Contents

1 Introduction 2
1.1 Password Authenticated Key Exchange 2
1.2 Related Work 3
1.3 Motivation and Contribution 3
1.4 Overview 3
1.5 Methodology (How to Read this Paper) 4

2 PAKE Definition and Security 6
2.1 PAKE Security 7
2.2 Balanced and Augmented PAKEs 7
2.3 Adversarial (Threat) Models and Frameworks 8

3 Post-Quantum Cryptography 9
3.1 Cryptographic Primitives and Hardness Assumptions 9
3.2 Key Agreement and Key Encapsulation 10

4 Security of Key Agreement and Key Encapsulation 11
4.1 Abstract Properties 11
4.2 KEM Security 13

5 Supporting Cryptographic Building Blocks 14
5.1 Smooth Projective Hash Functions 15
5.2 Oblivious Pseudo-Random Functions 15
5.3 Non-Interactive Zero-Knowledge Proofs 16

6 Security Models and Idealized Objects 17
6.1 Standard Model 17
6.2 Random Oracle Model 17
6.3 Ideal Cipher Model 17
6.4 Half Ideal Cipher 18
6.5 Programmable-Once Public Function 18
6.6 Common Reference String 19

7 PQC PAKEs 20
7.1 Classification 20
7.2 Lattice PAKEs 21
7.3 Isogeny PAKEs 23
7.4 Generic PAKEs 24
7.5 aPAKE Compilers (Symmetric to Asymmetric Transformation) 27
7.6 Security Overview 28
7.7 Performance Overview 28
7.8 Real World Use Cases 30

8 Honorable Mentions 32

9 Unexplored Territory 33

10 Observations and Discussion 36

11 Conclusion and Future Work 40

A Corresponding Classical Design Class Representatives 56

SoK: PQC PAKEs (Long Paper)
Cryptographic Primitives, Design and Security

Nouri Alnahawi[1,3,4], David Haas[2], Erik Mauß[1,3,4] and Alexander Wiesmaier[1,3,4]

[1] Darmstadt University of Applied Sciences, 64295 Darmstadt, Germany
\{nouri.alnahawi,erik.mauss,alexander.wiesmaier\}@h-da.de
[2] Technical University Darmstadt, 64289 Darmstadt, Germany
david.haas1@stud.tu-darmstadt.de
[3] National Research Center for Applied Cybersecurity ATHENE, 64295 Darmstadt, Germany
[4] European University of Technology, European Union

Abstract. Password Authenticated Key Exchange (PAKE) establishes secure communication channels using relatively short, often human memorable, passwords for authentication. The currently standardized PAKEs however rely on classical asymmetric (public key) cryptography. Thus, these classical PAKEs may become insecure, should the expected quantum threat become a reality. Despite the growing interest in realizing quantum-safe PAKEs, they did not receive much attention from the ongoing Post-Quantum Cryptography (PQC) integration efforts. Thus, there is a significant gap in awareness compared to PQC primitives subject to the official governmental and institutional standardization processes. In this work, we provide a comprehensive overview of the existing PQC PAKEs focusing on their design rationales, authentication methods and asymmetric key agreement primitives. Further, we classify PQC PAKEs w.r.t. their properties and security assurances. Finally, we address PAKE designs that are still unexplored in the PQC realm and discuss the possibility of their adaptation. Thus, we offer a detailed reference for future work on PQC PAKEs.

Keywords: Systematization of Knowledge · Password Authenticated Key Exchange · Post-Quantum Cryptography · Key Agreement · Key Encapsulation Mechanism.

1 Introduction

Ever since their emergence in the early nineties [BM92], *Password Authenticated Key Exchange* (PAKE) protocols became of great importance in the world of (applied) cryptography. According to [HvO22], PAKEs are present in several applications such as credential recovery (e.g.,*iCloud* and *ProtonMail*), device pairing (e.g., *E-Passport*, *bluetooth* and *WLAN*), and E2E secure communication (e.g., *Thread*). Over the last decade, the advances in quantum computing and its threat to (classical) asymmetric cryptography attracted many studies towards realizing PAKEs based on Post-Quantum Cryptography (PQC). In this paper, we provide an overview of the existing PQC PAKE proposals in the literature and classify them according to their design paradigms and cryptographic building blocks.

1.1 Password Authenticated Key Exchange

PAKEs are a special form of key agreement that add password authentication to key exchange [Jar22]. They establish secure communication over an insecure channel, where authentication is done using a password or a PIN. Hence, they pose the question of how password authentication can be achieved combined with a passively secure key agreement [Jar22]. The first PAKE protocol by Bellovin and Merritt, *Encrypted Key Exchange* (EKE) is based on the idea of combining symmetric encryption with asymmetric key agreement [BM92]. By encrypting the public key with a password (or a thereof derived value), an attacker cannot manipulate the key agreement [Jar22]. Similar approaches are also found in protocols such as the *Password Authenticated Key Agreement* (PAK) [Mac02], *Simple Password Exponential Key Exchange* (SPEKE) [Jab96], and *Password Authenticated Connection Establishment* (PACE) [BFK09]; typically instantiated with DH or ECDH[1].

1.2 Related Work

Azarderakhsh et al. [AJK$^+$20] discuss PAKEs from isogeny assumptions, and show the difficulty or even the impossibility of translating DH-based PAKEs to isogenies. To the best of our knowledge, no other works in the literature survey PQC PAKEs directly. The following works focus only on classical PAKEs, but nonetheless provide a valuable foundation for our own overview. For instance, the PAKE overview by Abdalla [Abd14] is one of the earliest works concerned with the design and security goals of PAKE protocols. This work describes the beginnings of PAKE design studies sufficiently (e.g., EKE [BM92] and its variants) and classifies following proposals based on their construction approach. In [Jar22], Jarecki provides a detailed overview on the methods used for constructing PAKEs from classical asymmetric cryptography. The overview offers a thorough analysis on the design and security of PAKEs in different security frameworks, as well as on their strengths and shortcomings. Hao and van Oorschot [HvO22] present a complete classification and a review of the state-of-the-art of classical PAKEs and provide practical information on their real-world applications.

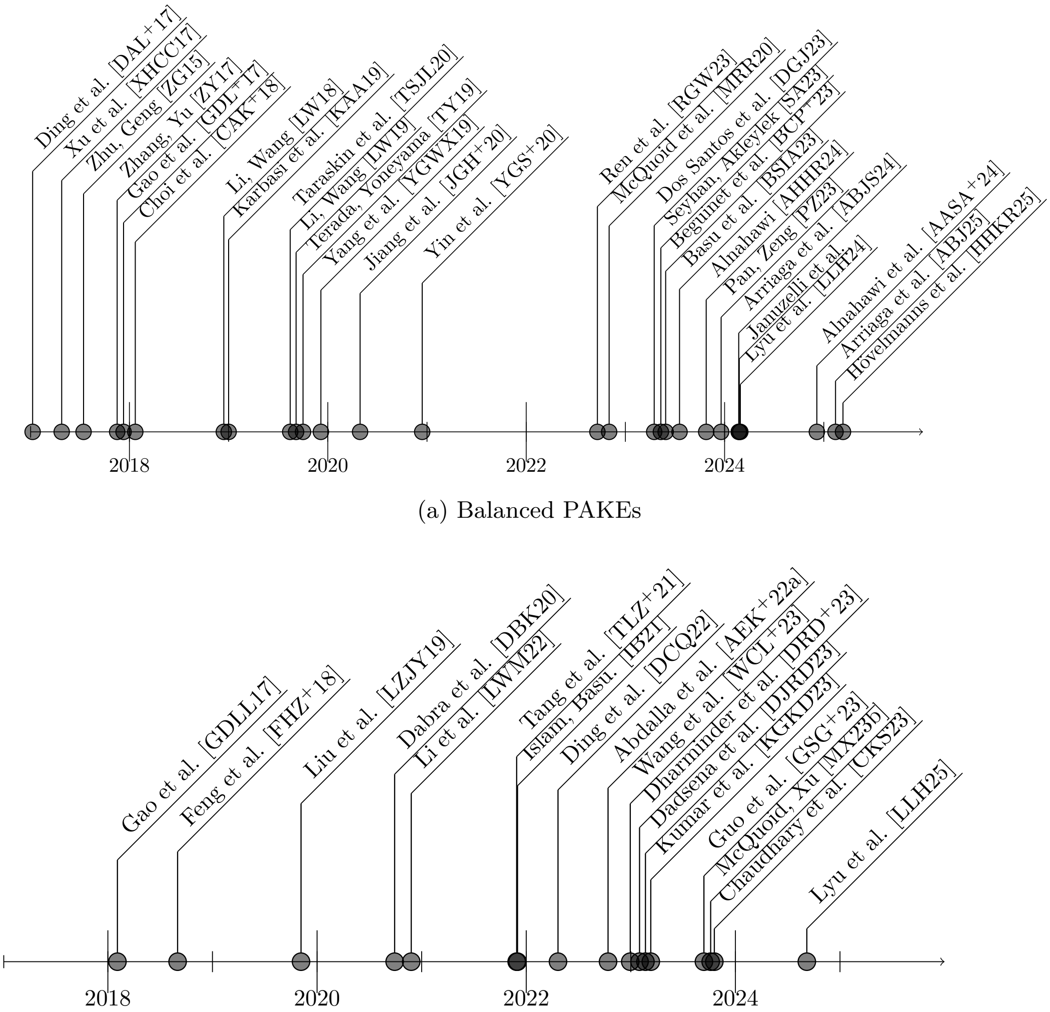
1.3 Motivation and Contribution

Currently, none of the PAKE schemes selected by IEEE (P1363.2 [IEE09]), ISO/IEC (11770-4 [ISO17]) or IETF (CFRG 104 [For19]) in the recent standardizations are quantum-safe. To the best of our knowledge, there does not exist any surveys concerned with PQC PAKEs. We provide an extensive and comprehensive overview of PQC PAKE proposals identified in the literature addressing PAKEs built directly from PQC primitives or on top of PQC *Key Encapsulation Mechanisms* (KEM). This includes PAKEs from lattice and isogeny assumptions, and generic and semi-generic PQC KEMs. We investigate said PAKEs and point out aspects we deem necessary to grasp the differences and issues of constructing PAKEs from PQC schemes. We build upon the systematic review in [HvO22] and adapt their classification of classical PAKEs to PQC ones. By doing so, we enable a direct comparison to classical designs, show the tight relations between classical and PQC PAKEs, and incorporate generic designs. Considering performance and security aspects, we collect analysis and benchmarks, and provide an overview as per the proposed instantiations, parameters, and security models. Further, we address PAKE designs that are currently non-present in the PQC realm and discuss the possibility of their adaptations and instantiations. Finally, we highlight open issues, takeaways, and future work.

