Post-Quantum Authenticated Encryption against Chosen-Ciphertext Side-Channel Attacks

Melissa Azouaoui, Yulia Kuzovkova, Tobias Schneider
and Christine van Vredendaal

NXP Semiconductors
melissa.azouaoui@nxp.com, yulia.kuzovkova_2@nxp.com,
tobias.schneider@nxp.com, christine.cloostermans@nxp.com

Abstract. Over the last years, the side-channel analysis of Post-Quantum Cryptography (PQC) candidates in the NIST standardization initiative has received increased attention. In particular, it has been shown that some post-quantum Key Encapsulation Mechanisms (KEMs) are vulnerable to Chosen-Ciphertext Side-Channel Attacks (CC-SCA). These powerful attacks target the re-encryption step in the Fujisaki-Okamoto (FO) transform, which is commonly used to achieve CCA security in such schemes. To sufficiently protect PQC KEMs on embedded devices against such a powerful CC-SCA, masking at increasingly higher order is required, which induces a considerable overhead. In this work, we propose to use a conceptually simple construction, the $\mathcal{E}t\mathcal{S}$ KEM, that alleviates the impact of CC-SCA. It uses the Encrypt-then-Sign ($\mathcal{E}t\mathcal{S}$) paradigm introduced by Zheng at ISW '97 and further analyzed by An, Dodis and Rabin at EUROCRYPT '02, and instantiates a post-quantum authenticated KEM in the outsider-security model. While the construction is generic, we apply it to the CRYSTALS-Kyber KEM, relying on the CRYSTALS-Dilithium and Falcon signature schemes. We show that a CC-SCA-protected $\mathcal{E}t\mathcal{S}$ KEM version of CRYSTALS-Kyber requires less than 10% of the cycles required for the CC-SCA-protected FO-based KEM, at the cost of additional data/communication overhead. We additionally show that the cost of protecting the $\mathcal{E}t\mathcal{S}$ KEM against fault injection attacks, necessarily due to the added signature verification, remains negligible compared to the large cost of masking the FO transform at higher orders. Lastly, we discuss relevant embedded use cases for our $\mathcal{E}t\mathcal{S}$ KEM construction.

Keywords: Post-Quantum Cryptography · Side-Channel Attacks · Chosen-Ciphertext Attacks · Authenticated Key Exchange

1 Introduction

Over the years, a range of efficient and secure instantiations of cryptographic primitives have been established. In particular for asymmetric cryptography, RSA and ECC are the dominating schemes in practice. However, with the advent of quantum computers the established solutions will no longer provide the desired security. Shor [Sho97] showed that their underlying hardness assumptions can be efficiently broken using a sufficiently powerful quantum computer. To prepare for this threat, the National Institute of Standards and Technology (NIST) has launched a standardization effort for cryptography resistant against quantum computers [Nat]. The goal is to select cryptographic algorithms that perform well in the considered performance metrics, while withstanding any known quantum attack threat. These *Post-Quantum Cryptography* (PQC) schemes and their implementations have become an active area of research in recent years.

One of the use cases where post-quantum cryptography is of interest for embedded devices is secure (firmware) update. If the secure update functionality of a device is not post-quantum secure, all functionality updated with it, including updates to post-quantum cryptography, cannot be trusted since the update might have been compromised by a quantum adversary. One way a secure update can be performed classically is by performing an ECC- or RSA-based key exchange to agree on a symmetric keypair, to then send the update in a second symmetric phase. The update can be made post-quantum secure by switching the key exchange out for a PQC KEM, and ensuring the second phase utilizes a symmetric cipher of sufficient post-quantum security. While the latter is straight-forward, the former poses a great challenge for implementations on constrained devices; the keys are larger, or the performance is lower, especially when physical attacks are in scope.

Indistinguishability against chosen-ciphertext attacks (IND-CCA1), or adaptive chosen-cipertext attacks (IND-CCA2), are common security notions for (post-quantum) cryptographic schemes [RS91]. It ensures that the ciphertext does not leak information on the encrypted message or the secret key when an attacker has access to a decryption oracle for chosen ciphertexts. A slightly weaker notion is that of indistinguishability against chosen-plaintext attacks (IND-CPA), where the adversary instead has control over the plaintext and an encryption oracle.

In the embedded context, although a post-quantum cryptographic scheme can be IND-CPA/CCA secure, this alone does not provide sufficient security. The implementation on a constrained device is an attractive target for physical adversaries that can either passively measure side-channel information or actively disturb the computation to extract sensitive information. Several post-quantum constructions are particularly vulnerable to side-channel attacks that exploit specifically chosen ciphertexts to amplify the observed leakage. This approach, denoted in the following as *Chosen-Ciphertext Side-Channel Analysis* (CC-SCA), has been shown to be a severe threat to even schemes that have countermeasures added to thwart side-channel adversaries [UXT+22].

The core issue for these schemes is the use of the so-called Fujisaki-Okamoto (FO) transform [FO99]. It allows to create an IND-CCA2-secure scheme from its CPA-secure counterpart. The transform adds adequate resistance against a black-box adversary, however does not account for leakage during its computation. In fact, its computation consists of multiple steps processing sensitive values, which allows a side-channel adversary numerous attack avenues. Countermeasures against side-channel attacks on the FO transform, like masking, therefore require many shares and are very costly with regard to performance [ABH+22].

There are alternatives to the FO transform, like the zero knowledge proof techniques presented in [BMV17]. In their solution, each message is encrypted using two independent cryptosystems, and both ciphertexts are sent along with a non-interactive zero-knowledge proof that they correspond to the encryption of the same message under different keys. No such solution is used for known PQC schemes since its instantiation is more challenging, less generic and presumably more expensive than the FO transform. D'Anvers, Orsini and Vercauteren developed alternative ciphertext transformations to the FO transform for lattice-based encryption in [DOV21]. These alternatives are based on error term checking and do not apply to schemes such as NewHope, Kyber and Saber. In the symmetric setting, one way to achieve CCA security with protection against leakage is to use a Message Authentication Code (MAC). The MAC can be computed after encryption with e.g., AES with a pre-shared key and can be used by the receiver to verify the validity of the ciphertext before decryption. In addition, the MAC computation with the shared symmetric key requires side-channel protection. However, in the asymmetric setting of post-quantum cryptography, for many use cases, there is no pre-shared symmetric key available to perform this authentication.

In this work, we propose an alternative approach based on *signcryption*, precisely

Encrypt-then-Sign ($\mathcal{E}t\mathcal{S}$) [Zhe97]. Simply put, a $\mathcal{E}t\mathcal{S}$ scheme adds a signature on top of an IND-CPA-secure encryption to lift the scheme to IND-CCA security in the outsider-security model [ADR02]. We apply this technique in the post-quantum setting and show that for embedded use cases where the outsider-security is realistic, e.g., secure update mechanisms, it can be used as an alternative to the FO transform. The resulting protocol avoids the costly side-channel protection requirements of the FO transform, and thereby provides a significant performance improvement in the considered use cases.

Related works. Research showing how side-channel attacks affect post-quantum cryptography has expanded in the recent years. Timing attacks were first shown to be applicable to lattice-based cryptography by Silverman and Whyte [SW07], and further exploited in [BHLY16, EFGT17]. Cache attacks were shown to be exploitable in [BBK+17]. Power analysis attacks on lattice-based schemes have been shown in a quickly expanding list of works, of which some are able to attack even side-channel protected implementations [XPRO20, RRCB20, HCY19, SRSW20, GJN20, RRCB20].

Work on protecting post-quantum schemes from side-channels is also gaining traction. One of the most well-known countermeasures against side-channel attacks is masking [CJRR99, PR13], which was first applied to a post-quantum ring-LWE (R-LWE) scheme at CHES'15 [RRVV15]. However, the target scheme analyzed considers only a CPA decryption of R-LWE. An initial first-order masking scheme of a complete Ring-LWE KEM including the FO transform was presented at CHES'18 [OSPG18]. An efficient first-order protected version of NIST-submitted Saber [DKR+20] was presented by Beirendonck, D'Anvers, Karmakar, Balasch and Verbauwhede in [BDK+21]. Similarly, first- and higher-order protection was added to NIST-submission Kyber [ABD+19] and was presented in [BGR+21, CGMZ22, DHP+22, FBR+22].

Signcryption schemes, including $\mathcal{E}t\mathcal{S}$, were introduced by Zheng [Zhe97] and its security further analyzed by An, Dodis and Rabin [ADR02]. They specify under which assumptions and security notions of the encryption scheme $\mathcal{E}$ and signature scheme $\mathcal{S}$, the resulting $\mathcal{E}t\mathcal{S}$ scheme is provably IND-CPA or IND-CCA secure. Its translation to post-quantum security notions was presented in [CPPS20] by Chatterjee, Pandit, Puria and Shah. In particular, they show that in the two-user outsider-model, which is our considered model for a secure update mechanism, post-quantum CPA security of $\mathcal{E}$ is amplified in the $\mathcal{E}t\mathcal{S}$ paradigm if the base signature scheme satisfies a stronger security definition.

Recent works combining or investigating the joint use of PQC KEMs and digital signatures include the proposal of Gérard and Merckx [GM18] of a lattice-based signcryption scheme reminiscent of Sign-then-Encrypt schemes, which offers improved bandwidth compared to a straightforward combination of signature and encryption. Schwabe, Stebila and Wiggers present KEMTLS as an alternative to the TLS handshake [SSW20]. KEMTLS uses IND-CCA-secure KEM for server authentication instead of signatures to reduce the bandwidth and the speed of the TLS handshake. In [BFG+21] an asynchronous deniable key exchange is built by combining PQC KEM and a designated verifier signature scheme for the Signal handshake.

