HIC is all you need: Practical Post-Quantum Password-Authenticated Public Key Encryption

Afonso Arriaga[1 [0000−0002−1967−3390]], David Mestel[2 [0000−0003−3186−8307]], Jan Oupický[1 [0009−0005−8204−7551] (✉)], Peter Browne Rønne[1 [0000−0002−2785−8301]], and Marjan Škrobot[1 [0000−0002−7132−7591]]

[1] University of Luxembourg
{afonso.delerue,jan.oupicky,peter.roenne}@uni.lu
marjanskrobot@gmail.com
[2] Maastricht University
david.mestel@maastrichtuniversity.nl

Abstract. Password-Authenticated Public Key Encryption (PAPKE) enables secure encryption using only a shared, human-memorable password—eliminating the need for trusted intermediaries or pre-established infrastructure. It allows a sender to encrypt a message for a recipient, using the recipient's password-authenticated public key and a shared password, while provably resisting man-in-the-middle and offline dictionary attacks. PAPKE's support for reusable password-authenticated public keys makes it especially suitable for asynchronous, PKI-free communication scenarios.

An important open problem is to construct PAPKE schemes that are secure against quantum adversaries, as existing instantiations rely on Diffie-Hellman assumptions. The PAPKE-IC construction (ACNS 2019) is generic and admits integration with post-quantum PKE schemes. However, the scheme assumes an Ideal Cipher (IC) over the public key domain, which is large for most post-quantum PKE schemes. While an IC is typically instantiated using a block cipher, standard block ciphers operate over much smaller domains (e.g., 128 or 256 bits). Alternatively, one can use an 8-round Feistel network, which achieves indifferentiability from an ideal cipher, or domain extenders. The latter are inefficient at the domain sizes required, making the efficient and secure instantiation of the IC in PAPKE-IC, in combination with post-quantum PKE, particularly challenging.

In this paper, we propose PAPKE-HIC, a UC-secure PAPKE scheme built from a PKE scheme and a Half-Ideal Cipher (HIC, introduced at EUROCRYPT 2023), which circumvents the challenges of instantiating ideal ciphers over large domains. We provide a detailed security proof of PAPKE-HIC and establish precise requirements for the underlying PKE: strong robustness, one-wayness, ciphertext anonymity, and pseudo-uniformity of public keys. Our analysis identifies a gap in the original PAPKE-IC security proof, motivating the introduction of a novel property, which we denote Decryption Robustness (DROB-CCA). Although DROB-CCA is implied by strong robustness (SROB-CCA), the reduction is not tight and incurs a quadratic security loss. We analyze

which PKE schemes directly satisfy DROB-CCA, and conclude by presenting concrete instantiations of PAPKE-HIC. To our knowledge, this is the first practical, post-quantum instantiation of the PAPKE primitive.

Keywords: Password-Authenticated Public Key Encryption (PAPKE) · Post-Quantum PAPKE · PAKE · Decryption Robustness (DROB-CCA).

1 Introduction

Password-based authentication remains an important and widely deployed mechanism in modern cryptographic systems. While traditional Authenticated Key Exchange (AKE) protocols rely on long-term private keys managed through Public Key Infrastructures (PKI), such frameworks assume stable device identity and secure key storage—assumptions that do not hold universally. Advances such as passkeys, secure enclaves, and biometric authentication have improved key management on end-user devices, but these solutions depend on proprietary hardware and platform support, and remain inaccessible in many contexts. In contrast, numerous real-world scenarios—such as onboarding in IoT protocols like Thread [33], ePassport inspection [25], peer-to-peer file transfers [35], and ephemeral wireless network sessions [36]—lack the infrastructure or trust anchors required for PKI-based schemes. In these settings, password-based authentication remains a practical and cryptographically meaningful tool as it enables secure communication using pre-shared low-entropy secrets without requiring persistent credentials, trusted intermediaries, or pre-established infrastructure.

PAKE. Password-Authenticated Key Exchange (PAKE) protocols address the core limitations of traditional password-based authentication, e.g., sending a password over a server-authenticated TLS channel. A PAKE enables two parties, who only share a low-entropy password, to establish a high-entropy session key, while provably resisting offline dictionary attacks. This is a core security guarantee: an adversary must interact with an honest party to test each password guess. Passive eavesdropping yields no advantage, even under full transcript access. This interaction-bound guessing constraint makes PAKE significantly more robust than simple password transmission schemes, and mutual authentication offers the additional benefit of protecting against phishing attacks.

PAPKE. Password-Authenticated Public Key Encryption (PAPKE) [12] is a cryptographic primitive that combines password authentication and CCA-secure encryption into a single mechanism, unifying two core security objectives: secrecy and authenticity. It enables a sender to encrypt a message for a recipient, using the recipient's password-authenticated public key and a shared password. The resulting ciphertext enforces password-based access control over the message, provides chosen-ciphertext security, and resists man-in-the-middle attacks, assuming the password remains unknown to the adversary.

Unlike PAKE, which derives a shared session key through mutual interaction, PAPKE supports *unilateral* encryption, enabling secure message delivery

without requiring real-time interaction or composition with a symmetric key scheme. PAPKE also provides a theoretically stronger abstraction than PAKE: a UC-secure PAKE protocol can be derived from a UC-secure PAPKE instance by encrypting a session key rather than an arbitrary message. The inverse, however, does not hold due to a defining feature of PAPKE: the use of a reusable authenticated public key.[3] Such reusability enables any sender, who holds an authenticated public key from the receiver and the corresponding password, to generate multiple valid ciphertexts without fresh interaction or per-session key derivation. The reusability of public keys makes PAPKE especially well-suited for scenarios requiring asynchronous communication, credential recovery, and one-to-many password-authenticated key exchange protocols such as SweetPAKE [6] and Oblivious PAKE [27, 6], where scalability and minimal communication overhead are essential. These PAPKE-based protocols are natural candidates for extending WPA3 to support multi-password authentication [34]. Importantly, such an extension should ensure post-quantum security with minimal communication and interaction overhead.

Two existing approaches to PAPKE and their limitations. PAPKE has been realized using two main approaches [12]. The first is a generic construction, PAPKE-IC, which relies on the Ideal Cipher (IC) model. The second, PAPKE-FO, is a direct construction based on the Decisional Diffie-Hellman (DDH) assumption and applies the Fujisaki-Okamoto (FO) transform, thus relying on the Random Oracle (RO) model instead.

When comparing the two approaches, PAPKE-IC is conceptually simpler and reflects the design of the classic Encrypted Key Exchange (EKE) protocol. However, it presents IC instantiability challenges. A poor IC instantiation may enable offline dictionary attacks. For example, if the IC is instantiated using a block cipher whose block size does not match the public key space, an adversary can decrypt the password-authenticated ciphertext under candidate passwords and discard those that do not yield a plaintext within the valid public key space.

To instantiate the ideal cipher over larger domains in PAPKE-IC, one may employ an 8-round Feistel network [17] or a domain extender from [15], both shown to achieve indifferentiability from an ideal cipher. These constructions are costly to implement over large algebraic domains (and in constant time). Consequently, practical and efficient instantiations of PAPKE-IC remain elusive in the post-quantum setting, underscoring the need for alternative designs.

1.1 Our Contributions

Main contribution: PAPKE-HIC with precise security requirements. We introduce a new UC-secure Password-Authenticated Public Key Encryption scheme, PAPKE-HIC, using the Half-Ideal Cipher (HIC) from [32]. Concretely, we show how to generically construct a PAPKE scheme from a public key encryption (PKE) scheme and a Half-Ideal Cipher (HIC) that can be instantiated using a

[3] A recent notion, Bare PAKE [7], captures a form of reusability within PAKEs.

modified 2-round Feistel construction m2F. The main advantage of m2F is that it employs an Ideal Cipher defined over a fixed-length bitstring domain, rather than over large algebraic domains required in prior IC-based constructions such as PAPKE-IC, making it significantly easier to instantiate efficiently. As a result, our design resolves a key open question: how to efficiently realize PAPKE in the post-quantum setting without sacrificing concrete security or performance.

We provide a detailed and rigorous proof that PAPKE-HIC realizes the UC-PAPKE functionality in the Universal Composability (UC) framework (Theorem 1). In doing so, we formally establish precise security requirements for the underlying public key encryption scheme. As in the PAPKE-IC proof [12], we identify three necessary properties: *anonymity and indistinguishability security* (AI-CCA), *strong robustness* (SROB-CCA), and *pseudo-uniformity of public keys* (UNI-PK). We further generalize the proof to accommodate public key encryption schemes with imperfect correctness and pseudo-uniform public key distributions (UNI-PK)—features that are relevant for several post-quantum public key encryption schemes.

Second contribution: Security gap in the original PAPKE-IC proof and new public key encryption property—Decryption Robustness. Our analysis reveals a subtle gap in the original analysis of PAPKE-IC [12]: the proof must ensure that an adversary, without access to the public key, cannot forge a ciphertext that decrypts successfully. We formalize this requirement as a new property called *decryption robustness* (DROB-CCA). While DROB-CCA is implied by strong robustness (SROB-CCA), as shown in Theorem 2, the reduction is not tight and incurs a quadratic security loss. Critically, when applying Theorem 2 to derive the concrete bound of PAPKE-HIC in Theorem 1, the DROB-CCA-related advantage becomes the dominant term, significantly weakening the overall security guarantee. In addition, we identify a further minor omission in the original analysis: the "badkeys abort" mechanism in the $\mathcal{F}_{\mathsf{PAPKE}}$ functionality was not accounted for in the proof.

Third contribution: Concrete instantiations of PAPKE-HIC with post-quantum PKE. Motivated by the limitations of the reduction in Theorem 2, we investigate which public key encryption schemes satisfy DROB-CCA directly with tighter bounds. We show that any PKE built using the standard KEM+DEM paradigm, where the KEM employs a variant of the *prefix-hashing* Fujisaki-Okamoto (FO) transform [19] and the DEM satisfies ciphertext integrity (INT-CTXT), achieves DROB-CCA without relying on the generic reduction (Theorem 5). This applies to post-quantum PKEs built from ML-KEM [30], or FrodoKEM [29].

We further show how to build DROB-CCA-secure PKE from KEMs that do not employ the prefix-hashing FO transform, such as Classic McEliece [2]. We also prove that the DHIES* [1], the hybrid PKE that was originally used to instantiate PAPKE-IC [12], also satisfies DROB-CCA with a tighter bound.

Finally, we implement and benchmark post-quantum PAPKE-IC and PAPKE-HIC using ML-KEM [30], FrodoKEM [29] and Classic McEliece [2] in various configurations.

2 Preliminaries

In this section, we present the cryptographic primitives relevant to our work. We begin with standard definitions of Public Key Encryption (PKE), Key Encapsulation Mechanism (KEM), and Data Encapsulation Mechanism (DEM). We then introduce Password-Authenticated Public Key Encryption (PAPKE) [12], followed by the Half-Ideal Cipher (HIC) [32], which forms the foundation of our construction.

All relevant PKE and DEM security definitions are provided in Figure 1. Let SCH be a scheme and EXP be a security experiment. For experiments where the adversary's ($\mathcal{A}$) goal is to guess the challenger's bit, we define its advantage as $\mathsf{Adv}^{\mathsf{EXP}}_{\mathcal{A},\mathsf{SCH}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{EXP}}_{\mathcal{A},\mathsf{SCH}}(\lambda) = 1\right] - \frac{1}{2}\right|$, and $\mathsf{Adv}^{\mathsf{EXP}}_{\mathcal{A},\mathsf{SCH}}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{EXP}}_{\mathcal{A},\mathsf{SCH}}(\lambda) = 1\right]$ otherwise. We say that SCH is EXP-secure if $\mathsf{Adv}^{\mathsf{EXP}}_{\mathcal{A},\mathsf{SCH}}(\lambda)$ is negligible for any PPT adversary $\mathcal{A}$.

2.1 Public Key Encryption (PKE)

We present the definition of a Public Key Encryption scheme (PKE) and relevant security notions.

Definition 1. *A Public Key Encryption scheme (PKE) is a triple of PPT algorithms* $(\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$: $\mathsf{Keygen} : \mathbb{N} \to \mathcal{PK} \times \mathcal{SK}$, $\mathsf{Enc} : \mathcal{PK} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$, $\mathsf{Dec} : \mathcal{SK} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$, *where $\mathcal{PK}$ is the public key space, $\mathcal{SK}$ is the private key space, $\mathcal{M}$ is the message space, $\mathcal{R}$ is the randomness space and $\mathcal{C}$ is the ciphertext space.*

Correctness. [23] We say a public key encryption scheme is δ-correct if

$$\mathbb{E}_{(pk,sk)\leftarrow\$\mathsf{Keygen}(\lambda)}\left[\max_{m\in\mathcal{M}} \Pr_{r\leftarrow\$\mathcal{R}}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m, r)) \neq m]\right] \leq \delta(\lambda).$$

PKE security properties. The standard security requirement for public key encryption is *indistinguishability under chosen-ciphertext attacks* ($\mathsf{IND\text{-}CCA}$). Another crucial property is *public key anonymity under chosen-ciphertext attacks* ($\mathsf{ANO\text{-}CCA}$), which ensures that ciphertexts reveal no information about the specific public key used during encryption. These properties can be combined into the *anonymity and indistinguishability under chosen-ciphertext attacks* ($\mathsf{AI\text{-}CCA}$), as done in [12].[4] A more recent property that has proven essential for the construction of PAKE from KEM is the *pseudo-uniformity of public keys* ($\mathsf{UNI\text{-}PK}$), which captures the notion that public keys produced by the key generation algorithm are (computationally) indistinguishable from uniformly random elements of the public key space $\mathcal{PK}$ (called *fuzziness* in [8]; see also [4] for reference). Another important property of a PKE is *strong robustness under chosen-ciphertext attacks* ($\mathsf{SROB\text{-}CCA}$), which ensures that it is computationally hard to create a

[4] Note that the notions are equivalent: $\mathsf{AI\text{-}CCA} \iff \mathsf{IND\text{-}CCA} + \mathsf{ANO\text{-}CCA}$.

ciphertext that decrypts successfully under two distinct secret keys, both honestly generated via Keygen. All of these properties are defined in Figure 1.

In this work, we introduce two novel properties: *decryption robustness* (DROB) and its chosen-ciphertext analogue, *decryption robustness under chosen-ciphertext attacks* (DROB-CCA). These properties formalize the requirement that it should be hard to craft a ciphertext that decrypts correctly *without knowledge of the public key*. While unconventional in standard PKE settings—where public keys are, by definition, public—this property becomes essential in PAPKE, where the public key is treated as secret. We provide exact definitions in Figure 1 and a detailed discussion in Section 4, including its relationship to SROB-CCA.

2.2 Key Encapsulation Mechanism (KEM)

We define a Key Encapsulation Mechanism (KEM). See the full version of the paper [5] for definitions of KEM security properties.

Definition 2. *A Key Encapsulation Mechanism (KEM) is a triple of PPT algorithms* (Keygen, Encap, Decap) *defined as:* $\mathsf{Keygen} : \mathbb{N} \to \mathcal{PK} \times \mathcal{SK}$, $\mathsf{Encap} : \mathcal{PK} \to \mathcal{K} \times \mathcal{C}$, $\mathsf{Decap} : \mathcal{SK} \times \mathcal{C} \to \mathcal{K} \cup \{\perp\}$, *where $\mathcal{PK}$ is the public key space, $\mathcal{SK}$ is the private key space, $\mathcal{K}$ is the symmetric key space, and $\mathcal{C}$ is the ciphertext space.*

We say a KEM is implicitly rejecting *if* Decap *never outputs the symbol $\perp$; otherwise, it is called* explicitly rejecting.

2.3 Data Encapsulation Mechanism (DEM)

We present the definition of a Data Encapsulation Mechanism (DEM) and relevant security notions.

Definition 3. *A DEM is a triple of PPT algorithms* (Keygen, Enc, Dec) *defined as:* $\mathsf{Keygen} : \mathbb{N} \to \mathcal{K}$, $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$, $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$, *where $\mathcal{K}$ is the symmetric key space, $\mathcal{M}$ is the message space, and $\mathcal{C}$ is the ciphertext space.*

A DEM is typically an authenticated symmetric-key encryption scheme used in hybrid encryption. We assume that the symmetric key space $\mathcal{K}$, resp. the message space $\mathcal{M}$, is the same in the DEM and the KEM, resp. PKE, definitions.

DEM security properties. In our proofs, we only directly work with the *integrity of ciphertexts* (INT-CTXT) property of DEM from [10] (Figure 1). This property captures the difficulty of generating a valid ciphertext (that decrypts to a valid message, i.e., $m \neq \perp$) without knowledge of the corresponding symmetric key.

2.4 Password-Authenticated Public Key Encryption (PAPKE)

A Password-Authenticated Public Key Encryption scheme (PAPKE) is a relatively recent primitive introduced in [12]. PAPKE extends public key encryption with password-based authentication.

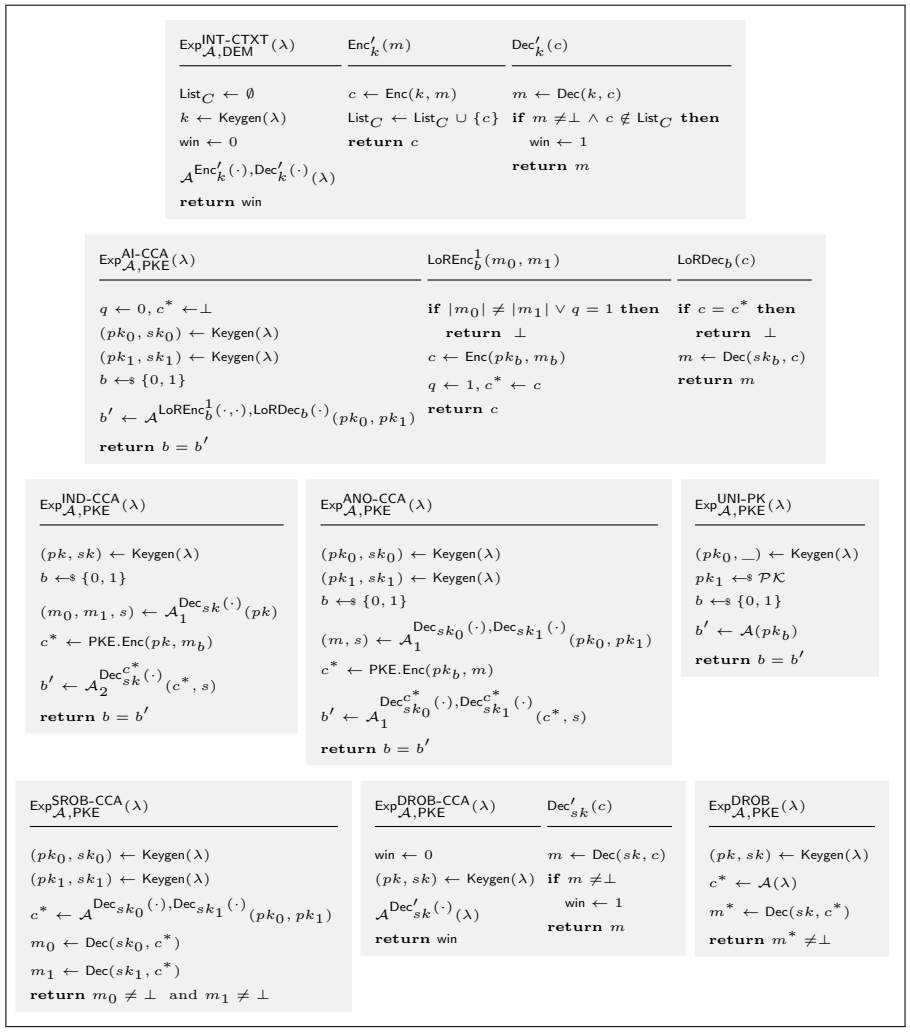


Fig. 1: Security experiments defining properties of PKEs and DEMs. (1) INT-CTXT is the standard *ciphertext integrity* from [10]; (2) AI-CCA is the *anonymity and indistinguishability* from [12, 1] which is equivalent to IND-CCA+ ANO-CCA; (3) IND-CCA is the standard *indistinguishability under chosen-ciphertext attacks*; (4) ANO-CCA for is the *anonymity under chosen-ciphertext attacks* from [9]; (5) UNI-PK is the *pseudo-uniformity of public keys* similar to the one from [4] and *fuzziness* from [8]; (6) SROB-CCA is the *strong-robustness* from [12]; (7) DROB-CCA is the *decryption robustness under chosen-ciphertext attacks*; (8) DROB is the *decryption robustness*. $\mathsf{Dec}^{c^*}_{sk}(\cdot)$ is a decryption oracle with a private key sk which returns $\perp$ if c^* is queried.

Definition 4. *A Password-Authenticated Public Key Encryption (PAPKE) is a triple of PPT algorithms* $(\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ *defined as:* $\mathsf{Keygen} : \mathbb{N} \times \mathcal{PW} \to \mathcal{SK} \times \mathcal{APK}$, $\mathsf{Enc} : \mathcal{APK} \times \mathcal{PW} \times \mathcal{M} \to \mathcal{C}$, $\mathsf{Dec} : \mathcal{SK} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$, *where* $\mathcal{PW}$ *is the password dictionary,* $\mathcal{SK}$ *is the private key space,* $\mathcal{APK}$ *is the authenticated public key space,* $\mathcal{M}$ *is the message space, and* $\mathcal{C}$ *is the ciphertext space.*

PAPKE security properties. The PAPKE paper [12] defines two game-based security definitions: *indistinguishability against chosen-ciphertext and chosen-key attacks* (IND-CCKA) and *ciphertext authenticity* (AUTH-CTXT). IND-CCKA captures resistance against offline dictionary attacks, secrecy of messages under chosen-ciphertext attacks, security against man-in-the-middle attacks, and long-term security. AUTH-CTXT ensures that an adversary cannot create a valid ciphertext without knowing the password.

We do not detail these game-based definitions, as we work within the Universal Composability (UC) framework [13], which implies both definitions. The ideal functionality of PAPKE ($\mathcal{F}_{\mathsf{PAPKE}}$) is defined in Figure 2.

Compared to the original functionality in [12], we slightly modify $\mathcal{F}_{\mathsf{PAPKE}}$ to ensure that the messages ENC-M and ENC-L also send apk' to $\mathcal{A}$ together with m and $|m|$, respectively. We believe this is an accidental omission in the original definition, since in the proofs of PAPKE-IC and PAPKE-FO the authors write "$\ldots$ $(\mathsf{ENC\text{-}M}, sid, apk, m)$ from $\mathcal{F}_{\mathsf{PAPKE}} \ldots$". In our proof of PAPKE-HIC (Theorem 1), we found that sending apk' in ENC-M is necessary because the simulator must encrypt the message "honestly" and otherwise cannot determine the correct apk' to use with certainty.

2.5 Half-Ideal Cipher (HIC)

The (randomized) Half-Ideal Cipher (HIC) is a Universal Composability (UC) security notion recently proposed in [32]. It serves as a relaxation of the Ideal Cipher (IC) notion with application to PAKEs. The HIC security model is formalized by an ideal functionality $\mathcal{F}_{\mathsf{HIC}}$ (Figure 4), which is parameterized by the domain $\mathcal{D} = \mathcal{R} \times \mathcal{G}$. Note that in our use case, $\mathcal{G} = \mathcal{PK}$, where $\mathcal{PK}$ is the public key space of a PKE. The HIC abstraction offers a practical alternative to ideal ciphers in the design of EKE-like protocols, eliminating the need for direct application of an IC over groups, which can be complex to instantiate [32].

Notably, HIC provides an *honest interface* accessible to honest parties, and an *adversarial interface* accessible to the adversary. The adversarial interface is more powerful because it allows the adversary to choose 1) the encryption randomness $r \in \mathcal{R}$ and half of the ciphertext $T \in \mathcal{G}$ during encryption and 2) the adversary also receives $r \in \mathcal{R}$ (compared to only $M \in \mathcal{G}$) during decryption.

It was shown in [32] that the *Modified 2-Feistel* (m2F) construction (Definition 5) realizes $\mathcal{F}_{\mathsf{HIC}}$ in the Random Oracle (RO) and Ideal Cipher (IC) model. Although m2F internally uses an ideal cipher, it is an ideal cipher over bitstrings of small size, which is easy to instantiate using a regular block cipher.

1. **Key Generation**. On input $(\mathsf{KEYGEN}, sid, pwd)$ from party $\mathcal{P}$:
   - If $sid \neq (\mathcal{P}, sid')$ or a record $(\mathsf{keyrec}, sid, \cdot, \cdot)$ exists, ignore.
   - Send (KEYGEN, sid) to $\mathcal{A}$ and wait for $(\mathsf{KEYCONF}, sid, apk, \mathcal{M})$ from $\mathcal{A}$.
   - If a record $(\mathsf{badkeys}, sid, apk', \cdot)$ with $apk' = apk$ exists, abort.
   - Create a record $(\mathsf{keyrec}, sid, apk, pwd)$ and output $(\mathsf{KEYCONF}, sid, apk)$ to $\mathcal{P}$.

2. **Encryption**. On input $(\mathsf{ENCRYPT}, sid, apk', pwd', m)$ from party $\mathcal{Q}$ where $m \in \mathcal{M}$:
   - If a record $(\mathsf{keyrec}, sid, apk, pwd)$ with $apk = apk'$ exists and $\mathcal{P}$ (from $(\mathcal{P}, sid') \leftarrow sid$) is honest, then:
     - Send $(\mathsf{ENC\text{-}L}, sid, apk', |m|)$ to $\mathcal{A}$ and wait for $(\mathsf{CIPHERTEXT}, sid, c)$ from $\mathcal{A}$.
     - Abort if a record $(\mathsf{enrec}, sid, \cdot, c)$ for c already exists.
     - Create a record $(\mathsf{enrec}, sid, m', c)$ with $m' \leftarrow m$ if $pwd' = pwd$, and $m \leftarrow \perp$, else.
   - Else:
     - If a record $(\mathsf{badkeys}, sid, apk_j, pwd_j)$ with $apk_j = apk'$ exists, set $pwd^* \leftarrow pwd_j$.
     - Else, send $(\mathsf{GUESS}, sid, apk')$ to $\mathcal{A}$, wait for $(\mathsf{GUESS}, sid, pwd^*)$ from $\mathcal{A}$ and create record $(\mathsf{badkeys}, sid, apk', pwd^*)$.
     - If $pwd' = pwd^*$, send $(\mathsf{ENC\text{-}M}, sid, apk', m)$ to $\mathcal{A}$, and wait for $(\mathsf{CIPHERTEXT}, sid, c)$ from $\mathcal{A}$.
     - If $pwd' \neq pwd^*$, send $(\mathsf{ENC\text{-}L}, sid, apk', |m|)$ to $\mathcal{A}$, and wait for $(\mathsf{CIPHERTEXT}, sid, c)$ from $\mathcal{A}$.
   - Output $(\mathsf{CIPHERTEXT}, sid, c)$ to $\mathcal{Q}$.
3. **Decryption**. On input $(\mathsf{DECRYPT}, sid, c)$ from party $\mathcal{P}$:
   - If $sid \neq (\mathcal{P}, sid')$ or no record $(\mathsf{keyrec}, sid, apk, pwd)$ exists, ignore.
   - If a record $(\mathsf{enrec}, sid, m, c)$ for c exists, where $m \in \mathcal{M} \cup \{\perp\}$:
     - Output $(\mathsf{PLAINTEXT}, sid, m)$ to $\mathcal{P}$.
   - Else:
     - Send $(\mathsf{DECRYPT}, sid, c)$ to $\mathcal{A}$ and wait for $(\mathsf{PLAINTEXT}, sid, m, pwd^*)$ from $\mathcal{A}$.
     - If $pwd^* = pwd$, set $m' \leftarrow m$ and $m' \leftarrow \perp$ otherwise.
     - Create a record $(\mathsf{enrec}, sid, m', c)$.
     - Output $(\mathsf{PLAINTEXT}, sid, m')$ to $\mathcal{P}$.

Fig. 2: Patched ideal functionality $\mathcal{F}_{\mathsf{PAPKE}}$ from [12]—the functionality additionally sends apk' for messages ENC-M and ENC-L. The functionality is parametrized by the message space $\mathcal{M}$.

Definition 5. *Let* $\mathsf{H} : \mathcal{PW} \times \mathcal{R} \to \mathbb{G}$ *and* $\mathsf{G} : \mathcal{PW} \times \mathbb{G} \to \mathcal{K}$ *be random oracles and let* $\mathsf{IC} = (\mathsf{IC.Enc}, \mathsf{IC.Dec})$ *be an ideal cipher over* $\mathcal{R}$*. A Modified 2-Feistel construction* $\mathsf{m2F} = \mathsf{m2F}[\mathsf{H}, \mathsf{G}, \mathsf{IC}]$ *is a tuple of PPT algorithms* $\mathsf{m2F.Enc}/\mathsf{m2F.Dec} : \mathcal{PW} \times \mathcal{R} \times \mathbb{G} \to \mathcal{R} \times \mathbb{G}$ *where* $\mathcal{PW}$ *is the dictionary of possible passwords,* $\mathcal{R} = \{0,1\}^n$*,* $\mathbb{G}$ *is a group and* $\mathcal{K}$ *is the keyspace of* IC*. The descriptions of* $\mathsf{m2F.Enc}$ *and* $\mathsf{m2F.Dec}$ *are in Figure 3.*

$\mathsf{m2F.Enc}(pw, r, M)$	$\mathsf{m2F.Dec}(pw, s, T)$
$R \leftarrow \mathsf{H}(pw, r)$	$t \leftarrow \mathsf{G}(pw, T)$
$T \leftarrow M \odot R$	$r \leftarrow \mathsf{IC.Dec}(t, s)$
$t \leftarrow \mathsf{G}(pw, T)$	$R \leftarrow \mathsf{H}(pw, r)$
$s \leftarrow \mathsf{IC.Enc}(t, r)$	$M \leftarrow T \odot R^{-1}$
return (s, T)	**return** (r, M)

Fig. 3: The modified 2-Feistel ($\mathsf{m2F}[\mathsf{H}, \mathsf{G}, \mathsf{IC}]$) [32], $\odot$ is the group operation in $\mathbb{G}$ and $(\cdot)^{-1}$ is the inverse in $\mathbb{G}$.

3 PAPKE-HIC—Practical Post-Quantum PAPKE

In this section, we present a new practical PAPKE construction, PAPKE-HIC (Figure 5), which achieves the UC-PAPKE notion, offers strong security guarantees, and avoids the need to instantiate an ideal cipher over groups or large domains. We show how to instantiate PAPKE-HIC using any AI-CCA, SROB-CCA, UNI-PK, and DROB-CCA-secure public key encryption scheme and a half-ideal cipher. Furthermore, we discuss subtle issues overlooked in the original proof of PAPKE-IC in [12].

Setup. Let $\mathsf{PKE} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key encryption scheme with a public key space $\mathcal{PK}$ and let $\mathsf{HIC} = (\mathsf{HIC.Enc}, \mathsf{HIC.Dec})$ be a half-ideal cipher with the domain $\mathcal{D} = \mathcal{R} \times \mathcal{PK}$.

$\mathsf{PAPKE.Keygen}(\lambda, pw)$.
Compute $(sk, pk) \leftarrow\!\!\$\ \mathsf{PKE.Keygen}(\lambda)$ and $apk \leftarrow \mathsf{HIC.Enc}(pw, pk)$. Output (sk, apk).

$\mathsf{PAPKE.Enc}(apk', pw', m)$.
Compute a public key $pk' \leftarrow \mathsf{HIC.Dec}(pw', apk')$, and output $c \leftarrow \mathsf{PKE.Enc}(pk', m)$.

$\mathsf{PAPKE.Dec}(sk, c')$.
Decrypt $m \leftarrow \mathsf{PKE.Dec}(sk, c')$ and output m.

Fig. 5: The generic PAPKE-HIC construction.

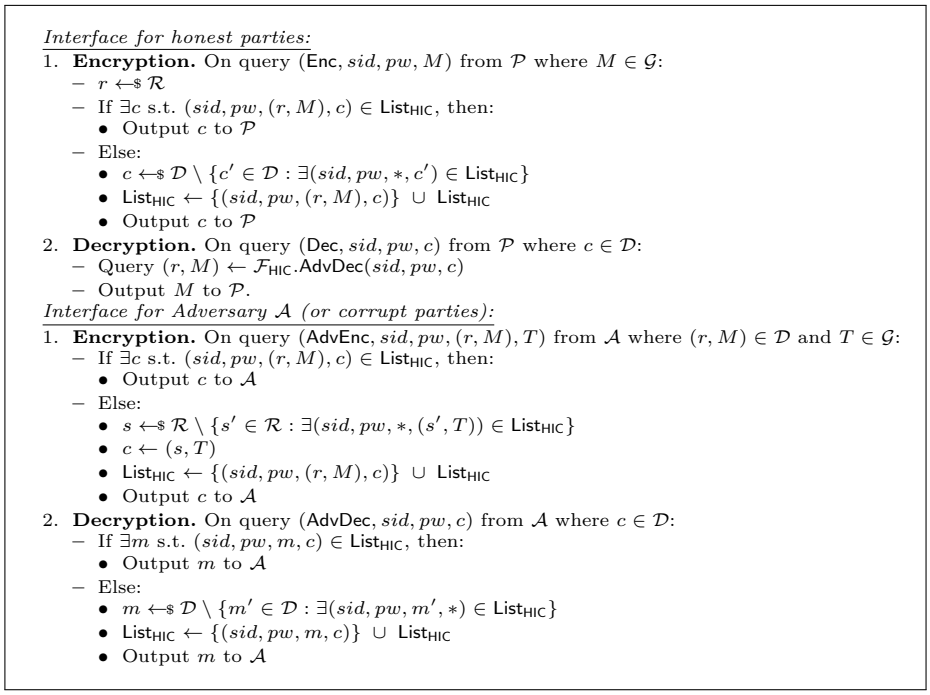


Fig. 4: Ideal functionality $\mathcal{F}_{\mathsf{HIC}}$ for the Half-Ideal Cipher from [32]. The functionality is parametrized by the domain $\mathcal{D} = \mathcal{R} \times \mathcal{G}$.

3.1 Benefits of PAPKE-HIC

PAPKE-HIC, depicted in Figure 5, is inspired by the original PAPKE-IC construction from [12], with the difference that a half-ideal cipher (HIC) [32] is used instead of an ideal cipher (IC). The main benefit of a HIC, or more precisely its instantiation m2F (Section 2.5), is that it avoids the need for an IC over the public key space. Constructing a post-quantum PAPKE requires a post-quantum PKE, and such schemes typically involve large public keys. As a result, instantiating a post-quantum PAPKE-IC in practice is challenging, making the HIC-based approach a more practical alternative.

Recall that m2F uses an ideal cipher internally, but this cipher operates over bitstrings of length 256 (assuming a 128-bit security), which can be instantiated using a regular block cipher. Even though some post-quantum KEMs, such as Classic McEliece [2], have public keys that are indistinguishable from random bitstrings, and there exist methods for encoding public keys as random bitstrings (e.g., Kemeleon for ML-KEM [22]), the main challenge remains their size. For instance, the size of a Kemeleon-encoded ML-KEM public key is 781 *bytes*, and for Classic McEliece it is 261120 *bytes* at NIST security level 1.

