SHADOWFAX: Hybrid Security and Deniability for AKEMs

Phillip Gajland
IBM Research Europe – Zurich

Vincent Hwang
MPI-SP
Radboud University

Jonas Janneck
Ruhr University Bochum

Abstract

As cryptographic protocols transition to post-quantum security, most adopt hybrid solutions combining classical and post-quantum assumptions. This shift often sacrifices efficiency, compactness, or even security. One such property is *deniability*, which enables users to plausibly deny authorship of potentially incriminating messages. While classical protocols like X3DH key agreement (used in Signal and WhatsApp) provide deniability, post-quantum protocols like PQXDH and Apple's iMessage with PQ3 do not.

This work addresses this gap by investigating how to efficiently preserve deniability in post-quantum protocols. Specifically, we propose two hybrid schemes for authenticated key encapsulation mechanisms (AKEMs). The first is a black-box construction that preserves deniability when both constituent AKEMs are deniable. The second is SHADOWFAX, a non-black-box AKEM that achieves hybrid security, integrating a classical non-interactive key exchange, a post-quantum key encapsulation mechanism, and a post-quantum ring signature. SHADOWFAX satisfies deniability in both dishonest and honest receiver settings, relying on statistical security in the former and on a single pre- or post-quantum assumption in the latter.

Finally, we provide several portable implementations of SHADOWFAX. When instantiated with standardised components (ML–KEM and FALCON), SHADOWFAX yields ciphertexts of 1 728 bytes and public keys of 2 036 bytes, with encapsulation and decapsulation costs of 1.8M and 0.7M cycles on an Apple M1 Pro.

1 Introduction

The global roll out of post-quantum cryptography (PQC) is a monumental challenge. While the multi-year National Institute of Standards and Technology (NIST) standardisation [62] has been a critical milestone, it marks only the beginning of a much larger effort. With several algorithms selected for standardisation (three standards have already been released [59, 60, 75]), the next phase of implementation and adaptation is now underway. Migrating countless systems to PQC will likely take decades[1]. Nevertheless, significant progress has been made towards adapting widely deployed protocols for post-quantum security. For instance, X3DH [58], which supports billions of WhatsApp and Messenger users, has been upgraded to the post-quantum variant, PQXDH [49]. PQXDH is already deployed in Signal [49], and Apple's iMessage now uses PQ3 [6]. However, these upgrades involve trade-offs: ciphertexts and keys grow larger, and deniability is often lost. PQXDH, for example, sacrifices the deniability of its predecessor X3DH due to a signature on the ephemeral key [33]. Similarly, the analysis of Apple's iMessage with PQ3 [53, 76], explicitly states that deniability is not a design goal. TLS [26, 68] has also been updated for post-quantum security by using key encapsulation mechanisms (KEMs) in multiple papers [15, 18, 64] and real-world deployments [50–52, 78].

A crucial aspect of all these adaptations is the *hybrid* approach, combining post-quantum algorithms with classical cryptographic methods. Post-quantum solutions, despite their potential, lack the decades of cryptographic analysis that traditional schemes like RSA and (EC)DH have undergone. Therefore, it is prudent to adopt hybrid solutions. This strategy is widely endorsed by national security agencies. The French National Agency for the Security of Information Systems (ANSSI) recommends a hybrid adoption of PQC [5], and the German Federal Office for Information Security (BSI) explicitly states that *"post-quantum cryptography should not be used in isolation if possible, but only in hybrid mode"* for both key agreement and authentication [21]. The BSI has reiterated this need in their recent updated technical guidelines, which *"only recommends the hybrid use of quantum-safe methods in combination with classical methods"* [22].

[1] Although it has been known for over twenty years that `MD5` [69] fails to provide collision resistance [77], recent research continues to exploit this insecurity in new vulnerabilities [41] within prevalent protocols.

1.1 Combiners

A combiner, or more generally hybrid scheme, ensures security as long as at least one of its components remains secure. For instance, if cryptographically relevant quantum computers (CRQCs) become available rendering classical schemes insecure [74], the hybrid scheme would still be secure as the post-quantum component remains intact. Conversely, if advances in cryptanalysis or implementation issues break the post-quantum scheme, the classical security of the hybrid scheme would still hold due to the hardness of the classical problem. In fact, recent work demonstrated several classical attacks on post-quantum schemes [16, 23, 57, 71] underscoring the importance of hybrids. Additionally, since only basic post-quantum primitives like KEMs and signatures have been standardised, many applications necessarily need to rely on non-standard primitives, further supporting hybrid configurations. Finally, to achieve *"cryptographic agility"* [63] in the long run, the permanent use of hybrid solutions may become a common practice.

Generic Combiners. Combiners are typically defined relative to a *primitive* and security *notion*, where security is *binary* – either the scheme is secure (Π) or insecure ($\neg\Pi$). Generic combiners allow statements like $\Pi_1 \vee \Pi_2 \implies \Pi$, meaning that as long as one of the schemes satisfies the security notion, the combined scheme also satisfies it. For instance, consider the *pseudorandomness* of a PRG, or *confidentiality* of a KEM. A generic combiner for the former is $\mathsf{PRG}(s_1 \| s_2) := \mathsf{PRG}_1(s_1) \oplus \mathsf{PRG}_2(s_2)$, and for the latter, KEM with $k = \mathsf{H}(k_1, k_2, c_1, c_2)$, when H is modelled as a random oracle. Such combiners are the most powerful as they are the most general.

PQC Migration. While generic combiners are sufficient for PQC migration, constructing them can be challenging in some cases, and we argue they may be unnecessarily restrictive. To understand why, one must consider the primary motivation for using combiners in PQC migration which is risk mitigation against the failure of *assumptions*. Therefore, instead of reasoning about combiners for security *notions*, one can relax this requirement to focus specifically on the underlying *assumptions* for PQC migration. Specifically, we argue that it suffices to construct combiners of the form $\text{Assumption}_{\text{pre-Q}} \vee \text{Assumption}_{\text{post-Q}} \implies \Pi$, where the security of the scheme holds as long as one of the classical or post-quantum *assumptions* is not broken. We call such a scheme a *hybrid*. This relaxation simplifies combiner design, enables constructions where generic approaches remain elusive [32, 42, 44, 45, 55] and better aligns with the primary motivation for adopting hybrid solutions in PQC migration.

Additional related work on combiners and hybrids is further discussed in the extend version of the paper.

1.2 AKEM

The integration of combiners for KEMs and signatures – ensuring confidentiality and authenticity – has already begun in the context of PQC migration. An Authenticated Key Encapsulation Mechanism (AKEM) shares the same interfaces as a standard KEM, with two key differences: encapsulation proves the sender's authenticity requiring their secret key, while decapsulation verifies the sender's authenticity using their public key. Therefore, the primitive includes both notions of confidentially and authenticity. Introduced in the HPKE standard [9], an AKEM draws inspiration from the signcryption literature [25] and generalises the split-KEM primitive [20].[2] An additional feature of several AKEM constructions [3, 4, 24, 39, 46], and later formalised in [24, 39] is *deniability*. This property allows a sender to deny having sent a particular ciphertext.[3] Beyond its application to HPKE, which is specified to be used in the Message Layer Security (MLS) [8] protocol, AKEMs find applications in authenticated key exchange and secure messaging [13, 20], for instance in K-WAAY [24].

Another related work discusses PQ deniable authenticated key exchange [43]. Unlike an AKEM which is a one-shot primitive, the protocol from [43] is interactive. However, the core part of the protocol could be viewed as an AKEM (with ephemeral AKEM keys). Similar to our scheme which we will present later, the construction from [43] is based on a KEM and a ring signature which can be efficiently instantiated from PQ primitives. A key difference to our approach is that we consider hybrid security guarantees along with the associated challenges.

More generally, despite the existence of both classical [3, 4] and post-quantum [4, 24, 39, 43, 46, 61] constructions, there are no known hybrid AKEMs. This leads to the natural question:

"Can hybrid AKEMs efficiently preserve deniability?"

1.3 Technical Overview

In this work, we answer the aforementioned question in the affirmative. The following section presents a high-level summary of our technical contributions.

AKEM. The two main security properties are confidentiality and authenticity. Another desirable property is (sender) deniability, allowing the sender to plausibly deny sending a ciphertext even though the receiver can verify the sender's identity. This is formalised by showing the existence of a simulator Sim, whose output ciphertext c and key k are indistinguishable from those produced by the encapsulation

[2] In fact, a symmetric split-KEM [20, Def. 4] is equivalent to an AKEM [3, Def. 9].

[3] To the best of our knowledge, combiners for deniability have not yet been studied. For background on deniability we refer to the extend version of the paper.

algorithm Enc, to any adversary $\mathcal{A}$. The model of deniability varies depending on the scenario [39]. For a *dishonest receiver*, Sim is given the receiver's secret key, modelling a scenario where the receiver could forge a ciphertext c to falsely attribute it to the sender. For *honest receivers*, the receiver is assumed to not simulate any values, so Sim is not given the receiver's secret key.

1.3.1 Black-Box Construction.

A natural way to construct a hybrid AKEM would be to use the *"parallel KEM combiner"*, where the ciphertext $c := (c_1, c_2)$ and key $k := \mathsf{H}(k_1, k_2, pk_s, pk_r, c)$. Here $(c_i, k_i) \xleftarrow{\$} \mathsf{Enc}_i(sk_i, pk_i)$ for $i \in \{1,2\}$ and $(sk_s = (sk_1, sk_2), pk_r = (pk_1, pk_2))$. Confidentiality is guaranteed as long as *one of* AKEM_1 *or* AKEM_2 provides confidentiality, akin to the KEM combiner from [40]. However, we additionally include the sender and receiver's public keys in H to satisfy the strong notion of insider CCA security (see Theorem 2). Similarly, for *authenticity*, we show that the resulting scheme inherits authenticity as long as *one of* AKEM_1 *or* AKEM_2 satisfies authenticity (see Theorem 3).

For *deniability*, we were only able to prove that the resulting scheme satisfies deniability if *both* AKEM_1 *and* AKEM_2 provide deniability (see Theorems 4 and 5). This result is unsatisfactory, as it undermines the purpose of the hybrid scheme. The challenge lies in constructing a simulator for the combined scheme without having a simulator for either AKEM_1 or AKEM_2. Specifically, such a simulator needs to output a ciphertext and key that are indistinguishable from those generated by the AKEM encapsulation. If one of the underlying AKEMs is not deniable (whether for an honest or dishonest receiver), then a distinguisher exists for any potential simulator. While the key can be simulated using known KEM combiner techniques, the problem arises when trying to simulate both ciphertext components, as no simulator exists for one of the components. That makes it hard to satisfy hybrid-like deniability that only relies on the security properties of one of the AKEMs. Interestingly, an AKEM can be constructed satisfying statistical *dishonest receiver deniability* from a ring signature scheme with information-theoretic anonymity. Although this is not true of schemes like SMILE [56] and EREBOR [17], GANDALF [39] does satisfy the necessary anonymity. However, it appears unlikely that *honest receiver deniability* can be achieved information theoretically, as prior techniques still require computational assumptions. For instance, in [39], the authors *"boost"* an AKEM to satisfy honest receiver deniability by symmetrically encrypting the ring signature with a KEM key, which, in turn, depends on a computational assumption. Therefore, instead of analysing security notions at a black-box level, we consider the underlying computational hardness assumptions needed to achieve these security properties. This requires examining AKEM constructions at a lower abstraction level.

1.3.2 SHADOWFAX.

The SHADOWFAX construction consists of two parts: the post-quantum component and the classical component. A high level overview of the construction is depicted in Figure 1.

The post-quantum component. The post-quantum part relies on a post-quantum KEM, a post-quantum ring signature scheme, a symmetric encryption scheme, and a PRF used as a key derivation function (KDF). For confidentiality, the receiver's KEM public key kpk_r is input to the encapsulation procedure, returning a KEM ciphertext kct and KEM key kk. For authenticity, the KEM ciphertext kct is signed using a ring signature scheme with the sender's signing key ssk_s, where the signing ring consists of the sender's public key spk_s and the receiver's public key spk_r. The anonymity property of the ring signature implies dishonest receiver deniability [39]. However, if the anonymity relies on computational assumptions as mentioned above, deniability would be lost if those assumptions turn out to be flawed. To avoid this, the ring signature scheme must provide statistical anonymity. However, this alone does not suffice for honest receiver deniability, since the signature σ is publicly verifiable. If the signature is unforgeable, the ciphertext must have originated from the sender or from the receiver, since only they can sign for the respective ring. However, since the receiver is honest (captured by the simulator not given the receiver's secret key), the adversary can deduce that the signature was issued by the sender. Therefore, the signature is symmetrically encrypted using the KEM key kk, ensuring that only the receiver can verify the signature. The symmetric ciphertext sct and KEM ciphertext kct become the AKEM ciphertext c in the SHADOWFAX construction, while the key k is derived from the KDF applied to sct and kk. This step is similar to the KEM combiner from [40]. To ensure strong confidentiality, specifically insider CCA security, the KDF also includes the public keys of both the sender and the receiver. Up to this point, this construction mirrors the PQ-AKEM from [39].

The classical component. To obtain a hybrid scheme we add the classical part of SHADOWFAX, that integrates two non-interactive key exchanges (NIKEs), similar to the DH-AKEM construction of [3], and intertwines them with the post-quantum component. During encapsulation, an ephemeral NIKE key pair (nsk_e, npk_e) is generated to provide confidentiality, with the NIKE public key npk_e appended to the ciphertext. The ephemeral NIKE secret key nsk_e is then used to compute a shared NIKE key nk_e with the receiver's NIKE public key npk_e and this key is fed into the KDF. That is, SHADOWFAX provides confidentiality as long

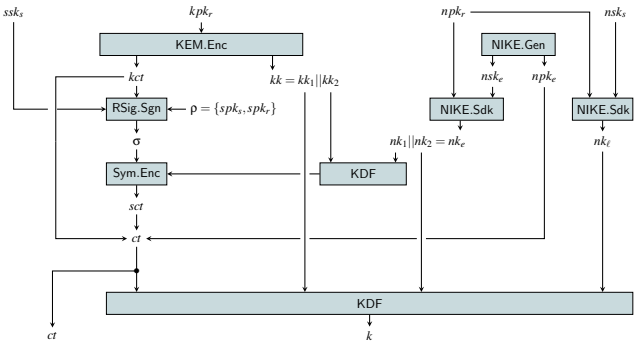

Figure 1: High level overview of the SHADOWFAX construction.

as *one of* the post-quantum KEM *or* the classical NIKE is secure (see Theorem 7). For (implicit) authentication, the sender's long-term NIKE secret key nsk_s is used with the receiver's NIKE public key npk_r to derive a NIKE shared key nk_ℓ, which is also used as input to the KDF. This ensures authenticity of SHADOWFAX as long as *one of* the post-quantum ring signature is unforgeable, *or* the classical NIKE remains secure (see Theorem 8).

The dishonest receiver deniability of SHADOWFAX holds as long as the ring signature scheme is information theoretically anonymous and the NIKE is correct (see Theorem 9). Finally, combining a KEM and ring signature scheme with two NIKEs would not satisfy honest receiver deniability, as the signature is publicly verifiable. Recall that we symmetrically encrypted the signature using the post-quantum KEM key. However, if the CCA security is compromised, then honest receiver deniability would be lost. To mitigate this, both the KEM key kk and the ephemeral NIKE shared key nk_e are run through another KDF before symmetrically encrypting the signature σ. This ensures that also the honest receiver deniability of SHADOWFAX is satisfied in a hybrid sense. Specifically, honest receiver deniability is preserved as long as *one of* the post-quantum KEM is CPA secure *or* the classical NIKE remains secure (see Theorem 10).

1.4 Contributions

This work considers hybrid schemes that preserve deniability, an area previously unexplored. Specifically, we focus on authenticated key encapsulation mechanisms (AKEMs) and present the following contributions.

BLACK-BOX CONSTRUCTION. At the highest level of abstraction, we present a black-box construction that combines two AKEMs. We prove that deniability is preserved if both underlying schemes are deniable. Moreover, confidentiality and authenticity are guaranteed as long as one of the AKEMs provides the desired notion, aligning with the expected behaviour of a combiner.

NON-BLACK-BOX CONSTRUCTION. At a lower level of abstraction, we introduce SHADOWFAX, a non-black-box AKEM that achieves hybrid security, built from a classical NIKE, a post-quantum KEM, and a post-quantum ring signature scheme. We show that SHADOWFAX achieves deniability in two distinct settings. In the dishonest receiver setting, deniability relies on the correctness of the NIKE and the (possibly statistical) anonymity of the ring signature. In the honest receiver setting, deniability holds under *one of* two computational assumptions: the security of the ephemeral NIKE *or* the KEM.

IMPLEMENTATION. Our final contribution is a set of portable implementations designed for compactness, reproducibility, and easy integration into existing cryptographic libraries. These include the first implementations of the GANDALF ring signature scheme and post-quantum AKEM from [39], as well as the hybrid AKEM SHADOWFAX.

Our GANDALF implementation with FALCON achieves 1.29× faster key generation and 1.63× faster signing compared to concurrent work [47, Tab. 3]. When instantiated with standardised components, such as FALCON and ML-KEM, SHADOWFAX achieves ciphertexts of 2 036 bytes and public keys of 1 728 bytes. With non-standard components, ciphertexts can be reduced to 1 781 bytes and public keys to 1 449 bytes (see Section 5).

For our instantiation with standardised components, encapsulation takes 1.8 million cycles, and decapsulation takes about 675 000 cycles on a Firestorm core running at 3 GHz on an Apple M1 Pro. Detailed parameters and benchmarks appear in Table 3, Table 4, and the project's GitHub repository at Shadowfax. [4]

2 Preliminaries

We introduce some relevant definitions used throughout the paper. Further notions can be found in the full version of the paper [37].

2.1 Notations

Sets and Algorithms. We write $s \xleftarrow{\$} \mathcal{S}$ to denote the uniform sampling of s from the finite set $\mathcal{S}$. For an integer n, we define $[n] := \{1, \ldots, n\}$. The notation $[\![b]\!]$, where b is a boolean statement, evaluates to 1 if the statement is true and 0 otherwise. We use uppercase letters $\mathcal{A}, \mathcal{B}, \ldots$ to denote algorithms. Unless otherwise stated, algorithms are probabilistic, and we write $(y_1, \ldots) \xleftarrow{\$} \mathcal{A}(x_1, \ldots)$ to denote that $\mathcal{A}$ returns $(y_1, \ldots)$ when run on input $(x_1, \ldots)$ and $t_{\mathcal{A}}$ to denote the time of $\mathcal{A}$. We write $\mathcal{A}^{\mathcal{B}}$ to denote that $\mathcal{A}$ has oracle access to $\mathcal{B}$ during its execution. For a randomised

[4] https://github.com/vincentvbh/shadowfax

algorithm $\mathcal{A}$, we use the notation $y \in \mathcal{A}(x)$ to denote that y is a possible output of $\mathcal{A}$ on input x. The support of a discrete random variable X is defined as $\sup(X) := \{x \in \mathbb{R} \mid \Pr[X = x] > 0\}$.

Security Games. We use standard code-based security games [12]. A *Game* G is a probability experiment in which an adversary $\mathcal{A}$ interacts with an implicit challenger that answers oracle queries issued by $\mathcal{A}$. The game G has one *main procedure* and an arbitrary amount of additional *oracle procedures* which describe how these oracle queries are answered. We denote the (binary) output b of game G between a challenger and an adversary $\mathcal{A}$ as $\mathsf{G}^{\mathcal{A}} \Rightarrow b$. $\mathcal{A}$ is said to *win* G if $\mathsf{G}^{\mathcal{A}} \Rightarrow 1$, or shortly $\mathsf{G} \Rightarrow 1$. Unless otherwise stated, the randomness in the probability term $\Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow 1]$ is over all the random coins in game G. If a game is aborted the output is either 0 or a random bit in case of an indistinguishability game, i.e. a game for which the advantage of an adversary is defined as the absolute difference of winning the game to $\frac{1}{2}$. To provide a cleaner description and avoid repetitions, we sometimes refer to procedures of different games. To call the oracle procedure `Oracle` of game G on input x, we shortly write $\mathsf{G}.\texttt{Oracle}(x)$. Throughout the proofs we rely on game hopping and the main difference lemma [12].

2.2 AKEM

Definition 1 (Authenticated Key Encapsulation Mechanism [3, Def. 9])**.** An *authenticated key encapsulation mechanism* AKEM is defined as a tuple $\mathsf{AKEM} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of the following algorithms.

$(sk, pk) \xleftarrow{\$} \mathsf{Gen}$**:** The probabilistic generation algorithm Gen returns a secret key sk and a corresponding public key pk. We implicitly assume that pk defines a shared key space $\mathcal{K}$.

$(c, k) \xleftarrow{\$} \mathsf{Enc}(sk_s, pk_r)$**:** Given a sender's secret key sk_s and a receiver's public key pk_r, the probabilistic encapsulation algorithm Enc returns a ciphertext c and a shared key $k \in \mathcal{K}$.

$k \leftarrow \mathsf{Dec}(pk_s, sk_r, c)$**:** Given a sender's public key pk_s, a receiver's secret key sk_r, and a ciphertext c, the deterministic decapsulation algorithm Dec returns a shared key $k \in \mathcal{K}$, or a failure symbol $\perp$.

The correctness error δ_{AKEM} is defined as

$$\delta_{\mathsf{AKEM}} := \Pr\left[\mathsf{Dec}(pk_s, sk_r, c) \neq k \;\middle|\; \begin{array}{l}(sk_s, pk_s) \xleftarrow{\$} \mathsf{Gen} \\ (sk_r, pk_r) \xleftarrow{\$} \mathsf{Gen} \\ (c, k) \xleftarrow{\$} \mathsf{Enc}(sk_s, pk_r)\end{array}\right],$$

where the probability is over the randomness of Gen and Enc.

Without loss of generality we assume the existence of an efficiently computable function μ such that for all $(sk, pk) \in \mathsf{Gen}$ it holds $\mu(sk) = pk$.

Confidentiality. We consider the strongest notion of CCA security for an AKEM, in particular that of insider security [3]. As a building block we will also need a weaker notion of CCA security, namely outsider security [3]. We formalise the notion of ciphertext indistinguishability for an authenticated key encapsulation mechanism AKEM via the games depicted in Figure 2 and Figure 3, respectively. The advantage of adversary $\mathcal{A}$ is defined as

$$\mathrm{Adv}^{Q_I\text{-}\mathbf{Ins\text{-}CCA}}_{\mathsf{AKEM},\mathcal{A}} := \left|\Pr\left[Q_I\text{-}\mathbf{Ins\text{-}CCA}_{\mathsf{AKEM}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right|,$$

$$\mathrm{Adv}^{Q_O\text{-}\mathbf{Out\text{-}CCA}}_{\mathsf{AKEM},\mathcal{A}} := \left|\Pr\left[Q_O\text{-}\mathbf{Out\text{-}CCA}_{\mathsf{AKEM}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right|,$$

for $Q_I = (n, Q_{\texttt{Enc}}, Q_{\texttt{Dec}}, Q_{\texttt{Chl}}), Q_O = (n, Q_{\texttt{Enc}}, Q_{\texttt{Dec}})$.

