Flush Code Eraser: Fast Attack Response
Invalidating Cryptographic Sensitive Data

Kazuo Sakiyama, Tatsuya Fujii, Kohei Matsuda, and Noriyuki Miura

Abstract—Fault-injection attacks against cryptographic devices have been studied in great detail. As a countermeasure to such attacks, the flush code eraser (FCE) has been proposed. The FCE realizes a new attack reaction in which sensitive data in a cryptographic device are quickly invalidated by redistributing the electrons in the circuits. In this letter, we discuss the abilities of the FCE by evaluating the data invalidation period with experiments using a test chip. As a result, the invalidation period is estimated to be approximately 80 ps, which enhances the security of the cryptographic device and also prevents the possibility of the future attacks.

Index Terms—Attack response, countermeasure, data invalidation, fault injection, physical attacks

I. INTRODUCTION

COUNTERMEASURES for fault-injection attacks against cryptographic devices have been studied to counteract physical attacks. This field of study has formed a different branch of research from fault-tolerant implementation, which aims at maintaining uninterrupted operation when a failure occurs in a computational resource unintentionally. The purpose is to prevent information leakage. Ultimately, even breaking the cryptographic device could be a countermeasure to keep a secret from the attacker.

Two types of countermeasures have been discussed to prevent information leakage occurring during the fault attack. One is based on a detection that is realized by a sensor, duplicated circuits, algorithmic verification, and so on. Then, triggered by a detection mechanism, we need to respond with actions, such as resetting a state in the circuit and destroying a part of the circuit. In this two-step approach consisting of the detection and reaction, a reaction mechanism should be embedded in the system as well as the detection mechanism. Unfortunately, the detection and response mechanisms are normally developed as separate functions. Furthermore, the reaction method is not deeply discussed in the literature compared to the detection methods. Notice that the attack reaction is closely related to the resilience of a system. Moreover, from a sustainability standpoint, we expect to restore the system to a normal state after an attack reaction.

The other type of countermeasure is called an infective countermeasure; it performs dummy and redundant operations in addition to operations performed by an original cryptographic algorithm so that an attacker cannot derive any information related to a secret key from a faulty output [1]. The infective countermeasure does not require any special reaction on detected faults, which provides prominent resistance against the fault attack. The drawback is the cost or performance overheads caused by the dummy and redundant operations.

Recently, the flush code eraser (FCE), a new attack reaction, was proposed in [2]. FCE is based on the detection-and-reaction countermeasure while enhancing the merit of the infective countermeasure. By redistributing the electrons in a floating module in a device, FCE enables us to invalidate sensitive data in circuits significantly fast. To measure the period of the data invalidation, in [2], they monitor the voltage drop during the data invalidation by FCE over several measurements as well as the averaged Hamming distance (HD) of an intermediate value in 128-bit advanced encryption standard (AES) encryption processor. The obtained result would accompany with a large margin, i.e., worst-case result, which is secure enough in many physical attack scenarios but could be insecure against more powerful attacks. The significance of understanding the actual data invalidation period is out of scope and hence not discussed in [2]. Therefore, in this letter, we detail the power of FCE and evaluate the performance from a security viewpoint.

II. BACKGROUND

A. Fault Attacks and Countermeasures

Among the fault attacks, the differential fault analysis proposed in [3] and [4] is considered as a powerful attack and a real threat. In the aforementioned attack, an error is induced in the intermediate value during applying a cryptographic algorithm due to a transient fault controlled by an attacker. Thereafter, the error data is processed by a correct encryption algorithm process, and erroneous data is outputted as a result. The attacker analyzes the information on the difference between the correct and faulty outputs based on a fault model to derive the secret key. The fault model here refers to the kind of fault the attacker induces on the intermediate
value. The safe error attacks that use faults that do not affect the calculation results, and the ineffective fault attacks are also famous. Fault sensitivity analysis (FSA), which infers the intermediate value from the fault sensitivity, opens a new direction of fault analysis [5].