1.4 Overview

PQC PAKE dates back before the start of the NIST PQC standardization process. The LWE PAKE by Katz and Vaikuntanathan [KV09] in 2009 and the isogeny-based PAKE by Zhu et al. [ZHS14] in 2014 may therefore, and to the best of our knowledge, be considered the very first PQC PAKEs. The RLWE PAKE by Ding et al. [DAL$^+$17] in 2017, however marks the emergence of many other PQC PAKEs (cf. Timeline in Fig. 1).

[1] We refer to [HvO22] for an overview of *classical* PAKEs and their real-world applications, as well as more details on the past official PAKE standardization processes.



(a) Balanced PAKEs

(b) Augmented PAKEs

Figure 1: Timeline of a) Balanced and b) Augmented PQC PAKE Publications

We collected roughly 50 papers (cf. Tab. 1), which were reviewed except for eight papers[2]. These include over 30 PAKEs based on lattice LWE and variants. The remaining works consist of five (C)SIDH isogeny PAKEs, and eight generic PAKEs. An overview of all surveyed PAKEs and their properties is presented in Tab. 1. There are two parts separating *balanced* and *augmented* PAKEs respectively. In each part we group PAKEs into design classes. We only deem a PAKE fully quantum-safe if its security proof is given in models that address quantum adversaries, regardless of the underlying primitives.

1.5 Methodology (How to Read this Paper)

Due to the large number of PQC PAKEs in the literature and the involved building blocks, we opted for an SoK that enables researchers and scholars to navigate relevant aspects from A to Z withing one paper. Further, we based our classification on the previous SoK

[2] Ye et al. [YHL13] (full-text unavailable), Ding and Fang [DF11] and Jheng et al. [JTCW18] (incomplete works), Yu et al. [YLZ+21] and Rewal et al. [RSM+23] (paid access), Zi-chen et al. [ZTJ21] and Shu et al. [SWL+21] (written in Chinese), and Seyhan and Akleylek [SA24] (unknown assumption).

Table 1: PQC PAKE Overview / Landscape

Class	Scheme	Year	Security Model	Assumptions	Proof Method	Properties							
						Generic[1]	Quantum-Safe[2]	Peer-Reviewed	2-Party	3-Party	Benchmarks	Provide Impl.	Recomm. Param.
Balanced													
C1	Terada, Yoneyama [TY19] ((C)SIDH-EKE)	2019	IC, ROM	(C)SIDH	BPR			✓	✓		✓		
	McQuoid et al. [MRR20] ((O)EKE)	2020	ROM	-	UC	✓		✓	✓				
	Beguinet et al. [BCP$^+$23] ((O)CAKE)	2023	IC, ROM	(M)LWE	UC	✓		✓	✓				
	Dos Santos et al. [DGJ23] (HIC-EKE)	2023	IC, ROM	MLWE(R)^3	UC	✓		✓	✓				
	Pan, Zeng [PZ23] (CAKE)	2023	IC, ROM	(M)LWE	BPR	✓		✓	✓				
	Alnahawi et al. [AHHR24] (OCAKE)	2023	IC, ROM	(M)LWE	BPR	✓		✓	✓		✓	✓	
	Januzelli et al. [JRX24] ((O)EKE)	2024	ROM	-	UC	✓		✓	✓				
	Arriaga et al. [ABJS24] (CHIC)	2024	IC, ROM	(M)LWE	UC	✓		✓	✓		✓	✓	✓
	Alnahawi et al. [AASA$^+$24] (NICE-PAKE)	2024	ROM	(M)LWE	BPR	✓			✓				✓
	Arriaga et al. [ABJ25] (NoIC-PAKE)	2025	ROM	-	UC	✓			✓				
	Hövelmanns et al. [HHKR25] ((O)CAKE)	2025	QROM	(M)LWE	BPR	✓			✓				
C2	Zhu, Geng [ZG15]	2015	-	(C)SIDH	CK				✓				
	Alsayigh [Als16]	2016	ROM	RLWE	BPR				✓				✓
	Ding et al. [DAL$^+$17, Din17] (RLWE-PAK-PPK)	2017	ROM	RLWE	BPR			✓	✓		✓		
	Gao et al. [GDL$^+$17] (RLWE-PAK-PPK)	2017	ROM	RLWE	BPR				✓		✓		
	Taraskin et al. [TSJL20] (SIDH-PAK)	2019	ROM	SIDH	BPR				✓		✓		✓
	Yang et al. [YGWX19] (RLWE-PAK)	2019	ROM	RLWE	BPR			✓	✓		✓		✓
	Jiang et al. [JGH$^+$20] (PAKEs)	2020	ROM	(R)LWE	BPR			✓	✓		✓		✓
	Ren et al. [RGW23]([RG22]) (MLWE-PAK)	2022	-	MLWE	Hybrid			✓	✓		✓	✓	✓
	Seyhan, Akleylek [SA23]	2023	ROM	MLWR	Hybrid				✓		✓	✓	✓
	Basu et al. [BSIA23] (MLWR-2PAKA)	2023	ROM	MLWR	DY			✓	✓				✓
C3	Katz, Vaikuntanathan [KV09]	2009	CRS	LWE	BPR		✓	✓	✓				
	Xu et al. [XHCC17] (RLWE-3PAKE)	2017	ROM	RLWE	BPR					✓	✓		✓
	Zhang, Yu [ZY17]	2017	CRS, ROM	LWE	BPR				✓				
	Choi et al. [CAK$^+$18] (AtLast)	2018	ROM	RLWE	BPR			✓		✓			
	Li, Wang [LW18]	2018	CRS	LWE	BPR		✓	✓	✓				
	Li, Wang [LW19]	2019	CRS	LWE	BPR		✓	✓	✓				
	Karbasi et al. [KAA19] (Ring-PAKE)	2019	CRS	RLWE	BPR		✓	✓	✓				
	Yin et al. [YGS$^+$20]	2020	ROM	LWE	BPR			✓	✓	✓			
	Lyu et al. [LLH24]	2024	(Q)ROM	(M)LWE^4	UC	✓	✓	✓	✓				
Augmented													
C2	Gao et al. [GDLL17]	2018	-	RLWE	UC			✓	✓		✓		✓
C3	Zhu et al. [ZHS14]	2014	-	SIDH	CK			✓		✓			
	Feng et al. [FHZ$^+$18]	2018	ROM	RLWE	BPR			✓	✓		✓		✓
	Liu et al. [LZJY19]	2019	ROM	RLWE	Hybrid					✓	✓	✓	✓
	Dabra et al. [DBK20] (LBA-PAKE)	2020	ROM	RLWE	FTG			✓	✓		✓		✓
	Li et al. [LWM22]	2020	CRS	LWE(R)^3	BPR		✓	✓	✓		✓		✓
	Tang et al. [TLZ$^+$21]	2021	ROM	RLWE	BPR			✓		✓	✓		✓
	Islam, Basu [IB21] (BP-3PAKA)	2021	ROM	RLWE	BPR			✓		✓	✓		
	Ding et al. [DCQ22]	2022	RoR	RLWE	FTG			✓	✓		✓	✓	✓
	Abdalla et al. [AEK$^+$22a] (X-GA-PAKE)	2022	CRS	CSIDH	BPR			✓	✓				
	Wang et al. [WCL$^+$23] (LB-ID-2PAKA)	2023	ROM	MLWE	BPR			✓	✓				
	Dharminder et al. [DRD$^+$23]	2023	Standard	RLWE	Hybrid			✓	✓		✓		✓
	Dadsena et al. [DJRD23]	2023	ROM	RLWE	BPR			✓	✓		✓		✓
	Kumar et al. [KGKD23]	2023	ROM	RLWE	BPR			✓	✓		✓		✓
	Guo et al. [GSG$^+$23]	2023	ROM	MLWE	BPR			✓		✓	✓		✓
	Chaudhary et al. [CKS23]	2023	ROM	RLWE	BPR			✓		✓	✓		✓
	McQuoid, Xu [MX23b]	2023	ROM	CSIDH	UC			✓	✓				
	Lyu et al. [LLH25]	2024	ROM	-	UC	✓		✓	✓		✓		
	Yang et al. [YZYW25] (K-PAKE)	2025	ROM	MLWE	UC			✓	✓		✓		

[1] Denotes whether a PAKE follows a generic design without relying on a specific scheme (e.g., using a KEM opposed to using an RLWE-KEX).

[2] Denotes whether the PAKE provides a proof considering quantum security models.

[3] (R) denotes the (M)LWR variant of (M)LWE as in Learning with Rounding.

[4] Also provides an instantiation based on the GA-DDH assumption.