Game $(n, Q_{\texttt{Enc}}, Q_{\texttt{Dec}}, Q_{\texttt{Chl}})$-**Ins-CCA**$_{\mathsf{AKEM}}(\mathcal{A})$

$\mathcal{D} := \emptyset$
for $i \in [n]$
  $(sk_i, pk_i) \xleftarrow{\$} \mathsf{Gen}$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}^{\texttt{Encps},\texttt{Decps},\texttt{Chall}}(pk_1, \ldots, pk_n)$
return $[\![b = b']\!]$

Oracle $\texttt{Encps}(s \in [n], pk)$

$(c, k) \xleftarrow{\$} \mathsf{Enc}(sk_s, pk)$
return (c, k)

Oracle $\texttt{Decps}(pk, r \in [n], c)$

if $\exists\, k : (pk, pk_r, c, k) \in \mathcal{D}$
  return k
$k \leftarrow \mathsf{Dec}(pk, sk_r, c)$
return k

Oracle $\texttt{Chall}(sk, r \in [n])$

$(c, k) \xleftarrow{\$} \mathsf{Enc}(sk, pk_r)$
if $b = 1$
  $k \xleftarrow{\$} \mathcal{K}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mu(sk), pk_r, c, k)\}$
return (c, k)

Figure 2: Game defining **Ins-CCA** for an authenticated key encapsulation mechanism $\mathsf{AKEM} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with adversary $\mathcal{A}$ making at most; $Q_{\texttt{Enc}}$ queries to `Encps`, $Q_{\texttt{Dec}}$ queries to `Decps`, $Q_{\texttt{CSK}}$ queries to `CorSK`, and $Q_{\texttt{Chl}}$ queries to `Chall`.

Game $(n, Q_{\texttt{Enc}}, Q_{\texttt{Dec}})$-**Out-CCA**$_{\mathsf{AKEM}}(\mathcal{A})$

$\mathcal{D} := \emptyset$
for $i \in [n]$
  $(sk_i, pk_i) \xleftarrow{\$} \mathsf{Gen}$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}^{\texttt{Encps},\texttt{Decps}}(pk_1, \ldots, pk_n)$
return $[\![b = b']\!]$

Oracle $\texttt{Encps}(s \in [n], pk)$

$(c, k) \xleftarrow{\$} \mathsf{Enc}(sk_s, pk)$
if $b = 1 \wedge pk \in \{pk_1, \ldots, pk_n\}$
  $k \xleftarrow{\$} \mathcal{K}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(pk_s, pk, c, k)\}$
return (c, k)

Oracle $\texttt{Decps}(pk, r \in [n], c)$

if $\exists\, k : (pk, pk_r, c, k) \in \mathcal{D}$
  return k
$k \leftarrow \mathsf{Dec}(pk, sk_r, c)$
return k

Figure 3: Game defining **Out-CCA** for an authenticated key encapsulation mechanism $\mathsf{AKEM} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with adversary $\mathcal{A}$ making at most; $Q_{\texttt{Enc}}$ queries to `Encps` and $Q_{\texttt{Dec}}$ queries to `Decps`.

Authenticity. We consider outsider authenticity from [3], the strongest notion that is achievable when also seeking

deniability [39]. We formalise the notion via the game depicted in Figure 4 and define the advantage of an adversary $\mathcal{A}$ as

$$\mathrm{Adv}^{Q\text{-}\mathbf{Out\text{-}Aut}}_{\mathsf{AKEM},\mathcal{A}} := \left|\Pr\left[Q\text{-}\mathbf{Out\text{-}Aut}_{\mathsf{AKEM}}(\mathcal{A}) \Rightarrow 1\right] - \tfrac{1}{2}\right|,$$

for $Q = (n, Q_{\mathtt{Enc}}, Q_{\mathtt{Chl}})$.

Games $(n, Q_{\mathtt{Enc}}, Q_{\mathtt{Chl}})$-**Out-Aut**$_{\mathsf{AKEM}}(\mathcal{A})$

$\mathcal{D} := \emptyset$
for $i \in [n]$
  $(sk_i, pk_i) \xleftarrow{\$} \mathsf{Gen}$
$b \xleftarrow{\$} \{0,1\}$
$b' \xleftarrow{\$} \mathcal{A}^{\mathtt{Encps},\mathtt{Chall}}(pk_1, \ldots, pk_n)$
return $[\![b = b']\!]$

Oracle $\mathtt{Encps}(s \in [n], pk)$

$(c,k) \xleftarrow{\$} \mathsf{Enc}(sk_s, pk)$
$\mathcal{D} \leftarrow \mathcal{D} \cup \{(pk_s, pk, c, k)\}$
return (c,k)

Oracle $\mathtt{Chall}(pk, r \in [n], c)$

if $\exists\, k : (pk, pk_r, c, k) \in \mathcal{D}$
  return k
$k \leftarrow \mathsf{Dec}(pk, sk_r, c)$
if $b = 1 \wedge pk \in \{pk_1, \ldots, pk_n\} \wedge k \neq \bot$
  $k \xleftarrow{\$} \mathcal{K}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(pk, pk_r, c, k)\}$
return k

Figure 4: Game defining **Out-Aut** for an authenticated key encapsulation mechanism $\mathsf{AKEM} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with adversary $\mathcal{A}$ making at most $Q_{\mathtt{Enc}}$ queries to `Encps` and $Q_{\mathtt{Chl}}$ queries to `Chall`.

Deniability. As in [39], we consider deniability in two independent settings. For *dishonest receiver* deniability, the receiver is potentially dishonest and capable of simulating ciphertexts. Therefore, the simulator is also given the receiver's secret key. In contrast, in the *honest receiver* setting, the receiver is assumed to behave honestly, and the simulator only has access to public key material. For an authenticated key encapsulation mechanism AKEM and a simulator Sim, we define deniability in the *dishonest receiver* setting and *honest receiver* setting via the games depicted in Figure 5. The advantage of an adversary $\mathcal{A}$ is then defined as

$$\mathrm{Adv}^{(n,Q_{\mathtt{Chl}})\text{-}\mathbf{DR\text{-}Den}}_{\mathsf{AKEM},\mathcal{A},\mathsf{Sim}} := \left|\Pr[(n, Q_{\mathtt{Chl}})\text{-}\mathbf{DR\text{-}Den}_{\mathsf{AKEM},\mathsf{Sim}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}\right|,$$

$$\mathrm{Adv}^{(n,Q_{\mathtt{Chl}})\text{-}\mathbf{HR\text{-}Den}}_{\mathsf{AKEM},\mathcal{A},\mathsf{Sim}} := \left|\Pr[(n, Q_{\mathtt{Chl}})\text{-}\mathbf{HR\text{-}Den}_{\mathsf{AKEM},\mathsf{Sim}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}\right|.$$

Games $(n, Q_{\mathtt{Chl}})$-**DR-Den**$_{\mathsf{AKEM},\mathsf{Sim}}(\mathcal{A})$ and $(n, Q_{\mathtt{Chl}})$-**HR-Den**$_{\mathsf{AKEM},\mathsf{Sim}}(\mathcal{A})$

$\mathcal{R}, \mathcal{C} \leftarrow \emptyset$
for $i \in [n]$
  $(sk_i, pk_i) \xleftarrow{\$} \mathsf{Gen}$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathtt{Rev},\mathtt{Chall}}(pk_1, \ldots, pk_n)$
if $\mathcal{R} \cap \mathcal{C} \neq \emptyset$ /**HR-Den**
  abort /**HR-Den**
return $[\![b = b']\!]$

Oracle $\mathtt{Chall}(s \in [n], r \in [n])$

if $s = r$ **return** $\bot$
$\mathcal{C} \leftarrow \mathcal{C} \cup \{r\}$
$(c,k) \xleftarrow{\$} \mathsf{Enc}(sk_s, pk_r)$
if $b = 1$
  $(c,k) \xleftarrow{\$} \mathsf{Sim}(pk_s, pk_r, sk_r)$ /**DR-Den**
  $(c,k) \xleftarrow{\$} \mathsf{Sim}(pk_s, pk_r)$ /**HR-Den**
return (c,k)

Oracle $\mathtt{Rev}(i \in [n])$

$\mathcal{R} \leftarrow \mathcal{R} \cup \{i\}$
return sk_i

Figure 5: Games defining **DR-Den** and **HR-Den** for an AKEM AKEM and a simulator Sim for adversary $\mathcal{A}$ where $\mathcal{A}$ makes at most $Q_{\mathtt{Chl}}$ queries to `Chall`.

2.3 Pseudorandom Function

Definition 2 (Pseudorandom Function)**.** A keyed function F with a finite key space $\mathcal{K}$, and finite output range $\mathcal{R}$ is a function $F : \mathcal{K} \times \{0,1\}^* \to \mathcal{R}$. We formalise the notion of *pseudorandomess* for a keyed function F via the game $(n, Q_{\mathtt{Eval}})$-**PRF** depicted in Figure 6 and define the advantage of adversary $\mathcal{A}$ as

$$\mathrm{Adv}^{(\mathrm{n},\mathrm{Q}_{\mathtt{Eval}})\text{-}\mathbf{PRF}}_{F,\mathcal{A}} := \left|\Pr\left[(n, Q_{\mathtt{Eval}})\text{-}\mathbf{PRF}_F(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right|.$$

Game $(n, Q_{\mathtt{Eval}})$-**PRF**$_F(\mathcal{A})$

for $i \in [n]$
  $k_i \xleftarrow{\$} \mathcal{K}$
  $f_i \xleftarrow{\$} \{f \mid f : \{0,1\}^* \to \mathcal{R}\}$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathtt{Eval}}$
return $[\![b = b']\!]$

Oracle $\mathtt{Eval}(i \in [n], x)$

if $b = 0$
  return $F(k_i, x)$
if $b = 1$
  return $f_i(x)$

Figure 6: Game defining **PRF** for a keyed function F with adversary $\mathcal{A}$ making at most $Q_{\mathtt{Eval}}$ queries to `Eval`.

Based on a PRF one can also define a dual-PRF [10, 11] which means that the function can be keyed on either the actual key or the (fixed-length) input. This was even further generalised as a split-key PRF [40]. To obtain a uniformly random key from an unpredictable input, a random oracle would be needed. However, if the secrets are uniformly random, then a split-key PRF is sufficient. In particular, the output of the sk-PRF will be pseudorandom. The idea is that adversary $\mathcal{A}$ can first choose the key that is attacked (position j) and is then playing the normal PRF game where the remaining keys (for positions $\ell \in [m] \setminus \{j\}$) that were not chosen as the attacked key act as the input to the function. Note that a sk-PRF can be generically instantiated by calling a dual-PRF multiple times sequentially.

Definition 3 (Split-Key Pseudorandom Function)**.** A multi-keyed function with $m \in \mathbb{N}$ inputs, input space $\mathcal{K}_1 \times \ldots \times \mathcal{K}_n$, and output space $\mathcal{R}$ is a function $F_m : \mathcal{K}_1 \times \ldots \times \mathcal{K}_n \to \mathcal{R}$. We formalise the notion of *split-key pseudorandomess* for a multi-keyed function F_m via the game $(n, Q_{\mathtt{Eval}})$-**PRF** depicted in Figure 7 and define the advantage of adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as

$$\mathrm{Adv}^{(\mathrm{n},\mathrm{Q}_{\mathtt{Eval}})\text{-}\mathbf{PRF}}_{F_m,\mathcal{A}} := \left|\Pr\left[(n, Q_{\mathtt{Eval}})\text{-}\mathbf{PRF}_{F_m}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right|.$$

2.4 Non-Interactive Key Exchange (NIKE)

Definition 4 ((Simplified) Non-Interactive Key Exchange [35, App. G])**.** A *simplified non-interactive key exchange* NIKE is defined as a tuple $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{Sdk})$ of the following algorithms.

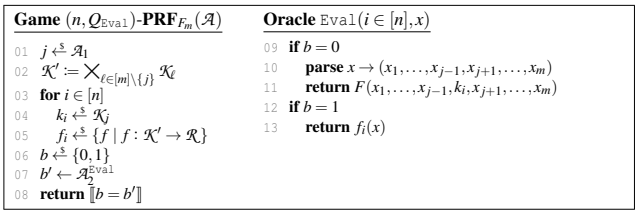

Figure 7: Game defining **PRF** for a multi-keyed function F_m with adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ making at most $Q_{\texttt{Eval}}$ queries to $\texttt{Eval}$.

$par \xleftarrow{\$} \mathsf{Stp}$**:** The probabilistic setup algorithm returns a set of system parameters par. We assume that par implicitly defines a shared key space $\mathcal{K}_{\mathsf{NIKE}}$ and is implicitly accessed by all other algorithms.

$(sk, pk) \xleftarrow{\$} \mathsf{Gen}$**:** Given system parameters par, the probabilistic key generation algorithm Gen returns a secret/public key pair (sk, pk).

$k \leftarrow \mathsf{Sdk}(sk, pk)$**:** Given a secret key sk and a public key pk, the deterministic shared key establishment algorithm Sdk returns a shared key $k \in \mathcal{K}_{\mathsf{NIKE}}$, or a failure symbol $\perp$. We assume that Sdk always returns $\perp$ if sk is the secret key corresponding to pk.

A NIKE is δ_{NIKE} correct if for all $par \in \mathsf{Stp}$

$$\Pr\left[\mathsf{Sdk}(sk_1, pk_2) \neq \mathsf{Sdk}(sk_2, pk_1) \;\middle|\; \begin{array}{l}(sk_1, pk_1) \xleftarrow{\$} \mathsf{Gen} \\ (sk_2, pk_2) \xleftarrow{\$} \mathsf{Gen}\end{array}\right] \leq \delta_{\mathsf{NIKE}}.$$

Security notions can be found in the full version of the paper [37].

2.5 Key Encapsulation Mechanism

Definition 5 (Key Encapsulation Mechanism)**.** A *key encapsulation mechanism* KEM is defined as a tuple $\mathsf{KEM} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of the following algorithms.

$(sk, pk) \xleftarrow{\$} \mathsf{Gen}$**:** The probabilistic key generation algorithm Gen returns a key pair (sk, pk) implicitly defining a shared key space $\mathcal{K}_{\mathsf{KEM}}$.

$(c, k) \xleftarrow{\$} \mathsf{Enc}(pk)$**:** The probabilistic encapsulation algorithm Enc takes as input a public key and returns a ciphertext c and a shared key $k \in \mathcal{K}_{\mathsf{KEM}}$.

$k \leftarrow \mathsf{Dec}(sk, c)$**:** The deterministic decapsulation algorithm Dec takes as input a secret key sk and a ciphertext c and returns a shared key $k \in \mathcal{K}_{\mathsf{KEM}}$ or a failure symbol $\perp$.

The correctness error δ_{KEM} is defined as

$$\delta_{\mathsf{KEM}} := \Pr\left[\mathsf{Dec}(sk, c) \neq k \;\middle|\; \begin{array}{ll}(sk, pk) & \xleftarrow{\$} \mathsf{Gen} \\ (c, k) & \xleftarrow{\$} \mathsf{Enc}(pk)\end{array}\right].$$

Security notions can be found in the full version of the paper [37].

2.6 Ring Signatures

Syntax. We recall syntax and standard security notions of ring signatures [70].

Definition 6 (Ring Signature)**.** A *ring signature* scheme RSig is defined as a tuple $(\mathsf{Stp}, \mathsf{Gen}, \mathsf{Sgn}, \mathsf{Ver})$ of the following algorithms.

$par \xleftarrow{\$} \mathsf{Stp}(\kappa)$**:** Given an upper bound, κ, on the ring size the probabilistic setup algorithm Stp returns system parameters par, where par defines a message space $\mathcal{M}$. We assume that all algorithms are implicitly given access to the system parameters par.

$(sk, pk) \xleftarrow{\$} \mathsf{Gen}$**:** The probabilistic key generation algorithm returns a secret key sk and a corresponding public key pk.

$\sigma \xleftarrow{\$} \mathsf{Sgn}(sk, \rho, m)$**:** Given a secret key sk, a ring $\rho = \{pk_1, \ldots, pk_k\}$ such that the public key pk corresponding to sk satisfies $pk \in \rho$ and $k \leq \kappa$, and a message $m \in \mathcal{M}$, the probabilistic signing algorithm Sgn returns a signature σ from a signature space $\mathcal{S}$.

$b \leftarrow \mathsf{Ver}(\sigma, \rho, m)$**:** Given a signature σ, a ring ρ, and a message m, the deterministic verification algorithm Ver returns a bit b, such that $b = 1$ if and only if σ is a valid signature on m and $b = 0$ otherwise.

RSig is $\delta(\kappa)$-*correct* or has *correctness error* $\delta(\kappa)$ if for all $\kappa \in \mathbb{N}$, $par \xleftarrow{\$} \mathsf{Stp}(\kappa)$, and $\{(sk_i, pk_i)\}_{i \in [k]} \in \sup(\mathsf{Gen})$, and for any $i \in [k]$ with $k \leq \kappa$,

$$\Pr[\mathsf{Ver}(\mathsf{Sgn}(sk_i, \rho, m), \rho, m) \neq 1] \leq \delta(\kappa),$$

where $\rho := \{pk_1, \ldots, pk_k\}$, and the probability is taken over the random choices of Stp, Gen and Sgn.

We assume (w.l.o.g.) that there is a mapping μ from the space of secret keys to the space of public keys such that for all $(sk, pk) \in \sup(\mathsf{Gen})$ it holds $\mu(sk) = pk$.

Security notions can be found in the full version of the paper [37].

2.7 Symmetric Encryption

Definition 7 (Symmetric Encryption)**.** A *symmetric encryption* Sym is defined as a tuple $\mathsf{Sym} := (\mathsf{Enc}, \mathsf{Dec})$ of the following algorithms and a key space $\mathcal{K}_{\mathsf{Sym}}$.

$c \leftarrow \mathsf{Enc}(k, m)$**:** The deterministic encryption algorithm Enc takes as input a symmetric key k and a message m and outputs a ciphertext c.

$m \leftarrow \mathsf{Dec}(k, c)$**:** The deterministic decryption algorithm Dec takes as input a symmetric key k and a ciphertext c and outputs a message m.

Sym is (perfectly) correct if for all $k \in \mathcal{K}_{\mathsf{Sym}}$ and all messages m it holds $m = \mathsf{Dec}(k, \mathsf{Enc}(k,m))$.

Security notions can be found in the full version of the paper [37].

3 Generic Construction

In this section, we present a generic construction for a deniable AKEM combiner derived from two deniable AKEMs, AKEM_1 and AKEM_2, and a multi-keyed function H with five inputs which is illustrated in Figure 8. This construction builds upon the natural approach proposed in [40]. Regarding security, our results are as follows: For confidentiality (see Theorem 2) and authenticity (see Theorem 3) the combiner requires only one of the underlying AKEMs to ensure confidentiality or authenticity, aligning with the expected behaviour of a combiner. However, for deniability, we prove that our generic black-box construction requires that both schemes be deniable. Specifically, Theorem 4 shows that if both schemes are *dishonest receiver* deniable, then the combiner inherits this property. Similarly, Theorem 5 establishes that the combiner maintains deniability in the *honest receiver* setting if both underlying schemes are honest receiver deniable.

Lemma 1 (Correctness)**.** *If* AKEM_1 *has correctness error* δ_1 *and* AKEM_2 *correctness error* δ_2*, then* $\delta_{\mathsf{AKEM}[\mathsf{AKEM}_1,\mathsf{AKEM}_2,\mathsf{H}]} \leq \delta_1 + \delta_2$.

Theorem 2 (Confidentiality)**.** *For any* **Ins-CCA** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{AKEM}_1,\mathsf{AKEM}_2,\mathsf{H}]$*, depicted in Figure 8, there exists an* **Ins-CCA** *adversary* $\mathcal{B}_1$ *against* AKEM_1*, an* **Ins-CCA** *adversary* $\mathcal{B}_2$ *against* AKEM_2*, and a* **PRF** *adversary* $\mathcal{C}$ *against* H *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}_1} \approx t_{\mathcal{B}_2} \approx t_{\mathcal{C}}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{\text{Q-Ins-CCA}}_{\Pi,\mathcal{A}} \leq 2 \cdot \min\left\{\mathrm{Adv}^{Q\text{-Ins-CCA}}_{\mathsf{AKEM}_1,\mathcal{B}_1},\ \mathrm{Adv}^{Q\text{-Ins-CCA}}_{\mathsf{AKEM}_2,\mathcal{B}_2}\right\} \\ + 2 \cdot \mathrm{Adv}^{Q_{\mathsf{H}}\text{-PRF}}_{\mathsf{H},\mathcal{C}} + Q_{Chl} \cdot \delta_{\Pi}\end{aligned}$$

for $Q = (n, Q_{Enc}, Q_{Dec}, Q_{Chl}), Q_{\mathsf{H}} = (Q_{Chl}, Q_{Dec} + Q_{Chl})$.

Proof (Sketch). If AKEM_1 or AKEM_2 is **Ins-CCA** secure, one of the input keys to H is uniformly random. With the **PRF** security of H, the key of the combiner is uniformly random too. The full proof can be found in the full version of the paper [37]. ■

Theorem 3 (Authenticity)**.** *For any* **Out-Aut** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{AKEM}_1,\mathsf{AKEM}_2,\mathsf{H}]$*, as depicted in Figure 8, there exists an* **Out-Aut** *adversary* $\mathcal{B}_1$ *against* AKEM_1*, an* **Out-Aut** *adversary* $\mathcal{B}_2$ *against* AKEM_2*, an* **Out-CCA** *adversary* $\mathcal{C}_1$ *against* AKEM_1*, an* **Out-CCA** *adversary* $\mathcal{C}_2$ *against* AKEM_2*, and a* **PRF** *adversary* $\mathcal{D}$ *against* H *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}_1} \approx t_{\mathcal{B}_2} \approx t_{\mathcal{C}_1} \approx t_{\mathcal{C}_2} \approx t_{\mathcal{D}}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{Q\text{-Out-Aut}}_{\Pi,\mathcal{A}} \leq 2 \cdot \min\Big\{&\mathrm{Adv}^{Q\text{-Out-Aut}}_{\mathsf{AKEM}_1,\mathcal{B}_1} + \mathrm{Adv}^{Q\text{-Out-CCA}}_{\mathsf{AKEM}_1,\mathcal{C}_1}, \\ &\mathrm{Adv}^{Q\text{-Out-Aut}}_{\mathsf{AKEM}_2,\mathcal{B}_2} + \mathrm{Adv}^{Q\text{-Out-CCA}}_{\mathsf{AKEM}_2,\mathcal{C}_2}\Big\} \\ &+ 2 \cdot \mathrm{Adv}^{Q_{\mathsf{H}}\text{-PRF}}_{\mathsf{H},\mathcal{D}} + Q_{Chl} \cdot \delta_{\Pi}\end{aligned}$$

for $Q = (n, Q_{Enc}, Q_{Chl}), Q_{\mathsf{H}} = (Q_{Enc} + Q_{Chl}, Q_{Enc} + Q_{Chl})$.

Proof (Sketch). If AKEM_1 or AKEM_2 is **Out-Aut** secure, an adversary cannot construct a ciphertext for which they can distinguish a real decapsulation from a uniformly random key. With the **PRF** security of H, the key of the combiner should be uniformly random too. However, in contrast to the confidentiality case these properties are not enough because an adversary could use the encapsulation oracle to produce a ciphertext for which they know the key and then recycle one part of the ciphertext. To avoid this, we also require confidentiality (**Out-CCA**) of AKEM_1 or AKEM_2. The full proof can be found in the full version of the paper [37]. ■

Theorem 4 (Dishonest Deniability)**.** *For any* **DR-Den** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{AKEM}_1,\mathsf{AKEM}_2,\mathsf{H}]$*, as depicted in Figure 8, any simulators* $\mathsf{Sim}_1, \mathsf{Sim}_2$*, and simulator* $\mathsf{Sim}[\mathsf{Sim}_1,\mathsf{Sim}_2]$ *as defined in the proof, there exists a* **DR-Den** *adversary* $\mathcal{B}_1$ *against* AKEM_1 *and a* **DR-Den** *adversary* $\mathcal{B}_2$ *against* AKEM_2 *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}_1} \approx t_{\mathcal{B}_2}$ *such that*

$$\mathrm{Adv}^{Q\text{-DR-Den}}_{\Pi,\mathsf{Sim},\mathcal{A}} \leq 2 \cdot \mathrm{Adv}^{Q\text{-DR-Den}}_{\mathsf{AKEM}_1,\mathsf{Sim}_1,\mathcal{B}_1} + 2 \cdot \mathrm{Adv}^{Q\text{-DR-Den}}_{\mathsf{AKEM}_2,\mathsf{Sim}_2,\mathcal{B}_2}$$

for $Q = (n, Q_{Chl})$.

Proof (Sketch). The simulator of the combiner can be constructed by using the simulators of the underlying schemes. For this reason the security relies on both the AKEMs. The full proof can be found in the full version of the paper [37]. ■

Theorem 5 (Honest Deniability)**.** *For any* **HR-Den** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{AKEM}_1,\mathsf{AKEM}_2,\mathsf{H}]$*, as depicted in Figure 8, any simulators* $\mathsf{Sim}_1, \mathsf{Sim}_2$*, and simulator* $\mathsf{Sim}[\mathsf{Sim}_1,\mathsf{Sim}_2]$ *as defined in the proof, there exists a* **HR-Den** *adversary* $\mathcal{B}_1$ *against* AKEM_1 *and a* **HR-Den** *adversary* $\mathcal{B}_2$ *against* AKEM_2 *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}_1} \approx t_{\mathcal{B}_2}$ *such that*

$$\mathrm{Adv}^{Q\text{-HR-Den}}_{\Pi,\mathsf{Sim},\mathcal{A}} \leq 2 \cdot \mathrm{Adv}^{Q\text{-HR-Den}}_{\mathsf{AKEM}_1,\mathsf{Sim}_1,\mathcal{B}_1} + 2 \cdot \mathrm{Adv}^{Q\text{-HR-Den}}_{\mathsf{AKEM}_2,\mathsf{Sim}_2,\mathcal{B}_2}$$

for $Q = (n, Q_{Chl})$.

Proof. The theorem can be proved analogously to Theorem 4. ■

$\underline{\mathsf{Gen}}$

01 $(sk_1, pk_1) \xleftarrow{\$} \mathsf{AKEM}_1.\mathsf{Gen}$
02 $(sk_2, pk_2) \xleftarrow{\$} \mathsf{AKEM}_2.\mathsf{Gen}$
03 $sk := (sk_1, sk_2)$
04 $pk := (pk_1, pk_2)$
05 **return** (sk, pk)

$\underline{\mathsf{Enc}(sk_s, pk_r)}$

06 **parse** $sk_s \to (sk_1, sk_2)$
07 **parse** $pk_r \to (pk_1, pk_2)$
08 $(c_1, k_1) \xleftarrow{\$} \mathsf{AKEM}_1.\mathsf{Enc}(sk_1, pk_1)$
09 $(c_2, k_2) \xleftarrow{\$} \mathsf{AKEM}_2.\mathsf{Enc}(sk_2, pk_2)$
10 $c := (c_1, c_2)$
11 $k := \mathsf{H}(k_1, k_2, (\mu(sk_1), \mu(sk_2)), (pk_1, pk_2), c)$
12 **return** (c, k)

$\underline{\mathsf{Dec}(pk_s, sk_r, c)}$

13 **parse** $pk_s \to (pk_1, pk_2)$
14 **parse** $sk_r \to (sk_1, sk_2)$
15 **parse** $c \to (c_1, c_2)$
16 $k_1 \leftarrow \mathsf{AKEM}_1.\mathsf{Dec}(pk_1, sk_1, c_1)$
17 $k_2 \leftarrow \mathsf{AKEM}_2.\mathsf{Dec}(pk_2, sk_2, c_2)$
18 $k := \mathsf{H}(k_1, k_2, (pk_1, pk_2), (\mu(sk_1), \mu(sk_2)), c)$
19 **return** k

Figure 8: Generic Construction of a deniable authenticated key encapsulation mechanism $\mathsf{AKEM}[\mathsf{AKEM}_1, \mathsf{AKEM}_2, \mathsf{H}]$.

4 Hybrid Construction: SHADOWFAX

In this section, we present a hybrid construction for a deniable AKEM based on a non-interactive key exchange NIKE, a key encapsulation mechanism KEM, a ring signature scheme RSig, a symmetric encryption scheme Sym, and two split-key PRFs H_1 and H_2 (H_1 having two inputs and H_2 having six inputs), as shown in Figure 9. This approach leverages well-known cryptographic primitives that can be instantiated from concrete schemes, providing a practical construction. Our security results are as follows: For both confidentiality (see Theorem 7) and authenticity (see Theorem 8), the combiner requires only one of the underlying AKEMs to ensure the respective property, consistent with the generic combiner. Confidentiality is provided by the security of either the ephemeral NIKE or the KEM. Authenticity comes from the static NIKE (providing implicit authentication) or the ring signature. For dishonest receiver deniability (see Theorem 9) we only rely on security advantages that can be instantiated with statistical security arguments, specifically the correctness property of the NIKE and the anonymity property of the ring signature. Finally, we achieve honest receiver deniability (see Theorem 10) by relying on just one of the underlying computational assumptions – specifically, the security of either the ephemeral NIKE or the KEM – to ensure deniability for the combiner. The main challenge arises from the public verifiability of the ring signature. [39] addresses this issue by symmetrically encrypting the ring signature using the KEM key. We implement a similar solution but derive the key material from *both* the NIKE and the KEM. This design mirrors our approach for confidentiality, ensuring that an adversary would need to compromise both the NIKE and KEM in order to verify the signature. Additionally H_1 is used twice in the construction to simplify the instantiation and used with a tag `"auth"` for domain separation in the proof. The setup of NIKE and RSig are implicitly done; for RSig by inputting maximum ring size 2.

Lemma 6 (Correctness). *If* NIKE *has correctness error* δ_{NIKE}, KEM *correctness error* δ_{KEM}, *and* RSig *correctness error* δ_{RSig} *and* Sym *is (perfectly) correct, then*

$$\delta_{\mathsf{AKEM}[\mathsf{NIKE},\mathsf{KEM},\mathsf{RSig},\mathsf{Sym},\mathsf{H}_1,\mathsf{H}_2]} \leq \delta_{\mathsf{NIKE}} + \delta_{\mathsf{KEM}} + \delta_{\mathsf{RSig}}.$$

Theorem 7 (Confidentiality). *For any* **Ins-CCA** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{NIKE},\mathsf{KEM},\mathsf{RSig},\mathsf{Sym},\mathsf{H}_1,\mathsf{H}_2]$, *as depicted in Figure 9, there exists a* **CKS** *adversary* $\mathcal{B}$ *against* NIKE, *a* **PRF** *adversary* $\mathcal{C}$ *against* H_1, *an* **PRF** *adversary* $\mathcal{D}$ *against* H_2, *and an* **IND-CCA** *adversary* $\mathcal{E}$ *against* KEM *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}} \approx t_{\mathcal{C}} \approx t_{\mathcal{D}} \approx t_{\mathcal{E}}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{Q\text{-}\mathbf{Ins\text{-}CCA}}_{\Pi,\mathcal{A}} \leq 2nQ_{Ch1} \cdot \Big(\min\Big\{&\mathrm{Adv}^{Q_{\mathsf{NIKE}}\text{-}\mathbf{CKS}}_{\mathsf{NIKE},\mathcal{B}} + \mathrm{Adv}^{(1,1)\text{-}\mathbf{PRF}}_{\mathsf{H}_1,\mathcal{C}}, \mathrm{Adv}^{(1,Q_{Dec},1)\text{-}\mathbf{IND\text{-}CCA}}_{\mathsf{KEM},\mathcal{E}}\Big\} \\ &+ 2 \cdot \mathrm{Adv}^{(1,Q_{Dec}+1)\text{-}\mathbf{PRF}}_{\mathsf{H}_2,\mathcal{D}} + (Q_{Enc} + Q_{Dec}) \cdot \eta_{\mathsf{NIKE}} \cdot \gamma_{\mathsf{KEM}} + Q_{Ch1} \cdot \delta_{\Pi}\Big)\end{aligned}$$

for $Q = (n, Q_{Enc}Q_{Dec}, Q_{Ch1}), Q_{\mathsf{NIKE}} = (Q_{Enc} + 2, 2Q_{Enc} + 2Q_{Dec}, 2Q_{Enc} + 2Q_{Enc} + 1)$.

Proof (Sketch). If the NIKE is **CKS** secure and H_1 a **PRF**, one of the keys for the split-key PRF H_2 is uniformly random and therefore the output of the hybrid AKEM construction. Analogously, the security can be based on the **IND-CCA** security of the KEM. In this case, another input key of H_2 is uniformly random and thus the hybrid's key. The full proof can be found in the extended version of the paper [37]. ■

Theorem 8 (Authenticity). *For any* **Out-Aut** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{NIKE},\mathsf{KEM},\mathsf{RSig},\mathsf{Sym},\mathsf{H}_1,\mathsf{H}_2]$, *as depicted in Figure 9, there exists an* **CKS** *adversary* $\mathcal{B}$ *against* NIKE, *a* **PRF** *adversary* $\mathcal{C}$ *against* H_1, *an* **PRF** *adversary* $\mathcal{D}$ *against* H_2, *a* **UF-CRA1** *adversary* $\mathcal{E}$ *against* RSig, *and an* **IND-CCA** *adversary* $\mathcal{F}$ *against* KEM *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}} \approx t_{\mathcal{C}} \approx t_{\mathcal{D}} \approx t_{\mathcal{E}} \approx t_{\mathcal{F}}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{Q\text{-}\mathbf{Out\text{-}Aut}}_{\Pi,\mathcal{A}} \leq \min\Big\{&2 \cdot \mathrm{Adv}^{Q_{\mathsf{NIKE}}\text{-}\mathbf{CKS}}_{\mathsf{NIKE},\mathcal{B}} + 2 \cdot \mathrm{Adv}^{(n^2,n^2)\text{-}\mathbf{PRF}}_{\mathsf{H}_1,\mathcal{C}}, \\ &\mathrm{Adv}^{(n,2,Q_{Enc})\text{-}\mathbf{UF\text{-}CRA1}}_{\mathsf{RSig},\mathcal{E}} + 2 \cdot \mathrm{Adv}^{Q\text{-}\mathbf{IND\text{-}CCA}}_{\mathsf{KEM},\mathcal{F}} + Q^2_{Enc} \cdot \gamma_{\mathsf{KEM}}\Big\} \\ &+ 2 \cdot \mathrm{Adv}^{(Q',Q')\text{-}\mathbf{PRF}}_{\mathsf{H}_2,\mathcal{D}} + Q_{Ch1} \cdot \delta_{\Pi} + Q_{Enc} \cdot Q' \cdot \eta_{\mathsf{NIKE}} \cdot \gamma_{\mathsf{KEM}}\end{aligned}$$

with $Q = (n, Q_{Enc}, Q_{Ch1}), Q_{\mathsf{NIKE}} = (Q_{Enc} + 2Q_{Ch1}, Q_{Enc} + 2Q_{Ch1}), Q' = Q_{Enc} + Q_{Ch1}$.

Proof (Sketch). If the RSig is **UF-CRA1** it is hard for an adversary to come up with a valid ciphertext unless it was produced by the encapsulation oracle. This can be a valid attack since the adversary can recycle one part of the encapsulation output and thus produce a fresh and valid ciperhetxt. Additionally requiring the KEM to be **IND-CCA** prevents this attack. An analogous argument can be made if the NIKE is **CKS** secure. Note that a NIKE is used for

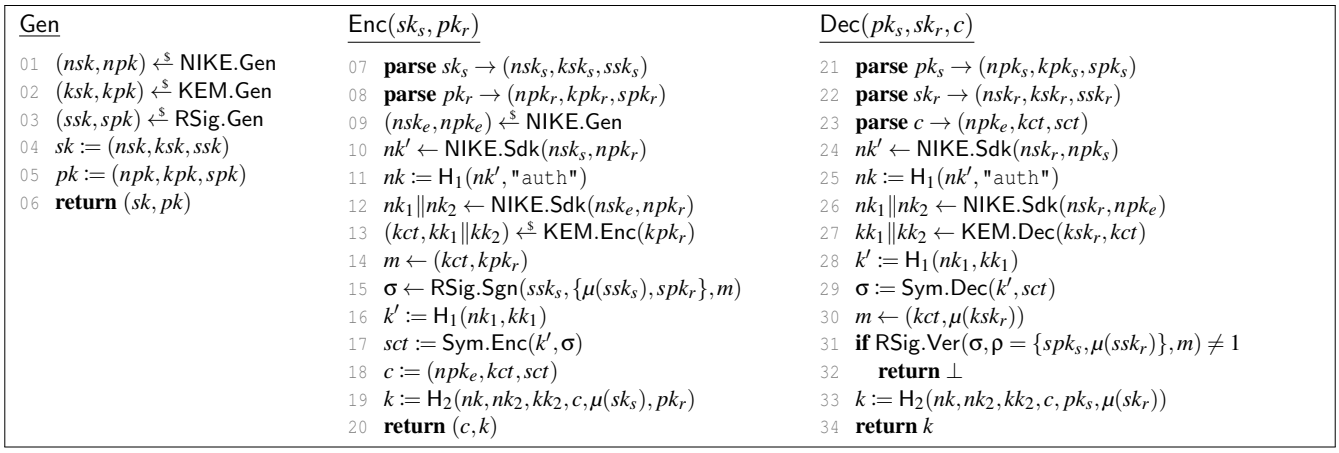

Figure 9: Concrete construction of a deniable AKEM $\mathsf{AKEM}[\mathsf{NIKE}, \mathsf{KEM}, \mathsf{RSig}, \mathsf{Sym}, \mathsf{H}_1, \mathsf{H}_2]$. By "$\|$" we denote that an output is split into two equal parts.

authenticity (analogue to the **UF-CRA1** argument) and for confidentiality (analogue to the **IND-CCA** argument). Applying the **PRF** security of H_2 makes the hybrid key uniformly random. The full proof can be found in the full version of the paper [37]. ■

Theorem 9 (Dishonest Deniability)**.** *For any* **DR-Den** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{NIKE}, \mathsf{KEM}, \mathsf{RSig}, \mathsf{Sym}, \mathsf{H}_1, \mathsf{H}_2]$*, as depicted in Figure 9, and simulator* Sim *as defined in the proof, there exists a* **MC-Ano** *adversary* $\mathcal{B}$ *against* RSig *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}}$ *such that*

$$\mathrm{Adv}^{(n,\mathrm{Q}_{Ch1})\text{-}\mathbf{DR\text{-}Den}}_{\Pi,\mathsf{Sim},\mathcal{A}} \leq \mathrm{Adv}^{(n,2,\mathrm{Q}_{Ch1})\text{-}\mathbf{MC\text{-}Ano}}_{\mathsf{RSig},\mathcal{B}} + Q_{Ch1} \cdot \delta_{\mathsf{NIKE}}.$$

Proof (Sketch). To achieve dishonest receiver deniability, the simulator must create an indistinguishable NIKE and RSig part. For the NIKE, this reduces to the NIKE's correctness since the simulator has access to the receiver's secret key. For the RSig, this reduces to **MC-Ano**. Note that both properties must be fulfilled because distinguishing one part is sufficient for the adversary to win their game. The full proof can be found in the full version of the paper [37]. ■

Theorem 10 (Honest Deniability)**.** *For any* **HR-Den** *adversary* $\mathcal{A}$ *against* $\Pi := \mathsf{AKEM}[\mathsf{NIKE}, \mathsf{KEM}, \mathsf{RSig}, \mathsf{Sym}, \mathsf{H}_1, \mathsf{H}_2]$*, as depicted in Figure 9,and simulator* Sim *as defined in the proof, there exists a* **CKS** *adversary* $\mathcal{B}$ *against* NIKE*, an* **IND-CPA** *adversary* $\mathcal{C}$ *against* KEM*,* **PRF** *adversaries* $\mathcal{D}$ *and* $\mathcal{E}$ *against* H_1 *and* H_2*, and a* **IND-CPA** *adversary* $\mathcal{F}$ *against* Sym *with* $t_{\mathcal{A}} \approx t_{\mathcal{B}} \approx t_{\mathcal{C}} \approx t_{\mathcal{D}} \approx t_{\mathcal{E}} \approx t_{\mathcal{F}}$ *such that*

$$\mathrm{Adv}^{(n,\mathrm{Q}_{Ch1})\text{-}\mathbf{HR\text{-}Den}}_{\Pi,\mathsf{Sim},\mathcal{A}} \leq 4n^2 \cdot Q_{Ch1} \cdot \Big(\min\Big\{\mathrm{Adv}^{(2,0,1)\text{-}\mathbf{CKS}}_{\mathsf{NIKE},\mathcal{B}}, \mathrm{Adv}^{(1,1)\text{-}\mathbf{IND\text{-}CPA}}_{\mathsf{KEM},\mathcal{C}}\Big\} + \mathrm{Adv}^{(1,1)\text{-}\mathbf{PRF}}_{\mathsf{H}_1,\mathcal{D}} + \mathrm{Adv}^{(1,1)\text{-}\mathbf{PRF}}_{\mathsf{H}_2,\mathcal{E}} + \mathrm{Adv}^{\mathbf{IND\text{-}CPA}}_{\mathsf{Sym},\mathcal{F}}\Big).$$

Proof (Sketch). Authenticity via a NIKE is not a problem for honest receiver deniability because authentication is made implicit and there is no information about it in the AKEM ciphertext. This is different from the ring signature which can be publicly verified which is why it is encrypted in the construction. Hence, security can be reduced to said encryption. Since the symmetric encryption key is derived from a NIKE key and a KEM key via a split-key PRF, we can reduce to **CKS** security of NIKE or **IND-CPA** of the KEM. Applying the **PRF** property of H_2 yields a uniformly random encryption key which is used to apply Sym's **IND-CPA** security. The full proof can be found in the full version of the paper [37]. ■

5 Implementation

In this paper, we instantiate the GANDALF ring signature scheme, the corresponding post-quantum AKEM by [39], and the hybrid AKEM SHADOWFAX of this paper with several choices for the underlying post-quantum KEM and trapdoor sampler.

Optimisation Goals. We aim for portability, the compactness of public key and ciphertext sizes, and compliance with standardised components in our instantiations of post-quantum and hybrid AKEMs. Since the rapid development of Post-Quantum Cryptography Standardisation by the National Institute of Standards and Technology (NIST), there are rich C reference implementations for several post-quantum cryptosystems.[5] We follow a similar paradigm and implement the AKEMs with the C programming language[6]. Since C is a high-level

[5] Additionally, standardised building blocks often provide deeper analyses [1, 2, 19, 38] providing more trust.

[6] The only exception is the reference implementation of FALCON: The latest source code by [66] comes with built-in selection for platform-specific intrinsics and falls back to C with integer emulation for the floating-point operations.

Table 1: Sizes (in bytes) of GANDALF ring signature instantiations, key-encapsulation mechanisms, the resulting deniable AKEMs, and SHADOWFAX hybrid instantiations derived from the initial AKEM. Sizes are from our implementation. Unless stated otherwise, "✓" denotes implementations provided in this work. The most compact instantiation and the one based on NIST standards are highlighted.

RSig				KEM				(PQ-)AKEM			NIKE			SHADOWFAX		
Scheme	Size		Impl.	Scheme	Size		Impl.	Size		Impl.	Scheme	Size	Impl.	Size		Impl.
	pk	σ			*pk*	*c*		*pk*	*c*			*pk*		*pk*	*c*	
GANDALF [ANTRAG, MITAKA]	896	1 276	✓	NTRU-A	768	768	✗	1 664	2 044	✗	Curve25519	32	✓ [14]	1 696	2 076	✗
				BAT	521	473	✓ [34]	1 417	1 749	✓				1 449	1 781	✓
				ML–KEM	800	768	✓ [73]	1 696	2 044	✓				1 728	2 076	✓
GANDALF [FALCON, FALCON]	896	1 276	✓	NTRU-A	768	768	✗	1 664	2 044	✗				1 696	2 076	✗
				BAT	521	473	✓ [34]	1 417	1 749	✓				1 449	1 781	✓
				ML–KEM	800	768	✓ [73]	1 696	2 044	✓				1 728	2 076	✓

Table 2: Parameter sets for BAT, ML–KEM, and FALCON in this paper.

Scheme	BAT	ML–KEM	FALCON
Parameter set	`bat-257-512`	`mlkem-512`	`falcon-512`

programming language, our instantiations are portable. When compactness is the main optimisation goal, we choose BAT [34], ANTRAG [30], and MITAKA [29] along with the latest NTRU solver by [65]. If standardised components are preferred, we choose ML–KEM [60] and the latest implementation of the fast Fourier sampler [28] from FALCON [66, 67]. Table 2 summarises the chosen parameter sets for BAT, ML–KEM, and FALCON in this paper. Our instantiation is also modular and one can replace the post-quantum KEM and the trapdoor sampler with other combinations after some wrapping for the API. Table 1 summarises the schemes used to instantiate the GANDALF ring signature and the KEM, the implemented components, and the resulting sizes of public keys, signatures, and ciphertexts. The signature size of one instantiation is 40 bytes larger than in the original proposal [39], due to different compression techniques. The authors of [39] assumed the techniques of [29, 31], but no implementations of those techniques were available. We therefore use the compression from the round-3 FALCON submission, where the signature has a fixed size of 666 bytes (including a 40-byte nonce). This results in a ring signature of 1 276 bytes with a 24-byte nonce for all GANDALF instantiations in this work. There is a concurrent work instantiating GANDALF [47]. Due to a different compression and a different salt size, they achieve slightly larger signatures of 1 288 bytes. All our source code is publicly available on the GitHub repository Shadowfax. [7] A comparison in security and size with different AKEMs from the literature is shown in Table 3.

[7] https://github.com/vincentvbh/shadowfax

5.1 Instantiation

HASH. Our instantiations use five distinct hash functions. `BLAKE2b` [72] which is shipped with BAT. `shake256` is used for converting the 32-byte shared key from ML–KEM to a 64-byte shared key. We use `shake128` [48] for hashing the message to a polynomial in the ring signature. Additionally, `SHA3-512` is used in the Non-Interactive Key Exchange (NIKE) construction, while `SHA3-256` is used for the hash functions H_1 and H_2 in the concrete construction (see Figure 9). Specifically, H_1 is implemented as the hmac `HMAC-SHA3-256` derived from `SHA3-256`. As for H_2, we implement it with three `HMAC-SHA3-256` calls as follows: $\mathsf{H}_2(nk, nk_2, kk_2, c, \mu(sk_s), pk_r) = \texttt{HMAC-SHA3-256}(nknk_2, [\texttt{rest}]_{kk_2})$ where

$$\begin{cases} nknk_2 = & \texttt{HMAC-SHA3-256}(nk, nk_2), \\ [\texttt{rest}]_{kk_2} = & \texttt{HMAC-SHA3-256}(kk_2, [\texttt{rest}]), \end{cases}$$

and `[rest]` is the concatenation of the rest of the inputs. Note that our instantiations of H_1 and H_2 align with what we actually proved in Theorem 7. HMAC has been proven to be a dual-PRF [7] and the consecutive calls as described above instantiate a split-key PRF.

SYMMETRIC ENCRYPTION, NIKE AND KEM. We choose the CTR mode of `AES-128` for the symmetric encryption. For the NIKE, we choose the `Curve25519` Diffie-Hellman [14] based on the `ref10` implementation of `crypto_scalarmult/curve25519` from `supercop-20240716` [27] and `SHA3-512`. After computing the raw Diffie-Hellman shared secret, we pass it through `SHA3-512` to derive the shared key for the NIKE. For the post-quantum KEM, we consider two options: ML–KEM [60] and BAT [34]. For ML–KEM, we pass the 32-byte shared key to `shake256` and expand it to a 64-byte shared key. For BAT, we integrate the latest NTRU solver by [65], enforce the use of `BLAKE2b` in encapsulation and decapsulation, and simplify the source code with the C preprocessor.

NTRU SOLVER. In BAT and ANTRAG, we have to solve for polynomials $F, G \in \mathbb{Z}[X]/\langle X^N+1\rangle$ satisfying the following NTRU equation: $g \cdot F - f \cdot G = q \bmod (X^N+1)$ for a power-of-two N, a positive integer q, and polynomials $g, f \in \mathbb{Z}_q[X]/\langle X^N+1\rangle$ with small coefficients. We integrate the latest NTRU solver by [65] into BAT and ANTRAG.