There are techniques to instantiate an ideal cipher over bitstrings > 256 [15, 17]; however, these are practically inefficient for large domains.

While m2F requires a *hash-to-group* function, hashing into the group—which in this case is the public key space—is easy in our setting, as most of the post-quantum KEMs derive their keys from a seed. This derivation mechanism can be used to implement the required hash-to-group function.

3.2 Proof of PAPKE-HIC and Issues with the PAPKE-IC Proof

We now explain why substituting an ideal cipher (IC) with a half-ideal cipher (HIC) in the PAPKE construction does not compromise security. First, a HIC ciphertext generated via the HIC.Enc interface effectively commits the adversary to a unique key (password) used to produce that ciphertext (authenticated public key). This commitment property holds just as it does in the IC setting, despite the adversary having additional power to specify part of a ciphertext using the adversarial HIC.AdvEnc interface. Crucially, the simulator can straight-line extract the associated key (password), ensuring that only a single password guess is possible per adversarially generated authenticated public key. Furthermore, as with IC, decrypting a HIC ciphertext (authenticated public key) using a key (password) different from the one used in encryption yields a random element in the HIC domain. This allows the simulator to embed a public key of its choice when simulating HIC.Dec responses.

In Theorem 1, we prove that PAPKE-HIC (Figure 5) securely realizes the PAPKE UC functionality (Figure 2) by leveraging the HIC functionality (Figure 4) and several security properties of the underlying PKE scheme.

Theorem 1. *The scheme* PAPKE-HIC *(Figure 5) UC-realizes* $\mathcal{F}_{\mathsf{PAPKE}}$ *(Figure 2) in the* $\mathcal{F}_{\mathsf{HIC}}$*-hybrid model, if the public key encryption scheme* PKE *is* AI-CCA*,*

SROB-CCA, UNI-PK, DROB-CCA-*secure and* δ*-correct. Specifically, there exists a simulator* SIM*, such that for all environments* $\mathcal{Z}$ *controlling a dummy adversary* $\mathcal{A}$*, its distinguishing advantage,* Adv*, between the real and ideal world is*

$$\begin{aligned}\mathsf{Adv}(\lambda) \leq\ & \frac{q_{\mathsf{HIC}}^2}{2 \cdot |\mathcal{R}|} + \frac{q_{\mathsf{HIC}}^2}{2 \cdot |\mathcal{PK}|} + \frac{q_{\mathsf{PAPKE}}}{|\mathcal{R} \times \mathcal{PK}|} + q_{\mathsf{PAPKE}} \cdot \delta(\lambda) \\ & + 2 \cdot q_{\mathsf{HIC}} \cdot (q_{\mathsf{HIC}} + 1) \cdot \mathsf{Adv}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda) \\ & + 2 \cdot (q_{\mathsf{HIC}} + 1) \cdot q_{\mathsf{PAPKE},\mathsf{Enc}} \cdot \mathsf{Adv}^{\mathsf{AI\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda) \\ & + q_{\mathsf{HIC}} \cdot \mathsf{Adv}^{\mathsf{UNI\text{-}PK}}_{\mathcal{B},\mathsf{PKE}}(\lambda) + \mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda).\end{aligned}$$

Here q_{HIC} *and* q_{PAPKE} *represent upper bounds on the number of queries to* $\mathcal{F}_{\mathsf{HIC}}$ *and* $\mathcal{F}_{\mathsf{PAPKE}}$*, respectively.* $\mathcal{R}$ *is the* HIC *randomness space and* $\mathcal{PK}$ *is the public key space of* PKE*.*

The proof of Theorem 1 is provided in the full version of the paper [5]. Given the similarities between the IC and the HIC, our proof is inspired by the original proof of PAPKE-IC (Theorem 3, [12]). However, we identified and addressed several subtle issues in the original proof.

The main issue concerns the treatment of the event bad_5 in Game_5 (using numbering from [12]). The original proof claims that this event cannot occur because there always exists a row in $\mathbf{L}$ for the honest apk such that $sk \neq \perp$. However, under the simulator's decryption procedure in Game_5, the corresponding row in $\mathbf{L}$ is generated via a call to $\mathsf{IC.Enc}(pw, pk)$ and therefore contains $sk = \perp$. One could alter the simulation of the key generation process to avoid this issue—for example, by sampling apk, querying $\mathsf{IC.Dec}(pw, apk)$, and retrieving (pk, sk) from $\mathbf{L}$ to ensure $sk \neq \perp$. While this would prevent bad_5 in Game_5, it does not extend to later games such as Game_8, where the generated (pk, sk) pair becomes independent of apk, and the simulator cannot rely on knowing the password. In such settings, the environment could potentially craft a valid ciphertext without knowing the public key, which reintroduces bad_5.

This motivates our *decryption robustness* property DROB-CCA (Section 4), which formalizes the requirement that an adversary cannot forge a valid ciphertext under a public key it does not know, even with access to a decryption oracle. In our proof, we argue the equivalent of the bad_5 event (in our proof referred to as bad_9) using the DROB-CCA-security of the PKE. This corresponds to the final term in the bound stated in Theorem 1.

We also observed that the "badkeys" abort (3rd line in the key generation of $\mathcal{F}_{\mathsf{PAPKE}}$) is not mentioned in the original proof. Although minor, this omission is relevant. The event occurs with negligible probability, assuming a large apk keyspace. In our proof, it contributes the $q_{\mathsf{PAPKE}} \cdot |\mathcal{R} \times \mathcal{PK}|^{-1}$ term in Theorem 1.

Additionally, we generalize the proof to consider non-perfectly correct public key encryption schemes with non-perfectly uniform public keys (UNI-PK). This is important when instantiating PAPKE-HIC with a post-quantum PKE, as most available schemes are neither perfectly correct nor have perfectly uniform public keys. Note that our proof of PAPKE-HIC can be adapted into a PAPKE-IC proof by modifying our Game_2 to accommodate for an IC instead of a HIC.

3.3 Why DROB-CCA Matters for the Security of UC-PAPKE?

Let us first explain why the DROB-CCA property is necessary in the context of PAPKE within the Universal Composability (UC) framework. Consider the following PAPKE attack in the UC framework. The environment $\mathcal{Z}$ initiates a party $\mathcal{P}$ with a *high-entropy password* pw by sending the message $(\mathsf{KEYGEN}, sid, pw)$. The simulator, upon receiving the message (KEYGEN, sid), must choose a (random) apk, without knowing the actual password pw.

Later, $\mathcal{Z}$ instructs $\mathcal{P}$ to decrypt a randomly chosen ciphertext c by sending $(\mathsf{DECRYPT}, sid, c)$. Since $\mathcal{P}$ was honestly initiated, there exists a key record $(sid, apk, pw) \in \mathsf{List}_{\mathsf{keyrec}}$. However, there is no corresponding encryption record in $\mathsf{List}_{\mathsf{encrec}}$, because the ciphertext c was chosen independently by the environment.

The simulator, upon receiving query $(\mathsf{DECRYPT}, sid, c)$, must respond with $(\mathsf{PLAINTEXT}, sid, m, pw^*)$. The probability that $pw^* = pw$, i.e., that the simulator guesses the correct password, is negligible because pw is hidden from the simulator and was chosen from a high-entropy distribution. Thus, with overwhelming probability, the response sent to $\mathcal{P}$ must be $\perp$, i.e., decryption fails.

At first glance, this may seem like an artificial attack: the environment is not even trying to learn the underlying public key, but rather to exploit behavioral inconsistencies between the real world and the simulated ideal world. Nevertheless, it pinpoints an important observation: we can only ensure a sound simulation if ciphertexts selected before key generation are also rejected in the real world. This motivates the need for our DROB-CCA property.

4 Decryption Robustness of Public Key Encryption

Motivated by the requirements of the proof of PAPKE-(H)IC (Section 3.2), we defined in Section 2 a new public key encryption property, called *decryption robustness under chosen ciphertext-attacks* (DROB-CCA). In this section, we examine how it relates the existing *strong robustness* (SROB-CCA) property and also which schemes satisfy DROB-CCA directly. To the best of our knowledge, this is a new property, relevant in a setting where the public key is not yet known—an atypical situation in public-key encryption security modeling, where public keys are generally assumed to be known to all parties.

First, we show that DROB-CCA is implied by SROB-CCA in Section 4.1. However, the reduction is not tight, with at least a quadratic security loss. A direct proof with a tighter bound is therefore preferable for concrete security analyses.

In Section 4.2, we show that a hybrid PKE constructed from a KEM and a DEM satisfies DROB-CCA if 1) the DEM is INT-CTXT-secure and 2) the KEM is constructed using a variant of an FO transform that is contributory, i.e., that utilizes the public key during the symmetric key derivation. Note that many post-quantum KEMs satisfy condition 2), e.g., ML-KEM [30] or FrodoKEM [29]. We prove the result for a specific variant of the FO transform introduced in [19]

called *Fujisaki-Okamoto Transformation with Prefix Hashing*. The transform is described in Figure 7.

It's also worth noting that not all KEMs satisfy condition 2). For example, Classic McEliece [2] does not involve the public key during the symmetric key derivation. More importantly, as observed in [20], there exist ciphertexts that decapsulate to the same key under any keypair. This implies that a hybrid PKE using Classic McEliece cannot satisfy DROB-CCA since an adversary can select one of these ciphertexts and use its decapsulated value as the DEM key.

This motivates our *hashed public key* KEM transform (HPK), presented in the full version of the paper [5], which transforms any KEM into one that can be used to build a DROB-CCA-secure PKE scheme.

In the full version [5], we show that DHIES*, originally used to instantiate PAPKE-IC in [12], also directly satisfies DROB-CCA.

4.1 Strong Robustness Implies Decryption Robustness

We separate the proof SROB-CCA $\implies$ DROB-CCA into two steps. First, in Theorem 2, we prove that SROB-CCA $\implies$ DROB, where DROB (Figure 1) is a weaker variant of DROB-CCA in which the adversary does not have access to the Dec oracle. Then, we prove that DROB is equivalent to DROB-CCA, albeit with a $q \in \mathsf{poly}(\lambda)$ security loss. The relationships are visualized in Figure 6.

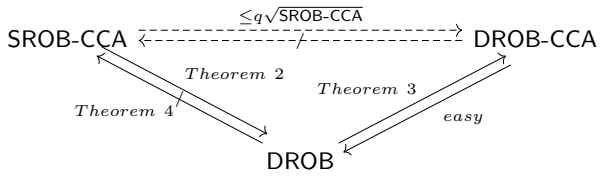


Fig. 6: Relations among SROB-CCA, DROB-CCA and DROB.

Theorem 2. *If a public key encryption scheme* PKE *satisfies* SROB-CCA*, then it also satisfies* DROB*. In other words, for any PPT adversary* $\mathcal{A}$ *against* DROB*, there exists a PPT adversary* $\mathcal{B}$ *against* SROB-CCA *such that*

$$\mathsf{Adv}^{\mathsf{DROB}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \sqrt{\mathsf{Adv}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda)}.$$

Proof. We prove the contrapositive: if there is an efficient adversary $\mathcal{A}$ against DROB, then there is an efficient adversary $\mathcal{B}$ against SROB-CCA.

The SROB-CCA adversary $\mathcal{B}$ receives the security parameter λ, two public keys pk_0, pk_1 corresponding to the private keys sk_0, sk_1, and access to two decryption oracles $\mathsf{Dec}(sk_0, \cdot), \mathsf{Dec}(sk_1, \cdot)$. $\mathcal{B}$ simply runs $\mathcal{A}$ with λ as input and once $\mathcal{A}$ outputs c^*, $\mathcal{B}$ submits it to the challenger. Let us analyze the advantage of $\mathcal{B}$.

We need to show that $f(\lambda) \coloneqq \Pr\left[\mathsf{Exp}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda) = 1\right]$ is non-negligible; specifically that

$$\exists d \in \mathbb{N}, \forall n_0 \in \mathbb{N}, \exists \lambda \geq n_0 : f(\lambda) > \frac{1}{\lambda^d}.$$

By assumption, $g(\lambda) \coloneqq \Pr\left[\mathsf{Exp}^{\mathsf{DROB}}_{\mathcal{A},\mathsf{PKE}}(\lambda) = 1\right]$ is non-negligible, i.e., there exists $d_{\mathsf{DROB}} \in \mathbb{N}$ s.t. for any $n_0 \in \mathbb{N}$ there exists $\lambda \geq n_0$ for which $g(\lambda) > \frac{1}{\lambda^{d_{\mathsf{DROB}}}}$. Note that

$$\begin{aligned} g(\lambda) &= \Pr_{\substack{(pk,sk)\sim\mathsf{Keygen}(\lambda) \\ r\sim U}}[\mathsf{Dec}(sk, \mathcal{A}(\lambda, r)) \neq \perp] \\ &= \sum_{c\in\mathcal{C}} \left(\Pr_{r\sim U}[\mathcal{A}(\lambda, r) = c] \cdot \Pr_{(pk,sk)\sim\mathsf{Keygen}(\lambda)}[\mathsf{Dec}(sk, c) \neq \perp] \right) \\ &> \frac{1}{\lambda^{d_{\mathsf{DROB}}}}, \end{aligned}$$

where $\mathcal{C}$ is the set of ciphertexts and r denotes the random coins for $\mathcal{A}$ from the uniform distribution over $\{0,1\}^{\mathsf{poly}(\lambda)}$ (denoted as U). By explicit definition of $\mathcal{B}$ in terms of $\mathcal{A}$, we have

$$\begin{aligned} f(\lambda) &= \Pr\left[\mathsf{Exp}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda) = 1\right] \\ &= \Pr_{\substack{(pk_0,sk_0)\sim\mathsf{Keygen}(\lambda) \\ (pk_1,sk_1)\sim\mathsf{Keygen}(\lambda) \\ r\sim U}}[\mathsf{Dec}(sk_0, \mathcal{A}(\lambda, r)) \neq \perp \wedge \mathsf{Dec}(sk_1, \mathcal{A}(\lambda, r)) \neq \perp] \\ &= \sum_{c\in\mathcal{C}} \left(\Pr_{r\sim U}[\mathcal{A}(\lambda, r) = c] \cdot \Pr_{\substack{(pk_0,sk_0)\sim\mathsf{Keygen}(\lambda) \\ (pk_1,sk_1)\sim\mathsf{Keygen}(\lambda)}}[\mathsf{Dec}(sk_0, c) \neq \perp \ \wedge \mathsf{Dec}(sk_1, c) \neq \perp] \right). \end{aligned}$$

For a fixed c, the probabilities that $\mathsf{Dec}(sk_0, c) \neq \perp$ and $\mathsf{Dec}(sk_1, c) \neq \perp$ are independent since sk_0 and sk_1 are independently sampled, therefore

$$f(\lambda) = \sum_{c\in\mathcal{C}} \left(\Pr_{r\sim U}[\mathcal{A}(\lambda, r) = c] \cdot \left(\Pr_{(pk,sk)\sim\mathsf{Keygen}(\lambda)}[\mathsf{Dec}(sk, c) \neq \perp] \right)^2 \right). \quad (1)$$

Jensen's inequality gives us that $\psi(\mathbb{E}[X]) \leq \mathbb{E}(\psi(X))$ where ψ is a real convex function and $X : \Omega \to \mathcal{X}$ a random variable. Furthermore, for any $h : \mathcal{X} \to \mathbb{R}$, $h(X)$ is a random variable and $\mathbb{E}[h(X)] = \sum_{x\in\mathcal{X}} h(x) \Pr[X = x]$. In our case $\psi(x) \coloneqq x^2$, $X \coloneqq \mathcal{A}(\lambda, r)$ where r is sampled uniformly at random from the appropriate domain and

$$h(c) \coloneqq \Pr_{(sk,pk)\sim\mathsf{Keygen}(\lambda)}[\mathsf{Dec}(sk, c) \neq \perp].$$

Therefore, by applying Jensen's inequality to Equation (1) we get

$$f(\lambda) \geq \left(\sum_{c \in \mathcal{C}} \Pr_{r \sim U} [\mathcal{A}(\lambda, r) = c] \cdot \Pr_{(sk,pk) \sim \mathsf{Keygen}(\lambda)} [\mathsf{Dec}(sk, c) \neq \perp] \right)^2$$
$$= g(\lambda)^2 > \frac{1}{\lambda^{2d_{\mathsf{DROB}}}}.$$

In other words, we have found $d \coloneqq 2d_{\mathsf{DROB}}$ which proves that $f(\lambda)$ is non-negligible.

Next, we show that DROB is nearly equivalent to DROB-CCA. The implication DROB-CCA $\implies$ DROB is clear as any successful DROB adversary always wins the DROB-CCA game by submitting its output c^* to the Dec oracle at the end. So, $\mathsf{Adv}^{\mathsf{DROB}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda)$. We prove the other implication in Theorem 3.

Theorem 3. *If a public key encryption scheme* PKE *satisfies* DROB*, then it also satisfies* DROB-CCA*. Specifically, for any PPT adversary $\mathcal{A}$ against* DROB-CCA*, there exists a PPT adversary $\mathcal{B}$ against* DROB *such that*

$$\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq q_D \cdot \mathsf{Adv}^{\mathsf{DROB}}_{\mathcal{B},\mathsf{PKE}}(\lambda).$$

where q_D is the number of $\mathcal{A}$'s Dec *oracle queries.*

Theorem 3 is proven in the full version of the paper [5].

Corollary 1. *If a public key encryption scheme* PKE *satisfies* SROB-CCA*, then it also satisfies* DROB-CCA*. More precisely, for any PPT adversary $\mathcal{A}$ against* DROB-CCA *there exists a PPT adversary $\mathcal{B}$ against* SROB-CCA *such that*

$$\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq q_D \cdot \sqrt{\mathsf{Adv}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda)}.$$

where q_D is the number of $\mathcal{A}$'s Dec *oracle queries.*

Theorem 4. *There exists a public key encryption scheme that is* DROB*-secure, but is not* SROB-CCA*-secure. In other words,* DROB *does not imply* SROB-CCA*.*

Theorem 4 is proven in the full version of the paper [5]. Note that Theorem 4 shows that DROB-CCA and SROB-CCA are not equivalent notions.

4.2 Building a DROB-CCA PKE from a Prefix-Hashing FO KEM

In Section 4.1, we have shown that SROB-CCA implies DROB-CCA. However, the reduction is not tight. Specifically, there is at least a quadratic security loss. This motivates Theorem 5, which shows that the standard hybrid (KEM+DEM) public key encryption construction (PKE^{hy} in Figure 8) directly satisfies DROB-CCA, provided that 1) the DEM is INT-CTXT-secure and 2) the KEM is constructed using the prefix-hashing Fujisaki-Okamoto transform pFO (Figure 7).

The pFO transform is a modified version of the transform introduced in [19, Figure 4]—this transform does not derive the "rejection key" $\bar{K}$ from $\mathsf{ID}(pk)$, which exactly models KEMs such as ML-KEM [30]. Note that in the proof and the description of the transform (Figure 7) we use Python-like syntax to indicate truncated outputs, e.g., the first 128 bits of the value $\mathsf{F}(x)$ is denoted as $\mathsf{F}(x)[0:128]$.

$\mathsf{pFO}[\mathsf{PKE}_1, \mathsf{ID}, \mathsf{F}].\mathsf{Keygen}(\lambda)$	$\mathsf{pFO}[\mathsf{PKE}_1, \mathsf{ID}, \mathsf{F}].\mathsf{Encap}(pk)$	$\mathsf{pFO}[\mathsf{PKE}_1, \mathsf{ID}, \mathsf{F}].\mathsf{Decap}(sk, c)$
$(pk, sk') \leftarrow \mathsf{PKE}_1.\mathsf{Keygen}(\lambda)$	$m \leftarrow\!\!\$\ \mathcal{M}$	$(sk', s, pk) \leftarrow sk$
$s \leftarrow\!\!\$\ \mathcal{M}$	$(K, r) \leftarrow \mathsf{F}(\mathsf{ID}(pk), m)$	$m' \leftarrow \mathsf{PKE}_1.\mathsf{Dec}(sk', c)$
$sk \leftarrow (sk', s, pk)$	$c \leftarrow \mathsf{PKE}_1.\mathsf{Enc}(pk, m, r)$	$(K, r) \leftarrow \mathsf{F}(\mathsf{ID}(pk), m')$
return (pk, sk)	**return** (K, c)	$\bar{K} \leftarrow \mathsf{F}(s, c)[0:k]$
		if $m' = \perp \vee \mathsf{PKE}_1.\mathsf{Enc}(pk, m', r) \neq c$
		return $\bar{K}$ $\boxed{/\perp}$
		return K

Fig. 7: $\mathsf{pFO}[\mathsf{PKE}_1, \mathsf{ID}, \mathsf{F}]$ is the prefix-hashing FO transform inspired by [19, Figure 4]. PKE_1 is a PKE, ID is a fixed-length output function and F is a random oracle. The inclusion of the boxed code defines the explicitly rejecting transform.

$\mathsf{PKE}^{hy}[\mathsf{KEM}, \mathsf{DEM}].\mathsf{Keygen}(\lambda)$	$\mathsf{PKE}^{hy}[\mathsf{KEM}, \mathsf{DEM}].\mathsf{Enc}(pk, m)$	$\mathsf{PKE}^{hy}[\mathsf{KEM}, \mathsf{DEM}].\mathsf{Dec}(sk, c)$
$(pk, sk) \leftarrow \mathsf{KEM}.\mathsf{Keygen}(\lambda)$	$(c_1, k) \leftarrow \mathsf{KEM}.\mathsf{Encap}(pk)$	$(c_1, c_2) \leftarrow c$
return (pk, sk)	$c_2 \leftarrow \mathsf{DEM}.\mathsf{Enc}(k, m)$	$k \leftarrow \mathsf{KEM}.\mathsf{Decap}(sk, c_1)$
	$c \leftarrow (c_1, c_2)$	**if** $k = \perp$ **then** **return** $\perp$
	return c	$m \leftarrow \mathsf{DEM}.\mathsf{Dec}(k, c_2)$
		return m

Fig. 8: Description of the standard hybrid PKE construction using a KEM and a DEM. The inclusion of the boxed code defines the construction for explicitly rejecting KEMs.

We prove Theorem 5 in the QROM model [11] using the following lemma from [31, Lemma 2.2] which shows that a quantum random oracle can be used as a PRF.

Lemma 1. *Let $l \in \mathbb{N}$. Let $\mathsf{F} : \{0,1\}^l \times \mathcal{X} \to \mathcal{Y}$ and $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$ be two independent random oracles. If an unbounded-time quantum adversary $\mathcal{D}$ makes at most q queries to its quantum-accessible random oracles, then we have*

$$\left|\Pr\left[1 \leftarrow \mathcal{D}^{\mathsf{F},\mathsf{F}(t,\cdot)}() | t \leftarrow\!\!\$\ \{0,1\}^l\right] - \Pr\left[1 \leftarrow \mathcal{D}^{\mathsf{F},\mathsf{H}}()\right]\right| \leq \frac{2 \cdot q}{\sqrt{2^l}}.$$

Theorem 5. *Let $\mathsf{PKE} = \mathsf{PKE}^{hy}[\mathsf{KEM}, \mathsf{DEM}]$ be a hybrid public key encryption scheme constructed from an implicitly rejecting key encapsulation mechanism* KEM *and a data encapsulation mechanism* DEM*. If* KEM *is constructed using the prefix-hashing FO transform* pFO *(Figure 7), i.e., $\mathsf{KEM} = \mathsf{pFO}[\mathsf{PKE}_1, \mathsf{ID}, \mathsf{F}]$ where PKE_1 is a public key encryption scheme, $\mathsf{ID} : \mathcal{PK} \to \{0,1\}^n$ is a fixed-length output function, $\mathsf{F} : \{0,1\}^* \to \{0,1\}^k \times \{0,1\}^{\overline{r}}$ is a quantum random oracle and* DEM *is* INT-CTXT*-secure, then for any QPT adversary $\mathcal{A}$ against* DROB-CCA *of* PKE *there exists a QPT adversary $\mathcal{B}$ against* INT-CTXT *of* DEM *such that*

$$\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \frac{4 \cdot q_{\mathsf{F}}}{\sqrt{2^{\min(m,k)}}} + q_D \cdot \mathsf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathcal{B},\mathsf{DEM}}(\lambda),$$

for $(pk, sk) \leftarrow \mathsf{PKE}_1.\mathsf{Keygen}(\lambda)$, where $m = H_\infty(\mathsf{ID}(pk))$ is the min-entropy of $\mathsf{ID}(pk)$, q_{F} denotes the number of queries to the quantum random oracle F *and q_D denotes the number of queries to the (classical) oracle* Dec*.*

Proof. Let $\mathcal{A}$ be an adversary against DROB-CCA of PKE. Denote by $\Pr[\mathsf{Game_i}]$ the probability that $\mathcal{A}$ wins the i-th game. See the full version of the paper [5] for a detailed description of the games.

Game_0. This is the unmodified DROB-CCA game with an unpacked PKE, i.e.,

$$\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) = \Pr[\mathsf{Game}_0].$$

Game_1. In Game_1, we replace the KEM key and randomness derivation in the decryption oracle $(K, r) \leftarrow \mathsf{F}(\mathsf{ID}(pk), m_1)$ with $(K, r) \leftarrow \mathsf{H}_1(m_1)$ where H_1 is an internal random oracle. This change is undetectable by $\mathcal{A}$ with probability given by Lemma 1.

We build a distinguisher $\mathcal{D}$ from Lemma 1 as follows. $\mathcal{D}$ has access to two quantum random oracles F, F' where $\mathsf{F}'(\cdot) = \mathsf{F}(t, \cdot)$ for $t \leftarrow\!\!\$\ \{0,1\}^m$, or $\mathsf{F}'(\cdot) = \mathsf{H}_1(\cdot)$. $\mathcal{D}$ runs $\mathcal{A}$ in a modified Game_0 where $(K, r) \leftarrow \mathsf{F}(\mathsf{ID}(pk), m_1)$ in the decryption oracle is replaced with $(K, r) \leftarrow \mathsf{F}'(m_1)$. $\mathcal{D}$ outputs 1 if $\mathcal{A}$ wins the game, i.e., submits a decrypting ciphertext. Clearly, if $\mathsf{F}'(\cdot) = \mathsf{F}(t, \cdot)$ then the game corresponds to Game_0 since t has the same min-entropy as $\mathsf{ID}(pk)$ and if $\mathsf{F}'(\cdot) = \mathsf{H}_1(\cdot)$, then it corresponds to Game_1.

By Lemma 1, $|\Pr[\mathsf{Game}_0] - \Pr[\mathsf{Game}_1]| =$

$$\left|\Pr\left[1 \leftarrow \mathcal{D}^{\mathsf{F},\mathsf{F}(t,\cdot)}() | t \leftarrow\!\!\$\ \{0,1\}^m\right] - \Pr\left[1 \leftarrow \mathcal{D}^{\mathsf{F},\mathsf{H}_1}()\right]\right| \leq \frac{2 \cdot q_{\mathsf{F}}}{\sqrt{2^m}}.$$

Game_2. In Game_2, we replace the rejection KEM key derivation in the decryption oracle $K' \leftarrow \mathsf{F}(s, c_1)$ with $K' \leftarrow \mathsf{H}_2(c_1)$ where H_2 is an internal random oracle. Analogously to the previous game hop, by Lemma 1 we get

$$|\Pr[\mathsf{Game}_1] - \Pr[\mathsf{Game}_2]| \leq \frac{2 \cdot q_{\mathsf{F}}}{\sqrt{2^k}}.$$

Game_3. In Game_3, we reorganize the decryption oracle code. K is sampled from the same distribution regardless of whether H_1 or H_2 is called. Therefore, we can "unroll" H_1 and H_2 and replace them with sampling of $K \leftarrow\!\!\$\ \{0,1\}^k$ with record keeping. We also remove the "rejection if branch."

We keep a list List_C with pairs (c_1, K) where c_1 is the queried KEM ciphertext. Observe that keeping the list indexed by c_1 is enough to preserve consistency. Consider that $\mathcal{A}$ queries $c = (c_1, c_2)$ and $c' = (c'_1, c'_2)$ to the Dec oracle:

- If $c_1 = c'_1$, then the lookup in List_C assures consistency.
- If $c_1 \neq c'_1$ and $\mathsf{PKE}_1.\mathsf{Dec}(sk, c_1) \neq \mathsf{PKE}_1.\mathsf{Dec}(sk, c'_1)$, then Game_2 also derives independent DEM keys.
- If $c_1 \neq c'_1$ and $\mathsf{PKE}_1.\mathsf{Dec}(sk, c_1) = m' = \mathsf{PKE}_1.\mathsf{Dec}(sk, c'_1)$, then observe that the derived DEM keys are also independent because c_1 or c'_1 must fail the FO re-encryption check given that $\mathsf{PKE}_1.\mathsf{Enc}(pk, m', r)$ is deterministic.

Since this game is just a code reorganization, we have $\Pr[\mathsf{Game}_2] = \Pr[\mathsf{Game}_3]$.

Observe that $\mathcal{A}$ wins Game_3 iff it queries $c = (c_1, c_2)$ where $\mathsf{DEM}.\mathsf{Dec}(K, c_2) \neq \perp$ where $K \leftarrow\!\!\$\ \{0,1\}^k$. We show that we can build an $\mathsf{INT\text{-}CTXT}$ adversary $\mathcal{B}$ using $\mathcal{A}$ in Game_2.

Let $i \in \{1, \ldots, q_D\}$ be the index of the first query from $\mathcal{A}$ that decrypts. $\mathcal{B}$ guesses $i \in \{1, \ldots, q_D\}$ and runs $\mathcal{A}$ in a modified Game_2 where $\mathcal{B}$ answers with $\perp$ to the first $i - 1$ queries. During the i-th query, it uses its Dec oracle from the $\mathsf{INT\text{-}CTXT}$ challenger instead of calling $\mathsf{DEM}.\mathsf{Dec}(K, \cdot)$ with $K \leftarrow\!\!\$\ \{0,1\}^k$. Observe that K is sampled from the same distribution in Game_2 and in the $\mathsf{INT\text{-}CTXT}$ game and $\mathcal{B}$ perfectly simulates Game_2 if it guessed i. Therefore, $\mathcal{B}$ always wins if it correctly guessed i and $\mathcal{A}$ wins. In other words,

$$\Pr[\mathsf{Game}_3] \leq q_D \cdot \mathsf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathcal{B},\mathsf{DEM}}(\lambda).$$

In total, we have

$$\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \frac{4 \cdot q_{\mathsf{F}}}{\sqrt{2^{\min(m,k)}}} + q_D \cdot \mathsf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathcal{B},\mathsf{DEM}}(\lambda).$$

Explicitly rejecting pFO. Note that a variant of Theorem 5 also holds for a PKE constructed from an explicitly rejecting pFO KEM. The proof is almost identical except we do not need to consider the rejection key. As a consequence there is no $s \in \{0,1\}^k$ and the bound is $\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \frac{2 \cdot q_{\mathsf{F}}}{\sqrt{2^m}} + q_D \cdot \mathsf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathcal{B},\mathsf{DEM}}(\lambda)$.

Application of Theorem 5 to existing KEMs. Recall that compared to prefix-hashing FO transform from [19], the pFO transform is slightly modified so that it directly models KEMs such as ML-KEM [30] and FrodoKEM [29]. However, the inclusion of $\mathsf{ID}(pk)$ in rejection key derivation only tightens the bound from Theorem 5—we get $4 \cdot q_{\mathsf{F}} \cdot \sqrt{2^{-m}}$ instead of $4 \cdot q_{\mathsf{F}} \cdot \sqrt{2^{-\min(m,k)}}$.

As discussed in [19], the function ID may either be a hash function or a function that selects the random portion of the public key. For instance, in ML-KEM, ID is instantiated as SHA3-256, meaning the entire public key is hashed—including a 256-bit random seed. In FrodoKEM, the key derivation is more involved, however, it still incorporates a hash of the entire public key.

Impact of Theorem 5 on concrete security. We compare the concrete $\mathsf{DROB\text{-}CCA}$ security bounds implied by Theorem 2 and by Theorem 5 considering a hybrid PKE instantiated with ML-KEM [30].

From [20, Theorem 2 + Theorem 12], we have $\mathsf{Adv}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \frac{4 \cdot q_{\mathsf{F}}}{2^{128}} + \epsilon$, where q_{F} denotes the number of queries to the quantum-accessible random oracle and $\epsilon \ll 2^{-128}$ a negligible term. By Corollary 1, we have $\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq q_D \cdot \sqrt{\mathsf{Adv}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{B},\mathsf{PKE}}(\lambda)}$, where q_D is the number of decryption queries. Replacing $\mathsf{Adv}^{\mathsf{SROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)$ term with the bound above, we get $\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq q_D \cdot \sqrt{\frac{4 \cdot q_{\mathsf{F}}}{2^{128}} + \epsilon}$. While Corollary 1 shows that $\mathsf{DROB\text{-}CCA}$ and $\mathsf{SROB\text{-}CCA}$ are asymptotically equivalent, a concrete security analysis reveals that even for arguably modest upper bounds on q_{F} and q_D (say $q_{\mathsf{F}} = 2^{40}$ and $q_D = 2^{16}$), the combination of [20, Theorem 2 + Theorem 12] and Corollary 1 provides limited insight into the actual security guarantees that ML-KEM delivers in terms of *decryption robustness*, as it does not yield an upper bound that would be regarded as meaningful under modern security standards.

In contrast, the direct analysis of the prefix-hashing FO transform allows us to establish security bounds via Theorem 5 that remain meaningful in practical settings, i.e., we have that $\mathsf{Adv}^{\mathsf{DROB\text{-}CCA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \frac{4 \cdot q_{\mathsf{F}}}{\sqrt{2^{\min(m,k)}}} + q_D \cdot \mathsf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathcal{B},\mathsf{DEM}}(\lambda)$, where $m = k = 256$ for ML-KEM. Since the DEM can be selected such that $\mathsf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathcal{B},\mathsf{DEM}}(\lambda)$ becomes the non-dominant term, this gives a meaningful upper bound on the decryption-robustness advantage against PKE from ML-KEM in practical settings.

5 Implementation and Performance

In this section, we discuss which KEMs are suitable to build a post-quantum $\mathsf{PAPKE\text{-}HIC}$. The main advantage of $\mathsf{PAPKE\text{-}HIC}$ compared to $\mathsf{PAPKE\text{-}IC}$ is the fact that it does not require an ideal cipher over large algebraic domains, which affects performance. To demonstrate the concrete advantage of our HIC-based approach, we implemented post-quantum $\mathsf{PAPKE\text{-}HIC}$ and $\mathsf{PAPKE\text{-}IC}$ using ML-KEM [30], FrodoKEM [29] and Classic McEliece [2]. We focus on ML-KEM implementation details and performance results in this section, while further results for FrodoKEM and Classic McEliece are provided in the full version [5].

5.1 Possible PAPKE-HIC instantiations

We consider three post-quantum KEMs: ML-KEM [30], FrodoKEM [29] and Classic McEliece [2] as candidates for PAPKE-HIC instantiation. Recall that PAPKE-HIC requires a PKE scheme that is IND-CCA, ANO-CCA, SROB-CCA, UNI-PK, and DROB-CCA (Theorem 1). We construct a post-quantum PKE using the standard KEM+DEM construction (Figure 8). Note that IND-CCA-security of the PKE is easily achievable since each mentioned KEM is IND-CCA. As shown in Section 4.1, deriving DROB-CCA from SROB-CCA incurs a quadratic security degradation, which then becomes the dominant term in the security bound of Theorem 1. Considering concrete security, this may require choosing larger security parameters for PAPKE-HIC instantiations. However, as shown in Section 4.2, this is unnecessary for KEMs with the prefix-hashing FO transform (pFO) described in Figure 7. We now analyze each KEM separately.

ML-KEM. The hybrid PKE with ML-KEM is ANO-CCA [28] and SROB-CCA [20]. Moreover, in ML-KEM, the public key is computationally indistinguishable from uniform under the MLWE assumption [8], thus supporting the UNI-PK property in practice. ML-KEM uses the pFO transform, and thus directly satisfies DROB-CCA with the bound from Theorem 5.

FrodoKEM. The hybrid PKE with FrodoKEM is ANO-CCA and SROB-CCA as shown in [37]. FrodoKEM uses the pFO transform, so it satisfies DROB-CCA with the bound from Theorem 5. FrodoKEM satisfies UNI-PK under the LWE assumption by an analogous argument as for ML-KEM in [8].

Classic McEliece. It was shown in [37] that Classic McEliece satisfies ANO-CCA under additional assumptions. UNI-PK is considered a standard assumption for Classic McEliece [2]. However, [20] demonstrated that Classic McEliece does not satisfy SCFR-CCA, which means that the hybrid PKE with Classic McEliece is not SROB-CCA. A modified variant of Classic McEliece, proposed in [37], yields a hybrid PKE that satisfies SROB-CCA. This modified variant essentially integrates the pFO transform internally into Classic McEliece, thus directly satisfying DROB-CCA per Theorem 5.

Still, modifying the internals of an existing scheme can be impractical. That is why we introduce the *hashed public key* (HPK) KEM transform in the full version of the paper [5], a simpler black-box modification resulting in a DROB-CCA-secure scheme. We show that HPK preserves anonymity and that any HPK-transformed KEM is SCFR-CCA and consequently the hybrid PKE is SROB-CCA.

5.2 ML-KEM PAPKE implementation

The crux of implementing PAPKE-IC is the instantiation of the ideal cipher over the ML-KEM public key space, which is a group. There exist two approaches to implement an ideal cipher over the ML-KEM public key space. The first approach is to combine an ideal cipher over bitstrings with a quasi-bijective

encoding of the public keys to bitstrings [21]. This requires an ideal cipher over a large bitstring domain, which can be realized either via domain extenders [15] or via an 8-round Feistel network [17]. The second approach is to use the 8-round Feistel construction [17] to build an ideal cipher over the public key space directly, which in turn requires 4 hash-to-group operations.[5]

Regarding the PAPKE-HIC implementation, there currently exists only one HIC construction, m2F (Figure 3), which requires one hash-to-group operation, i.e., hashing to the ML-KEM public key space. To realize this, we implemented two hash-to-group approaches. The first one employs the SampleNTT subroutine from [30, Algorithm 7], the same rejection sampling routine used by ML-KEM key generation. The second approach is to use Kemeleon [22], specifically Kemeleon decoding, which maps bitstrings to ML-KEM public keys.

To summarize, we have implemented and benchmarked the following implementations of post-quantum PAPKE with ML-KEM:

(a) PAPKE-IC is implemented by combining Kemeleon—used to encode ML-KEM as a pseudo-uniform bitstring—with an ideal cipher instantiated over bitstrings, either via a domain extender or an 8-round Feistel network.
(b) PAPKE-IC is also implemented by directly employing an 8-round Feistel, in which four of the hashing operations are hash-to-group operations realized using either SampleNTT or Kemeleon decoding.
(c) PAPKE-HIC is implemented with a 2-round modified Feistel m2F, in which the single hash-to-group operation is implemented either via SampleNTT or Kemeleon decoding.

Note that the underlying public key encryption scheme is the KEM+DEM hybrid construction PKE^{hy} (Figure 8) with ML-KEM serving as the KEM. Recall from Section 3 that we require the PKE to be SROB-CCA-secure, which in this case requires the DEM to be FROB-secure [20, Theorem 2]. Standard AEAD constructions, such as AES-GCM, are typically not FROB-secure [18], therefore, we use the CTX construction introduced in [14, Figure 2] which builds an FROB-secure[6] DEM from an AEAD and a collision-resistant hash function. We use AES-256-GCM as the underlying AEAD and SHA-256 as the hash function.

Our C++ implementation uses libraries Botan (https://botan.randombit.net/) and Catch2 (https://github.com/catchorg/Catch2). Specifically, we use Botan for cryptographic primitives, such as ML-KEM and AES-256-GCM, and Catch2 for benchmarking. More details and the full code is available at https://github.com/PAPKE-HIC/benchmarks.

5.3 ML-KEM PAPKE Performance Results

We executed the benchmarks on a machine with AMD EPYC 7543 (2,8 GHz) as a CPU, running Ubuntu 24.04 and compiled using GCC 13. We used the `--benchmark-samples 1000` Catch2 parameter during benchmarking.

[5] One wire carries the group element while the second carries a random bitstring.
[6] The paper in question [14] uses a different security notion ($\mathsf{CAE_{XX}}$) which implies FROB from [20].

Table 1 provides performance results of our PAPKE-IC and PAPKE-HIC implementations with ML-KEM-512. For ML-KEM-768, ML-KEM-1024 see Tables 2 and 3.

Table 1: ML-KEM-512 PAPKE performance. Values are in microseconds in the format: `mean` ± `std. dev.`; "bitstring Extender/Feistel" refers to variants (a), "Feistel+H2G" refers to variants (b) from Section 5.2.

Variant	**Key Gen.**	**Encryption**	**Decryption**
PAPKE-HIC, SampleNTT	116 ± 8	116 ± 11	98 ± 7
PAPKE-IC, Feistel+H2G, SampleNTT	170 ± 6	189 ± 15	98 ± 10
PAPKE-IC, bitstring Feistel	417 ± 258	$14\,606 \pm 126$	103 ± 13
PAPKE-IC, bitstring Extender	974 ± 227	$14\,816 \pm 114$	99 ± 10
PAPKE-HIC, Kemeleon	$14\,572 \pm 222$	$14\,534 \pm 185$	100 ± 10
PAPKE-IC, Feistel+H2G, Kemeleon	$56\,932 \pm 503$	$57\,449 \pm 864$	98 ± 9

Table 2: ML-KEM-768 PAPKE performance. Values are in microseconds in the format: `mean` ± `std. dev.`; "bitstring Extender/Feistel" refer to variants (a), "Feistel+H2G" refer to variants (b) from Section 5.2.

Variant	**Key Gen.**	**Encryption**	**Decryption**
PAPKE-HIC, SampleNTT	162 ± 9	163 ± 10	171 ± 13
PAPKE-IC, Feistel+H2G, SampleNTT	248 ± 10	257 ± 11	170 ± 9
PAPKE-IC, bitstring Feistel	471 ± 174	$40\,628 \pm 308$	179 ± 10
PAPKE-IC, bitstring Extender	$2\,183 \pm 181$	$43\,534 \pm 431$	178 ± 13
PAPKE-HIC, Kemeleon	$40\,480 \pm 268$	$41\,082 \pm 632$	183 ± 15
PAPKE-IC, Feistel+H2G, Kemeleon	$166\,040 \pm 1\,810$	$163\,714 \pm 2\,374$	197 ± 19

Table 3: ML-KEM-1024 PAPKE performance. Values are in microseconds in the format: `mean` ± `std. dev.`; "bitstring Extender/Feistel" refer to variants (a), "Feistel+H2G" refer to variants (b) from Section 5.2.

Variant	**Key Gen.**	**Encryption**	**Decryption**
PAPKE-HIC, SampleNTT	215 ± 9	222 ± 14	269 ± 18
PAPKE-IC, Feistel+H2G, SampleNTT	327 ± 10	345 ± 13	266 ± 14
PAPKE-IC, bitstring Feistel	984 ± 577	$88\,959 \pm 1\,228$	280 ± 21
PAPKE-IC, bitstring Extender	$2\,767 \pm 657$	$89\,841 \pm 972$	282 ± 20
PAPKE-HIC, Kemeleon	$87\,908 \pm 372$	$89\,760 \pm 1\,157$	284 ± 21
PAPKE-IC, Feistel+H2G, Kemeleon	$354\,004 \pm 4\,455$	$353\,089 \pm 3\,801$	290 ± 16

We found that the Kemeleon-based hash-to-group is the bottleneck in variants that use it. This is because Kemeleon decoding requires division of large integers (around 6000 bits for ML-KEM-512). Note that PAPKE-IC with bitstring ideal cipher constructions use Kemeleon *encoding* during key generation and *decoding* during encryption. This explains why those instantiations perform better than PAPKE-HIC (Kemeleon variant), which relies on Kemeleon decoding for the hash-to-group, at key generation, but perform essentially the same at encryption.

Overall, the best performing constructions rely on the SampleNTT-based hashing to group, and of those, PAPKE-HIC performs better than PAPKE-IC. This is to be expected when we trade four hash-to-group operations for one hash-to-group operation and a block cipher.

It's also worth noting that the (relative) high variance of PAPKE-IC variants with bitstring ideal cipher is caused by rejection of keypairs—only certain ML-KEM public keys can be Kemeleon-encoded. For ML-KEM-512 the success probability is $\approx 56\%$ [22], but for ML-KEM-768 it is $\approx 83\%$, which explains the smaller relative variance for ML-KEM-768 in Table 2.

6 Conclusion

Why was DROB-CCA not needed in KEM-to-PAKE compilers? The new DROB notion introduced here targets PKE schemes, but a similar property could trivially be adapted for explicitly rejecting KEMs. One might even consider an analogue for implicitly rejecting KEMs, although this would require examining the internal workings of the KEM to identify the winning condition in decapsulation and determine when the decapsulation did not fail. Given the long line of work compiling KEMs to PAKE in the UC framework [8, 32, 4, 26, 3], why has decryption robustness (or another similar property) not previously emerged as a required property in the PAKE setting?

In OEKE-style protocols (e.g., OCAKE, EKE-KEM, CHIC, and NoIC [8, 32, 4, 3]), the simulator can reject ciphertexts based on authentication tags derived from a random oracle. This effectively forces the adversary to commit to a specific password, which can then be extracted by the simulator. This password allows the simulator to associate an *apk* to a unique public key with knowledge of the secret key; a tag not returned by the random oracle will only be valid with negligible probability over the oracle's output space.

In EKE-style compilers (e.g., CAKE, EKE-PRF [8, 26]), the KEM ciphertext is further encrypted with an ideal cipher (with domain separation from the session identifier and the password attributed to the party). As a result, if the adversary submits a ciphertext that was never queried to the IC, it is impossible to extract a valid password. (Due to the IC's behavior, the underlying KEM ciphertext is also completely random.) This appears to be precisely what [26] identified as being overlooked in CAKE [8], which they fix by requiring "strong pseudorandomness" from the underlying key agreement (KA). According to the

authors, this property on the KA is satisfied when the KA is built from KEM satisfying both SPR-CCA and SMT-CCA security properties [37].

We leave it as future work to investigate whether DROB-CCA (or a variant thereof) would be sufficient for EKE-style compilers from KEMs. The issues appear related, and strong pseudorandomness may be unnecessarily strong if DROB-CCA alone suffices to prevent such attacks, provided the session key is derived from a KDF that binds the full communication transcript and the underlying common secrets (i.e. the password and the KEM key).