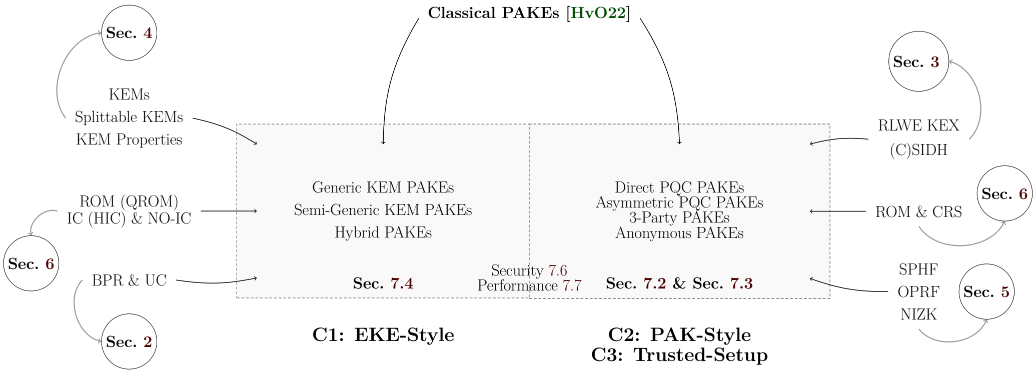


Figure 2: Overview of PQC PAKE Design Classes, Primitives and Security Models.

dealing with classical PAKEs by Hao and van Oorschot [HvO22], to facilitate both the transition and the comparison of design paradigms. That being said, we highly recommend [HvO22] for further, or prior, reading. Fig. 2 illustrates the methodology and structure of this paper. We recommend one of the following reading approaches:

1. Readers new to PAKE research can get familiar the basic concepts in Sec. 2.

2. Classical readers can explore PQC building blocks in Sec. 3 and Sec. 5.

3. Readers interested in KEM research and abstract properties may consult Sec. 4

4. Readers new to security models and idealized models may look at Sec. 6.

5. PAKE experts may directly skip to Sec. 7.

2 PAKE Definition and Security

At its core, a PAKE protocol involves two parties sharing a password π, who intend to establish a session key for secure communication over an unauthenticated channel. Based on [Gjö24], we can formally define a PAKE as follows.

Definition 1 (Password Authenticated Key Exchange [Gjö24])**.** A PAKE is a cryptographic protocol, $PAKE = (\mathcal{I}, \mathcal{R}, \mathcal{K}, \mathcal{D}, AD)$, executed between Initiator $\mathcal{I}$ and Receiver $\mathcal{R}$ and consists of key space $\mathcal{K}$, password dictionary $\mathcal{D}$, associated data AD , and the algorithms:

- $\mathcal{I}$: An interactive algorithms that takes as input the associated data $ad \in AD$ and a password $\pi \in \mathcal{D}$. It sends and receives messages, and eventually either outputs a session key $K \in \mathcal{K}$ or $\perp$ signaling rejection.

- $\mathcal{R}$: An interactive algorithms that takes as input the associated data $ad \in AD$ and a password $\pi \in \mathcal{D}$. It sends and receives messages, and eventually either outputs a session key $K \in \mathcal{K}$ or $\perp$ signaling rejection.

2.1 PAKE Security

PAKE security is essentially defined w.r.t. whether a low entropy password suffices to protect a key exchange (KEX) [Jar22]. According to Jablon [Jab96], this can be achieved if the small password size space is multiplied by the size of the key space of the asymmetric primitive. It follows that a PAKE should have the following security properties:

- Provide session key security (indistinguishability) targeting a transcript of a protocol execution (i.e. a session) [Jar22].
- Resist *offline dictionary attacks* passively targeting the password and a transcript of a protocol execution [Jab96, Jar22].
- Survive *online dictionary attacks* actively targeting the password [Jab96].
- Prevent *Man-in-the-Middle* (MitM) attacks actively targeting the KEX [Jab96].
- Provide explicit mutual authentication (MA), or at least implicit MA [Jar22].
- Provide *Perfect Forward Secrecy* (PFS), or at least weak PFS [Jar22].

Similar to common KEX, the session key security in PAKEs is defined in terms of indistinguishability in the real-or-random model; that is, for passive (eavesdropping) adversaries. Intuitively, this property implies that the passive security of a PAKE must not be worse than the underlying KEX, in the sense that it does not leak information regarding the final key [Jar22]. However, KEX are usually used in authenticated settings (e.g. in TLS). Hence, PAKE adversarial models assume authentication using a password with min-entropy t, such that the success probability of password guessing is upper bounded by 2^{-t} [Jar22], which prevents active MitM adversaries from successfully impersonating any of the honest parties. This so called online dictionary attack must be an adversary's best chance at deceiving an honest party to agree on a session key with them. Since online attacks are relatively easy to deal with through limiting the number of password entry trials, the security of a PAKE protocol heavily relies on its resistance to offline dictionary attacks [Jar22]. Therefore, even if honest parties re-use the same password across multiple sessions, an attacker with substantial resources and enough time should not be able to test further passwords against a public transcript of a PAKE execution [Jar22]. Additionally, PAKEs requires protecting past executions, should the password or ephemeral secrets of parties get leaked or compromised in following sessions. This requirement is referred to as *Perfect Forward Secrecy* (PFS), and roughly corresponds to *adaptive compromise* (corruption). Hence a PAKE that protects against corruption is said to provide adaptive security. Restricting PFS to protecting only passively observed previous executions is called weak PFS [Jar22]. However, the concrete security properties that can be achieved depend on the proof model, in which the formal analysis is conducted (cf. Sec. 2.3).

2.2 Balanced and Augmented PAKEs

PAKEs fulfilling the afore mentioned security properties are referred to as *Balanced* or symmetric PAKEs [HvO22]. The symmetry lies within the fact that two honest parties share the same password in its raw form. This usually applies to a so called user-user or client-client setting, where no server holding many passwords from different clients is involved. However, in the client-server setting, assuming users keep their passwords safe, a stolen or pre-computed password could be used directly in impersonation attacks. PAKEs that also protect against server compromise and pre-computation attacks are thus referred to as *Augmented* or asymmetric PAKEs [HvO22], where the value of the pre-shared password is stored only as a one-way transformation on the server. It follows that Def. 1 does not reflect asymmetry [Gjö24]. Hence, an additional algorithm is required that takes a password and some associated data as input, and outputs a password representation and some client-related data. Then, the initiator can use these parameters as input, and the responder can take the associated data and the password representation as input [Gjö24].

2.3 Adversarial (Threat) Models and Frameworks

The security of PAKEs is commonly shown in one of the following frameworks, which model cryptographic protocols and adversaries w.r.t. their capabilities:

1) *Bellare-Pointcheval-Rogaway* (BPR) model [BPR00] and extended eBPR [AFP05].

2) *Boyko-MacKenzie-Patel* (BMP) model [BMP00].

3) *Universal Composability* (UC) model [Can01].

While BPR is a game based model, BPM and UC are simulation based [Jar22]. This leads to some subtle differences in adversarial capabilities, and consequently in the degree of security captured within each model, where UC is claimed to provide more security. However, almost all of the current PQC PAKEs are analyized in BPR (cf. Tab. 1). Other less used frameworks in the context of PAKEs are the *relaxed* UC [ABB⁺20], *Dolev-Yao* (DY) [DY83], *Canetti-Krawczyk* (CK) [CK01] and *Benhamouda-Blazy-Ducas-Quach* (BBDQ) [BBDQ18] models. We briefly describe BPR and UC only, since BPM can be viewed as a specialization of UC [Jar22]. Further, we refer the reader to [Jar22] for a complete review and comparison between the advantages and disadvantages of each model.

BPR. The BPR model defines a game where an adversary $\mathcal{A}$ interacts with honest parties running multiple protocol instances. The PAKE provides so called oracles that implement the protocol on behalf of the parties and and interaction is performed via a set of *queries* that an adversary may submit throughout a series of game changes. The adversary's goal is to break the protocol through distinguishing real session keys from random ones. At the end of $\mathcal{A}$'s interaction with the protocol, they test a chosen session by outputing a test bit b, which determines if they receive the real session key or a random one from the protocol oracle. $\mathcal{A}$'s advantage in succeeding is defined as $\mathsf{Adv}_{\mathcal{A}}^{\text{pake}} = (\Pr[b' = b] - \frac{1}{2})$, bounding $\mathcal{A}$'s chances at being better than random guessing. Hence, the BPR and eBPR models are often referred to as *Find-then-Guess* (FTG), and *Real-or-Random* (RoR) respectively for single and multiple test queries. Nonetheless, BPR suffers from a limitation on the password distribution and requires that parties choose the same password from a uniform distribution over the password dictionary, which contradicts real world usage of passwords [Jar22].

Definition 2 (PAKE Security in the BPR Model [Jar22])**.** A PAKE protocol is secure in the Bellare-Pointcheval-Rogaway (BPR) model, if for all password dictionary $\mathcal{D}$ and all efficient algorithms (adversaries) $\mathcal{A}$,

$$\mathsf{Adv}_{\mathcal{A}}^{\text{pake}}(1^{\kappa}) \leq \frac{q_s}{|\mathcal{D}|} + \mathsf{negl}(1^{\kappa})$$

where q_s is the number of sessions that $\mathcal{A}$ actively interacts with, and negl is a negligible function for some security parameter κ.

UC. In contrast, the BMP and UC model assure security for arbitrary password distributions and leakages on correlated and mistyped passwords [Jar22]. These models define two distinct games called the *real-world game* and the *ideal-world game* in BPM, and similarly called the *real-world game* and the *real-world adversary* in UC. Further, the UC model defines *ideal* functionalities emulating a protocol, and and environment acting as a distinguisher interacting with honest parties and adversaries. The environment outputs a bit at the end of an execution indicating whether it was interacting with the real protocol or its ideal emulation. This modular approach allows using a UC secure PAKE as an ideal sub-routine in black-box manner within other cryptographic compositions [Jar22]. Nonetheless, UC also includes changes in behavior through steps (i.e., game hops). In each game, the protocol is changed in a way that an adversary's advantage in breaking the original protocol remains statistically negligible. These changes are defined based on the protocol interface (i.e., exchanged messages) and the defined ideal functionalities.

Definition 3 (PAKE Security in the BMP Model [Jar22])**.** A PAKE protocol Π is secure in the Boyko-MacKenzie-Patel (BMP) model, if for all efficient algorithms (adversaries) $\mathcal{A}$there exists an efficient algorithm $\mathcal{A}^*$ s.t.,

$$\mathsf{transcript}^{\mathsf{real}}[\mathcal{A}, \Pi] \overset{(c)}{\approx} \mathsf{transcript}^{\mathsf{ideal}}[\mathcal{A}^*, \mathcal{F}_{\mathsf{PAKE}}]$$

i.e., transcripts of the real-world and the ideal-world are computationally indistinguishable.

Definition 4 (UC Security [Tue13])**.** Let $\mathcal{P}$ be a real protocol and $\mathcal{F}$ be an ideal protocol. $\mathcal{P}$ realizes $\mathcal{F}$ ($\mathcal{P} \leq \mathcal{F}$) only if there exists a simulator $\mathcal{S} \in \mathsf{Sim}^{\mathcal{P}}(\mathcal{F})$ (an ideal adversary) s.t. $\mathcal{E}|\mathcal{P} \equiv \mathcal{E}|\mathcal{S}|\mathcal{F}$ for every environment in $\mathcal{E} \in \mathsf{Env}(\mathcal{P})$.

3 Post-Quantum Cryptography

The term classical cryptography is used in reference to public key crypto-systems based on the hardness assumptions from the integer factorization and discrete logarithm problems. The most prominent classical asymmetric algorithms and schemes are the *Rivest-Shamir-Adleman* (RSA) crypto-system [RSA78], the renowned *Diffie-Hellman* (DH) key agreement scheme [DH76], as well as variants using *Elliptic Curve Cryptography* such as as *Elliptic Curve Diffie-Hellman* (ECDH) and the *EC Digital Signature Algorithm* (ECDSA) [JMV01]. However, these public-key encryption (PKE) and digital signature schemes are expected to be broken by quantum computers utilizing Grover's [Gro96] and Shor's [Sho97] algorithms in the near future [Ber09, CCJ+16, KNW18]. Hence, PQC became of interest even before the start of the NIST PQC standardization process [CCJ+16] in 2016. Unlike classical cryptography, PQC is based on mathematical problems that presumably cannot be solved efficiently using quantum computers, nor classical ones for that matter [Ber09, BL17].

3.1 Cryptographic Primitives and Hardness Assumptions

Including generic KEMs, current PQC PAKEs come from two PQC algorithm families. These are *Supersingular Isogenies* (SI) e.g., *Supersingular Isogeny Diffie–Hellman* (SIDH) or *Commutative SIDH* (CSIDH), and the lattice *Learning-with-Errors* (LWE) problem and *Ring* (RLWE), *Module* (MLWE), and *Module Learning-with-Rounding* (MLWR).

SIDH and CSIDH. Generally speaking, isogenies are non-zero homomorphisms of elliptic curves (EC) [Smi18]. One may imagine that they resemble a mapping between ECs yielding a specific mathematical relation (isomorphism). SIs and CSIs are more or less special cases of isogenies. Mainly, they rely on the difficulty of identifying isogenies between ECs, i.e. curves that have the same j-invariants (if two ECs are connected by an isogeny, they will have the same j-invariant). Finding such mappings is considered a hard problem, since there can be infinitely many of them. Their attractiveness in asymmetric cryptography is that they enable building key agreement schemes that are very similar to the classical DH. They are hence used to construct the computational and decisional problems as well. However, we refrain from elaborate explanation due to a complexity beyond the scope of this paper. That being said, we refer to [Smi18] for further reading. Note that *SIKE* [CCH+19], which builds on SIDH, was a NIST round four candidate until its withdrawal due to its sudden break[3]. CSIDH [CLM+18] on the other hand is still secure.

Definition 5 (Problems on Isogenies [HPA21a])**.** Given tow elliptic curves E, E_A the SIDH and CSIDH problems are defined as follows:

- **SIDH:** Given the isogeny value $\varphi : E \rightarrow E_A$ on $E[\ell^e]$, find φ.
- **CSIDH:** Find an efficiently computable isogeny $\varphi \in \mathcal{C}\ell(\mathcal{O})$ s.t. $E_A = \varphi \cdot E$, where $\mathcal{C}\ell(\mathcal{O})$ is the class group of $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$.

LWE. A lattice is a discrete subgroup of a multidimensional vector space over real numbers under addition and can be described as a set of points in an n-dimensional space with a periodic structure. There are several classical computational problems in lattices, upon which crypto-systems are based [RHCB21], such as the *Shortest Vector Problem* (SVP) and the *Smallest Integer Solution* (SIS). The most relevant problem for PQC PAKEs is LWE (cf. Tab. 1), which was first introduced by Regev in [Reg05, Reg06]. The decisional LWE problem is basically to distinguish between random linear equations (called samples) and uniform equations, after applying a small amount of noise (called errors), as found in the Regev plain LWE crypto-system [Reg10]. LWE can also be defined on integral lattices, where the lattice base is a matrix of integers (e.g., modular lattices). Moreover, a base can be replaced by an irreducible polynomial and thus define as a special subset where all vectors form an ideal in a certain ring (e.g., ideal lattices), which was introduced in the *Lyubashevsky-Peikert-Regev* ideal RLWE crypto-system [LPR10]. The MLWE crypto-system was defined by Brakerski et al. [BGV14] and further studied by Langlois and Stehlé [LS15]. It takes the construction idea of RLWE a step further to replace the integers by a ring of algebraic integers of a number field. The most notable LWE schemes are the NIST standardization finalist *ML-KEM (CRYSTALS-Kyber)* [BDK+18] (MLWE), the round three candidates *FrodoKEM* [BCD+16] (LWE) and *SABER* [DKSRV18] (MLWR), and the RLWE key exchange scheme of Ding et al. [DXL12] (was not a NIST submission).

Definition 6 (Problems on LWE [HPA21a])**.** Given a monic polynomial $\varphi \in \mathbb{Z}[x]$ and an integer modulus q, let $\mathcal{R} = \mathbb{Z}_q[x]/(\varphi(x))$ be a ring, and $\mathbf{A} \in \mathcal{R}^{n\times m}$ be uniformly random. Further, let $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$, where $\mathbf{s} \in \mathcal{R}^n$ and $\mathbf{e} \in \mathcal{R}^m$ are sampled from secret and error distribution respectively. The **decision** LWE problem is to distinguish $(\mathbf{A}, \mathbf{b})$ from uniform, and **search** LWE is to find $\mathbf{s}$.

3.2 Key Agreement and Key Encapsulation

Current PQC PAKEs make use of two types of key agreement schemes:

1) *Key Exchange* (KEX): Protocol parties combine the exchanged key materials from their respective ephemeral public keys to create a session key. This method is found in classical DH (Fig. 3a), SIDH (Fig. 3b), and LWE (Fig. 3c).

2) *Key Encapsulation Mechanism* (KEM): Regardless of the used primitive, a receiver encapsulates a session key based on the public key of the initiator into a ciphertext, and then sends it to the initiator, who in turn decapsualtes the session key from the ciphertext using their secret key (Fig. 3d).

Generally speaking, key agreement from isogenies and lattices follow a similar concept as in DH, where both parties provide key shares through their public keys, which are (mathematically) combined to obtain the same final shared key [HPA21b]. Nevertheless,

[3] https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/sike-team-note-insecure.pdf

Alice		**Bob**
a	g, p	b
$A = g^a \bmod p$	$\xrightarrow{A}$	
	$\xleftarrow{B}$	$B = g^b \bmod p$
$K = B^a \bmod p$		$K = A^b \bmod p$

(a) DH Key Agreement

Alice		**Bob**
$k_A \in_R SK_A$	P_A, P_B, Q_A, Q_B	$k_B \in_R SK_B$
$R_A = P_A + k_A Q_A$ $\varphi_A : E \to E_A = E/\langle R_A \rangle$		$R_B = P_B + k_B Q_B$ $\varphi_B : E \to E_B = E/\langle R_B \rangle$
	$\xrightarrow{E_A, \varphi_A(P_B), \varphi_A(Q_B)}$	
	$\xleftarrow{E_B, \varphi_B(P_A), \varphi_B(Q_A)}$	
$R_{BA} = \varphi_B(P_A) + k_A \varphi_B(Q_A)$ $K_{\text{Alice}} = j(E_B/\langle R_{BA} \rangle)$		$R_{AB} = \varphi_A(P_B) + k_B \varphi_A(Q_B)$ $K_{\text{Bob}} = j(E_A/\langle R_{AB} \rangle)$

(b) SIDH Key Agreement [FTTY19]

Alice		**Bob**
	$\mathbf{A} \in R_q^{k \times k}$	
$s_a, e_a \in R_q$		$s_b, e_b \in R_q$
$\mathbf{b}_a = \mathbf{A} s_a + e_a$	$\xleftrightarrow{\mathbf{b}_b, \mathbf{b}_a}$	$\mathbf{b}_b = \mathbf{A} s_b + e_b$
$K = s_a^T \cdot \mathbf{b}_b$		$K = \mathbf{b}_a \cdot s_b$
$s_a^T(\mathbf{A} s_b + e_b)$	$\approx$	$(s_a^T \mathbf{A} + e_a^T) s_b$

(c) LWE Key Agreement

Alice		**Bob**
	KEM Parameters	
$(pk, sk) \leftarrow \text{KGen}(1^\kappa)$		
	$\xrightarrow{pk}$	
		$(c, K) \leftarrow \text{Encap}(pk)$
	$\xleftarrow{c}$	
key := Decap(sk, c)		key := K

(d) Abstract KEM Algorithm

Figure 3: High level description of key exchange and encapsulation algorithms.

this relation is not very straightforward on isogenies due to some extra information required to agree on a common curve [Smi18, HPA21b]. Further, (noisy) lattices does not produce identical final keys in non-interactive settings, as the error terms produce noisy secrets that differ on their lower bits [HPA21b]. Thus, they require a so-called (interactive) reconciliation step through signaling (or hinting), where Bob tells Alice how to round the noisy secrets [HPA21b]. Abstract KEM algorithms allow for non-interactive key agreement, since they act as a key transport scheme, where the parties are not concerned with reconciliation. For instance, lattice-based KEMs utilize PKEs that allows a KEM to produce identical shared keys with a negligible margin of error (decryption failures [HPA21b]) without reconciliation [Pei14], as opposed to reconciliation through signaling [DXL12].