5.1.1 Ring Signature.

We choose GANDALF [39] for the ring signature. According to [39], GANDALF achieves the smallest signature size for the ring of size 2, which suits well for constructing our AKEM. We consider two options: (i) components from FALCON and (ii) ANTRAG with MITAKA. For (i), we reorganise the construction of the signature generation and extract the trapdoor sampler. For (ii), we integrate the latest NTRU solver by [65] to the ANTRAG trapdoor generation [30] and outline below the necessary changes for achieving a compact signature size.

Modifications of MITAKA implementation. In the reference implementation of MITAKA released in [29], the signatures are stored as double-precision floating-point numbers with non-zero fractional parts, as opposed to integers. Therefore, existing compression techniques, which are defined over integers, cannot be straightforwardly deployed. Furthermore, there is no implementation for the latest compression technique [31] required by [29] and later used in [39]. Instead, we pull everything back to integers whenever the remaining computation can be defined entirely over $\mathbb{Z}$ and plug in the signature compression from the round 3 submission package of Falcon [67]. This results in a 40-byte increase of signature size compared to the original GANDALF by [39]. In the reference implementation of MITAKA, the program proceeds with double-precision floating-point arithmetic entirely, verifies the validity of signatures with double-precision floating-point arithmetic, and skips the signature compression. Finally, we also tweak the output of the sampler so it aligns with the definition of the trapdoor sampler. In the description of the MITAKA sampler, the output of the trapdoor sampler is negated and cannot be used directly in the ring signature scheme, as samples are supposed to be indistinguishable between parties. Therefore, we negate the output of the sampler.

5.2 Performance

Benchmarking Environment. We benchmark our portable implementations on the Firestorm core of an Apple M1 Pro with the operating system macOS Sonoma 14.6.1. Firestorm is the "big" core of the "big.LITTLE" computing architecture prevalent in Arm-based architecture in personal computing devices. It runs at the frequency of 3GHz and comes with a dedicated cryptographic extension. As we aim for portable implementations, we do not use the cryptographic extension. All programs are compiled with `GCC 13.3.0` with the optimisation flag `-O3`.

Cycle counts. Table 4 summarises the cycle counts of the C implementations of DH-AKEM, PQ-AKEM, and SHADOWFAX. For the key generations PQ-AKEM and SHADOWFAX, the cycle count is dominated by the NTRU solver used in BAT, MITAKA, and FALCON. For the encapsulation, the cycle count is dominated by the signing of GANDALF. As for the decapsulation, the cycle count is dominated by the NIKE in SHADOWFAX and by ML–KEM/BAT in PQ-AKEM. We also give the cycle counts of our portable implementations of the ring signature GANDALF, and benchmark the C implementations of Raptor by [54] and the implementation by [47] on our platform. We stress that the C implementation of Raptor is based on an earlier implementation of Falcon, which had been significantly refactored after the publication of [54].

Conclusion. The dominant cost in terms of ciphertext size arises from the post-quantum ring signature, followed by the post-quantum KEM ciphertext. Public key sizes are less of a concern, and the overhead of the pre-quantum AKEM is minimal. Notably, this implies that with a post-quantum deniable AKEM, the cost of constructing a hybrid with strong security properties is virtually negligible. In conclusion, whenever a post-quantum AKEM is needed (regardless of whether it needs to be deniable), incorporating a hybrid should always be considered.

Table 3: Comparison of AKEMs, their security notions, and reliance on pre-quantum (pre-Q) or post-quantum (post-Q) assumptions. Deniability notions marked with "†" are not formally proven in the original works. Entries marked "⋆" indicate theoretically achievable sizes. The SWOOSH [36] size refers to a passively secure NIKE; achieving active security requires a NIZK, which must be added to the NIKE public key. DUALRING [79, Tab. 3] is included because the parameters of GANDALF would need to be slightly increased for stronger concrete anonymity (see [39] for details).

Scheme	Instantiation	Confidentiality	Authenticity	Deniability	Assumption		Size (in bytes)	
					pre-Q	post-Q	*pk*	*c*
DH-AKEM [3, Lst. 10]	X25519	**Ins-CCA**	**Out-Aut**	**DR-Den**†	✓	✗	32	32
ETSTH-AKEM [4, Lst. 18]	BAT + ANTRAG	**Ins-CCA**	**Out-Aut**	✗	✗	✓	1 417	1 119
	ML–KEM + FALCON						1 697	1 434
NIKE-AKEM [4, Lst. 19]	SWOOSH [36]	**Ins-CCA**	**Out-Aut**	**DR-Den**†	✗	✓	> 221 184	> 221 184
EANTH-AKEM [46]	BAT + SWOOSH [36]	**Ins-CCA**	**Out-Aut**	**DR-Den**†	✗	✓	> 221 705	473
FrodoKEX+ [24, Fig. 12]	N/A	**IND-1BatchCCA**	**UNF-1KCA**	**DR-Den**	✗	✓	21 300	72
SPARROW-KEM [61, Fig. 7]	N/A	**IND-1BatchCCA**	**UNF-1KCA**	**DR-Den**	✗	✓	2 592	40
PQ-AKEM [39, Fig. 10]	NTRU-A + GANDALF [ANTRAG, MITAKA]	**Ins-CCA**	**Out-Aut**	**HR-Den** & **DR-Den**	✗	✓	1 664	2 004⋆/2 044
	BAT + GANDALF [ANTRAG, MITAKA]						1 417	1 709⋆/1 749
	BAT + DUALRING						3 361	4 953
SHADOWFAX	X25519 + BAT + GANDALF [ANTRAG, MITAKA]	**Ins-CCA**	**Out-Aut**	**HR-Den** & **DR-Den**	✓	✓	1 449	1 741⋆/1 781
	X25519 + ML–KEM + GANDALF [FALCON,FALCON]						1 728	2 076

Table 4: Cycle counts (in thousands) of different authenticated key encapsulation mechanisms AKEM and ring signature schemes RSig run on a Firestorm core of an Apple M1 Pro running at 3GHz.

RSig	Unit	Gen	Sgn	Ver
Raptor [54, 80]	kcc	71 420	7 980	505
	ms	23.81	2.66	0.17
GANDALF [FALCON, FALCON]* [47]	kcc	-	-	-
	ms	5.20	0.49	0.02
GANDALF [FALCON, FALCON] (this work)	kcc	12 283	911	84
	ms	4.04	0.30	0.03
GANDALF [ANTRAG, MITAKA]	kcc	13 441	1 137	85
	ms	4.45	0.38	0.03
AKEM	**Unit**	**Gen**	**Enc**	**Dec**
DH-AKEM [X25519]	kcc	227	679	457
	ms	0.08	0.23	0.15
PQ-AKEM [BAT, GANDALF [ANTRAG, MITAKA]]	kcc	25 496	1 310	346
	ms	8.50	0.43	0.12
PQ-AKEM [ML–KEM, GANDALF [FALCON, FALCON]]	kcc	13 483	1 337	233
	ms	4.09	0.59	0.08
SHADOWFAX [X25519, BAT, GANDALF [ANTRAG, MITAKA]]	kcc	25 876	2 12	795
	ms	8.57	0.66	0.26
SHADOWFAX [X25519, ML–KEM, GANDALF [FALCON, FALCON]]	kcc	12 569	1 792	674
	ms	4.18	0.59	0.22

* Our benchmark. In [47], the authors accessed the timing counters through the standard C library on our platform. We benchmark their implementation on our platform. For other implementations, we access the cycle counters through the macOS API with a fallback to assembly for cycle counters on other operating systems. As mentioned before, the implementation of [47] leads to slightly larger signatures (1 288 bytes).

Ethical Considerations

Our research project focuses on the theoretical aspects and the practical feasibility of cryptographic combiners, specifically in the context of deniable authenticated key encapsulation mechanisms (AKEMs). The future stakeholders of the proposed protocols include individuals, for-profit and non-profit organizations, and other entities that protect their privacy. The proposed cryptographic primitives will likely be deployed in standard large-scale services, such as messaging services. Protecting people's privacy is an essential and ethical task. It should also be noted that this can be used by criminals, for instance, by allowing them to communicate securely and later denying having participated in an incriminating conversation. Despite the potential misuse of these new cryptographic constructions by criminals, the results should be published because the resulting privacy benefits to society and its individuals are substantially greater. This analysis was performed post-facto. With an a priori analysis, the authors would have chosen the same methodology.

Open Science

For open science, we include our portable implementations in our artifact. The artifact can be accessed through the github repository Shadowfax and the permant archive Zenodo[8]. Our artifact requires the following software

- `gcc` for compiling the source code.
- `make` for compiling with the `Makefiles`.
- `bash` for running the benchmark.

The software versions can be checked with the following commands:

- `gcc ---version`.
- `make ---version`.
- `bash ---version`.

The cycle counts were benchmarked in the following environment:

- Apple M1 Pro, Sonoma 14.6.1.
- `gcc (Homebrew GCC 13.3.0) 13.3.0`.
- `GNU Make 3.81`.
- `GNU bash, version 3.2.57(1)-release (arm64-apple-darwin23)`.

Additionally, our artifact is also tested in the following environment:

- Apple M1 Pro, Sonoma 14.6.1, `Apple clang version 15.0.0 (clang-1500.3.9.4)`, same `make` and `bash` as above.
- Dell Inc. XPS 9320, Ubuntu 22.04.5 LTS:
  - `gcc (Ubuntu 11.4.0-1ubuntu1 22.04) 11.4.0`.
  - `GNU Make 4.3`.
  - `GNU bash, version 5.1.16(1)-release (x86_64-pc-linux-gnu)`.
- Dell Inc. XPS 9320, Ubuntu 22.04.5 LTS, `Ubuntu clang version 14.0.0-1ubuntu1.1`, same `make` and `bash` as above.
- MacBook Pro 2020 (Intel(R) Core(TM) i7-1068NG7 CPU @ 2.30GHz):
  - `Apple clang version 12.0.0 (clang-1200.0.32.28)`.
  - `GNU Make 3.81`.
  - `GNU bash, version 5.2.37(1)-release (x86_64-apple-darwin22.6.0)`.

Finally, we would like to clarify that the implementations of this artifact are to demonstrate the feasibility of compact ring signatures with reasonable performance. The artifact does NOT aim to provide heavily-optimised implementations on personal computing devices as this requires a huge set of background knowledge on various assembly implementations (Armv8-A Neon, AVX2, AVX-512), resource usages (memory optimisation alongside register pressure), and execution port contention (instruction scheduling) on the target platforms.

[8] https://doi.org/10.5281/zenodo.17975204

References

[1] José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Vincent Laporte, Jean-Christophe Léchenet, Tiago Oliveira, Hugo Pacheco, Miguel Quaresma, Peter Schwabe, Antoine Séré, and Pierre-Yves Strub. Formally verifying Kyber episode IV: Implementation correctness. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3):164–193, 2023.

[2] José Bacelar Almeida, Santiago Arranz Olmos, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Jean-Christophe Léchenet, Cameron Low, Tiago Oliveira, Hugo Pacheco, Miguel Quaresma, Peter Schwabe, and Pierre-Yves Strub. Formally verifying Kyber - episode V: Machine-checked IND-CCA security and correctness of ML-KEM in EasyCrypt. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part II*, volume 14921 of *Lecture Notes in Computer Science*, pages 384–421, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[3] Joël Alwen, Bruno Blanchet, Eduard Hauck, Eike Kiltz, Benjamin Lipp, and Doreen Riepel. Analysing the HPKE standard. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 87–116, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland.

[4] Joël Alwen, Jonas Janneck, Eike Kiltz, and Benjamin Lipp. The pre-shared key modes of HPKE. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VI*, volume 14443 of *Lecture Notes in Computer Science*, pages 329–360, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.

[5] ANSSI. Anssi views on the post-quantum cryptography transition (2023 follow up), 2023.

[6] Apple. iMessage with PQ3: The new state of the art in quantum-secure messaging at scale, February 2024.