To protect the secret key from the fault attacks, it is necessary to detect a physical phenomenon, which may cause a fault, or a fault itself, and then to react fast enough to prevent the leakage of sensitive information linked to the secret key. The role of this reaction is to invalidate the data in a cryptographic circuit. This is also known as zeroization described in [6]. In the data invalidation methods that are made public, several methods are adopted, such as initializing the circuit by resetting after fault detection (state initialization) and setting the output data to a random number (output randomization). These reactions are normally performed at the algorithm level. Accordingly, invalidation of complete data relies on an additional algorithm-level computation, which requires time. Moreover, sensitive data, which should be invalidated instantly, partially remains in the circuit somewhere.

III. Flush Code Eraser

FCE is a reactive countermeasure proposed against laser fault attacks [2]. The purpose of FCE is the same as the previously proposed countermeasures that do not allow attackers to exploit sensitive data. Conventional countermeasures normally employ an invalidation method based on logical computations, such as reset for digital circuits, XOR masking, infective computation, and so on. Strong attackers exploit faulty intermediate data before such initialization and invalidation processes are completed. On the other hand, FCE invalidates all of the sensitive data instantaneously by coercive redistribution of electrons via the power and ground lines of a circuit. Given that FCE removes the root cause of information leakage, it will enhance security not only against the attacks proposed so far but also against all future physical attacks.

The principle of FCE is simple. FCE controls two switches, a floating switch and a shunting switch, on the power line and the ground line of the cryptographic core, respectively. The process performed by these control signals has not been discussed well in the algorithm-level countermeasures.

A. Block Diagram

A block diagram of FCE for a cryptographic core module is shown in Fig. 1. In the normal state, the floating switches are on, and the shunting switches are off so that the cryptographic core can operate. When a fault attack is detected, the cryptographic core receives an alarm signal, and switches off the floating switches. At the same time, a reset signal is provided to the sequential logics [flip-flops (FFs)]. This is because the data stored in FFs can remain for a while without power due to the effect of the cross-coupled inverters. In this state, the cryptographic core is isolated from the power lines.

Immediately after floating the core, the shunting switch is set on so that data invalidation is forced to accelerated. This is because, even if this state is retained, sensitive data might not be invalidated quickly in the combinatorial logics as there are no electrical drivers to the core. As a result, the charges in the combinatorial logics in the cryptographic core are redistributed, and all the gates become irrelevant to the secret key.

The power domain of the cryptographic core is supposed to be different from other circuits on a chip, which keeps the chip functional even in the data invalidation process. This feature is important in the given resilience of devices. The additional circuits required for FCE are mainly the floating and shunting switches, which leads to ignorable overhead.

B. Timing Diagram

Fig. 2 shows a timing diagram for the data invalidation process by FCE. As a conventional countermeasure, the data initialization using a reset signal is described in Fig. 2(a). The alarm signal is used for resetting FFs asynchronously in this case. After all the FFs are initialized, the initialized data propagates through combinatorial logics. The time $t_p$ corresponds to the critical path delay of the combinatorial logics. All sensitive data are initialized in $t_p$.

The case for using FCE is illustrated in Fig. 2(b). When the alarm signal becomes high, the data in the sequential circuits (FFs) are reset by an asynchronous reset, which is the same procedure as the conventional case in Fig. 2(a). The floating switch turns off in $t_f$, and the cryptographic core, which calculates sensitive data, transitions into the floating state. Subsequently, the shunt switch turns on in $t_a$, and charges are forcibly redistributed in the combinatorial logics, which takes $t_p$, which is much shorter than $t_f$. Eventually, all the sensitive data are invalidated in $t_b$, and $V_{DD,CORE}$ voltage drops down

1In the case of FF with a synchronous reset, we need to wait for the arrival of a clock signal after the data in the combinational circuit is invalidated. In this sense, asynchronous reset is superior to synchronous reset as a countermeasure.
to approximately 0.6 V. Notice that measuring $t_q$ is one of the major objectives of this letter because it is directly related to FCE.