Contributions. In contrast to the previously mentioned works, in this paper we take a look at speeding up a *masked post-quantum key exchange* from an embedded device perspective *in the use cases where the communicating parties can be authenticated* and in particular against the threat of powerful CC-SCA. Previous work shows that masking a complete PQC KEM can be very costly. Especially when many shares are necessary, as is the case to protect the FO transform, the performance is greatly impacted. In this work, we provide the following related contributions:

- We propose a conceptually very simple solution, to instantiate post-quantum authenticated encryption, based on the $\mathcal{E}t\mathcal{S}$ construction [Zhe97, ADR02]. The resulting

scheme, that we call $\mathcal{E}t\mathcal{S}$ KEM, has improved resistance against side-channel attacks. This is achieved by replacing the FO transform, which manipulates a large number of sensitive variables, by a signature verification that only uses public data. This improvement comes with a data overhead of one PQC signature.

- We discuss the relevance of the $\mathcal{E}t\mathcal{S}$-based scheme for embedded use cases. The $\mathcal{E}t\mathcal{S}$ construction makes security assumptions that do not work for all use cases. We discuss these and show that, most notably, $\mathcal{E}t\mathcal{S}$ KEM can be applied to secure (over-the-air) update and consider other potential applications.
- We apply the scheme to the CRYSTALS-Kyber PKE and KEM [ABD+19] to illustrate and analyze our proposal. We show that in the $\mathcal{E}t\mathcal{S}$ KEM less components require the application of costly side-channel countermeasures such as masking compared to the standard FO-transform-based KEM (FO KEM). This decreases the cost of CC-SCA protection.
- Finally, we give performance estimates for the $\mathcal{E}t\mathcal{S}$ KEM implementation compared to the FO KEM when combining CRYSTALS-Kyber with either CRYSTALS-Dilithium or Falcon. We show that when 3 or more masking shares are required (which is likely the case for standard microcontrollers), the cost of the $\mathcal{E}t\mathcal{S}$ KEM is less than 10% compared to that of the FO KEM. This is including the impact of signature recomputation, an ad hoc countermeasure against fault injection attacks, and added SPA countermeasures.

2 Background

In this section, we introduce notations used and relevant definitions of security notions. We describe the Kyber KEM [ABD+19] since we use it in following descriptions, illustrations and discussions, but our proposal can be adapted to other PQC KEMs using the FO transform to achieve CCA security, such as Saber [DKRV18].

2.1 Notation

We denote the ring of integers modulo q by $\mathbb{Z}_q$ and the corresponding ring of polynomials $\mathbb{Z}_q[X]/(x^n+1)$ by R_q. We use lowercase letters (e.g., x) to denote elements in R_q; bold lower-case letters (e.g., $\boldsymbol{b}$) represent vectors and bold upper-case letters (e.g., $\boldsymbol{A}$) represent matrices with coefficients in R_q. Sampling x according to a distribution χ is denoted by $x \leftarrow \chi$. Sampling of matrices of polynomials is represented by $\boldsymbol{X} \leftarrow \chi(R^{l_1 \times l_2})$, where all the coefficients of $\boldsymbol{X}$ are sampled independently from the distribution χ. The uniform distribution is denoted by $\mathcal{U}$. We denote the centered binomial distribution as β_η, for a positive integer η.

2.2 Security definitions

2.2.1 Indistinguishability under Chosen-Plaintext Attacks (IND-CPA)

The security of Public Key Encryption (PKE) is defined in the sense of indistinguishability under chosen-plaintext attacks. Formally, security in terms of indistinguishability is presented as a cryptographic game [Sho04, BR06], where a cryptosystem is considered secure, if no adversaries can win the game with probability significantly greater than of random guessing. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary, that runs in two stages and aims to win the $\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}}$ game, described below. In a first stage, $\mathcal{A}$ is given access to an encryption oracle $\mathsf{Enc}()$ to encrypt arbitrary (polynomially bounded) number of messages of its choice. In the second stage, $\mathcal{A}$ submits two distinct fresh messages m_0,

m_1, and gets an encryption of one of the messages, c_b. The adversary's goal is to decide which message m_b is encrypted in a given ciphertext:

Game $\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}}$:

$(pk, sk) \leftarrow \mathsf{KeyGen}()$
$b \leftarrow \{0, 1\}$
$(m_0, m_1) \leftarrow \mathcal{A}(pk)$
$c_b \leftarrow \mathsf{Enc}(pk, m_b)$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}()}(pk, c_b)$
return $b \stackrel{?}{=} b'$.

A PKE is considered IND-CPA-secure, if for all efficient adversaries $\mathcal{A}$ there exists some negligible function $\mathsf{negl}(n)$ of the security parameter n, such that the advantage of $\mathcal{A}$ in winning the $\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}}$ game is given by:

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) = \left| \Pr\left(\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}} = 1\right) - \frac{1}{2} \right| < \mathsf{negl}(n).$$

2.2.2 Indistinguishability under (adaptive) Chosen-Ciphertext Attacks (IND-CCA)

The standard security notion for KEMs is indistinguishability under (adaptive) chosen-ciphertext attacks [RS91]. Similarly to IND-CPA, an adversary is given access to an encapsulation oracle $\mathsf{Encaps}()$ throughout the attack, such that it can encapsulate an arbitrary number of keys of its choice. In addition, the attacker is given access to a decapsulation oracle $\mathsf{Decaps}()$. IND-CCA-security provides stronger security guarantees compared to IND-CPA and is formalized in the following game:

Game $\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}$:

$(pk, sk) \leftarrow \mathsf{KeyGen}()$
$b \leftarrow \{0, 1\}$
$K'_1 \leftarrow \mathcal{K}$
$(c', K'_0) \leftarrow \mathsf{Encaps}(pk)$
$b' \leftarrow \mathcal{A}^{\mathsf{Decaps}()}(pk, c', K'_b)$
return $b \stackrel{?}{=} b'$.

A KEM is considered IND-CCA-secure, if for all efficient adversaries $\mathcal{A}$ the probability of winning the $\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}$ game is negligible. More precisely, given some negligible function $\mathsf{negl}(n)$ of the security parameter n:

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) = \left| \Pr\left(\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}} = 1\right) - \frac{1}{2} \right| < \mathsf{negl}(n).$$

2.3 CRYSTALS-Kyber KEM

The PKE of Kyber consists of three operations: key generation, encryption and decryption, given in Algorithms 1, 3 and 2, respectively.

Algorithm 1 Kyber.CPAPKE.KeyGen()

Ensure: Public key $pk = (seed_{\boldsymbol{A}}, \boldsymbol{b})$,
secret key $sk = \boldsymbol{s}$
1: $(seed_{\boldsymbol{A}}, \sigma) \leftarrow \mathcal{U}(\{0,1\}^n) \times \mathcal{U}(\{0,1\}^n)$
2: $\boldsymbol{A} \leftarrow \mathcal{U}(R_q^{k \times k}; seed_{\boldsymbol{A}})$
3: $(\boldsymbol{s}, \boldsymbol{e}) \leftarrow \beta_{\eta_1}(R_q^k; \sigma) \ \times \ \beta_{\eta_1}(R_q^k; \sigma)$
4: $\boldsymbol{b} := \boldsymbol{A} \cdot \boldsymbol{s} + \boldsymbol{e}$
5: **return** $(pk, sk) = ((seed_{\boldsymbol{A}}, \boldsymbol{b}), \boldsymbol{s})$

Algorithm 2 Kyber.CPAPKE.Dec(sk, c)

Require: Secret key $sk = \boldsymbol{s}$,
ciphertext $\boldsymbol{c} = (\boldsymbol{c_1}, c_2)$
Ensure: Message m
1: $\boldsymbol{u} \leftarrow \mathsf{Decompress}_q(\boldsymbol{c_1}, d_u)$
2: $v \leftarrow \mathsf{Decompress}_q(c_2, d_v)$
3: $m = \mathsf{Compress}_q(v - \boldsymbol{s} \cdot \boldsymbol{u}^T, 1)$
4: **return** m

Algorithm 3 Kyber.CPAPKE.Enc(pk, m, σ)

Require: Public key $pk = (seed_{\boldsymbol{A}}, \boldsymbol{b})$,
message m,
random coins σ
Ensure: Ciphertext $\boldsymbol{c} = (\boldsymbol{c_1}, c_2)$
1: $\boldsymbol{A} \leftarrow \mathcal{U}(R_q^{k \times k}; seed_{\boldsymbol{A}})$
2: $\boldsymbol{s'} \leftarrow \beta_{\eta_1}(R_q^k; \sigma)$
3: $(\boldsymbol{e_1}, e_2) \leftarrow \beta_{\eta_2}(R_q^k; \sigma) \ \times \ \beta_{\eta_3}(R_q; \sigma)$
4: $\boldsymbol{u} \leftarrow \boldsymbol{A} \cdot \boldsymbol{s'} + \boldsymbol{e_1}$
5: $v \leftarrow \boldsymbol{b}^T \cdot \boldsymbol{s'} + e_2 + \mathsf{Decompress}_q(m)$
6: $\boldsymbol{c_1} \leftarrow \mathsf{Compress}_q(\boldsymbol{u}, d_u)$
7: $c_2 \leftarrow \mathsf{Compress}_q(v, d_v)$
8: **return** $\boldsymbol{c} = (\boldsymbol{c_1}, c_2)$

The IND-CPA-secure PKE scheme in the previous section can be converted into an IND-CCA secure KEM by applying an appropriate transformation. Kyber and many lattice-based KEMs use a post-quantum variant of the Fujisaki-Okamoto (FO) transform [FO99] by Hofheinz, Hövelmanns and Kiltz [HHK17], however other transformations could be used to achieve CCA security [BMV17, RS91].

The resulting KEMs consist of a triplet of operations (KeyGen, Encaps, Decaps), given in Algorithms 4, 5 and 6, respectively. The CCA-transformation requires access to three hash functions $\mathcal{G}$, $\mathcal{H}$, $\mathcal{H}'$, modeled as random oracles, as well as the PKE scheme CPAPKE = (KeyGen, Enc, Dec). The only difference is the instantiation of these functions. In Kyber the hash-functions are instantiated with different symmetric primitives, based on the SHA3 standard. The key generation is similar to the one for Kyber.CPAPKE, with the difference that the secret key sk also includes the public key pk, the hash of pk and a secret random seed z. During encapsulation, a ciphertext $\boldsymbol{c}$ is returned together with a shared key K, where $\boldsymbol{c}$ is obtained by encrypting a random message m, sampled from the uniform distribution, and K is derived by hashing together the message, the public key and the ciphertext.