[7] Matilda Backendal, Mihir Bellare, Felix Günther, and Matteo Scarlata. When messages are keys: Is HMAC a dual-PRF? In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 661–693, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[8] Richard Barnes, Benjamin Beurdouche, Raphael Robert, Jon Millican, Emad Omara, and Katriel Cohn-Gordon. The Messaging Layer Security (MLS) Protocol. RFC 9420, July 2023.

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

[10] Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 602–619, Santa Barbara, CA, USA, August 20–24, 2006. Springer Berlin Heidelberg, Germany.

[11] Mihir Bellare. New proofs for NMAC and HMAC: Security without collision resistance. *Journal of Cryptology*, 28(4):844–878, October 2015.

[12] 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*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer Berlin Heidelberg, Germany.

[13] Mihir Bellare and Igors Stepanovs. Security under message-derived keys: Signcryption in iMessage. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 507–537, Zagreb, Croatia, May 10–14, 2020. Springer, Cham, Switzerland.

[14] Daniel J. Bernstein. Curve25519: New Diffie-Hellman speed records. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228, New York, NY, USA, April 24–26, 2006. Springer Berlin Heidelberg, Germany.

[15] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, and Nicola Tuveri. OpenSSLNTRU: Faster post-quantum TLS key exchange. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022: 31st USENIX Security Symposium*, pages 845–862, Boston, MA, USA, August 10–12, 2022. USENIX Association.

[16] Ward Beullens. Breaking rainbow takes a weekend on a laptop. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 464–479, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.

[17] Giacomo Borin, Yi-Fu Lai, and Antonin Leroux. Erebor and durian: Full anonymous ring signatures from quaternions and isogenies. Cryptology ePrint Archive, Report 2024/1185, 2024.

[18] Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy*, pages 553–570, San Jose, CA, USA, May 17–21, 2015. IEEE Computer Society Press.

[19] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367, London, United Kingdom, April 24–26, 2018. IEEE Computer Society Press.

[20] Jacqueline Brendel, Marc Fischlin, Felix Günther, Christian Janson, and Douglas Stebila. Towards post-quantum security for Signal's X3DH handshake. In Orr Dunkelman, Michael J. Jacobson, Jr., and Colin O'Flynn, editors, *SAC 2020: 27th Annual International Workshop on Selected Areas in Cryptography*, volume 12804 of *Lecture Notes in Computer Science*, pages 404–430, Halifax, NS, Canada (Virtual Event), October 21-23, 2020. Springer, Cham, Switzerland.

[21] BSI. Quantum-safe cryptography – fundamentals, current developments and recommendations, 2022.

[22] BSI. Cryptographic mechanisms: Recommendations and key lengths - bsi tr-02102-1, 2024.

[23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[24] Daniel Collins, Loïs Huguenin-Dumittan, Ngoc Khanh Nguyen, Nicolas Rolin, and Serge Vaudenay. K-waay: Fast and deniable post-quantum X3DH without ring signatures. In Davide Balzarotti and Wenyuan Xu, editors, *USENIX Security 2024: 33rd USENIX Security Symposium*, Philadelphia, PA, USA, August 14–16, 2024. USENIX Association.

[25] Alexander W. Dent and Yuliang Zheng, editors. *Practical Signcryption*. Springer Berlin Heidelberg, 2010.

[26] Tim Dierks and Christopher Allen. *RFC 2246 - The TLS Protocol Version 1.0*. Internet Activities Board, January 1999.

[27] D.J.Bernstein and T.Lange. ebacs:ecrypt benchmarking of cryptographic systems, 2024. accessed 16 July 2024.

[28] Léo Ducas and Thomas Prest. Fast Fourier Orthogonalization. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 191–198, 2016.

[29] Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Mitaka: A simpler, parallelizable, maskable variant of falcon. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 222–253, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland.

[30] Thomas Espitau, Thi Thu Quyen Nguyen, Chao Sun, Mehdi Tibouchi, and Alexandre Wallet. Antrag: Annular NTRU trapdoor generation - making mitaka as secure as falcon. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VII*, volume 14444 of *Lecture Notes in Computer Science*, pages 3–36, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.

[31] Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Shorter Hash-and-Sign Lattice-Based Signatures. In *Annual International Cryptology Conference*, pages 245–275. Springer, 2022.

[32] Sebastian Faller and Julia Hesse. How to (not) combine oblivious pseudorandom functions. Cryptology ePrint Archive, Paper 2025/1084, 2025.

[33] Rune Fiedler and Christian Janson. A deniability analysis of Signal's initial handshake PQXDH. *Proceedings on Privacy Enhancing Technologies*, 2024(4):907–928, October 2024.

[34] Pierre-Alain Fouque, Paul Kirchner, Thomas Pornin, and Yang Yu. BAT: Small and fast KEM over NTRU lattices. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(2):240–265, 2022.

[35] Eduarda S. V. Freire, Dennis Hofheinz, Eike Kiltz, and Kenneth G. Paterson. Non-interactive key exchange. Cryptology ePrint Archive, Paper 2012/732/20130101:143205, 2012.

[36] Phillip Gajland, Bor de Kock, Miguel Quaresma, Giulio Malavolta, and Peter Schwabe. SWOOSH: Efficient lattice-based non-interactive key exchange. In Davide

Balzarotti and Wenyuan Xu, editors, *USENIX Security 2024: 33rd USENIX Security Symposium*, Philadelphia, PA, USA, August 14–16, 2024. USENIX Association.

[37] Phillip Gajland, Vincent Hwang, and Jonas Janneck. Shadowfax: Hybrid security and deniability for AKEMs. Cryptology ePrint Archive, Paper 2025/154, 2025.

[38] Phillip Gajland, Jonas Janneck, and Eike Kiltz. A closer look at falcon. Cryptology ePrint Archive, Report 2024/1769, 2024.

[39] Phillip Gajland, Jonas Janneck, and Eike Kiltz. Ring signatures for deniable AKEM: Gandalf's fellowship. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part I*, volume 14920 of *Lecture Notes in Computer Science*, pages 305–338, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[40] Federico Giacon, Felix Heuer, and Bertram Poettering. KEM combiners. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018: 21st International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 10769 of *Lecture Notes in Computer Science*, pages 190–218, Rio de Janeiro, Brazil, March 25–29, 2018. Springer, Cham, Switzerland.

[41] Sharon Goldberg, Miro Haller, Nadia Heninger, Mike Milano, Dan Shumow, Marc Stevens, and Adam Suhl. RADIUS/UDP considered harmful. In Davide Balzarotti and Wenyuan Xu, editors, *USENIX Security 2024: 33rd USENIX Security Symposium*, Philadelphia, PA, USA, August 14–16, 2024. USENIX Association.

[42] Felix Günther, Michael Rosenberg, Douglas Stebila, and Shannon Veitch. Hybrid obfuscated key exchange and KEMs. Cryptology ePrint Archive, Report 2025/408, 2025.

[43] Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski, and Thomas Prest. An efficient and generic construction for Signal's handshake (X3DH): Post-quantum, state leakage secure, and deniable. *Journal of Cryptology*, 35(3):17, July 2022.

[44] Julia Hesse and Michael Rosenberg. PAKE combiners and efficient post-quantum instantiations. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part II*, volume 15602 of *Lecture Notes in Computer Science*, pages 395–420, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland.

[45] Jonas Janneck. Bird of prey: Practical signature combiners preserving strong unforgeability. Cryptology ePrint Archive, Paper 2025/1844, 2025.

[46] Jonas Janneck, Jonas Meers, Massimo Ostuzzi, and Doreen Riepel. Snake mackerel: An isogeny-based AKEM leveraging randomness reuse. Cryptology ePrint Archive, Paper 2025/1474, 2025.

[47] Shuichi Katsumata, Guilhem Niot, Ida Tucker, and Thom Wiggers. Comprehensive deniability analysis of signal handshake protocols: X3DH, PQXDH to fully post-quantum with deniable ring signatures. Cryptology ePrint Archive, Paper 2025/1090, 2025. To appear in USENIX Security 2025: 34th USENIX Security Symposium.

[48] John Kelsey, Shu jen Change, and Ray Perlner. SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash. Technical report, National Institute of Standards and Technology, December 2016.

[49] Ehren Kret and Rolfe Schmidt. The pqxdh key agreement protocol, 2024.

[50] Kris Kwiatkowski and Luke Valenta. The TLS post-quantum experiment. Post on the Cloudflare blog, 2019. https://blog.cloudflare.com/the-tls-post-quantum-experiment/.

[51] Adam Langley. CECPQ1 results. Blog post, 2016. https://www.imperialviolet.org/2016/11/28/cecpq1.html.

[52] Adam Langley. CECPQ2. Blog post, 2018. https://www.imperialviolet.org/2018/12/12/cecpq2.html.

[53] Felix Linker, Ralf Sasse, and David Basin. A formal analysis of apple's iMessage PQ3 protocol. Cryptology ePrint Archive, Paper 2024/1395, 2024.

[54] Xingye Lu, Man Ho Au, and Zhenfei Zhang. Raptor: A practical lattice-based (linkable) ring signature. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 2019: 17th International Conference on Applied Cryptography and Network Security*, volume 11464 of *Lecture Notes in Computer Science*, pages 110–130, Bogota, Colombia, June 5–7, 2019. Springer, Cham, Switzerland.

[55] You Lyu and Shengli Liu. Hybrid password authentication key exchange in the UC framework. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part II*, volume 15602 of *Lecture Notes in Computer Science*, pages 421–450, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland.

[56] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. SMILE: Set membership from ideal lattices with applications to ring signatures and confidential

transactions. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 611–640, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

[57] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[58] Moxie Marlinspike and Trevor Perrin. The x3dh key agreement protocol, 2016.

[59] Module-lattice-based digital signature standard. National Institute of Standards and Technology NIST FIPS PUB 204, U.S. Department of Commerce, August 2024.

[60] Module-lattice-based key-encapsulation mechanism standard. National Institute of Standards and Technology NIST FIPS PUB 203, U.S. Department of Commerce, August 2024.

[61] Guilhem Niot. Practical deniable post-quantum X3DH: A lightweight split-KEM for k-waay. In *ASIACCS 2025*. ACM Press, 2025.

[62] NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[63] David Ott and Christopher Peikert. Identifying research challenges in post quantum cryptography migration and cryptographic agility, 2019.

[64] Christian Paquin, Douglas Stebila, and Goutam Tamvada. Benchmarking post-quantum cryptography in TLS. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 72–91, Paris, France, April 15–17, 2020. Springer, Cham, Switzerland.

[65] Thomas Pornin. Improved key pair generation for falcon, BAT and hawk. Cryptology ePrint Archive, Report 2023/290, 2023.

[66] Thomas Pornin. Falcon on ARM cortex-m4: an update. Cryptology ePrint Archive, Paper 2025/123, 2025.

[67] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon. Submission to the NIST Post-Quantum Cryptography Standardization Project, 2020.

[68] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018.

[69] Ronald L. Rivest. *RFC 1321: The MD5 Message-Digest Algorithm*. Internet Activities Board, April 1992.

[70] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, Gold Coast, Australia, December 9–13, 2001. Springer Berlin Heidelberg, Germany.

[71] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[72] Markku-Juhani O. Saarinen and Jean-Philippe Aumasson. The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC). RFC 7693, November 2015.

[73] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[74] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

[75] Stateless hash-based digital signature standard. National Institute of Standards and Technology NIST FIPS PUB 205, U.S. Department of Commerce, August 2024.

[76] Douglas Stebila. Security analysis of the iMessage PQ3 protocol. Cryptology ePrint Archive, Report 2024/357, 2024.

[77] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, Report 2004/199, 2004.

[78] Bas Westerbaan and Cefan Daniel Rubin. Defending against future threats: Cloudflare goes post-quantum. Post on the Cloudflare blog, 2019. https://blog.cloudflare.com/post-quantum-for-all/.

[79] Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. DualRing: Generic construction of ring signatures with efficient instantiations. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 251–281, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

[80] Zhenfei Zhang. Raptor. https://github.com/zhenfeizhang/raptor, 2020.