IV. ELABORATE EXPERIMENTS

The attacker attempts to exploit sensitive information from a faulty output, which stems from erroneous data induced in the cryptographic core. Notice that there are two types of the erroneous data; one is produced by an attacker’s intentional fault, and the other is by FCE. That is, FCE holds a certain risk that could be exploited by an attacker. This means that the faulty data should be invalidated fast enough to avoid the potential risk of fault analysis.

Here, we focus on the latter threat, which is related to the time period $t_q$ in Fig. 2(b). One of the performance requirements is the short transition period of data invalidation by FCE. More precisely, it is required that all the intermediate values in a combinational circuit settle as fast as possible in the FCE process. Measuring $t_q$ for a single measurement, which is determined by elaborate experiments, will be explained in the following sections.

A. Preparation for Experiments

To observe the capabilities of FCE, we designed and fabricated a test chip using the 0.18 μm standard CMOS process. The chip has the FCE mechanism to protect a loop-architecture-based 128-bit AES encryption processor, which performs one AES round per cycle. We have prepared test pins for controlling the alarm signal externally and for providing clock signal to the AES core. The intermediate values of the AES core stored in registers, i.e., output of an intermediate round, are observable via an interface block that does not have the FCE mechanism. In the experiments, assuming that a laser fault is injected during the third-round AES operation, the alarm signal is activated during the third round.

In [2], it is shown that the period of the data invalidation, $t_q$, is in approximately 2 ns by a simulation, which is confirmed by an actual measurement of voltage drop by an on-chip monitor circuit. Also, the time period is derived based on the averaged HD of an intermediate value in AES over several measurements. That is, the reported result does not indicate the time from the start to the end of the data disappearance observed by a single measurement. Therefore, there may be a case where a measurement error occurs in obtaining the intermediate value, and the time period of the data invalidation appears to be longer than the actual value. In other words, the actual data invalidation period could be shorter than 2 ns.

It is difficult to realize a worst-case situation as for the time $t_q$. It is heavily dependent on the data to be processed in the combinational circuit, i.e., the stored data in FFs and the switching activities in the circuit. Herein, assuming that one of the worst-cases happens when the 128-bit intermediate value in FFs is all one, i.e., $1^{128}$, $t_q$ is measured. Thus, in the experiments, the plaintext and the secret key are set so that the result of the third round operation becomes $1^{128}$. This is because the output of the combinational logics become all zeros, i.e., $0^{128}$ in the result of FCE, and we consider that discharge is a major factor that determines $t_q$.

B. Experimental Results of $t_q$ Using Test Chip

The experiments are performed with the environment illustrated in Fig. 3. An field programmable gate array
(FPGA) (Spartan XC3S1400A) provides a clock signal to the test chip, configures the cryptographic core, and controls the data communication with PC. The alarm signal is provided via a pin of the test chip. Therefore, we did not emulate detection mechanisms, such as bulk built-in current (BBIC) sensor for laser fault attacks in [2]. The reason is as follows. Sensors in the test chip, which activate the alarm signal when detecting laser injection, introduce additional measurement errors. To exclude them, we could measure the actual period time easily and precisely.

The test chip does not have an external port to synchronously reset FFs in the cryptographic core. Therefore, we followed the sequence shown in Fig. 2(c) by controlling the delay time of $t_d$. In other words, data in FFs are invalidated synchronously. We now describe the experiment in detail. The system clock signal was generated using a function generator (Agilent Technologies 33250A), and the delay time of $t_d$, which is the time period between the alarm signal and the positive edge of the clock signal as shown in Fig. 2(c), was controlled by an FPGA. The alarm signal was generated using another function generator (Keysight 33612A), and was provided to the test chip, triggered by a control signal.

