POSTER: SMAUG-SCA: Machine Learning Based Power Side-Channel Attack on SMAUG-T*

Madhumitha Ramaswamy[1], Rishav Saha[1], Suparna Kundu[2], Anupam Golder[3], Angshuman Karmakar[4], Debayan Das[1]
[1]Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore, India
[2]KU Leuven, Belgium
[3]Intel Corporation, Hillsboro, Oregon, USA
{madhumithar,debayandas}@iisc.ac.in,suparna.kundu@esat.kuleuven.be,anupam.golder@intel.com,angshuman@cse.iitk.ac.in

Abstract

This work presents SMAUG-SCA: the first power side-channel analysis (SCA) attack on SMAUG-T, the Korean post-quantum crypto (KpqC) Key Encapsulation Mechanism (KEM) standard. Although KpqC standardization was concluded in 2025, the side-channel analysis (SCA) of SMAUG-T has not been investigated in detail until now. SMAUG-SCA utilizes Random Forest, XGBoost, and a Fully Connected Neural Network (FCN) to achieve a test accuracy of $> 94\%$, showing that the secret key can be recovered with < 10 traces with a success probability of $> 99.9\%$.

CCS Concepts

• **Security and privacy → Embedded systems security**; **Side-channel analysis and countermeasures**.

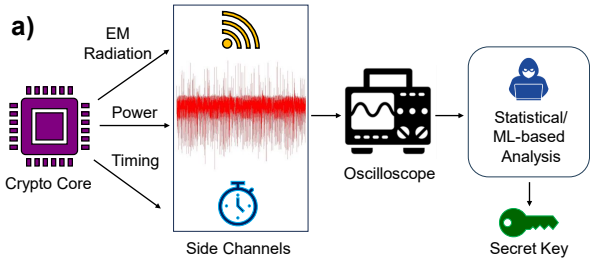
Keywords

Side-Channel Analysis, Post-Quantum Cryptography, SMAUG-T, Machine Learning, Power Analysis attacks

ACM Reference Format:
Madhumitha Ramaswamy[1], Rishav Saha[1], Suparna Kundu[2], Anupam Golder[3], Angshuman Karmakar[4], Debayan Das[1]. 2026. POSTER: SMAUG-SCA: Machine Learning Based Power Side-Channel Attack on SMAUG-T. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '26), June 01–05, 2026, Bangalore, India.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3779208.3804885

1 Introduction

Advancements in quantum computing pose a significant threat to classical public-key cryptography, as schemes such as RSA and ECC can be efficiently broken using Shor's algorithm[6]. To address this challenge, worldwide efforts have focused on developing post-quantum cryptography (PQC) to replace pre-quantum cryptographic primitives. In parallel with the NIST PQC standardization process, the Korean Post-Quantum Cryptography (KpqC) Competition was established to develop and evaluate PQC algorithms for domestic standards. The KpqC competition concluded in 2025, declaring SMAUG-T as the key encapsulation mechanism (KEM) standard. SMAUG-T is a lattice-based, IND-CCA2-secure key encapsulation mechanism. SMAUG-T is a merger of SMAUG and TiGER, combining module-lattice and ring-lattice constructions. Its security relies on the hardness of the Module Learning With Errors (MLWE) and Module Learning With Rounding (MLWR) assumptions with sparse secrets[5]. While SMAUG-T is mathematically secure against classical and quantum adversaries, its complex polynomial arithmetic is yet to be thoroughly evaluated against side-channel analysis (SCA) attacks. This work presents SMAUG-SCA: the first ML-based power SCA attack on SMAUG-T running on a 32-bit ARM microcontroller.



b)

$\underline{\texttt{KeyGen}}(1^\lambda)$:
$1: \mathbf{A} \leftarrow \mathcal{R}_q^{k\times k}, \mathbf{s} \leftarrow \texttt{HWT}_{h_s} \in S_\eta^k$
$2: \mathbf{e} \leftarrow \widetilde{D}_\sigma \in \mathcal{R}^k$
$3: \mathbf{b} \leftarrow -\mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e} \in \mathcal{R}_q^k$
$4:$ **return** $\texttt{pk} = (\mathbf{A}, \mathbf{b}),\ \texttt{sk} = \mathbf{s}$