To achieve the NIST security levels, CRYSTALS-Kyber has three parameter sets: Kyber512, Kyber768 and Kyber1024 in order of increasing security. The corresponding parameter sets for each version are provided in Table 1.

Table 1: Kyber parameter sets.

	NIST security level	n	k	q	η_1	η_2	(d_u, d_v)
Kyber512	1	256	2	3329	3	2	(10, 4)
Kyber768	3	256	3	3329	2	2	(10, 4)
Kyber1024	5	256	4	3329	2	2	(11, 5)

Algorithm 4 Kyber.CCAKEM.KeyGen()

Ensure: Public key $pk = (seed_{\boldsymbol{A}}, \boldsymbol{b})$,
secret key $sk = (\boldsymbol{s}, \mathcal{H}(pk), pk, z)$
1: $z \leftarrow \mathcal{U}(\{0,1\}^n)$
2: $((seed_{\boldsymbol{A}}, \boldsymbol{b}), \boldsymbol{s}) \leftarrow$ Kyber.CPAPKE.KeyGen()
3: $sk = (s, pk, \mathcal{H}(pk), z)$
4: **return** $(pk, sk) = ((seed_{\boldsymbol{A}}, \boldsymbol{b}), (\boldsymbol{s}, pk, \mathcal{H}(pk), z))$

Algorithm 5 Kyber.CCAKEM.Encaps(pk)

Require: Public key $pk = (seed_{\boldsymbol{A}}, \boldsymbol{b})$
Ensure: Ciphertext $\boldsymbol{c}$, key K
1: $m \leftarrow \mathcal{U}(\{0,1\}^n)$
2: $(\bar{K}, r) := \mathcal{G}(m \mid\mid \mathcal{H}(pk))$
3: $\boldsymbol{c} :=$ Kyber.CPAPKE.Enc$(pk, m; r)$
4: $K := \mathcal{H}'(\bar{K} \mid\mid \mathcal{H}(\boldsymbol{c}))$
5: **return** $(\boldsymbol{c}, K)$

Algorithm 6 Kyber.CCAKEM.Decaps(sk, c)

Require: Ciphertext $\boldsymbol{c}$,
secret key $sk = (\boldsymbol{s}, pk, \mathcal{H}(pk), z)$
Ensure: Key K
1: $m' :=$ Kyber.CPAPKE.Dec$(\boldsymbol{s}, \boldsymbol{c})$
2: $(\bar{K}', r') := \mathcal{G}(m' \mid\mid \mathcal{H}(pk))$
3: $\boldsymbol{c'} :=$ Kyber.CPAPKE.Enc$(\boldsymbol{pk}, m'; r')$
4: **if** $\boldsymbol{c} = \boldsymbol{c'}$ **then**
5: **return** $K := \mathcal{H}'(\bar{K}' \mid\mid \mathcal{H}(c))$
6: **else**
7: **return** $K := \mathcal{H}'(z \mid\mid \mathcal{H}(\boldsymbol{c}))$
8: **end if**

2.4 Side-channel security notions

In this section, we introduce the main side-channel security definitions and notions that we use in the remainder of this paper to define the protection profiles for the $\mathcal{E}t\mathcal{S}$ KEM and the FO KEM.

SPA. Simple Power Analysis (SPA) analyses a limited number of measurements to extract a secret value. It has been used to attack both symmetric and asymmetric cryptographic primitives, and has been shown to be particularly powerful for some post-quantum schemes exploiting chosen ciphertext leakage [XPRO20]. In its most extreme variant, the attack is limited to one single trace and specific attack strategies are employed to maximize the extraction of sensitive information [KPP20]. Note that in some scenarios, it is possible to repeat the measurement with the same inputs and intermediates. This is used to average the traces and significantly reduce the noise in the measurements. So while an SPA attacker might have access to a large number of traces, the amount of distinct leakages is still limited. Therefore, countermeasures against this type of attack usually do not rely on masking, but rather on more cost-efficient shuffling [HOM06] or, if possible, exploit parallel leakages [BMPS21]. Note that in case of CC-SCA on Kyber, the SPA is still very powerful and requires costly protection to achieve the desired security level [ABH+22].

DPA. In contrast to SPA, a Differential Power Analysis (DPA) adversary can measure the leakage of a large number of different intermediate values. This enables very powerful attacks [KJJ99] to extract long-term secret values and, therefore, requires costly protection measures to thwart them. Commonly, masking [CJRR99, PR13] is used, sometimes in combination with other countermeasures, e.g., shuffling, to increase the noise level.

Leveling. There have been some works that try to level the protection profile of a target scheme [ABH+22, BBC+20]. Instead of protecting every operation at the maximum level, e.g., with strong DPA countermeasures, the underlying algorithm is analyzed and protected at different levels. The parts that leak about ephemeral secrets, which cannot be targeted with DPA, are only hardened using more cost-efficient SPA countermeasures. This enables more efficient protected implementations, especially for schemes that have been designed with leveling in mind. Azouaoui et al. [ABH+22] have shown that for

standard Kyber leveling protection is negligible as all parts need to be protected with costly countermeasures given the potency of the CC-SCA SPA on the FO transform. In this work, we show that by relying on a public signature verification check, it is possible (for some use cases) to prevent CC-SCA and exploit leveling to significantly speed-up protected implementations.

2.5 Chosen-ciphertext SCA on the FO transform

The FO transform is a simple and well suited approach for lattice-based PQC PKE to reach standard CCA security. However, several recent works showed that its use still leaves a very powerful attack vector when physical attacks are considered [RRCB20, XPRO20, UXT+22, NDGJ21]. In the following, we first provide a brief description of the FO transform as used in Kyber. Then, we give a short description of chosen-ciphertext side-channel attacks on the FO transform, that we refer to for conciseness as CC-SCA. Finally, we highlight recent results in the literature assessing and improving the cost of protecting PQC KEM implementations against these attacks [ABH+22, BC22].

2.5.1 Fujisaki-Okamoto transform in Kyber

We illustrate the basic working of the FO transform in Figure 1. The core idea is to check the validity of a decrypted message m' in the decapsulation phase by performing re-encryption. The obtained candidate ciphertext $\boldsymbol{c'}$ is then compared with the original ciphertext $\boldsymbol{c}$. If both ciphertexts are equal, the session key K is derived from the message m', the ciphertext c and the hash of the public key pk, otherwise a pseudo-random string $K = \mathcal{H}'(z, \mathcal{H}(c))$ is returned. Since the decapsulation never indicates failure explicitly (e.g., by returning a failure symbol $\perp$), the rejection of malformed ciphertexts is implicit. The FO transform is used in many PQC schemes because of its simplicity and efficiency.

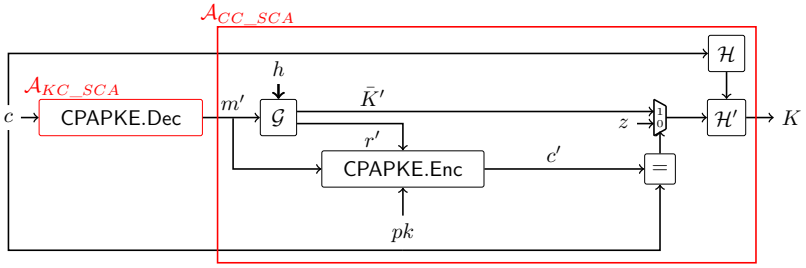


Figure 1: FO-transform-based CCAKEM.Decaps (based on [ABH+22]).

2.5.2 Attack description

While classic chosen-ciphertext attacks are not possible on CCA-secure KEMs thanks to the FO transform, the attacks presented in [RRCB20, XPRO20, UXT+22, NDGJ21] are able to use the side-channel leakage of the FO transform computation to target only the CPA-secure encryption. To do so, the adversary carefully crafts ciphertexts such that one bit of the decrypted message m depends on a single secret key coefficient. Since this message m is used as input for the deterministic re-encryption, the adversary then only has to distinguish between an encryption of 0 or 1 given leakage of the re-encryption, which includes a large number of leaking intermediates. The number of traces required for successful attacks on both unprotected and masked implementations are of the order of a few thousands for many PQC KEMs (see the results of Ueno et al. [UXT+22], Table 7).

2.5.3 The cost of protecting the FO transform

Azouaoui et al. [ABH+22] initiate a study in modeling attacks on lattice-based KEMs. In particular, the authors derive a shortcut formula to approximate the minimal number of traces required for a successful worst-case (exploiting all intermediates) chosen-ciphertext attack. The approximation is parameterized by the number of shares and the noise level to assess the impact of the masking countermeasure. For instance, based on [ABH+22], Figure 2, to achieve resistance against an attack with 10^6 traces for standard 32-bit MCUs, high-order masking with $d \geq 7$ shares is required. This entails approx. 350 million clock cycles for an ARM Cortex-M4 based implementation of the Kyber.CCAKEM decapsulation ([ABH+22], Appendix A, Table 2).

3 Authenticated key encapsulation against CC-SCA

In the previous section, we have recalled that while the FO transform grants CCA security to PQC KEMs like Kyber, it comes with significant drawbacks with respect to its side-channel security, and in particular the great cost its protection against such attacks implies. In this section, we introduce a different construction based on the Encrypt-then-Sign ($\mathcal{E}t\mathcal{S}$) method studied by An, Dodis and Rabin and shown to provide CCA security to CPA-secure PKE [ADR02].

We first describe the relevant security notions for the signcryption of [ADR02]. We then introduce the $\mathcal{E}t\mathcal{S}$ KEM in Section 3.2.1, which is a straightforward application of signcryption to the post-quantum setting. The application areas that benefit from the $\mathcal{E}t\mathcal{S}$ KEM are discussed in Section 3.2.2. We then discuss the side-channel security of the $\mathcal{E}t\mathcal{S}$ KEM and how it compares to the standard FO KEM in Section 3.2.3.

3.1 The Encrypt-then-Sign paradigm

In this section, we first introduce the relevant security notions necessary to define the security of the Encrypt-then-Sign paradigm. We will end with the security guarantees and theorems relevant for this work.

Signcryption. We denote the sender by S and the receiver by R. We assume S uses signcryption, i.e., a scheme in which a message m is first encrypted by an encryption scheme $\mathcal{E}$ and then signed by a signature scheme $\mathcal{S}$ as $u = \mathsf{SigEnc}(m)$. R can then verify and decrypt with a deterministic de-signcryption algorithm $m = \mathsf{VerDec}(u)$. In this setting, beyond the integrity and confidentiality of the message, we would like to protect S's authenticity and R's privacy.

IND-gCCA2-security. In Section 2.3 we discussed the IND-CCA2 security of the Kyber KEM. Generalized CCA2 security (gCCA2) was introduced in [ADR02] (also called 'benign malleability' in [Sho01]) and offers a slightly weaker notion of security. It is defined as having some relation $\mathcal{R}$ for which it holds that for distinct ciphertexts e_1, e_2, if $\mathcal{R}(e_1, e_2) = true$, then $Dec(e_1) = Dec(e_2)$. Such a relation $\mathcal{R}$ is called a *decryption-respecting* relation.

An example is to append a ciphertext with an arbitrary byte, which is ignored during decryption. This cipher is then not CCA2 secure, since the ciphertext can be adapted, but it can be considered secure in almost all use cases of CCA2 encryption, since the adaptation is 'benign'. Note that since the notion of gCCA2 security is a relaxation of CCA2 security, any IND-CCA2 secure encryption scheme is also gCCA2 secure.

UF-NMA/CMA-security. For signature schemes, UnForgeability against No Message Attack (UF-NMA) security describes the notion in which the adversary $\mathcal{A}$ attempts to

create a forged signature from the scheme's public key without accessing a signing oracle. A slightly different notion is that of UnForgeability against Chosen Message Attack (UF-CMA). In this setting $\mathcal{A}$ can make queries to a signing oracle and must forge the signature of a previously unqueried message. UF-CMA and UF-NMA security have been shown to be tightly equivalent in certain settings like deterministic signature schemes [KLS18].

In the context of signature schemes, we can also distinguish weak UF-CMA-security (wCMA) and strong UF-CMA-security (sCMA). The weak case is equivalent to the description above, where an adversary wants to forge the signature of a previously unqueried message. In the strong case, the adversary is deemed successful even when they forge a previously queried message, as long as the signature differs from the queried result.

Insider- vs. outsider-security. The third security notion we need is the distinction between insider- and outsider-security. In the outsider-security setting, we assume that $\mathcal{A}$ is privy only to public information, i.e., the public keys of S and R, pk_S and pk_R, and oracle access to the functionalities of S and R. Specifically, they can query (the functionality of) S, a signed encryption u of a chosen message m (i.e., the *signcryption oracle* computing $u = \mathsf{SigEnc}(m)$). Similarly, they can query (the functionality) of R by providing a signed encryption u and receiving the result m, which could be $\perp$ (i.e., the *de-signcryption oracle* computing $m = \mathsf{VerDec}(u)$). This setting is called outsider-security, because it aims at an adversary that is outside of the protocol.

The stronger notion of insider-security also includes the option that $\mathcal{A}$ is R or S. It aims to protect S's authenticity (respectively, R's privacy) even in the case that $\mathcal{A}$ is using the system as R (respectively, S).

Encrypt-then-Sign ($\mathcal{E}t\mathcal{S}$) security. Given these notions, we have the following theorem on the security of signcryption.

> **Theorem 1** ([ADR02]). *If $\mathcal{E}$ is IND-CPA-secure, and $\mathcal{S}$ is UF-CMA-secure, then $\mathcal{E}t\mathcal{S}$ is INDgCCA2-secure in the Insider- and UF-CMA-secure in the Outsider-security model.*

We see that under security assumptions on the schemes $\mathcal{E}$ and $\mathcal{S}$, the signcryption scheme $\mathcal{E}t\mathcal{S}$ is also secure (in specific security models). In Section 3.2, we will leverage this theorem for secure communication in specific use cases.

Quantum security of $\mathcal{E}t\mathcal{S}$. In [CPPS20] the security of signcryption was shown under the extension of the security model to include a quantum adversary. In particular, they show that in the outsider-security setting, the post-quantum CPA security is amplified with $\mathcal{E}t\mathcal{S}$ if the base signature scheme satisfies slightly stricter security definitions.

> **Theorem 2** (result from [CPPS20]). *If a post-quantum PKE $\mathcal{E}$ is pqIND-CPA-secure, and a post-quantum signature scheme $\mathcal{S}$ is (w/s)UF-CMA-secure, then $\mathcal{E}t\mathcal{S}$ is IND(-g/-)qCCA2-secure and UF-qCMA-secure in the Outsider-security model.*

Here the pq and q notation denote the CPA/CCA2 security in the quantum setting. Note that in the insider-security setting, pqIND-CPA-security of the PKE does not suffice, and IND-qgCCA security is required.

3.2 Leveraging $\mathcal{E}t\mathcal{S}$ to thwart CC-SCA

3.2.1 Scheme description

We first describe the $\mathcal{E}t\mathcal{S}$ key encapsulation scheme in Figure 2. We refer to the encapsulator as the server and to the decapsulator as device, to highlight that in the relevant use cases of this scheme, power-based side-channel attacks can only be a concern on the embedded device's side. In the following, we detail the steps of the scheme:

- First, a signing keypair (sk_s, pk_s) is generated by the server and shared with the device. The dashed horizontal line and gray background emphasizes that this step can be performed off-line and the public key can be pre-provisioned onto the device. Otherwise, a root certificate is pre-provisioned and (possibly ephemeral) signing keys can be generated by the server and verified by the device given their corresponding certificate.
- Next, the KEM key generation and encapsulation are performed sequentially by the device and the server, respectively. However, in this new construction, the ciphertext c is signencrypted (SigEnc) using the device's public key pk_d and the server's secret key sk_s. The ciphertext along with its signature σ are transmitted to the device.
- On receiving the ciphertext and its signature, the device starts the verification and decryption process (VerDec). Prior to initiating any decryption using sk_d, the device first authenticates the ciphertext's source, by verifying its signature σ, using the server's authenticated public key pk_s. If the ciphertext is verified, then it is decrypted and the shared key is derived from the decrypted message m[1]. Otherwise, an implicit rejection is performed as in the original KEM.

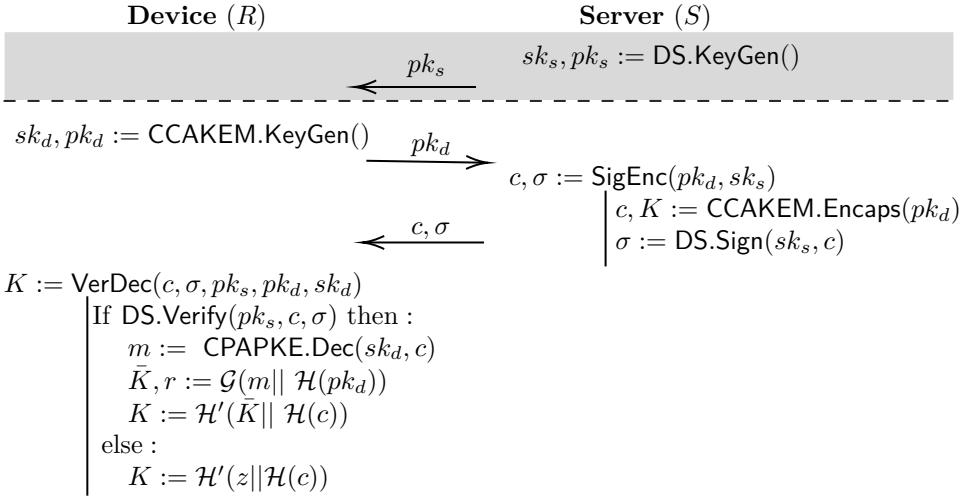


Figure 2: KEM construction using $\mathcal{E}t\mathcal{S}$ to achieve CCA security.

Security of the $\mathcal{E}t\mathcal{S}$ KEM. We see that the presented $\mathcal{E}t\mathcal{S}$ KEM is a direct application of the $\mathcal{E}t\mathcal{S}$ scheme of [ADR02]. Therefore, by Theorem 2, if the used PKE is pqIND-CPA-secure and the used signature scheme DS is strongly UF-CMA-secure, then the $\mathcal{E}t\mathcal{S}$ scheme

[1]The random coins r are no longer required since no re-encryption is performed. The function $\mathcal{G}$ can be modified on the device's side to only return $\bar{K}$

of Figure 2 is IND-qCCA2-secure and UF-qCMA-secure in the outsider-security model. If a weakly UF-CMA-secure signature scheme is used instead, the $\mathcal{E}t\mathcal{S}$ scheme can be proven IND-qgCCA2-secure.

Two-user vs. Multi-user setting. It should be noted that the $\mathcal{E}t\mathcal{S}$ KEM can be extended to a multi-user setting. In this case, the concept of identity needs to be introduced to the scheme to be able to differentiate different actors in the communication. In [ADR02], this is achieved by adding the sender's identity (which can potentially be their public key) to the encrypted message, the receiver identity to the signed message, and having R output $\perp$ if the identities are not as expected.

Although we focus in this work on the two-user setting, we remark that the scheme of Figure 2 will suffer a slight performance impact from allowing multiple users. Since the identities need to be included in the ephemeral key K encryption, this will increase the size of the message and therefore in the worst-case increase the encryption time. It might be possible to (non-trivially) include the identities in a manner maintaining the message length, but we leave this aspect and the applications of $\mathcal{E}t\mathcal{S}$ authenticated post-quantum CCAKEM in the multi-user setting for future research.