4 Security of Key Agreement and Key Encapsulation

The design of PAKEs, and especially generic constructions, takes several security properties (goals) into account, which the underlying key agreement scheme must fulfill. These properties address the semantics of a KEX regarding the security of the shared key, the public key, and the ciphertext. For a KEM, similar notions also define the security of the encapsulated (or decapsulated) key, the public (encapsulating) key, and the (decapsulation) ciphertext. We note that the exact definitions and security assurances of the following notions differ across the existing literature. In this work, they solely serve the purpose of understanding their usage in PAKE designs[4].

4.1 Abstract Properties

Mainly, *Indistinguishability* is considered the minimum degree of security required for the final session key. According to [Jar22], the underlying PKE in a PAKE has to be at least IND-CPA *(Indistinguishability under Chosen-Plaintext Attacks)* secure, and thus this property is present in all PAKE constructions. However, unlike PAKEs built directly from PQC harndess assumptions (e.g., [KV09, DAL$^+$17]), the emphasis on abstract security properties is mostly found in generic PAKE constructions. This is due to the impossibility of involving the hardness assumptions in black-box reductions (cf. Sec. 10). Hence,

[4] We refer to [Poi22] for more details on formal definitions of security notions in asymmetric cryptography.

generic constructions define a set of required properties, and utilize KEMs that fulfill them in concrete instantiations. Since all NSIT PQC KEMs are considered IND-CCA *(Indistinguishability under Chosen-Ciphertext Attacks)* secure[5], recent works focus on PQC KEMs that also fulfill additional properties, which we address in the following.

Encryption (Key) Security. The *Indistinguishability* (IND) notion expresses the core security of the final session key in a KEX or a KEM. Basically, it indicates the infeasibility for an attacker to differentiate (distinguish) real, honestly generated values, from random ones [GM84]. The *One-wayness* (OW) property plays a similar role, and denotes the irreversibility of an encryption, i.e., the infeasibility of recovering a value from its corresponding encryption [ABP15].

Public Key Security. Public key *Uniformity or Fuzziness* (PKU) denotes the indistinguishability of real generated public keys from uniform ones, i.e., honestly generated public keys have an equal probability distribution to random ones from the same public key space [BCP$^+$23, AHHR24]. Further, public key *Anonymity* (ANO) expresses the public key privacy, i.e., the difficulty of utilizing information from a ciphertext to reveal which public key was used to create it [GMP22, CDM23]. Some works refer to the anonymity of a ciphertext, and not of a public key (e.g., [BCP$^+$23, ABJS24]).

Ciphertext Security. The ciphertext *Robustness* (ROB) and *Collision Freenes* (CFR) notions denotes the binding property of a ciphertext / secret key relationship, i.e., a ciphertext does not decrypt to a valid plaintext for two distinct secret keys [GMP22, CDM23]. Further, ciphertext *Pseudo Randomness* (PR) denotes the indistinguishability of ciphertext and shared key pairs from random ones. i.e., it is infeasible to distinguish between said pairs under knowledge of the plaintext message and the public key used for encryption [Xag22, CDM23]. Finally, *Non-malleability* (NM) expresses the infeasibility to construct ciphertexts that are meaningfully related to a known ciphertext and its initailly corresponding plaintext [DDN91].

Adversarial Capabilities (in Related Security Experiments). Multiple variants of the afore mentioned properties can be obtained under a variety of attacks, which define the capabilities of an adversary interacting with challenges constructed from said notions, and hence provide different degrees of security:

- *Plaintext Checking Attack* (PCA): Adversaries have access to a plaintext-checking oracle (PCO) that answers queries as to whether given plaintext-ciphertext pairs correspond to each other [OP01].

- *Chosen Plaintext Attack* (CPA): Adversaries can generate arbitrary valid ciphertexts from plaintexts, of their choosing, using an honest public key [GM84].

- *Non-adaptive Chosen Ciphertext Attack* (CCA1): Same as CPA, but adversaries additionally have access to a decryption oracle before accessing the challenge ciphertext, i.e., adversaries may choose a set of ciphertexts distinct from the challenge ciphertext and query the decryption oracle to obtain the corresponding plaintexts [NY90].

- *Adaptive Chosen Ciphertext Attack* (CCA2): Adversaries have permanent access to a decryption oracle that only disallows querying the challenge ciphertext itself to directly obtain the corresponding plaintext [RS91].

[5] NIST PQC KEMs apply the generic Fujisaki-Okamoto (FO) transform [FO99] to lift a PQC PKE from CPA to CCA2 security [Unr20] as required in the NIST process.

On this basis, it is possible to obtain different notions of security by combining goals with attack models. There exist a multitude of pathways in which these notions relate to each other (i.e., implications and separations) as shown in [BDPR98, OP01, ABP15, CDM23]. For instance, IND-CPA provides a weaker, yet more efficient alternative to IND-CCA for scenarios where the ability to check plaintexts is sufficient for the application's security requirements. As a result, IND-CCA implies IND-CPA and OW-PCA. Further, a cryptosystem may exhibit semantic security against CPA, or even CCA1, yet it may remain malleable if does not satisfy ANO-CCA. However, satisfying CCA2 combined with ANO-CCA is equivalent to achieving pseudo-randomness and non-malleability [JRX24, ABJ25].

4.2 KEM Security

Considering the increasing interest in abstract KEM security, we provide definitions for KEM security properties addressed in recent generic PAKEs (cf. Sec. 7.6). The respective security experiments (games) for the following notions are illustrated in Fig. 4.

Definition 7 (Key Uniformity of KEMs [AHHR24])**.** For a key encapsulation mechanism KEM with public key space $\mathcal{PK}$, we define the advantage of an adversary $\mathcal{A}$ in distinguishing honestly generated public keys from uniform random ones as $\mathsf{Adv}^{\mathsf{PKU}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$.

Definition 8 ($\mathsf{IND\text{-}CCA}$ security of KEM [AHHR24])**.** For a key encapsulation mechanism KEM with session key space $\mathcal{K}$, define the advantage of an adversary $\mathcal{A}$ in distinguishing genuinely encapsualted session keys from uniform random ones as $\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$.

Definition 9 ($\mathsf{OW\text{-}PCA}$ security of KEM [ABJS24])**.** For a key encapsulation mechanism KEM with session key space $\mathcal{K}$, define the advantage of an adversary $\mathcal{A}$ in recovering an encpasualted key from known public key and ciphertext as $\mathsf{Adv}^{\mathsf{OW\text{-}PCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$.

Definition 10 ($\mathsf{ANO\text{-}CCA}$ security of KEM [AASA+24])**.** For a key encapsulation mechanism KEM with public key space $\mathcal{PK}$, define the advantage of an adversary $\mathcal{A}$ in distinguishing public keys used to probabilistically encapsulate keys into ciphertexts as $\mathsf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ ANO-PCA is obtained by replacing the decryption oracle with a PCO.

Definition 11 ($\mathsf{SCFR\text{-}CCA}$ security of KEM (also $\mathsf{SROB\text{-}CCA}$) [AASA+24])**.** For a key encapsulation mechanism KEM with public key space $\mathcal{PK}$ and session key space $\mathcal{K}$, define the advantage of an adversary $\mathcal{A}$ in probabilistically generating a ciphertext C that decapsualtes correctly under two unique public keys and their corresponding secret keys as $\mathsf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ Robustness (SROB-CCA) is identical except for the changed last line.

Generalized Key Agreement. McQuoid et al. [MRR20] and Januzelli et al. [JRX24] define similar security properties, however w.r.t. any key agreement (KA) scheme used in an instantiation of EKE-style PAKE protocols:

- KA security: Equivalent to indistinguishability of final session key.
- First Pseudo-Randomness (PR): Equivalent to PK).
- Second Pseudo-Randomness (PR): Similar to wANO-CCA.
- Strong Pseudo-Randomness (SPR): Similar to ANO-CCA.
- Pseudo-Random Non-Malleability (PR-NM): Equivalent to strong pseudo-randomness (SPR-CCA) as defined in [Xag22], or simultaneously IND-CCA and ANO-CCA.
- Collision-Resistance (CR): Similar to SCFR-CCA.

```
$\mathsf{Exp}^{\mathsf{PKU}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $pk_1 \leftarrow\!\!\$\ \mathcal{PK}$
    $b' \leftarrow \mathcal{A}(pk_b)$
    return $b = b'$
```

```
$\mathsf{Exp}^{\mathsf{IND\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(C^*, K_0) \leftarrow \mathsf{Encap}(pk)$
    $K_1 \leftarrow\!\!\$\ \mathcal{K}$
    $b' \leftarrow \mathcal{A}^{\mathsf{D}(sk,pk,.)}(pk, C^*, K_b)$
    return $b = b'$
```

```
$\mathsf{Exp}^{\mathsf{OW\text{-}PCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(C^*, -) \leftarrow \mathsf{Encap}(pk)$
    $K \leftarrow \mathcal{A}^{\mathsf{PCO}_\perp(sk,.,.)}(pk, C^*))$
    return $K = \mathsf{Decap}(sk, C^*)$
```

```
$\mathsf{Exp}^{\mathsf{ANO\text{-}PCA}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(pk_1, sk_1) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(C^*, -) \leftarrow \mathsf{Encap}(pk_b)$
    $b' \leftarrow \mathcal{A}^{\mathsf{PCO}_{C^*}(.,.)}(pk_0, pk_1, C^*))$
    return $b = b'$
```

```
$\mathsf{Exp}^{\mathsf{ANO\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(pk_1, sk_1) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(C^*, K^*) \leftarrow \mathsf{Encap}(pk_b)$
    $b' \leftarrow \mathcal{A}^{\mathsf{D}(.,.)}(pk_0, pk_1, (C^*, K^*))$
    return $b = b'$
```

```
$\mathsf{Exp}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(pk_1, sk_1) \leftarrow \mathsf{KGen}(1^\kappa)$
    $C \leftarrow \mathcal{A}^{\mathsf{D}(.,.)}(pk_0, pk_1)$
    $K_0 \leftarrow \mathsf{Decap}(pk_0, sk_0, C)$
    $K_1 \leftarrow \mathsf{Decap}(pk_1, sk_1, C)$
    return $K_0 = K_1 \neq \perp$
    return $K_0 \neq \perp \wedge K_1 \neq \perp$
```

Figure 4: KEM Security Experiments - Adopted from [GMP22, AASA+24, ABJ25]. CPA versions can be obtained by removing decryption oracles D, and weaker versions (e.g., wANO or wCFR) are obtained by not revealing the encapsulated key K.

Splittable KEM. Arriaga et al. [ABJS24] and Alnahawi et al. [AASA+24] address specific properties w.r.t. lattice-based KEMs with splittable public keys (splittable KEMs):

- Splittable and pseudo-uniform public keys (UNI-PK) [ABJS24]: Similar to PKU.
- Splittable public key uniformity (SPLIT-PKU) [AASA+24]: Similar to PKU.
- A-Part-Secrecy (A-SEC-CCA) [AASA+24]: Similar to ANO-CCA.
- A-Part-Collision Freeness (A-CFR-CCA): Similar to SCFR-CCA.

5 Supporting Cryptographic Building Blocks

Mainly, *Smooth Projective Hash Functions* (SPHF), *Oblivious Pseudo Random Functions* (OPRF), and *Non-Interactive Zero-Knowledge* proofs (NIZK) serve as supporting components in PAKEs. That is, either in establishing a trusted setup, or in realizing commitment schemes w.r.t. users and public keys. For instance, one of the earliest PQC PAKEs leveraging such building blocks is the LWE PAKE by Katz and Vaikuntanathan [KV09], which adapts classical *KOY-GL* PAKEs (Katz, Ostrovsky, and Yung [KOY01], and Gennaro and

Lindell [GL03]) however using *Approximate* SPHs (ASPH) from LWE (cf. Sec. 7.2). An example of the usage of OPRF in PAKEs is the augmented *OPAQUE* protocol [JKX18], which was instantiated in PQC setting using an isogeny-based OPRF in [HHM+24]. Last but not least, NIZKs have also appeared in classical PAKEs such as *J-PAKE* [HR10], which employs the Schnorr protocol [Sch91]. Nevertheless, there have been sofar no attempts at realizing J-PAKE using PQC primitives or KEMs.

5.1 Smooth Projective Hash Functions

Originally, Cramer and Shoup introduced the SPHFs [CS02] in order to obtain hash proof systems with IND-CCA security. Gennaro and Lindell proposed a generalized SPHF [GL03] for its many attractive properties and purposes such as implicit designated verifier proofs of membership. The use of SPHFs in PAKEs was presented by Katz, Ostrovsky, and Yung [KOY01] and also Gennaro and Lindell [GL03], which is known as the *KOY-GL* paradigm. The basic idea is that an SPHF defined over a language allows to hash a value (word) in two different ways, with the hashing key (secret key), or with the associated projection key (public key). If the word is in the language, both ways of hashing will yield the same hash value. However, if it is outside the language, the hash obtained using the secret must be statistically indistinguishable from a random value, even with knowledge of the public key. Hence, computing the projection key via hashing a value in the language does not reveal any information about the hash for a random value outside the language.

Definition 12 (Projective Hash Function [GL03])**.** The family $(\mathcal{H}, K, X, L, G, S, \alpha)$ is a projective hash family if for all $k \in K$ and $x \in L$, it holds that the value of $H_k(x)$ is uniquely determined by $\alpha(k)$ and x.

Definition 13 (Smooth Projective Hash Function [GL03])**.** Let $(\mathcal{H}, K, X, L, G, S, \alpha)$ be a projective hash family. Then let $V(x, \alpha(k), H_k(x))$ be the following random variable: choose $x \in X \setminus L$ according to $D(X \setminus L)$, $k \in_R K$ and output $(x, \alpha(k), H_k(x))$. Similarly, define $V(x, \alpha(k), g)$ as follows: choose $x \in X \setminus L$ according to $D(X \setminus L)$, $k \in_R K$, $g \in_R G$ and output $(x, \alpha(k), g)$. Then the projective hash family $(\mathcal{H}, K, X, L, G, S, \alpha)$ is smooth if

$$\Big\{V(x, \alpha(k), H_k(x))\Big\}_{n\in\mathsf{N}} \overset{s}{\equiv} \Big\{V(x, \alpha(k), g)\Big\}_{n\in\mathsf{N}}$$

Definition 14 (Approximate Smooth Projective Hash Function [KV09])**.** An approximate smooth projective hash function is a collection of keyed functions $H_k : X \to 0, 1^n{}_{k\in K}$, along with a projection function $\alpha : K \times (0, 1^* \times \mathcal{C}) \to S$, satisfying notions of (approximate) correctness and smoothness:

- **Approximate correctness:** If $x = (\text{label}, C, m) \in \overline{L}$, then the value of $H_k(x)$ is approximately determined by $\alpha(k, \text{label}, C)$ and x.
- **Smoothness:** If $x \in X\ L$ then the value of $H_k(x)$ is statistically close to uniform given $\alpha(k, \mathit{label}, C)$ and x (assuming k was chosen uniformly in K).

5.2 Oblivious Pseudo-Random Functions

The first *Pseudo-Random Function* (PRF) by Goldreich et al. [GGM86] dates back to 1986, and 18 years later, Naor and Reingold introduced the idea of interactive and oblivious evaluation of such functions [NR04]. Based on the Naor-Reingold PRF, Freedman et al. formalized the definition of the first two-party protocol known as an *Oblivious* PRF [FIPR05]. The general idea of an OPRF is to associate a keyed PRF with a protocol execution between a server and a user. The server holds the key for the function and the user provides an input to that function. The main catch is that the user can learn the output of the PRF at the end of the execution, whereas the server does not learn anything about the input of the user. We refer the reader to [CHL22] for a complete overview on the evolution, applications and different types of OPRFs. *OPAQUE* [JKX18] for instance, utilizes the verifiable hash DH-OPRF as proven secure by Jarecki et al. in [JKK14, JKKX16]. As far as post-quantum security is concerned, the majority practical OPRF instantiations rely on classical hardness assumptions [CHL22], with the exception of ones based on symmetric primitives and garbled circuit. Nevertheless, many recent works provide OPRF constructions based on PQC such as OT adn NIZKs [FOO23, BDFH24], Legendre PRFs [SHB21, KCM24, YBH$^+$24], lattices [ADDS21, ADDG24] and isogenies [BKW20, Bas23, HHM$^+$24, DdSGP24, BM25]. We refer to [HHM$^+$24] for a comprehensive and thorough overview on PQC OPRFs[6]. Currently, there are no PQC PAKE construction utilizing quantum-secure OPRFs that provide a full formal proof.

Definition 15 (Pseudorandom Function [HHM$^+$24])**.** A Pseudorandom Function is a deterministic and polynomial time function $F : 0, 1^k \times 0, 1^x \to 0, 1^n$ such that there is no probabilistic polynomial-time algorithm to distinguish any output N from a randomly chosen element from $0, 1^n$.

Definition 16 (Oblivious Pseudorandom Function [HHM$^+$24])**.** An Oblivious Pseudorandom Function is a protocol between two parties. One holds the secret key K and the other holds their secret input X. The OPRF privately realizes the joint computation outputting $F(K, X)$ for a PRF F to the party holding X, and nothing to the party holding K.

5.3 Non-Interactive Zero-Knowledge Proofs

Zero-Knowledge Proofs (ZKP), first introduced by Goldwasser et al. [GMR85], are cryptographic constructions that enable one party (a prover) to exhibit its knowledge of a certain property to another party (a verifier), without revealing the property itself. In other words, ZKPs allow sharing a proof of holding a specific secret without sharing the actual secret. In PAKEs, this property or secret could be the long lived key (e.g., the password) or an honest public key pair. While interactive proofs require multiple rounds of interaction, NIZKs, first introduced by Blum et al. [BFM88], require only one message from a prover to a verifier, as they make use of a *Common Reference String* (CRS) for an initial setup. Benhamouda et al. [BBDQ18] provide an overview of lattice-based hash proof systems and discuss how to construct 3, 2, and 1-round PAKEs using ASPHs and NIZKs, which we recommend for further reading.

Definition 17 (Non-Interactive Zero-Knowledge Proof System [WW14])**.** For a pair of probabilistic Turing machines (P, V), in which P is probabilistic polynomial time and V is deterministic polynomial time, (P, V) is called the Non-Interactive Zero Knowledge proof system for language L if it provides:

- **Completeness:** For any common input $x \in L$ and polynomial $p(.)$,
$$Pr[V(x, R, P(x, R)) = 1] \geq 1 - \frac{1}{p(|x|)}$$
- **Soundness:** For any common input $x \notin L$, any interactive Turing machine P', and polynomial $p(.)$,
$$Pr[V(x, R, P'(x, R)) = 1] < 1 - \frac{1}{P(|x|)}$$
- **Zero knowledge:** For any $x \in L$, there is a probabilistic polynomial time algorithm M such that
$$V(x) = (x, R \in 0, 1^{c(|x|)}, P(x, R)) \approx \ _cM(x)_{x \in L}$$

[6] https://heimberger.xyz/oprfs.html provides a regularly updated list of PQ friendly OPRFs.

6 Security Models and Idealized Objects

6.1 Standard Model

The standard model enables proofs based solely on complexity (hardness) assumptions. That is, the attacker's capabilities are only limited by their computational power, i.e., an arbitrary polynomial-time machine [CGH04]. However, idealized models are often used, as they allow for additional complexity-theoretic hardness assumptions [Bla06].