$\underline{\texttt{Enc}}(\texttt{pk}, \mu)$: $\quad \triangleright\ \texttt{pk} = (\mathbf{A}, \mathbf{b}),\ \mu \in \mathcal{R}_t$
$1: r \leftarrow \texttt{spCBD}_r \in S_\eta^k$
$2: \mathbf{c_1} = \lfloor p/q \cdot \mathbf{A} \cdot \mathbf{r} \rceil \in \mathcal{R}_p^k$
$3: c_2 = \lfloor p'/q \cdot \langle \mathbf{b}, \mathbf{r} \rangle + p'/t \cdot \mu \rceil \in \mathcal{R}_{p'}$
$4:$ **return** $\texttt{ct} = (\mathbf{c}_1, c_2)$

$\underline{\texttt{Dec}}(\texttt{sk}, \mathbf{c})$: $\quad \triangleright\ \texttt{sk} = \mathbf{s},\ \mathbf{c} = (\mathbf{c}_1, c_2)$
$1: \mu' = \lfloor t/p \cdot \langle \mathbf{c}_1, \mathbf{s} \rangle + t/p' \cdot c_2 \rceil \in \mathcal{R}_t$
$2:$ **return** μ'

Point of Attack

Figure 1: a) Overview of Side-channel analysis (SCA) attack on a cryptographic device. b) SMAUG-T decapsulation highlighting the secret-dependent multiplication targeted by the proposed SMAUG-SCA attack.

1.1 Motivation

SCA attacks exploit *unintended physical leakage*, such as power consumption, arising from secret-dependent computations in cryptographic implementations as shown in Figure 1(a). In lattice-based

*This work was supported in part by the Cyber Security Karnataka (CySecK) initiative by the Govt of Karnataka, India. Our SMAUG-SCA dataset can be found here: https://github.com/CSRL-IISc/SMAUG

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
ASIA CCS '26, Bangalore, India
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2356-8/2026/06
https://doi.org/10.1145/3779208.3804885

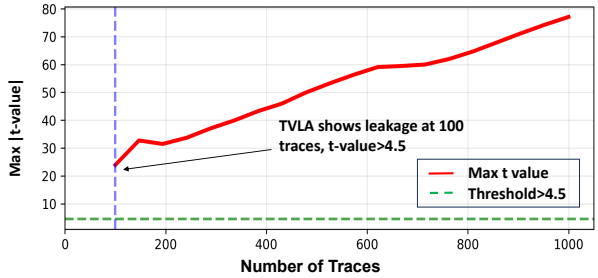


Figure 2: TVLA showing data-dependent leakage crossing threshold of 4.5 even with 100 traces.

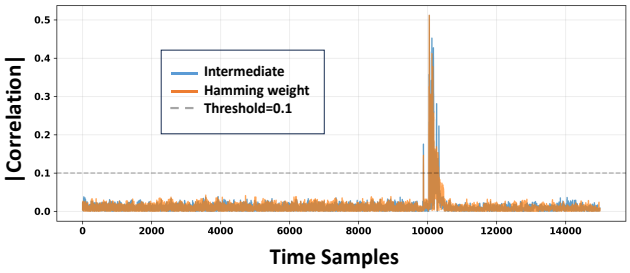


Figure 3: Known value correlation between the power samples trace and the secret dependent intermediate value (polynomial multiplication) shows distinct spike.

schemes such as SMAUG-T, the *decapsulation phase* which internally invokes the *decryption* algorithm as shown in Figure 1(b) is a particularly attractive attack target because it involves the multiplication of the ciphertext with the long-term secret key, directly exposing secret-dependent intermediate values. Unlike key generation and encapsulation, which are typically executed only once, decapsulation is performed repeatedly in real-world deployments, allowing an adversary to collect a large number of power traces under chosen ciphertexts. This makes decapsulation significantly more vulnerable to statistical and machine learning-based SCA.