3.2.2 Applications

In Section 3.2.1, we presented a pq$\mathcal{E}t\mathcal{S}$ KEM scheme that is CCA2 secure in the outsider-security model. It offers significant advantages to the 'standard' PQC KEM with regard to side-channel protection, but only works for use cases where the outsider-security model holds. In this section, we discuss some possible application areas.

Secure update mechanism. Secure encrypted updates are critical to ensure that a device is running in a secure manner with optimal performance throughout its lifecycle. Firmware updates are often administered locally via a network, or Over-the-Air (OTA). During the update process an embedded device is in its most vulnerable state; if an adversary is able to compromise the content of the update, it can render the device insecure for the remainder of its lifecycle. Therefore, a secure update protocol is essential.

Different strategies can be taken to securely update a device. The provisioning method, the updater scheme and the underlying cryptography can differ greatly per use case. However, a high-level depiction of an update protocol is depicted in Figure 3. To perform the update in a post-quantum secure manner, a general strategy can be to perform a PQC Key EXchange or KEM first, to agree on a shared secret key, and then send the encrypted update by way of symmetric cryptography. In many cases, this is much faster than sending the entire update asymmetrically encrypted and signed.

We notice that secure update is an excellent candidate to apply the pq$\mathcal{E}t\mathcal{S}$ KEM scheme. Firstly, if a KEX/KEM mechanism is used, it needs to be protected against side-channel attacks and in particular against CC-SCA. If not, the device is vulnerable against an adversary tampering with the update, and thereby reducing the security. Secondly, the assumptions made in the outsider-security model hold. For a firmware update both parties are trusted, and the necessary digital signature certificates can be provisioned in Step 1 of Figure 3. Therefore, if the pq$\mathcal{E}t\mathcal{S}$ KEM is applied, Theorem 2 ensures its post-quantum security[2].

When we apply the pq$\mathcal{E}t\mathcal{S}$ KEM scheme, it replaces Step 4 in Figure 3. The costly-to-mask CPAPKE.Enc is replaced with a PQC digital signature. This does mean that the size of the initial transaction grows; PQC digital signatures that are well-suited for embedded applications usually have signature sizes of thousands of bytes. However, since

[2]Note that outsider-security is even a strong notion here; often the public keys might not be readily available to an adversary.

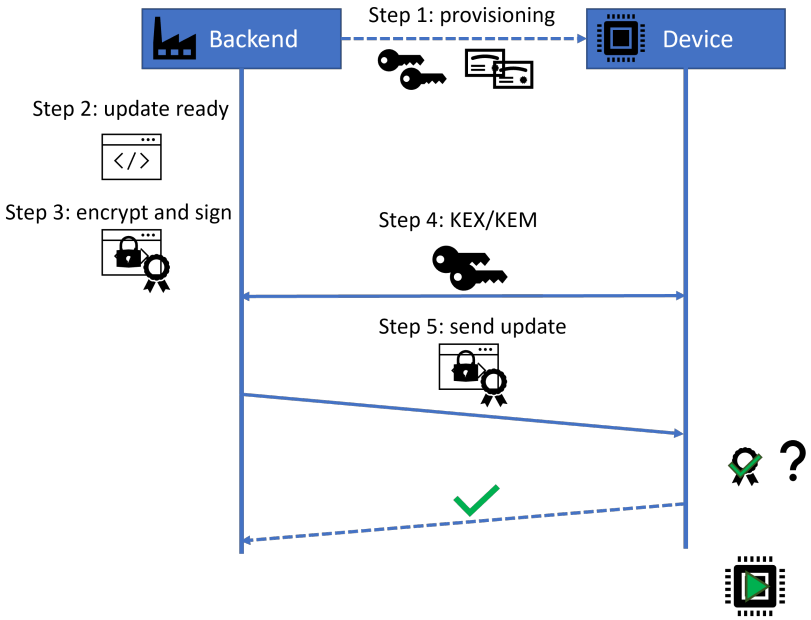


Figure 3: General device secure update mechanism flow.

the subsequent transaction (the sending of the firmware update) also consists of sending large amounts of data, the DS size is negligible for most use cases. In Section 4, we will show that using the pq$\mathcal{E}t\mathcal{S}$ KEM scheme in a secure update context can improve performance by at least a factor 10 (for 3 shares and upwards).

A concrete example of such an update protocol is the SUIT (Software Updates for Internet of Things) solution [MTBM21]. The PQC considerations related to SUIT without firmware encryption were recently analyzed by Banegas et al. [BZH+21]. The version of SUIT with firmware encryption [THM22] uses a HPKE (Hybrid Public Key Encryption) [BBLW22] to establish a shared symmetric key to encrypt/decrypt the firmware image. This HPKE can be instantiated with a PQC CCA-secure KEM, hence when side-channel attacks are in scope the $\mathcal{E}t\mathcal{S}$ KEM can be used for this purpose as well.

Other applications. Although secure update is our main focus in this work, there are other (embedded) applications of the pq$\mathcal{E}t\mathcal{S}$ KEM scheme. One area would be that of secure element to MCU communication. For this communication the Secure Channel Protocol [Glo] can be utilized which, similarly to the secure update, starts with a KEX or KEM to establish the session keys. In this case, the secure element is a trusted party and the outsider-security model applies. This communication also needs to be side-channel-secured. Since the secure element often acts as a trust anchor for the System-on-Chip, its compromise means the entire device is compromised. This would therefore be a good candidate to apply a pq$\mathcal{E}t\mathcal{S}$ KEM scheme, although it has to be taken into account that the resulting protocol is not standardized (yet).

A second area we see applications for the pq$\mathcal{E}t\mathcal{S}$ KEM scheme is in edge computing for the Internet-of-Things. This is in essence the simple concept of performing computations at the edge node of a network instead of in the cloud. This improves security and privacy since less data is sent over an Internet connection while at the same time being able to offload heavier computation like machine learning from more-restrained embedded devices. In such a network, the edge node acts as a trusted party and often it is not necessary for all devices to communicate pairwise: only communication with the edge node is necessary,

and therefore the two-user pq$\mathcal{EtS}$ KEM scheme can be applied. This can be generalized to a multi-user setting where all devices do communicate, like a private network, however this does slow down the protocol a little, as was discussed in Section 3.2.1.

In general, whenever power-based side-channel attack protection is required on the decapsulator/receiver's side and not on the encapsulator/sender's side, and where the outsider-security model applies the use of a pq$\mathcal{EtS}$ KEM scheme can be considered.

3.2.3 Side-channel security of the $\mathcal{EtS}$ KEM

To argue about the side-channel security and the different levels of side-channel protections necessary for all the components of the $\mathcal{EtS}$ KEM, we introduce the two main SCA adversaries on KEMs:

- $\mathcal{A}_{KC-SCA}$: The Known-Ciphertext (KC) adversary can only provide known, but valid ciphertexts to the decapsulation operation. This adversary can mainly target the decryption of the ciphertext c, which depends explicitly on the secret key sk_d as shown on Figure 1.
- $\mathcal{A}_{CC-SCA}$: The Chosen-Ciphertext (CC) adversary can craft their own specifically chosen ciphertexts and input them to the decapsulation operation. This adversary can force the whole FO transform computation, including the long re-encryption, to depend on a single bit of the secret key, as shown on Figure 1.

The goal of both adversaries is to extract either the long-term secret sk_d or the ephemeral encapsulated key m (or the secret keys derived from it, K' and K). Sticking to our chosen use case, the key K is not explicitly returned to the adversary but rather further used internally to decrypt the update image. In accordance with [BDK+21, BGR+21], we exclude the protection of the value z from our analysis[3]. It is obvious, that $\mathcal{A}_{CC-SCA}$ is strictly stronger than $\mathcal{A}_{KC-SCA}$, which leads to costly protection requirements for CCAKEM.Decaps. In the following, we show that for our proposed scheme, the two adversaries are equivalent and, thus, it avoids the costly protection overhead.

For both the $\mathcal{EtS}$ KEM and the standard FO KEM, we analyze each step of the decapsulation computation and argue about their protection requirements, i.e., SPA, DPA or no protection required, against the two SCA adversaries. The high-level intuition is that DPA protection is commonly more costly than SPA protection (or no protection at all) and it is therefore beneficial to limit the number of modules that require this level of protection. Note that we do not distinguish between SPA with and without averaging like some prior works [BBC+20], as in our scenario averaging is always possible.

The FO KEM. The protection level assignment for an FO-based KEM, following its description in Figure 4, is provided in Table 2, and we provide some rationale about the coloring in the following. As the initial CPAPKE.Dec(sk_d, c) manipulates the long-term secret sk_d together with an adversary-controlled input c, it requires DPA protection for both cases. For $\mathcal{A}_{KC-SCA}$, the following intermediates are both mostly unknown, and more importantly independent of the long-term secret sk_d. Therefore, most require only SPA protection. Since the comparison will be true up to a negligible failure probability of the underlying scheme, i.e., $c' = c$, it can be made public and does not require dedicated SCA protection. $\mathcal{A}_{CC-SCA}$ can craft specific ciphertexts such that m' and the values derived from it leak information about the long-term secret sk_d. Therefore, it is necessary to protect the modules processing these values with strong protection measures. Note that the comparison for these chosen ciphertext is always false up to a negligible failure

[3] If this should be considered, it would require DPA protection for the key derivation involving z due to the $\mathcal{A}_{CC-SCA}$ adversary controlling the value of $\mathcal{H}(c)$ when $\mathcal{H}'(z||\mathcal{H}(c))$ is computed.

probability. Therefore, for these inputs K is always derived from z and the public value c, which can be left unprotected. The key derivation based on K' is only computed for valid ciphertexts and requires only SPA protection as in the case of $\mathcal{A}_{KC-SCA}$.