In the experiments, a timing error was observed in the alarm signal, which was mainly due to the jitter of the function generators. Here, we assume that the time period $t_d$ follows a standard distribution as

$$t_d \sim \mathcal{N}\left(t_b - \frac{t_q}{2}, \sigma^2_g\right).$$

Three types of results were observed in the 128-bit read data; the Hamming weights are 0, 128, and between 0 and 128. Here, the number of results categorized in these three types are denoted as $H_0$, $H_{128}$, and $H_2$, respectively. We set the delay time $t_d$ such that $H_0$ and $H_{128}$ are as close as possible. That is, $H_0 \approx H_{128}$. In total, 10,000 measurements were performed. The value of $t_q$ can be calculated as

$$t_q = \sigma_g(z_{128} - z_0)$$

where $z_{128}$ and $z_0$ are read from the standard normal table. Fig. 4 visualizes the relationship of variables.

The ambient temperature was set to 20°C and 80°C using temperature controller (MISUMI MTCS). Table I summarizes the results for $t_q$. Although $\sigma_g = 330$ ps was observed in the experiments; however, results for several different $\sigma_g$ are listed for the purpose of reference.

### Table I

| Temp. | $\sigma_g$ (ps) | 320 | 330 | 340 | 350 |
|-------|----------------|-----|-----|-----|-----|
| 20°C  | $t_q = 70$ ps  | 72  | 75  | 77  | 79  |
| 80°C  | $t_q = 76$ ps  | 78  | 81  | 83  | 86  |

### V. Discussion on Security Evaluation

The validity of the proposed FCE countermeasure is discussed here. We assume possible future attacks although they are not implemented yet. Here, we briefly review the limitation of FCE, including its misuse and clarify the problems.

FCE can be seen as an on-chip fault generator. For instance, consider a case that the attacker has a full access to the floating switch, and forces the shunt switch off. By switching-off the floating switch off in a short period, meaningful faulty data might be leaked because the invalidation process is relatively slow without shunting. To prevent this type of attack, some protection mechanisms, such as an attack-tolerant control of the floating switch, would be required.

A powerful and straightforward attack is also probable. As a typical example, an invasive attacker delays or interrupts the alarm signals connected to FCE so that the attacker can obtain useful information by a fault injection. In this regard, on-the-fly malfunction diagnosis is necessary, and it is feasible just by checking the connectivity between FCEs and sensors.

### VI. Conclusion

With elaborate experiments using a test chip, the period of AES data validation by FCE was shown to be approximately 80 ps. This result infers that FCE can avoid a serious leakage by invalidating all sensitive data instantaneously. In the future, we intend to consider a secure implementation of FCE to prevent future possible attacks. Moreover, we would like to clarify the sensitivity of a sensor to arise an alarm signal such that FCE can also invalidate attackers’ intended faults, which would require proactive fault detections.

### References

[1] B. Gierlichs, J.-M. Schmidt, and M. Tunstall, “Infective computation and dummy rounds: Fault protection for block ciphers without check-before-output,” in Proc. LATINCRYPT, vol. 7533, 2012, pp. 305–321.

[2] K. Matsuda et al., “A 286 F°/cell distributed bulk-current sensor and secure flush code eraser against laser fault injection attack on cryptographic processor,” IEEE J. Solid-State Circuits, vol. 53, no. 11, pp. 3174–3182, Nov. 2018.

[3] D. Boneh, R. A. DeMillo, and R. J. Lipton, “On the importance of checking cryptographic protocols for faults (extended abstract),” in Proc. EUROCRYPT, vol. 1233, 1997, pp. 37–51.

[4] E. Biham and A. Shamir, “Differential fault analysis of secret key cryptosystems,” in Proc. CRYPTO, vol. 1294, 1997, pp. 513–525.

[5] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta, “Fault sensitivity analysis,” in Proc. CHES, vol. 6225, 2010, pp. 320–334.

[6] National Institute of Standards and Technology, FIPS PUB 140-2: Security Requirements for Cryptographic Modules. Accessed: Feb. 20, 2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf