On the robustness of the ISAP mode against physical attacks

ISAP Team

The Isap mode for lightweight authenticated encryption is designed to provide increased robustness against a large class of implementation attacks, such as Differential Power Analysis (DPA) [14], Simple Power Analysis (SPA) [4], Differential Fault Attacks (DFA) [2], Statistical Fault Attacks (SFA) [11], and Statistical Ineffective Fault Attacks (SIFA) [6] entirely on mode-level. In the following, we briefly explain how the Isap mode achieves this hardening/protection against physical attacks and provide further references to literature. For a detailed discussion of the features of ISAP we refer to the specification [7].

Differential Power Analysis (DPA). In the context of DPA attacks we disscuss the protection of the ISAP mode against 3 different kinds of DPA attacks, i.e., classical DPA-based key recovery attacks, DPA-based plaintext recovery attacks, and tag recovery attacks.

One of the main design goals of Isap is the inherent resilience against classical DPA-based key recovery attacks [14]. This is achieved through the usage of the leakage-resilient re-keying function that derives unique session keys for encryption/authentication from the long term key and the nonce. It can be viewed as a sponge variant of the classical GGM construction [12]. By limiting the absorption rate during re-keying, one can reduce the number of possible inputs to a permutation call per inner part to 2, which renders classical DPA attacks impractical. A more detailed analysis of classic DPA attacks on an Isap-like mode is discussed in Section 4.2 of [10].

Another quite unique feature of Isap's mode in the context of DPA attacks is the fact that it does not enable DPA-based plaintext recovery attacks during authenticated decryption. This is essential in situations such as firmware updates where the plaintexts could carry sensitive information like cryptographic keys. In case of an online/single-pass AEAD scheme, an attacker could query the decryption with a constant nonce and varying ciphertexts, therewith forcing constant key stream blocks that get combined with varying ciphertext blocks. A simple DPA-style attack could then be used to learn the key stream blocks, and thus the corresponding plaintext blocks. Such attacks do not require the extraction of cryptographic keys, but can still be used to undermine the security and integrity of security-critical systems. The two-pass construction of Isap prevents this type of attack by starting the decryption only after the authenticity of the ciphertext and nonce was successfully verified.

Last but not least, we discuss the case of DPA-based message forgery attacks by recovering the correct tag. Given an implementation of authenticated decrytpion without DPA-secure tag verification, it is possible to forge valid messages without knowledge of the master key [1], which is for example critical for secure bootloading [17]. In case of ISAP, such an attack can be prevented without the need of algorithmic countermeasures by utilizing additional permutation calls before the tag verification (as suggested in Section 6.1.4 of [7]). This idea was later analyzed in more detail as the StP (SuKS then PVP) construction in Section 7.2 of [9].

Simple Power Analysis (SPA). The ISAP mode does not provide inherent protection against SPA, however, it was designed such that SPA leakage is reduced as described in Section 6.1.2 of [7]. Besides that, as pointed out by many papers in the past, the exploitation of SPA Leakage becomes considerable harder as the word size of a processor increases. Performing SPA attacks on e.g. 32-bit devices is thus generally considered to require much more sophisticated attacks than in the case of e.g. 8-bit devices [13, 19, 16, 3]. This fact stands especially true in case of hardware implementations that calculate the permutation concurrently on all bits of the state. The resilience of Isap against SPA attacks in such a scenario was analyzed in [18].

Differential Fault Analysis (DFA). DFA [2] attacks exploit the difference between results of repeated executions of cryptographic computations with and without fault injection. During authenticated encryption, fresh nonces ensure that the session keys are unique for each encryption, which precludes DFA attacks.

In the case that the attacker can force multiple queries with the same inputs to Isap (e.g., same ciphertext/nonce/tag during decryption), Isap provides enhanced resilience against the straightforward application of DFA attacks. While Luo et al. [15] show how DFA attacks can be applied to Keccak-based MAC constructions, in the case of Isap, a single fault injection per decryption is not sufficient to learn information about the long-term key. The long term key is only used within IsapRK, which by itself cannot be directly attacked via classical DFA since the attacker never gets to see any output directly. A multi-fault strategy, as outlined in Appendix A of [10], is still possible but requires roughly the quadratic amount of faulted decryptions when compared to the numbers reported in [15]. More importantly, it requires precise combinations of multiple fault injections, both in terms of timing and location, which is considered to be impractical.

Statistical (Ineffective) Fault Attacks (SFA/SIFA). SFA [11] and SIFA [6] are fault attack techniques that are, in contrast to DFA, applicable to many AEAD schemes, including online/single-pass variants, and without assumptions such as nonce repetition or release of unverified plaintext. These attacks are especially interesting since it was shown that they are also applicable to masked implementations, whereas SIFA can even work in cases where masking is combined with typical fault countermeasure techniques [6].

Both attacks have in common that they require the attacker to call a certain cryptographic building block (e.g., permutation) with varying inputs. In principle, SFA can be applicable when AEAD schemes perform a final key addition before generating an output [5], which is not the case for Isap. SIFA, on the other hand, can be used in the initialization phase of almost all AEAD schemes, similarly to what was shown for the Keccak-based AEAD schemes Ketje and Keyak [8]. However, in the case of Isap, the 1-bit rate during IsapRK limits the number of inputs per permutation call to 2 and thus severely limits the capabilities of SIFA, which usually requires several hundred calls with varying inputs [8] in practice. A more detailed analysis of SIFA attacks on an Isap-like mode is discussed in Section 4.2 and Section 4.3 of [10].

References

[1] F. Berti, O. Pereira, T. Peters, and F.-X. Standaert. “On Leakage-Resilient Authenticated Encryption with Decryption Leakages”. In: *IACR Trans. Symmetric Cryptol.* 2017.3 (2017), pp. 271–293.

[2] E. Biham and A. Shamir. “Differential Fault Analysis of Secret Key Cryptosystems”. In: *CRYPTO ’97*. Ed. by B. S. Kaliski Jr. Vol. 1294. LNCS. Springer, 1997, pp. 513–525.

[3] O. Bronchain and F. Standaert. “Breaking Masked Implementations with Many Shares on 32-bit Software Platforms or When the Security Order Does Not Matter”. In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021.3 (2021), pp. 202–234.

[4] S. Chari, J. R. Rao, and P. Rohatgi. “Template Attacks”. In: *CHES*. Vol. 2523. Lecture Notes in Computer Science. Springer, 2002, pp. 13–28.

[5] C. Dobraunig, M. Eichlseder, T. Korak, V. Lomné, and F. Mendel. “Statistical Fault Attacks on Nonce-Based Authenticated Encryption Schemes”. In: *ASIACRYPT 2016*. Vol. 10031. LNCS. 2016, pp. 369–395.

[6] C. Dobraunig, M. Eichlseder, T. Korak, S. Mangard, F. Mendel, and R. Primas. “SIFA: Exploiting Ineffective Fault Inductions on Symmetric Cryptography”. In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018.3 (2018), pp. 547–572.

[7] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, B. Mennink, R. Primas, and T. Unterluggauer. “Isap v2.0”. In: *IACR Trans. Symmetric Cryptol.* 2020.S1 (2020), pp. 390–416.

[8] C. Dobraunig, S. Mangard, F. Mendel, and R. Primas. “Fault Attacks on Nonce-Based Authenticated Encryption: Application to Keyak and Ketje”. In: *SAC 2018*. Vol. 11349. LNCS. Springer, 2018, pp. 257–277.

[9] C. Dobraunig and B. Mennink. “Leakage Resilient Value Comparison with Application to Message Authentication”. In: *EUROCRYPT (2)*. Vol. 12697. Lecture Notes in Computer Science. Springer, 2021, pp. 377–407.

[10] C. Dobraunig, B. Mennink, and R. Primas. “Leakage and Tamper Resilient Permutation-Based Cryptography”. In: *IACR Cryptol. ePrint Arch.* (2020), p. 200. URL: https://eprint.iacr.org/2020/200.

[11] T. Fuhr, É. Jaulmes, V. Lomné, and A. Thillard. “Fault Attacks on AES with Faulty Ciphertexts Only”. In: *FDTC*. IEEE Computer Society, 2013, pp. 108–118.

[12] O. Goldreich, S. Goldwasser, and S. Micali. “How to construct random functions”. In: *J. ACM* 33.4 (1986), pp. 792–807.

[13] M. J. Kannwischer, P. Pessl, and R. Primas. “Single-Trace Attacks on Keccak”. In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2020.3 (2020), pp. 243–268.

[14] P. C. Kocher, J. Jaffe, and B. Jun. “Differential Power Analysis”. In: *CRYPTO ’99*. Ed. by M. J. Wiener. Vol. 1666. LNCS. Springer, 1999, pp. 388–397.

[15] P. Luo, Y. Fei, L. Zhang, and A. A. Ding. “Differential Fault Analysis of SHA-3 Under Relaxed Fault Models”. In: *J. Hardw. Syst. Secur.* 1.2 (2017), pp. 156–172.

[16] M. Medwed, F. Standaert, and A. Joux. “Towards Super-Exponential Side-Channel Security with Efficient Leakage-Resilient PRFs”. In: *CHES*. Vol. 7428. Lecture Notes in Computer Science. Springer, 2012, pp. 193–212.

[17] C. O’Flynn and Z. (Chen. “Side channel power analysis of an AES-256 bootloader”. In: *CCECE*. IEEE, 2015, pp. 750–755.

[18] S. Steinegger and R. Primas. “A Fast and Compact RISC-V Accelerator for Ascon and Friends”. In: *CARDIS*. Vol. 12609. Lecture Notes in Computer Science. Springer, 2020, pp. 53–67.

[19] S. You and M. G. Kuhn. “Single-Trace Fragment Template Attack on a 32-Bit Implementation of Keccak”. In: *CARDIS*. Vol. 13173. Lecture Notes in Computer Science. Springer, 2021, pp. 3–23.