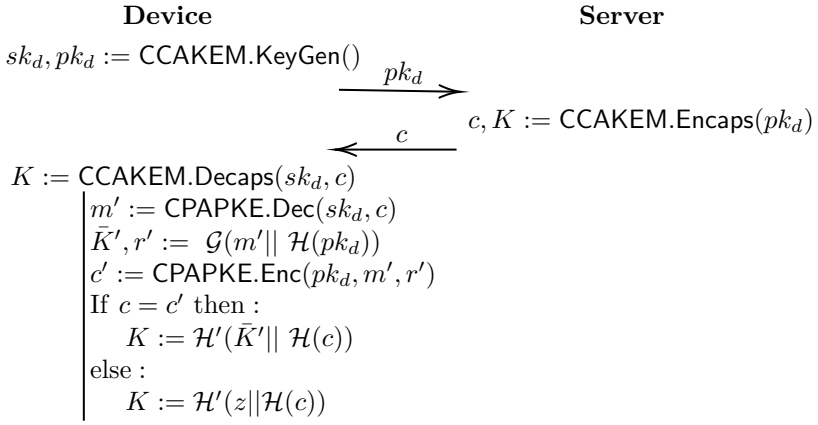


Figure 4: Classic KEM construction using the FO transform to achieve CCA security.

Table 2: Leveled protection profile for an FO KEM given the two SCA adversaries $\mathcal{A}_{KC-SCA}$ and $\mathcal{A}_{CC-SCA}$. No protection requirement is depicted in blue, SPA protection in orange, and DPA protection in red.

$\mathcal{A}_{KC-SCA}$	$\mathcal{A}_{CC-SCA}$
$m' :=$ CPAPKE.Dec(sk_d, c)	$m' :=$ CPAPKE.Dec(sk_d, c)
$\bar{K}', r' := \mathcal{G}(m' \| \mathcal{H}(pk_d))$	$\bar{K}', r' := \mathcal{G}(m' \| \mathcal{H}(pk_d))$
$c' :=$ CPAPKE.Enc(pk_d, m', r')	$c' :=$ CPAPKE.Enc(pk_d, m', r')
$c = c'$	$c = c'$
$K := \mathcal{H}'(\bar{K}' \| \mathcal{H}(c))$	$K := \mathcal{H}'(\bar{K}' \| \mathcal{H}(c))$
$K := \mathcal{H}'(z \| \mathcal{H}(c))$	$K := \mathcal{H}'(z \| \mathcal{H}(c))$

The $\mathcal{E}t\mathcal{S}$ KEM. The description of the $\mathcal{E}t\mathcal{S}$ KEM was provided in Figure 2. The security level assignment for the $\mathcal{E}t\mathcal{S}$ KEM is provided in Table 3, and again we provide some rationale about the coloring in the following. Since the verification processes only public values (i.e., pk_s, c, σ), it does not require SCA protection for any of the two adversaries. For $\mathcal{A}_{KC-SCA}$, the protection profile stays the same, as valid ciphertexts pass the verification and the intermediates after CPAPKE.Dec do not leak about the long-term secret sk_d. For $\mathcal{A}_{CC-SCA}$, the specifically-crafted chosen ciphertexts do not pass the verification, as the adversary does not have access to the secret signing key. Therefore, these inputs directly lead to the key derivation of K based only on z and c, which is unprotected as before. The only inputs that trigger operations processing the long-term secret sk_d are valid ciphertexts, and for these the intermediates after CPAPKE.Dec require only cost-efficient SPA protection.

Comparison. The impact of our proposed scheme over the original approach is visually noticeable from Tables 2 and 3. In the original scheme, the adversary $\mathcal{A}_{CC-SCA}$ is strictly stronger than $\mathcal{A}_{KC-SCA}$ and requires very costly DPA protection for many parts of CCAKEM.Decaps. Leveling is only partially possible with marginal impact on the overall performance. By introducing an explicit authenticity check based only on public values,

Table 3: Leveled protection profile for an $\mathcal{E}t\mathcal{S}$ KEM given the two SCA adversaries $\mathcal{A}_{KC-SCA}$ and $\mathcal{A}_{CC-SCA}$. No protection requirement is depicted in blue, SPA protection in orange, and DPA protection in red.

$\mathcal{A}_{KC-SCA}$	$\mathcal{A}_{CC-SCA}$
DS.Verify(pk_s, c, σ)	DS.Verify(pk_s, c, σ)
$m' :=$ CPAPKE.Dec(sk_d, c)	$m' :=$ CPAPKE.Dec(sk_d, c)
$\bar{K}', r' := \mathcal{G}(m' \| \mathcal{H}(pk_d))$	$\bar{K}', r' := \mathcal{G}(m' \| \mathcal{H}(pk_d))$
$K := \mathcal{H}'(\bar{K}' \| \mathcal{H}(c))$	$K := \mathcal{H}'(\bar{K}' \| \mathcal{H}(c))$
$K := \mathcal{H}'(z \| \mathcal{H}(c))$	$K := \mathcal{H}'(z \| \mathcal{H}(c))$

the potency of CC-SCA is completely negated. This is visible in Table 3, which shows that $\mathcal{A}_{CC-SCA}$ requires an equivalent protection profile to $\mathcal{A}_{KC-SCA}$. Effectively, this leaves only CPAPKE.Dec as a module with DPA protection requirements enabling significantly more efficient hardened implementations as will be demonstrated in Section 4. It should be noted, however, that the signature verification is the single point of failure for chosen ciphertext attacks. If this is skipped, e.g., due to an injected fault, the powerful attack vector is possible again. Therefore, it requires dedicated fault protection measures, if fault attacks are in scope. However, typical fault attack countermeasures, such as recomputing and comparing the results, induce a linear overhead on the total cost, as opposed to DPA countermeasures such as masking, which are significantly more expensive.

4 Illustration with lattice-based cryptography schemes

In this section, we provide a comparison of the FO KEM and the $\mathcal{E}t\mathcal{S}$-based KEM in terms of performance and communication overhead. For this purpose, we first describe in Section 4.1 the different parameters affecting these overhead measures and later provide a comparison for the STM32F4 ARM Cortex-M4 MCU used to benchmark NIST PQC candidates in pqm4 [KRSS19] in Section 4.2. We consider two different combinations of lattice-based schemes for the $\mathcal{E}t\mathcal{S}$-based KEM: instantiating the PKE with CRYSTALS-Kyber.CPAPKE and the digital signature with either CRYSTALS-Dilithium or Falcon. We additionally discuss the impact of introducing a signature verification function which requires fault attack protection in Section 4.3.

4.1 Parameters

Our comparison involves different parameters, including the choice of post-quantum PKE/KEM, digital signature, side-channel and fault attack countermeasures.

For lattice-based FO KEM, we consider the CRYSTALS-Kyber KEM [ABD+19], described previously. Table 4 shows the cost in kCycles for masking the Kyber decapsulation and relevant subroutines. As studied in [ABH+22] the signal-to-noise ratio of the leakage of the device and the target security level determine the number of shares required for masking. To capture the effect of both parameters, we consider different share numbers $d \in \{2, 3, 4, 5, 6, 7\}$ for higher-order masking. We also provide numbers for unprotected decapsulation for comparison.

When it comes to lattice-based signatures, we consider two options: the CRYSTALS-Dilithium [DLL+17] signature scheme for NIST level 3 and the Falcon-1024 signature scheme [FHK+19] for NIST level 5. Since, our running example and performance figures are provided for the Kyber level 3 instance, we use the corresponding Dilithium level 3. However, Falcon does not have a level 3 instance, and only a level 1 (Falcon-512) and a level 5 (Falcon-1024) parameter set. Since the security of the $\mathcal{E}t\mathcal{S}$ scheme relies first

Table 4: STM32F4 ARM Cortex-M4 MCU Performance numbers for masked Kyber.CCAKEM.Dec and its subroutines in kCycles.

Operation	**Number of shares**						
	unprotected	*2*	*3*	*4*	*5*	*6*	*7*
Kyber.CCAKEM.Decaps	850	3 178	57 141	97 294	174 220	258 437	350 529
Kyber.CPAPKE.Dec	64	200	4 203	7 047	13 542	20 323	27 230
Kyber.CPAPKE.Enc	647	2 024	18 879	32 594	53 298	75 692	104 191
comparison ($c = c'$)	3	693	32 293	54 725	102 922	156 075	210 518
$\mathcal{G}$	13	98	1 639	2 801	4 489	6 456	8 794
$\mathcal{H}$	113	113	113	113	113	113	113
$\mathcal{H}'$	13	13	13	13	13	13	13

and foremost on the security of the signature, we consider Falcon-1024 with NIST level 5. In the following analysis, we are mainly interested in the performance on the device's side, and hence in the signature verification. For this, we provide in Table 5 the speed in kCycles, public key and signature sizes based on the pqm4 benchmark [KRSS19]. While hash-based signature schemes (e.g., SPHINCS+, LMS, XMSS) can also be used for the $\mathcal{E}t\mathcal{S}$ KEM, we do not consider them in our analysis since they are more expensive (in terms of signature size, generation and verification) compared to lattice-based signature schemes.

Table 5: STM32F4 ARM Cortex-M4 MCU performance figures for CRYSTALS-Dilithium 3 and Falcon-1024 signature verification (sizes are given in bytes and speed in kCycles).

	CRYSTALS-Dilithium 3	**Falcon-1024**
Signature verification speed	2 229	977
Public key size	1 952	1 793
Signature size	3 293	1 280

While the $\mathcal{E}t\mathcal{S}$ scheme allows us to get rid of the leaky re-encryption, it introduces a new attack vector which is the signature check. Specifically, a signature verification performs a validity check based on the input message (more accurately its hash) and signature, that can be bypassed by fault injection, and grant adversaries the possibility to force the validation of any message-signature pair. The straightforward countermeasure against fault injection attacks is re-computation. Re-computing or duplicating the verification protects against the injection of a single fault. In general, by re-computing f times, we protect the target function against $\mathsf{f} - 1$ faults.

