IEMI-based fault injection method created a fault caused by a setup time violation due to clock glitching. However, it is likely that IEMI may cause faults because of causes other than a setup time violation. In the future, it is desirable to study the security degradation caused by faults other than fault injection methods.

References

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

[2] J.-M. Schmidt and M. Hutter, “Optical and EM fault-attacks on CRT-based RSA: Concrete results,” in *Proc. Austrian Workshop Microelectron. (Austrochip)*, 2007, pp. 61–67.

[3] N. Moro, A. Dehbaoui, K. Heydemann, B. Robisson, and E. Encrenaz, “Electromagnetic fault injection: Towards a fault model on a 32-bit microcontroller,” in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr.*, 2013, pp. 77–88.

[4] L. Zussa, J.-M. Dutertre, J. Clediere, and A. Tria, “Power supply glitch induced faults on FPGA: An in-depth analysis of the injection mechanism,” in *Proc. IEEE 19th Int. -Line Testing Symp.*, 2013, pp. 110–115.

[5] A. Dehbaoui, J.-M. Dutertre, B. Robisson, and A. Tria, “Electromagnetic transient faults injection on a hardware and a software implementations of AES,” in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr.*, 2012, pp. 7–15.

[6] Y.-I. Hayashi, N. Homma, T. Mizuki, T. Aoki, and H. Sone, “Transient IEMI threats for cryptographic devices,” *IEEE Trans. Electromagn. Compat.*, vol. 55, no. 1, pp. 140–148, Feb. 2013.

[7] Y. Shinoda, M. Takenouchi, Y. Hayashi, T. Mizuki, and H. Sone, “Measurement on effect of controlled wave phase in EM fault injection method,” in *Proc. Int. Symp. Electromagn. Compat. - Europe*, 2020, pp. 1–5.

[8] H. Nishiyama, D. Fujimoto, Y. Kim, H. Sone, and Y. Hayashi, “IEMI fault injection method using continuous sinusoidal wave with controlled frequency, amplitude, and phase,” in *Proc. 13th Int. Workshop Electromagn. Compat. Integr. Circuits*, 2022, pp. 97–101.

[9] E. Biham and A. Shamir, “Differential fault analysis of secret key cryptosystems,” in *Proc. Ann. Int. Cryptol. Conf.*, 1997, pp. 513–525.

[10] G. Piret and J.-J. Quisquater, “A differential fault attack technique against SPN structures, with application to the AES and khazad,” in *Cryptographic Hardware and Embedded Systems—CHES (LNCS 2779)*. Berlin, Germany: Springer, Sep. 2003, pp. 77–88.

[11] C. Giraud and A. Thillard, “Piret and quisquater’s DFA on AES revisited,” *Cryptol. ePrint Arch.*, Tech. Rep. 2010/440, 2010. [Online]. Available: https://ia.cr/2010/440

[12] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta, “Fault sensitivity analysis,” in *Proc. Int. Workshop Cryptograph. Hardware Embedded Syst.*, 2010, pp. 320–334.

[13] T. Fuhr, E. Jaulmes, V. Lomné, and A. Thillard, “Fault attacks on AES with faulty ciphertexts only,” in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr.*, 2013, pp. 108–118.

[14] N. Ghalaty, B. Yuce, M. Taha, and P. Schaumont, “Differential fault intensity analysis,” in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr.*, 2014, pp. 49–58.

[15] A. Spruyt, A. Milburn, and Ł. Chmielewski, “Fault injection as an oscilloscope: Fault correlation analysis,” *IACR Trans. Cryptographic Hardware Embedded Syst.*, pp. 192–216, 2021.

[16] L. Zussa, D. Jean-Max, C. Jessy, R. Bruno, and T. Assia, “Investigation of timing constraints violation as a fault injection means,” in *Proc. 27th Conf. Des. Circuits Integr. Syst.*, 2012, pp. 1–16.

[17] C. R. Paul, *Introduction to Electromagnetic Compatibility*. New York, NY, USA: Wiley, 1992, pp. 335–400.

[18] S. Endo et al., “A silicon-level countermeasure against fault sensitivity analysis and its evaluation,” *IEEE Trans. Very Large Scale Integration Syst.*, vol. 23, no. 8, pp. 1429–1438, Aug. 2015.

[19] NIST. “Specification for the advanced encryption standard (AES),” Technical Report FIPS PUB 197, 2001. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.197

[20] “Side-channel attack standard evaluation board,” 2007. [Online]. Available: https://www.risec.aist.go.jp/project/sasebo/

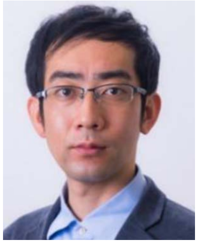
[21] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A compact rijndael hardware architecture with S-Box optimization,” in *Proc. Adv. Cryptol.*, 2001, pp. 239–254.

[22] J. Takahashi, Y.-I. Hayashi, N. Homma, H. Fuji, and T. Aoki, “Feasibility of fault analysis based on intentional electromagnetic interference,” in *Proc. IEEE Int. Symp. Electromagn. Compat.*, 2012, pp. 782–787.

[23] N. Saga, T. Itoh, Y. Hayashi, T. Mizuki, and H. Sone, “Study on the effect of clock rise time on fault occurrence under IEMI,” in *Proc. IEEE Int. Symp. Electromagn. Compat. IEEE Asia-Pacific Symp. Electromagn. Compat.*, 2018, Art. no. 9.

Hikaru Nishiyama received the B.E. degree in engineering from National Institute of Technology, Sasebo College, Sasebo, Japan, in 2019 and the M.E. degree in engineering in 2021 from Nara Institute of Science and Technology, Ikoma, Japan, where he is currently working toward the Ph.D. degree in engineering.

His research interests include hardware security and electromagnetic compatibility.



Daisuke Fujimoto (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in engineering from Kobe University, Kobe, Japan, in 2009, 2011, and 2014, respectively.

He is currently an Assistant Professor with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Japan. He is also a Visiting Assistant Professor with the Institute of Advanced Sciences, Yokohama National University, Yokohama, Japan. His research interests include hardware security and implementation of security cores.

Dr. Fujimoto is a Member of the Institute of Electronics, Information, and Communication Engineers.

Hideaki Sone (Life Member, IEEE) received the B.E. degree in electrical engineering and the M.E. and Ph.D. degrees in electrical communications from Tohoku University, Sendai, Japan, in 1978, 1980, and 1992, respectively. He was the Chapter Chair of Sendai Section Chapter, EMC Society, and is currently a Professor Emeritus with Tohoku University.

His main research interests include electromechanical device components, information telecommunication systems, and instrumentation electronics.

Yuichi Hayashi (Senior Member, IEEE) received the M.S. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2005 and 2009, respectively.

He is currently a Professor with the Nara Institute of Science and Technology, Ikoma, Japan. His research interests include electromagnetic compatibility and information security.

Dr. Hayashi was the recipient of many awards and honors, including the IEEE Electromagnetic Compatibility Society Technical Achievement Award, IEEE International Symposium on EMC Best Symposium Paper Award, and Workshop on Cryptographic Hardware and Embedded Systems Best Paper Award. He is the Chair of Electromagnetic Information Leakage Subcommittee in IEEE EMC Technical Committee 5.