It is also worth noting that the standard $\mathcal{F}_{\mathsf{PAKE}}$ functionality is implicitly rejecting, whereas $\mathcal{F}_{\mathsf{PAPKE}}$ explicitly rejects. This means that in PAPKE, the simulator must decide whether decryption fails, while in PAKE, the simulator can output a random session key and defer consistency of the simulation to a later stage. Another important difference is that PAPKE allows reuse of the authenticated public key and supports multiple decryption queries, whereas PAKE built from KEM demands only a single decryption/decapsulation per party (assuming the appropriate domain separation in idealized objects is in place).

Quantum adversaries. Although we have lifted some simpler theorems to the QROM (Theorem 5 and Theorem 6 in the full version [5]), the analysis of the main PAPKE-HIC construction remains in the classical setting, as we do not consider a quantum-accessible Half Ideal Cipher. Currently we have a somewhat limited set of techniques for lifting proofs to the QROM and even fewer for the QICM. The lack of perfect independence between input-output pairs and the possibility of quantum queries to the inverse permutation makes the analysis of quantum-accessible ideal ciphers usually a challenging barrier to overcome. (The same applies to the Half Ideal Cipher.) This is likely why lifting the security proofs of generic PAKE compilers from KEMs remains an open problem and an active area of research. New efforts in this direction are emerging, namely with a partial quantum proof for OCAKE protocol [24] and a generic compiler that avoids the IC [3]. Since PAPKE implies PAKE, a generic PAPKE construction proven secure in the quantum setting would also resolve the open problem for PAKE. We leave this as an interesting direction for future work.

Benefits of contributory KEMs. KEMs that incorporate the public key into the symmetric key derivation are typically referred to as *contributory* [19, 16] meaning that both parties contribute to the derived symmetric key during a key exchange. In this paper, we identify an additional benefit: contributory KEMs naturally give rise to hybrid public key encryption schemes that satisfy DROB-CCA when paired with an integrity-protecting DEM.

Acknowledgments. Jan Oupický was supported by the industrial partnership project between the interdisciplinary research center SnT and LuxTrust. Afonso Arriaga and Peter Rønne received support from the Luxembourg National Research Fund (FNR) under the CORE project (C21/IS/16221219/ImPAKT). Afonso Arriaga and Marjan Škrobot also received support from FNR Core Junior project (C21/IS/16236053/FuturePass).

Bibliography

[1] Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010: 7th Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 5978, pp. 480–497. Springer Berlin Heidelberg, Germany, Zurich, Switzerland (Feb 9–11, 2010). https://doi.org/10.1007/978-3-642-11799-2_28

[2] Albrecht, M.R., Bernstein, D.J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., von Maurich, I., Misoczki, R., Niederhagen, R., Paterson, K.G., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., Szefer, J., Tjhai, C.J., Tomlinson, M., Wang, W.: Classic McEliece. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions

[3] Arriaga, A., Barbosa, M., Jarecki, S.: NoIC: PAKE from KEM without ideal ciphers. Cryptology ePrint Archive, Report 2025/231 (2025), https://eprint.iacr.org/2025/231

[4] Arriaga, A., Barbosa, M., Jarecki, S., Skrobot, M.: C'est Très CHIC: A compact password-authenticated key exchange from lattice-based KEM. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part V. Lecture Notes in Computer Science, vol. 15488, pp. 3–33. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0935-2_1

[5] Arriaga, A., Mestel, D., Oupický, J., Rønne, P.B., Škrobot, M.: HIC is all you need: Practical Post-Quantum Password-Authenticated Public Key Encryption. Cryptology ePrint Archive, Report 2026/020 (2026), https://eprint.iacr.org/2026/020

[6] Arriaga, A., Ryan, P.Y.A., Skrobot, M.: SweetPAKE: Key exchange with decoy passwords. In: Zhou, J., Quek, T.Q.S., Gao, D., Cárdenas, A.A. (eds.) ASIACCS 24: 19th ACM Symposium on Information, Computer and Communications Security. ACM Press, Singapore (Jul 1–5, 2024). https://doi.org/10.1145/3634737.3645009

[7] Barbosa, M., Gellert, K., Hesse, J., Jarecki, S.: Bare PAKE: Universally composable key exchange from just passwords. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part II. Lecture Notes in Computer Science, vol. 14921, pp. 183–217. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68379-4_6

[8] Beguinet, H., Chevalier, C., Pointcheval, D., Ricosset, T., Rossi, M.: GeT a CAKE: Generic transformations from key encaspulation mechanisms to password authenticated key exchanges. In: Tibouchi, M., Wang, X. (eds.) ACNS 2023: 21st International Conference on Applied Cryptography and Network Security, Part II. Lecture Notes in Computer Science, vol. 13906, pp. 516–538. Springer, Cham, Switzerland, Kyoto, Japan (Jun 19–22, 2023). https://doi.org/10.1007/978-3-031-33491-7_19

[9] Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) Advances in Cryptology – ASI-

ACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 566–582. Springer Berlin Heidelberg, Germany, Gold Coast, Australia (Dec 9–13, 2001). https://doi.org/10.1007/3-540-45682-1_33

[10] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. Journal of Cryptology **21**(4), 469–491 (Oct 2008). https://doi.org/10.1007/s00145-008-9026-x

[11] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer Berlin Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011). https://doi.org/10.1007/978-3-642-25385-0_3

[12] Bradley, T., Camenisch, J., Jarecki, S., Lehmann, A., Neven, G., Xu, J.: Password-authenticated public-key encryption. Cryptology ePrint Archive, Report 2019/199 (2019), https://eprint.iacr.org/2019/199

[13] Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.D.: Universally composable password-based key exchange. In: Cramer, R. (ed.) Advances in Cryptology – EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 404–421. Springer Berlin Heidelberg, Germany, Aarhus, Denmark (May 22–26, 2005). https://doi.org/10.1007/11426639_24

[14] Chan, J., Rogaway, P.: On committing authenticated-encryption. In: Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W. (eds.) ESORICS 2022: 27th European Symposium on Research in Computer Security, Part II. Lecture Notes in Computer Science, vol. 13555, pp. 275–294. Springer, Cham, Switzerland, Copenhagen, Denmark (Sep 26–30, 2022). https://doi.org/10.1007/978-3-031-17146-8_14

[15] Coron, J.S., Dodis, Y., Mandal, A., Seurin, Y.: A domain extender for the ideal cipher. In: Micciancio, D. (ed.) TCC 2010: 7th Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 5978, pp. 273–289. Springer Berlin Heidelberg, Germany, Zurich, Switzerland (Feb 9–11, 2010). https://doi.org/10.1007/978-3-642-11799-2_17

[16] Cremers, C., Dax, A., Medinger, N.: Keeping up with the KEMs: Stronger security notions for KEMs and automated analysis of KEM-based protocols. In: Luo, B., Liao, X., Xu, J., Kirda, E., Lie, D. (eds.) ACM CCS 2024: 31st Conference on Computer and Communications Security. pp. 1046–1060. ACM Press, Salt Lake City, UT, USA (Oct 14–18, 2024). https://doi.org/10.1145/3658644.3670283

[17] Dai, Y., Steinberger, J.P.: Indifferentiability of 8-round Feistel networks. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 95–120. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53018-4_4

[18] Dodis, Y., Grubbs, P., Ristenpart, T., Woodage, J.: Fast message franking: From invisible salamanders to encryptment. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part I. Lecture Notes

in Computer Science, vol. 10991, pp. 155–186. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 19–23, 2018). https://doi.org/10.1007/978-3-319-96884-1_6

[19] Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G.: Faster lattice-based KEMs via a generic Fujisaki-Okamoto transform using prefix hashing. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021: 28th Conference on Computer and Communications Security. pp. 2722–2737. ACM Press, Virtual Event, Republic of Korea (Nov 15–19, 2021). https://doi.org/10.1145/3460120.3484819

[20] Grubbs, P., Maram, V., Paterson, K.G.: Anonymous, robust post-quantum public key encryption. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 402–432. Springer, Cham, Switzerland, Trondheim, Norway (May 30 – Jun 3, 2022). https://doi.org/10.1007/978-3-031-07082-2_15

[21] Gu, Y., Jarecki, S., Krawczyk, H.: KHAPE: Asymmetric PAKE from key-hiding key exchange. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021, Part IV. Lecture Notes in Computer Science, vol. 12828, pp. 701–730. Springer, Cham, Switzerland, Virtual Event (Aug 16–20, 2021). https://doi.org/10.1007/978-3-030-84259-8_24

[22] Günther, F., Stebila, D., Veitch, S.: Obfuscated key exchange. In: Luo, B., Liao, X., Xu, J., Kirda, E., Lie, D. (eds.) ACM CCS 2024: 31st Conference on Computer and Communications Security. pp. 2385–2399. ACM Press, Salt Lake City, UT, USA (Oct 14–18, 2024). https://doi.org/10.1145/3658644.3690220

[23] Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017: 15th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer, Cham, Switzerland, Baltimore, MD, USA (Nov 12–15, 2017). https://doi.org/10.1007/978-3-319-70500-2_12

[24] Hövelmanns, K., Hülsing, A., Kudinov, M., Ritsch, S.: CAKE requires programming - on the provable post-quantum security of (O)CAKE. Cryptology ePrint Archive, Report 2025/458 (2025), https://eprint.iacr.org/2025/458

[25] International Civil Aviation Organization (ICAO): Doc 9303: Machine readable travel documents (2021), https://www.icao.int/publications/Documents/9303_p11_cons_en.pdf

[26] Januzelli, J., Roy, L., Xu, J.: Under what conditions is encrypted key exchange actually secure? In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part II. Lecture Notes in Computer Science, vol. 15602, pp. 451–481. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91124-8_16

[27] Kiefer, F., Manulis, M.: Oblivious PAKE: Efficient handling of password trials. In: Lopez, J., Mitchell, C.J. (eds.) ISC 2015: 18th International Conference on Information Security. Lecture Notes in Computer Science, vol. 9290,

pp. 191–208. Springer, Cham, Switzerland, Trondheim, Norway (Sep 9–11, 2015). https://doi.org/10.1007/978-3-319-23318-5_11
[28] Maram, V., Xagawa, K.: Post-quantum anonymity of Kyber. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023: 26th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 13940, pp. 3–35. Springer, Cham, Switzerland, Atlanta, GA, USA (May 7–10, 2023). https://doi.org/10.1007/978-3-031-31368-4_1
[29] Naehrig, M., Alkim, E., Bos, J., Ducas, L., Easterbrook, K., LaMacchia, B., Longa, P., Mironov, I., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: FrodoKEM. Tech. rep., National Institute of Standards and Technology (2020), available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
[30] NIST: Module-Lattice-Based Key-Encapsulation Mechanism Standard. Tech. Rep. Federal Information Processing Standard (FIPS) 203, U.S. Department of Commerce (Aug 2024). https://doi.org/10.6028/NIST.FIPS.203
[31] Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2017/1005 (2017), https://eprint.iacr.org/2017/1005
[32] Santos, B.F.D., Gu, Y., Jarecki, S.: Randomized half-ideal cipher on groups with applications to UC (a)PAKE. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 128–156. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_5
[33] Thread Group: Thread Networking Protocol (2020), https://www.threadgroup.org/
[34] Vanhoef, M.: Multiple Passwords in WPA3: Use Cases & Initial Proposals (2025), https://papers.mathyvanhoef.com/pake2025-slides.pdf
[35] Warner, B.: Magic Wormhole: Simple Secure File Transfer. In: PyCon US (2016), https://magic-wormhole.readthedocs.io/
[36] Wi-Fi Alliance: Wi-Fi Certified WPA3 Security (2021), https://www.wi-fi.org/discover-wi-fi/security
[37] Xagawa, K.: Anonymity of NIST PQC round-3 KEMs. Cryptology ePrint Archive, Report 2021/1323 (2021), https://eprint.iacr.org/2021/1323