As previously discussed, compared to a FO-based KEMs, the $\mathcal{E}t\mathcal{S}$ KEM reduces the attack surface for DPA, and instead only requires SPA protection for some parts of the scheme. Countermeasures against SPA include shuffling [HOM06], which randomizes the order of the performed operations. The addition of side-channel noise can be achieved by different hardware or algorithmic means and aims to reduce the signal-to-noise ratio in the side-channel measurement. Overall, SPA counteremasures are typically less expensive (induce a linear overhead) than stronger DPA countermeasures such as masking (with quadratic overhead).

4.2 Performance comparison

We provide in Table 6 the performance values in kCycles for the $\mathcal{E}t\mathcal{S}$ KEM VerDec function using Dilithium 3 or Falcon-1024, for different masking orders. The signature verification,

the decryption and the key derivation steps are performed sequentially and the overall cost of VerDec is the sum of all its subroutines. The first column of the table corresponds to the masked Kyber.CCAKEM regular decapsulation. The first row corresponds to an uprotected implementation.

Table 6: Performance numbers for protected Kyber.CCAKEM.Decaps, $\mathcal{E}t\mathcal{S}$ Kyber.CPAPKE + Dilithium 3 and $\mathcal{E}t\mathcal{S}$ Kyber.CPAPKE + Falcon-1024 in kCycles for varying number of shares. The relative kCycles percentage w.r.t. Kyber.CCAKEM.Decaps is given in parentheses.

Num. of shares	**Scheme**		
	Kyber.Decaps	$\mathcal{E}t\mathcal{S}$ Kyber + Dilithium 3	$\mathcal{E}t\mathcal{S}$ Kyber + Falcon-1024
unprotected	850	2 432 (286%)	1 180 (139%)
2	3 178	2 568 (80.8%)	1 316 (41.4%)
3	57 141	6 571 (11.5%)	5 319 (9.3%)
4	97 294	9 415 (9.7%)	8 163 (8.4%)
5	174 220	15 910 (9.1%)	14 658 (8,4%)
6	258 437	22 691 (8.9%)	21 439 (8.3%)
7	350 529	29 598 (8.4%)	28 346 (8.1%)

First and with no surprise, when considering the unprotected case the FO-based Kyber KEM is more efficient than its $\mathcal{E}t\mathcal{S}$ counterpart. This is due to the large cost of signature verification for PQC signature schemes. However, in the masked case we observe that the $\mathcal{E}t\mathcal{S}$ schemes are significantly more efficient than the Kyber.CCAKEM decapsulation. When the noise level on the device decreases and, thus, the number of shares increases accordingly to achieve a target security level, the $\mathcal{E}t\mathcal{S}$ schemes become more and more efficient, since they do not require a costly masked re-encryption. For instance, when masking with only 2 shares is sufficient, the $\mathcal{E}t\mathcal{S}$ schemes with Dilithium and Falcon achieve an improvement of approximately 20% and 60%, respectively. When considering more sensible and larger share numbers to protect implementations on standard MCUs, the $\mathcal{E}t\mathcal{S}$ schemes achieve similar improvements ranging from 90% to 92%. The improvement gap between the $\mathcal{E}t\mathcal{S}$ scheme using Dilithium and the one using Falcon shrinks with the number of shares, since the cost of the signature verification becomes minimal next to the cost of the Kyber.CPA decryption and the KDF.

The main drawbacks of the $\mathcal{E}t\mathcal{S}$ schemes lie in the encapsulation and communication overheads. First, since the $\mathcal{E}t\mathcal{S}$ KEM encapsulation process includes a signature generation, its cost increases from 786 kCycles to 10 075 and 84 269 for $\mathcal{E}t\mathcal{S}$ with Dilithium and Falcon, respectively. In the relevant usecases of the $\mathcal{E}t\mathcal{S}$ KEM this cost is less detrimental than the one of the masked decapsulation. Regarding the data overhead, for Kyber level 3, the ciphertext size is 1088 bytes. For the $\mathcal{E}t\mathcal{S}$ schemes using Dilithium level 3 and Falcon level 5, the total ciphertext sizes (including the signature) are 4,381 and 2368 bytes, respectively. As discussed previously, based on the usecase, this can be worthwhile[4]. Interestingly, the choice of signature scheme can be based on a compromise between the data and performance overheads, e.g., while Falcon has relatively small signatures and fast verification, its signature generation is significantly more expensive than Dilithium's.

[4]Notably, NIST has indicated the need for PQC signature schemes with very short signatures, and will issue a new call for signature schemes independently of the 4^{th} round of the current PQC competition. Signature schemes with shorter signatures can be used to replace Dilithium or Falcon in the $\mathcal{E}t\mathcal{S}$ scheme.

4.3 Fault attacks mitigation for signature verification

Next, we examine the impact of the introduction of a signature verification in the $\mathcal{E}t\mathcal{S}$ schemes with respect to fault attack resistance. We show on Figure 5 the efficiency impact on the $\mathcal{E}t\mathcal{S}$ schemes, when the signature verification is protected against $\mathsf{f}-1$ faults, which requires recomputing the verification f times, and comparing the results, to detect any injected fault.

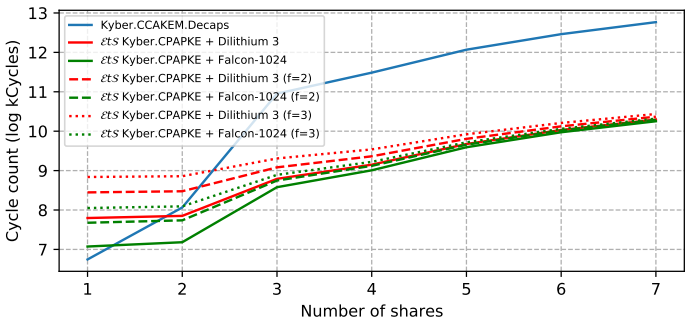


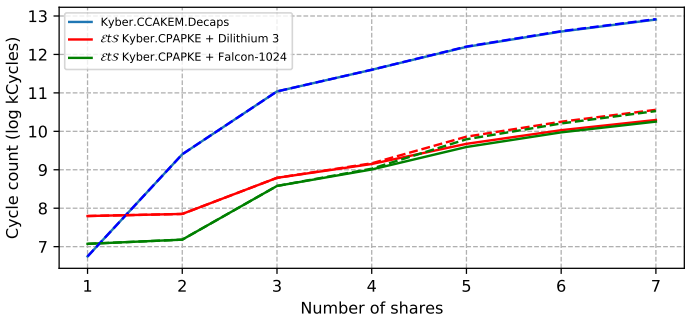
Figure 5: Performance of Kyber.CCAKEM.Decaps, $\mathcal{E}t\mathcal{S}$ KEM (with Dilithium and Falcon) as function of the number shares and the number f of signature verifications computed.

From Figure 5, we see that for a low number of shares the $\mathcal{E}t\mathcal{S}$ schemes are potentially less efficient than the Kyber.CCAKEM.Decaps when the signature verification is relatively costly. This can be remedied by using a more efficient signature algorithm such as Falcon. However, when the number of shares increases (d>3), the gap between Kyber.CCAKM.Decaps and the $\mathcal{E}t\mathcal{S}$ scheme with multiple signature verification re-computations increases rapidly. The impact of the re-computation on the overall cost diminishes with the number of shares. This is as expected since the cost of the signature verification is smaller than the cost of the masked decapsulation at high orders and the re-computation only induces a linear overhead, whereas the high order masking on the other hand induces a quadratic overhead.

4.4 Considerations for SPA security

As discussed in section 3.2.3, the $\mathcal{E}t\mathcal{S}$ KEM increases the number of operations to protect against SPA. Accordingly, in this section we take a closer look at the impact of SPA countermeasures on the FO-based KEM and the $\mathcal{E}t\mathcal{S}$ KEM. While the kind of SPA countermeasure to implement and its parameters are determined by the noise level on the considered device and the target security level, we adopt a general simplification to study its impact. Precisely, we assume that to achieve the same security for the SPA targets as for the DPA ones, we can mask the SPA targets using 2 less shares than the targets requiring DPA protection. Arguably, a cheaper countermeasure such as shuffling can also be used to achieve adequate security, however it is highly dependent on the number of independent operations at each stage of the considered function.

Figure 6 shows the extrapolated costs from Table 4. We see the overhead introduced by the SPA mitigation is more pronounced for the $\mathcal{E}t\mathcal{S}$ schemes compared to the FO-based schemes. This is expected since the $\mathcal{E}t\mathcal{S}$ schemes have more SPA targets. The main conclusion from this analysis, is that despite protecting the SPA targets with an expensive countermeasure such as masking, the $\mathcal{E}t\mathcal{S}$ KEM still remains signficantly more efficient than its FO-based counterpart.



.

Figure 6: Performance of Kyber.CCAKEM.Decaps, $\mathcal{E}t\mathcal{S}$ KEM (with Dilithium and Falcon) as function of the number shares for DPA protection. Dashed lines correspond to the schemes with added SPA protection (masking with $d-2$ shares) excluding $d \in \{1, 2\}$

5 Conclusion

In this work, we combine a standard cryptographic construction, namely the $\mathcal{E}t\mathcal{S}$ paradigm, and observations from recent side-channel analysis of post-quantum KEMs. Our main result is to enable efficiently hardened authenticated post-quantum public key encryption, that can be used to instantiate a KEM, without the need to protect the costly FO transform against CC-SCA. While the initial concept is simple, it surprisingly allows speeding up the KEM by a factor 10. However, the $\mathcal{E}t\mathcal{S}$ construction can only lift CPA security to CCA security under the outsider-security model. Therefore, we discuss applications of the $\mathcal{E}t\mathcal{S}$ KEM that conform to this model. The most notable is the secure update mechanism, which is essential in maintaining the security and the reliability of embedded and IoT devices.

The side-channel protection of post-quantum schemes is a recent, but quite active research direction for the academic community. Accordingly, we expect more efficient masked KEM implementations in the next few years. However, the main bottleneck when masking FO-based KEMs, such as Kyber or Saber, stems from the multiple calls to the hash functions in the re-encryption introduced by the FO transform, that require high-order masking to hinder CC-SCA. Since the $\mathcal{E}t\mathcal{S}$ KEM gets rid of the need of re-encryption, we expect that the improvement brought by using $\mathcal{E}t\mathcal{S}$ should transfer to more optimized implementations as well in the future.

