IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN XXXX-XXXX, Vol. 0, No. 0, pp. 1–2. DOI:XXXXXXXX

XXX for XXX Against Fault Attacks

Jiahao Xiang[1] and Lang Li[1]

Hengyang Normal University, College of Computer Science and Technology, Hengyang, China

Abstract.

Keywords: Fault attack · countermeasures

1 Introduction

1.1 Related Work

The [Gen23] shows both in theory and experimentally that the countermeasures based on *caching the intermediate WOTS*$^+$*s* offer a marginally greater protection against unintentional faults. For the [CM09] show that the Bellcore attack cannot be applied to the *PSS encoding*; namely we show that PSS is provably secure against random fault attacks in the random oracle model, assuming that inverting RSA is hard. The [THN$^+$24] formalizes the *k-fault-resistant partitioning* notion to solve the fault propagation problem when assessing *redundancy-based hardware countermeasures* in a first step. Proven security guarantees can then reduce the remaining hardware attack surface when introducing the software in a second step. which combines the software and hardware countermeasures to provide a more robust solution against fault attacks. The [DOT24] propose a fault countermeasure, StaTI, based on *threshold implementations and linear encoding techniques*. The proposed countermeasure protects the implementations of cryptographic algorithms against both side-channel and fault adversaries in a non-combined attack setting.

2 Preliminary

2.1 Fault Attack Model

Consider a cryptographic computation $\mathcal{C} : \mathcal{K} \times \mathcal{M} \to \mathcal{O}$ executing on a target device, where $\mathcal{K}$, $\mathcal{M}$, and $\mathcal{O}$ denote the key, message, and output spaces, respectively. Let $\mathcal{S} = \{s_0, s_1, \ldots, s_n\}$ represent the sequence of internal computational states during execution.

Attacker Model. The adversary $\mathcal{A}$ controls a fault injection oracle $\mathcal{F}(t, \sigma, \phi, \alpha)$ parameterized by timing $t \in [0, T]$ within execution window T, target computational domain $\sigma \in \Sigma$ where $\Sigma = \{\text{ALU}, \ldots, \text{control}\}$ represents functional units, injection mechanism $\phi \in \{\text{EM}, \ldots, \text{voltage}\}$, and intensity $\alpha \in \mathbb{R}^+$. The fault oracle induces state transitions $s_i \mapsto s_i^{\text{fault}}$ with probability $P_{\text{fault}}(t, \sigma, \phi, \alpha)$. The adversary observes output pairs $(o_{\text{clean}}, o_{\text{fault}})$ where $o_{\text{clean}} = \mathcal{C}(k, m)$ and $o_{\text{fault}} = \mathcal{C}^{\text{fault}}(k, m)$ represents computation under fault influence.

Security Assumptions. The internal states $s_i \in \mathcal{S}$ remain opaque to $\mathcal{A}$, formally expressed as $\mathcal{A}(s_i) = \perp$ for all $i \in [0, n]$. Fault effects manifest probabilistically according to $P(\Delta | t, \sigma, \phi, \alpha)$ where Δ represents the computational deviation induced by the fault oracle. The adversary cannot deterministically control fault propagation through the computational pipeline, acknowledging stochastic fault models: transient bit corruption $\Delta_{\text{bit}} \sim \text{Bernoulli}(p_\sigma)$, instruction disruption $\Delta_{\text{instr}} \sim \text{Geometric}(q_\sigma)$, or data corruption

Licensed under Creative Commons License CC-BY 4.0.

$\Delta_{\text{data}} \sim \text{Uniform}(\mathbb{F}_2^w)$ for w-bit word operations, where success probabilities p_σ, q_σ depend on the target domain σ.

This attack model aligns with practical fault injection scenarios encountered in hardware security evaluations and provides a realistic framework for analyzing the effectiveness of proposed countermeasures.

References

[CM09] Jean-Sébastien Coron and Avradip Mandal. PSS is secure against random fault attacks. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 653–666. Springer, Berlin, Heidelberg, December 2009.

[DOT24] Siemen Dhooghe, Artemii Ovchinnikov, and Dilara Toprakhisar. StaTI: Protecting against fault attacks using stable threshold implementations. *IACR TCHES*, 2024(1):229–263, 2024.

[Gen23] Aymeric Genêt. On protecting SPHINCS+ against fault attacks. *IACR TCHES*, 2023(2):80–114, 2023.

[THN+24] Simon Tollec, Vedad Hadzic, Pascal Nasahl, Mihail Asavoae, Roderick Bloem, Damien Couroussé, Karine Heydemann, Mathieu Jan, and Stefan Mangard. Fault-resistant partitioning of secure CPUs for system co-verification against faults. *IACR TCHES*, 2024(4):179–204, 2024.