1.2 Contributions

The key contributions of SMAUG-SCA are summarized as follows:

- We identify the attack point in the SMAUG-T decapsulation, specifically the `poly_mul_accumulate_toomcook` routine, where secret-dependent polynomial multiplications occur.
- Recognizing the limitation of Correlation Power Analysis (CPA) model, we evaluate different ML-based SMAUG-SCA attacks (XGBoost, Random Forest (RF), and Fully Connected Neural Network (FCN)) to recover the secret key coefficients.
- Using the SMAUG-SCA attack, we demonstrate that the secret key coefficients can be recovered with $> 99.9\%$ success probability with less than 10 power traces, with FCN showing the highest attack accuracy.

2 Related Work and Background on SMAUG-T

SMAUG-T is constructed as a key encapsulation mechanism (KEM) by applying a Fujisaki–Okamoto (FO) transform to an underlying public-key encryption (PKE) scheme [1]. The PKE layer defines the core algorithms for key generation, encryption, and decryption, while the KEM layer provides chosen-ciphertext (IND-CCA2) security through hashing, re-encryption, and ciphertext verification.

During encryption, an ephemeral secret is sampled and used to generate a ciphertext consisting of two components: a polynomial vector $\mathbf{c}_1$ and a polynomial $\mathbf{c}_2$, both derived through scaled and rounded polynomial multiplications. Decryption recovers the plaintext by computing a secret-dependent polynomial inner product between $\mathbf{c}_1$ and the long-term secret key $\mathbf{s}$, followed by rounding and decoding. In the KEM construction, encapsulation corresponds to encrypting a random message and deriving a shared secret via hashing, while decapsulation performs the corresponding decryption and ciphertext verification. The initial version of SMAUG-T utilized fast polynomial multiplication by exploiting the sparsity of the non-zero secret-key coefficients. However, this multiplication introduces a timing SCA attack; therefore, the authors switched to constant-time Toom–Cook based multiplication [5]. Although SCA on various PQC algorithms have been demonstrated [8, 9], SCA on the implementation of the KpqC KEM standard SMAUG-T has not been evaluated yet to the best of the authors' knowledge. Since decapsulation repeatedly executes the secret-dependent decryption procedure using attacker-controlled ciphertexts, it constitutes the primary SCA attack surface. In SMAUG-T, this computation is implemented using optimized Toom–Cook and Karatsuba polynomial multiplications, and hence SMAUG-SCA targets these operations.

3 SMAUG-SCA Leakage Analysis

As discussed in the previous section, the target of the attack is the SMAUG-T decapsulation routine where the secret key is multiplied with publicly available ciphertexts. The SMAUG-T decapsulation is implemented on the ARM 32-bit microcontroller, and traces are collected using the Chipwhisperer Lite capture board.

3.1 Test Vector Leakage Assessment

Test Vector Leakage Assessment (TVLA) [3, 10] on the SMAUG-T is performed with fixed vs random ciphertexts. Figure 2 shows that the t-value exceeds the threshold of 4.5 with < 100 traces, revealing considerable data-dependent leakage.

3.2 SMAUG-SCA: Points of Attack

In SMAUG-T, the decapsulation phase performs a secret-dependent polynomial inner-product between the ciphertext component $\mathbf{c}_1$ and the secret key vector $\mathbf{s}$. The ciphertext consists of a polynomial vector $\mathbf{c}_1$, containing two polynomials with 256 coefficients each, and a polynomial $\mathbf{c}_2$ with 256 coefficients. During decapsulation, only $\mathbf{c}_1$ interacts with the secret key, while $\mathbf{c}_2$ is used solely in public scaling and rounding operations. Consequently, the inner-product involving $\mathbf{c}_1$ and the ternary secret polynomials constitutes the primary source of side-channel leakage and forms the target of the attack. This inner-product is implemented through a polynomial accumulation routine using a four-way Toom–Cook multiplication. Each 256-coefficient polynomial is split into four 64-coefficient sub-polynomials, resulting in seven intermediate products computed via Karatsuba multiplication, followed by interpolation and recomposition. Due to the Toom–Cook evaluation and interpolation stages, each secret coefficient contributes to multiple Karatsuba multiplications, producing a long sequence of secret-dependent intermediate values. Since the ciphertext is attacker-controlled and the secret key remains fixed across repeated decapsulation queries, this multiplication routine becomes an ideal target for power side-channel analysis, enabling statistical and machine learning–based recovery of individual secret key coefficients.