6.2 Random Oracle Model

A *Random Oracle* (RO) is an ideal primitive that models a random hash function that responds to each query to a given fixed-length input value with a corresponding random output [BR93]. Additionally, a RO keeps a record of all placed queries, and responds with the same value for a previously queried input. Basically, the ROM is present in almost all PAKEs, since its usage is required to model hash and key derivation functions. Nearly all ROM-based PAKEs are analyzed in classical settings (cf. Tab. 1), and only two constructions provide security proofs in the *Quantum* ROM (QROM).

Definition 18 (Random Oracle)**.** *RO* is a function f that maps elements over the function space $\{0,1\}^* \to \{0,1\}^{poly(1^\kappa)}$, where:

$$\textbf{if } f(x) \neq \perp \textbf{ return } f(x) \textbf{ else return } y \leftarrow\!\!\$\, \{0,1\}^{poly(1^\kappa)}$$

QROM. Boneh et al. [BDF+11] proposed the QROM and addressed the issues with the classical ROM in the presence of adversaries with quantum capabilities, who may evaluate hash functions in superposition [BCMR19].

Definition 19 (Quantum(-Accessible) Random Oracle [BDF+11])**.** Evaluating a RO in superposition by submitting a quantum state s.t. $|\varphi\rangle = \sum \alpha_x |x\rangle$ to an oracle $\mathcal{O}$ an receiving the evaluated state $\sum \alpha_x |\mathcal{O}(x)\rangle$ is called a quantum-accessible random oracle.

Many ROM techniques were not directly applicable in the QROM as per quantum information fundamental concepts; such as adaptive programmability, extractability, and rewinding [BDF+11, BCMR19]. Thus, it was not clear, whether classical proof techniques in the ROM [BDF+11, BCMR19] also hold in the QROM. Nevertheless, recent works clarified that many ROM constructions can also be shown secure in the QROM [Zha19, YZ21, DFMS22, HM24]. In the context of PAKE formal analysis, the most important QROM proof techniques relate to bounding collisions on search queries and extracting inputs to hash functions [LLH24, HHKR25]. For instance, and considering the *no-cloning principle*, it was not possible to keep record of query transcripts (also referred to as the *recording barrier*), which renders searching for a certain query input to a RO infeasible [DFMS22]. However, *Compressed Oracles* (CO) [Zha19] overcame the recording barrier by allowing efficient on-the-fly simulation of random oracles, similar to the classical simulations [Zha19]. This technique was shown to enable proofs for applications involving pre-image search and collision finding [Zha19]. The *Extractable* CO (eCO) [DFMS22] on the other hand, takes this idea a step further by giving a form of observable QROM, where one can check for queries that satisfy some function [HHKR25]. Hence, the eCO allows the extraction of RO based commitments and simulating routines involving RO queries (e.g. decapsulation in an FO-KEM or hash functions in PAKEs [DFMS22, HM24, HHKR25]). Nevertheless, there are still no proofs in UC involving the QROM [MX23b].

6.3 Ideal Cipher Model

An *Ideal Cipher* (IC) serves modeling block ciphers (e.g., AES) as idealized objects similar to hash functions in the ROM with some exceptions [Bla06]. Its main advantage is defining the behavior of a cipher, where each encryption maps to an independently random permutation that belongs to the same set of possible input values. An IC provides oracle access for forward queries on encryption, and for backward queries on decryption as well, all of which are recorded. IC in PAKEs is mostly found in the design class C1 (cf. Tab. 1), most of which also address the absence of a *Quantum* IC model.

Definition 20 (Ideal Cipher)**.** *IC is an invertible permutation function* $\mathcal{C} : \{0,1\}^{\kappa} \times \{0,1\}^{n} \rightarrow \{0,1\}^{n}$, where each key $k \in \{0,1\}^{\kappa}$ defines a unique and independent random permutation $\mathcal{C}_k = \mathcal{C}(k, .)$ on $\{0,1\}^{n}$.

QIC. Many works in the literature [AR17, HY18, CMSZ19, SS19, Unr21, Ros21, ABPS23, Unr23, MMW24] address the notion of the *Quantum* IC model (QICM), yet do not show how to fully obtain the capabilities of a classical IC. The work by Unruh [Unr23] builds upon the idea of COs and takes a step forward in modeling keyed invertible permutations (i.e., IC) in quantum settings by introducing the *Compressed Permutation Oracle* (CPO). Nevertheless, and despite the novelty of the proposed approach, it is not yet formally proven that a CPO is indistinguishable from a truly random permutation [Unr23]. A recent breakthrough in realizing the QIC is the *Permutation Superposition Oracle* (PSO) introduced by Majenz et al. [MMW24]. The PSO can be considered similar to Zhandry's CO, but additionally allowing to (approximately) determine queried inputs to permutation based on the output [MMW24]. Nevertheless, these works aid sofar only in bounding collisions on search queries, yet do not enable programming and extraction.

Definition 21 (Quantum(-Accessible) Random Permutation [MMW24])**.** For a group of elements $[N] = \{1, 2, ..., N\}$, the permutation (bijection) group on N elements S_N, a permutation $\pi \in S_N$, and the corresponding unitary permutation operator $U^{\pi}|x, y\rangle = |x, y \oplus \pi(x)\rangle : \forall x, y \in [N]$ for the quantum space $\mathbb{C}^N \otimes \mathbb{C}^N$, a Quantum-Accessible Random Permutation consists of query access to U^{π} and to $U^{-\pi}$, for a permutation $\pi \in S_N$ chosen uniformly at random.

6.4 Half Ideal Cipher

Dos Santos et al. [DGJK22] introduced a relaxation of the IC model called *Half Ideal Cipher* (HIC), which is realized through a modified 2-Feistel construction (m2f) as shown in Fig. 9. The m2f extends the IC domain to bit strings using a block cipher and hash functions. This construction also inspired the work by Arriaga et al. [ABJS24], where a compact m2f is used over uniform bit-strings. HIC PAKEs also belong to class C1.

Definition 22 (Modified 2-Round Feistel Construction [ABJS24])**.** A modified 2-round Feistel (m2f) is constructed from: 1. A block cipher denoted by the tuple $(\mathsf{IC.Enc}, \mathsf{IC.Dec})$, wiht key space $\mathcal{K}$ and input/output space N, 2. a hash function $\mathcal{H}_1$ with output space G, 3. a hash function $\mathcal{H}_2$, with output space $\mathcal{K}$, 4. and two efficiently computable functions $\mathsf{m2F}_{\pi} : N \times G \rightarrow N \times G$ and its inverse $\mathsf{m2F}_{\pi}^{-1}$ (cf. Sec. 7.4 Fig.9).

6.5 Programmable-Once Public Function

McQuoid et al. [MRR20] introduced an adaptation of an Oblivious Programmable Pseudo-Random Function (OPPRF) [KMP+17] called the Programmable-Once Public Function (POPF). An initiator can generate a value based on a keyed function and make the evaluation of the function public for any input in non-interactive manner. Hence, POPF serves as an alternative to an IC on keyed permutations, however relying on hash functions [ABJ25]. McQuoid et al. [MRR20] showed that a 2-Fesitel (2f) construction realizes POPF in the ROM and used it in a game-based PAKE proof and Januzelli et al. reformulated POPF in a UC PAKE, which was also adopted by Arriaga et al. [ABJ25].

Definition 23 (Programmable-Once Public Function [ABJ25])**.** For a keyed function family $F_\varphi : X \to Y$ Programmable-Once Public Function (POPF) is a pair of sufficient algorithms $\mathsf{Program}$ and Eval s.t.:

- $\mathsf{Program}$ is a randomized algorithm s.t. $\varphi \leftarrow\$\ \mathsf{Program(x,y)}$ fixes a POPF isntance F_φ, a random function constrained by $F_\varphi(x) = y$.
- Eval is a deterministic algorithm that evaluates the POPF instance on arbitrary inputs, i.e. $\mathsf{Eval}(\varphi, x) = F_\varphi(x)$.

Definition 24 (2-Round Feistel Network [ABJ25])**.** A 2-Feistel (2f) is constructed from: 1. A hash function $\mathcal{H}_1$ whose output space is an algebraic group G, 2. a hash function $\mathcal{H}_2$ whose output space is a set N of fixed-length bit strings used in sampling the randomness r, 3. and two efficiently computable functions $\mathsf{2F}_k : N \times G \to N \times G$ and its inverses $\mathsf{2F}_k^{-1}$ (cf. Sec. 7.4 Fig. 10).

6.6 Common Reference String

The *Common Reference String* (CRS) was introduced by Blum et al. [BFM88] to realize NIZKs by providing a shared string accessible to and trusted by all protocol parties. Usually, a CRS is drawn from a pre-defined uniform distribution (randomness source) that does not reveal any information about the way the string is generated. A *Structured Reference String* (SRS) is a variant of a CRS where the string is structured, which is mainly used in NIZK design. In the context of PAKEs, a pre-shared public key of the protocol initiator is used as a CRS in a trusted-setup (with a prior registration phase). Most CRS PQC PAKEs constructions make use of SPHFs and ASPHs or NIZKs such as [KV09, ZY17, BBDQ18, KAA19], and belong to the design class C3. Note that CRS (often) implies security in the standard model.

RO or CRS. In 2008, Groth and Ostrovsky [GO14] introduced the *Multi-String* (Multi-CRS) model to mitigate the risks of having only one trusted authority generating a random string. Following this work, other methods and ideas were developed to distribute trust among multiple setups as discussed in [XZZ24]. The *CRS-or-RO-Setup* (CoR) was unified in 2014 by Katz et al. [KKZZ14], who showed the impossibility of obtaining a secure construction from a straightforward setup combining one CRS and one RO [XZZ24]. While this result also holds for PAKEs, Xiao et al. [XZZ24] showed how to maximize the utility of a such setup by extending the model to a so called *Fine Grained* CoR-Setup, where either the CRS or the RO may fail, yet it remains possible to build a secure PAKE. Still, this type of setup has no representatives in the PQC realm.