Eventually, future work could explore other practical applications of the $\mathcal{E}t\mathcal{S}$ KEM, and additionally, in other contexts, e.g., the multi-user setting, and whether it is suitable for some specific purpose protocols (e.g., secure element to MCU communication or IoT edge computing). Another research direction could be to design post-quantum cryptography schemes that are naturally resistant against implementation attacks by leveraging the large body of work related to leakage resilience, instead of focusing on protecting schemes that were not designed with physical attacks in mind.

Acknowledgements. The authors would like to thank the reviewers for their helpful comments and for pointing out the SUIT protocol.

References

[ABD+19] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien

Stehlé. Crystals-kyber algorithm specifications and supporting documentation. *NIST PQC Round*, 3:4, 2019.

[ABH+22] Melissa Azouaoui, Olivier Bronchain, Clément Hoffmann, Yulia Kuzovkova, Tobias Schneider, and François-Xavier Standaert. Systematic study of decryption and re-encryption leakage: the case of kyber. *IACR Cryptol. ePrint Arch.*, page 36, 2022.

[ADR02] Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer, 2002.

[BBC+20] Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Mode-level vs. implementation-level physical security in symmetric cryptography - A practical guide through the leakage-resistance jungle. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 369–400. Springer, Heidelberg, August 2020.

[BBK+17] Nina Bindel, Johannes Buchmann, Juliane Krämer, Heiko Mantel, Johannes Schickel, and Alexandra Weber. Bounding the cache-side-channel leakage of lattice-based signature schemes using program semantics. In Abdessamad Imine, José M. Fernandez, Jean-Yves Marion, Luigi Logrippo, and Joaquín García-Alfaro, editors, *Foundations and Practice of Security - 10th International Symposium, FPS 2017, Nancy, France, October 23-25, 2017, Revised Selected Papers*, volume 10723 of *Lecture Notes in Computer Science*, pages 225–241. Springer, 2017.

[BBLW22] Richard Barnes, Karthikeyan Bhargavan, Benjamin Lipp, and Christopher A. Wood. Hybrid Public Key Encryption. RFC 9180, February 2022.

[BC22] Olivier Bronchain and Gaëtan Cassiers. Bitslicing arithmetic/boolean masking conversions for fun and profit with application to lattice-based kems. Cryptology ePrint Archive, Report 2022/158, 2022. https://ia.cr/2022/158.

[BDK+21] Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede. A side-channel-resistant implementation of SABER. *ACM J. Emerg. Technol. Comput. Syst.*, 17(2):10:1–10:26, 2021.

[BFG+21] Jacqueline Brendel, Rune Fiedler, Felix Günther, Christian Janson, and Douglas Stebila. Post-quantum asynchronous deniable key exchange and the signal handshake. *IACR Cryptol. ePrint Arch.*, page 769, 2021.

[BGR+21] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider, and Christine van Vredendaal. Masking kyber: First- and higher-order implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):173–214, 2021.

[BHLY16] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 323–345. Springer, 2016.

[BMPS21] Olivier Bronchain, Charles Momin, Thomas Peters, and François-Xavier Standaert. Improved leakage-resistant authenticated encryption based on hardware AES coprocessors. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):641–676, 2021.

[BMV17] Silvio Biagioni, Daniel Masny, and Daniele Venturi. Naor-yung paradigm with shared randomness and applications. *Theor. Comput. Sci.*, 692:90–113, 2017.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.

[BZH+21] Gustavo Banegas, Koen Zandberg, Adrian Herrmann, Emmanuel Baccelli, and Benjamin Smith. Quantum-resistant security for software updates on low-power networked embedded devices. *IACR Cryptol. ePrint Arch.*, page 781, 2021.

[CGMZ22] Jean-Sébastien Coron, François Gérard, Simon Montoya, and Rina Zeitoun. High-order table-based conversion algorithms and masking lattice-based encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(2):1–40, 2022.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[CPPS20] Sanjit Chatterjee, Tapas Pandit, Shravan Kumar Parshuram Puria, and Akash Shah. Signcryption in a quantum world. *IACR Cryptol. ePrint Arch.*, page 1388, 2020.

[DHP+22] Jan-Pieter D'Anvers, Daniel Heinz, Peter Pessl, Michiel Van Beirendonck, and Ingrid Verbauwhede. Higher-order masked ciphertext comparison for lattice-based cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(2):115–139, 2022.

[DKR+20] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[DKRV18] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure KEM. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2018 - 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7-9, 2018, Proceedings*, volume 10831 of *Lecture Notes in Computer Science*, pages 282–305. Springer, 2018.

[DLL+17] Léo Ducas, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - dilithium: Digital signatures from module lattices. *IACR Cryptol. ePrint Arch.*, page 633, 2017.

[DOV21] Jan-Pieter D'Anvers, Emmanuela Orsini, and Frederik Vercauteren. Error term checking: Towards chosen ciphertext security without re-encryption. In Keita Emura and Yuntao Wang, editors, *Proceedings of the 8th on ASIA Public-Key Cryptography Workshop, APKC@AsiaCCS 2021, Virtual Event Hong Kong, 7 June, 2021*, pages 3–12. ACM, 2021.

[EFGT17] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. pages 1857–1874, 2017.

[FBR+22] Tim Fritzmann, Michiel Van Beirendonck, Debapriya Basu Roy, Patrick Karl, Thomas Schamberger, Ingrid Verbauwhede, and Georg Sigl. Masked accelerators and instruction set extensions for post-quantum cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):414–460, 2022.

[FHK+19] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. 2019.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.

[GJN20] Qian Guo, Thomas Johansson, and Alexander Nilsson. A key-recovery timing attack on post-quantum primitives using the fujisaki-okamoto transformation and its application on frodokem. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 359–386. Springer, 2020.

[Glo] GlobalPlatform Technology. Secure channel protocol '03'. https://globalplatform.org/wp-content/uploads/2014/07/GPC_2.3_D_SCP03_v1.1.2_PublicRelease.pdf.

[GM18] François Gérard and Keno Merckx. SETLA: signature and encryption from lattices. In Jan Camenisch and Panos Papadimitratos, editors, *Cryptology and Network Security - 17th International Conference, CANS 2018, Naples, Italy, September 30 - October 3, 2018, Proceedings*, volume 11124 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2018.

[HCY19] Wei-Lun Huang, Jiun-Peng Chen, and Bo-Yin Yang. Power analysis on NTRU prime. 2020(1):123–151, 2019. https://tches.iacr.org/index.php/TCHES/article/view/8395.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, 2017.

[HOM06] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252, 2006.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, Heidelberg, August 1999.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 552–586. Springer, 2018.

[KPP20] Matthias J. Kannwischer, Peter Pessl, and Robert Primas. Single-trace attacks on Keccak. *IACR TCHES*, 2020(3):243–268, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8590.

[KRSS19] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. pqm4: Testing and benchmarking NIST PQC on ARM cortex-m4. *IACR Cryptol. ePrint Arch.*, page 844, 2019.

[MTBM21] Brendan Moran, Hannes Tschofenig, David Brown, and Milosch Meriac. A Firmware Update Architecture for Internet of Things. RFC 9019, April 2021.

[Nat] National Institute of Standards and Technology. Post-quantum cryptography standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

[NDGJ21] Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. A side-channel attack on a masked IND-CCA secure saber KEM implementation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):676–707, 2021.

[OSPG18] Tobias Oder, Tobias Schneider, Thomas Pöppelmann, and Tim Güneysu. Practical CCA2-secure masked Ring-LWE implementations. 2018(1):142–174, 2018. https://tches.iacr.org/index.php/TCHES/article/view/836.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013.

[RRCB20] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs. 2020(3):307–335, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8592.

[RRVV15] Oscar Reparaz, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. A masked ring-lwe implementation. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES*

2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, volume 9293 of *Lecture Notes in Computer Science*, pages 683–702. Springer, 2015.

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sho01] Victor Shoup. A proposal for an ISO standard for public key encryption. *IACR Cryptol. ePrint Arch.*, page 112, 2001.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptol. ePrint Arch.*, page 332, 2004.

[SRSW20] Thomas Schamberger, Julian Renner, Georg Sigl, and Antonia Wachter-Zeh. A power side-channel attack on the cca2-secure HQC KEM. In Pierre-Yvan Liardet and Nele Mentens, editors, *Smart Card Research and Advanced Applications - 19th International Conference, CARDIS 2020, Virtual Event, November 18-19, 2020, Revised Selected Papers*, volume 12609 of *Lecture Notes in Computer Science*, pages 119–134. Springer, 2020.

[SSW20] Peter Schwabe, Douglas Stebila, and Thom Wiggers. Post-quantum TLS without handshake signatures. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1461–1480. ACM, 2020.

[SW07] Joseph H. Silverman and William Whyte. Timing attacks on NTRUEncrypt via variation in the number of hash calls. pages 208–224, 2007.

[THM22] Hannes Tschofenig, Russ Housley, and Brendan Moran. Firmware Encryption with SUIT Manifests. Internet-Draft draft-ietf-suit-firmware-encryption-04, Internet Engineering Task Force, April 2022. Work in Progress.

[UXT+22] Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. Curse of re-encryption: A generic power/em analysis on post-quantum kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):296–322, 2022.

[XPRO20] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, and David Oswald. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber. Cryptology ePrint Archive, Report 2020/912, 2020. https://eprint.iacr.org/2020/912.

[Zhe97] Yuliang Zheng. Signcryption and its applications in efficient public key solutions. In Eiji Okamoto, George I. Davida, and Masahiro Mambo, editors, *Information Security, First International Workshop, ISW '97, Tatsunokuchi, Japan, September 17-19, 1997, Proceedings*, volume 1396 of *Lecture Notes in Computer Science*, pages 291–312. Springer, 1997.