3.3 Known Value Correlation & CPA Attack

After confirming the presence of data-dependent leakage using TVLA, Known Value Correlation (KVC) analysis is performed to localize and characterize the leakage source [3, 10]. SMAUG-SCA

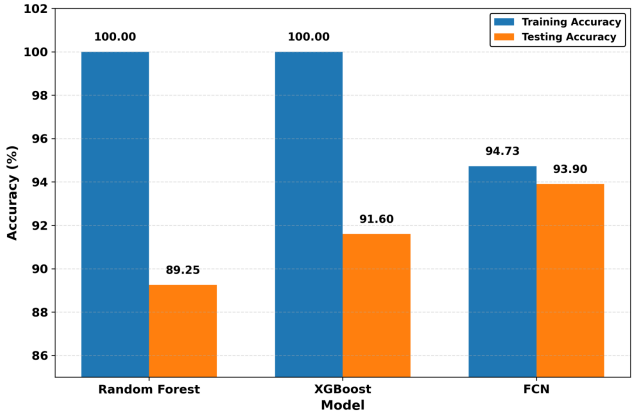


Figure 4: Training and testing accuracy of ML models. FCN achives the highest accuracy with less overfitting.

targets the decapsulation phase, where the ciphertext polynomial coefficients are multiplied with the ternary secret key coefficients during the polynomial inner-product computation. Since the ciphertext is fully known to the attacker, an intermediate value can be modeled as the product of a ciphertext coefficient and a secret key coefficient hypothesis.

Figure 3 shows the correlation spike from the secret-dependent ciphertext–secret key multiplication for both the absolute intermediate value and its hamming weight. However, it is important to note that due to the ternary nature of the secret key coefficients $\{-1, 0, +1\}$, coefficients with value zero exhibit no correlation under conventional Hamming-weight-based correlation power analysis (CPA), as the intermediate value which is a product of the cipher text coefficient and secret key is zero, motivating the use of ML-based SMAUG-SCA attacks.

4 ML-Based SMAUG-SCA Attack

Based on leakage localization from TVLA and KVC, we perform ML-based SMAUG-SCA attacks to recover coefficient-wise secret keys. The attack is demonstrated on a randomly selected coefficient (66th), taking values in $\{-1, 0, +1\}$.

A total of 1000 power traces are collected during decapsulation, with fixed ciphertext and randomized secret key seeds. Traces are labeled by the targeted secret coefficient, with class balancing ensured during collection. The dataset is split into training, validation, and test sets (72:8:20) and normalized prior to training.

4.1 Machine Learning Models

We evaluate two tree-based machine learning models for SMAUG-SCA: RF and XGBoost, both well suited for power SCA due to their ability to capture nonlinear relationships in high-dimensional, noisy traces [7, 11].

In addition, SMAUG-SCA employs a Fully Connected Neural Network (FCN) to model complex leakage patterns [2, 4]. The FCN takes normalized power traces as input and consists of three dense layers (256, 128, 64 neurons) with ReLU activation, followed by a softmax output layer for three-class classification $\{-1, 0, +1\}$. Regularization techniques, including L_2 weight decay, batch normalization, and dropout, are applied to improve generalization. The model is trained using the Adam optimizer with categorical cross-entropy loss.

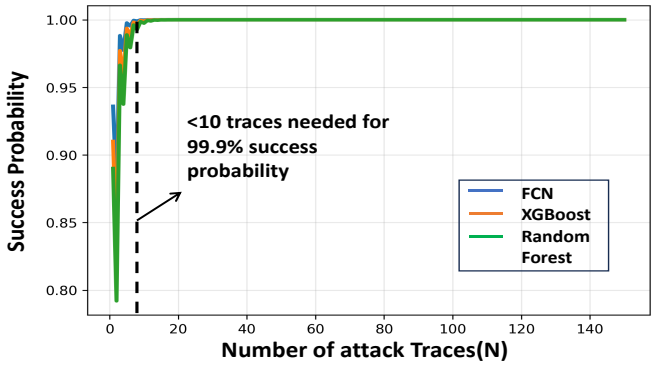


Figure 5: SMAUG-SCA requires < 10 traces to achieve $> 99.9\%$ attack success probability for the different ML models (FCN, XGBoost, RF).

4.2 SMAUG-SCA Attack Accuracy & Success Probability

While RF and XGBoost achieve perfect training accuracy, their test accuracy is lower, indicating mild overfitting. In contrast, FCN exhibits strong generalization, achieving the highest test accuracy, as shown in Figure 4. Utilizing the concept of majority voting [2], Fig. 5 shows that the ML-based SMAUG-SCA attack achieves $> 99.9\%$ success probability of key recovery with < 10 traces.

5 Future Works & Conclusion

This work demonstrates SMAUG-SCA by exploiting secret-dependent polynomial multiplications in the SMAUG-T decapsulation phase. Among the ML models (RF, XGBoost, FCN), FCN generalizes best and achieves the highest test accuracy, enabling secret key recovery with < 10 traces at $> 99.9\%$ success probability.

As part of the future works, to mitigate these side-channel vulnerabilities, countermeasures such as masking and shuffling can be applied. Masking is a provably secure and effective defense, widely used in lattice-based schemes such as SABER and Kyber, and is applicable to Toom–Cook and Karatsuba multiplications in SMAUG-T. Shuffling and dummy operations can further increase attack complexity, albeit with performance overheads.

References

[1] Jung Hee Cheon et al. [n. d.]. SMAUG: the Key Exchange Algorithm based on Module-LWE and Module-LWR. ([n. d.]).
[2] Debayan Das et al. 2019. X-DeepSCA: Cross-Device Deep Learning Side Channel Attack. https://eprint.iacr.org/2019/818 IEEE/ACM DAC 2019.
[3] Deepak et al. 2025. Machine Learning Power Side-Channel Attack on SNOW-V. doi:10.48550/arXiv.2512.21737 IEEE VLSID 2026.
[4] Anupam Golder et al. 2019. Practical Approaches Toward Deep-Learning-Based Cross-Device Power Side-Channel Attack. *IEEE TVLSI* 27, 12 (2019), 2720–2733.
[5] Jung Hee Cheon et al. 2024. SMAUG (-T), Revisited: Timing-Secure, More Compact, Less Failure. *IEEE Access* 12 (2024), 188386–188397. doi:10.1109/ACCESS.2024.3511346
[6] Tendayi Kamucheka et al. 2021. Power-based Side Channel Attack Analysis on PQC Algorithms. https://eprint.iacr.org/2021/1021 Publication info: Preprint. MINOR revision..
[7] Liran Lerman et al. 2015. Template Attacks vs. Machine Learning Revisited (and the Curse of Dimensionality in Side-Channel Analysis). In *COSADE*. Springer, Cham, 20–33. doi:10.1007/978-3-319-21476-4_2
[8] Catinca Mujdei et al. 2022. Side-Channel Analysis of Lattice-Based Post-Quantum Cryptography: Exploiting Polynomial Multiplication. https://eprint.iacr.org/2022/474
[9] Kalle Ngo et al. 2021. A Side-Channel Attack on a Masked IND-CCA Secure Saber KEM. https://eprint.iacr.org/2021/079
[10] Harshit Saurabh et al. 2024. SNOW-SCA: ML-assisted Side-Channel Attack on SNOW-V. https://eprint.iacr.org/2024/428 IEEE HOST 2024.
[11] Léo Weissbart et al. 2021. Systematic Side-channel Analysis of Curve25519 with Machine Learning. https://eprint.iacr.org/2021/944