7 PQC PAKEs

7.1 Classification

The classification is based on the way the password is used to provide authentication, which we refer to as the design paradigm. The most important aspect to observe here is the used representation of the password, and how it is (cryptographically) applied to protect the asymmetric key agreement scheme. Generally, the password is rarely used in its bare form, but rather as input to an Extended Output Function (XOF) to obtain a fixed-length value (e.g., obtaining an AES key using a Key Derivation Function (KDF)), or as a group element (e.g., a number) in an operation on the underlying number theoretic assumptions. Further, we observe whether a trusted setup is required in a client-server model. We adopt the classification system in [HvO22], and adapt it for PQC PAKEs:

C1) **Password Encrypted Public Key** (aka *EKE-style*): The password (or a password-derived symmetric key) is used directly to encrypt or mask the public key. e.g., *(O)CAKE* [BCP+23], *CHIC* [ABJS24], and *NICE-PAKE* [AASA+24]. The classical EKE design protects an asymmetric (e.g., DH) key agreement through encrypting the public keys of the communication parties. The encryption is done using a symmetric key (e.g. AES), where the blockcipher is modeled as an IC, and hash functions and KDFs are modeled as ROs. The core session key security (IND-CPA) is then achieved based on the hardness of DH. The CAKE protocol is built similar to EKE2, however using a KEM instead of DH for the key agreement (cf. Fig. 8).

C2) **Password Modified Public Key** (aka *PAK-style*): The password is used to modify the public key preserving the underlying hardness assumption of the key exchange primitive. e.g., *RLWE-PAK* [DAL+17] and *SIDH-PAK* [TSJL20]. The classical PAK design shifts the value of the public keys (e.g. a DH group element) through multiplication with the hash of the password. The RO-modeled hash value is then interpreted as a valid group element and hence it is called Hash-to-Group (H2G), and the modified public key is a also valid group element. Here as well, the session key security is based on the hardness of DH. The RLWE-PAK protocol utilizes the lattice RLWE crypto-system to realize this design by adding the password hash to an LWE sample, which works almost identically as in a regular DH (cf. Fig. 5).

C3) **Trusted Setup**: Multiple approaches, where predefined trusted parameters are required to achieve authentication. e.g., *KV-PAKE* [KV09] and *GA-PAKE* [AEK+22a]. Both C1 and C2 designs can be combined with a previously established parameter known only to communication parties to realize a trusted-setup, where the authentication is verified based on said pre-established value (e.g., a CRS). However, it is sofar only found in PAKEs following C2 (e.g., RLWE-PAK). Other variations, not based on C2, also exist, and rely on more complex supporting cryptographic building blocks (e.g., SPHFs and NIZKs), or isogenies (cf. Fig. 6 and Fig. 7).

Although [HvO22] introduced five design classes, only classes C1, C2, and C3 apply to the current PQC PAKEs, as classes C4 and C5 do not (yet) have any representatives in the PQC realm (cf. Sec. 9). Moreover, some works could be categorized into more than one class at the same time (e.g., a RLWE-PAK in a trusted setup). Generic PAKE constructions may be considered a class of their own, however, they also adhere to the same classes w.r.t. their password usage. For instance, both CAKE and NoIC are generic, yet CAKE relies on blockciphers (IC) to encrypt public keys, whereas NoIC utilizes an H2G in the ROM to modify them.

7.2 Lattice PAKEs

C2: PAK-Style with (R)LWE and Variants in the ROM.

Ding et al. [DAL$^+$17] proposed the first RLWE PAKE based on Ding's RLWE KEX and reconciliation mechanism [DXL12]. The PAKE is inspired by MacKenzie's classical PAK and PKK [Mac02] and follows a similar security analysis. The variant following PAK is a 3-pass (i.e., 3 messages), and provides explicit MA. The other variant following PPK is a 2-flow and provides implicit MA. The authors also introduce the *Pairing with Errors* (PWE) and the decisional (D)PWE problems relying mainly on the H2G method for the public key authentication, which can be reduced to the RLWE problem. In other words, the hash value of the password is added to the public key, which results in shifting (masking) the RLWE sample to a new one within the same lattice. The RLWE-PAK design witnessed numerous adaptations and modifications in follow-up works in the literature [GDL$^+$17, JZ16, GDLL17, JGH$^+$20, RGW23, BSIA23, SA23, YZYW25].

- *Observation 1:* Ding's RLWE KEX strongly resembles the classic DH, which enables a high level of flexibility in its usage in PAKE design.
- *Observation 2:* Ding's RLWE KEX requires signaling for key reconciliation, which is a critical attack surface (e.g., offline dictionary and impersonation attacks), if the signal is sent in plain text [DCQ22, CL24].
- *Observation 3:* Utilizing less expensive LWE variants (e.g., MLWR) and implementation optimizations lead to performance gains.

C3 Adaptations.

The protocol was also adapted to the *3-party setting* (i.e., client-server-client) in [XHCC17, CAK$^+$18], and to *augmented 3-party* in [LZJY19, GSG$^+$23] with prior registration. Another adaptation of RLWE-PAK is to the *anonymous-augmented* (mostly 3-party) setting, focusing on user registration, login, and password updates for mobile devices with key reuse as in [FHZ$^+$18, DBK20, IB21, CKS23, KGKD23, DJRD23, DRD$^+$23].

- *Observation 4:* Many of the RLWE PAKEs are vulnerable to signal-leakage, key-reuse, and pre-computation attacks [DCQ22, CKS23, DRD$^+$23, CL24].
- *Observation 5:* Using reconciliation without signaling mitigates the risk of signal-leakage attacks but may induce decryption failures [LZJY19, GSG$^+$23].
- *Observation 6:* RLWE-PAK and adaptations do not provide proofs in the QROM, although all of the PAKEs rely mainly on a RO-modeled H2G.

Technical Description. The 3-pass RLWE-PAK protocol (Fig. 5) resembles the original DH-based PAK protocol (Fig. 13). The client initiates the protocol by randomly sampling s_C, e_C (the RLWE secret and error terms) and computing α (the RLWE public key), γ (the hashed password), $m = \alpha + \gamma$ (the shifted RLWE sample i.e., public key), and finally sending $< C, m >$ to the Server. The server verifies if $m \in R_q$, aborting if the check fails. It then similarly samples s_S, e_S, computes μ and recovers $\alpha = m + \gamma'$. In the following steps, the server computes k_S (DH-like key agreement) to finally compute $w = \mathrm{Cha}(k_S) \in \{0,1\}^n$ and $\sigma = \mathrm{Mod}_2(k_S, w)$. The server sends μ, w and k to the client, who in turn checks if $\mu \in R_q$, aborting if the check fails. Else, the client computes k_C and σ. Both client and server check if the pre-keys match using two hash functions. Finally, they can derive the session key $sk_C = sk_S = H_4(C, S, m, \mu, \sigma, \gamma')$.

Client $\mathcal{C}$		**Server** $\mathcal{S}$
Input $\mathcal{S}, pw$		$\gamma' = -\mathrm{H}_1(pw_{\mathcal{C}})$
Sample $s_{\mathcal{C}}, e_{\mathcal{C}} \leftarrow \mathcal{X}_\beta$		Sample $s_{\mathcal{S}}, e_{\mathcal{S}} \leftarrow \mathcal{X}_\beta$
$\alpha = as_{\mathcal{C}} + 2e_{\mathcal{C}} \in R_q$		$\mu = as_{\mathcal{S}} + 2e_{\mathcal{S}} \in R_q$
$\gamma = \mathrm{H}_1(pw_{\mathcal{C}})$		Abort if $m \notin R_q$
$m = \alpha + \gamma$	$\xrightarrow{<\mathcal{C},m>}$	$\alpha = m + \gamma'$
Abort if $\mu \notin R_q$		$k_{\mathcal{S}} = \alpha s_{\mathcal{S}}$
$k_{\mathcal{C}} = s_{\mathcal{C}}\mu$		$w = \mathbf{Cha}(k_{\mathcal{S}}) \in \{0,1\}^n$
$\sigma = \mathbf{Mod}_2(k_{\mathcal{C}}, w)$	$\xleftarrow{\mu,w,k}$	$\sigma = \mathbf{Mod}_2(k_{\mathcal{S}}, w)$
$\gamma' = -\gamma$		$k = \mathrm{H}_2(\mathcal{C}, \mathcal{S}, m, \mu, \sigma, \gamma')$
		$k'' = \mathrm{H}_3(\mathcal{C}, \mathcal{S}, m, \mu, \sigma, \gamma')$
Abort if $k \neq \mathrm{H}_2(\mathcal{C}, \mathcal{S}, m, \mu, \sigma, \gamma')$		
else $k' = \mathrm{H}_3(\mathcal{C}, \mathcal{S}, m, \mu, \sigma, \gamma')$	$\xrightarrow{k'}$	Abort if $k' \neq k''$
$sk_{\mathcal{C}} = \mathrm{H}_4(\mathcal{C}, \mathcal{S}, m, \mu, \sigma, \gamma')$		$sk_{\mathcal{S}} = \mathrm{H}_4(\mathcal{C}, \mathcal{S}, m, \mu, \sigma, \gamma')$

Figure 5: RLWE-PAK [Din17]

C3: KOY-GL-Style with (A)SPHFs and NIZKs (mostly with CRS).

The PAKE by Katz and Vaikuntanathan [KV09] is most likely the first PQC PAKE built from an LWE PKE. The authors showed how a modification of the *Genarro-Lindell* (GL) framework [GL03] is used to construct a PAKE from an IND-CCA encryption scheme and an associated ASPH. The protocol consists of three messages and relies on a CRS for a trusted-setup using an LWE public key combined with an ASPH. The protocol parties exchange CCA secure ciphertexts, from which they attempt to compute ASPH values and execute a reconciliation to derive a final key. The reconciliation first extracts a bit from the noisy ASPH value and then applies an *Error-Correcting Code* (ECC) to finish the reconciliation. This PAKE inspired several CRS adaptations using different variations of LWE crypto-systems, (A)SPHFs, and NIZKs (or without NIZKs) [ZY17, KAA19, LW18, LW19, YGS+20, LWM22] and in *augmented* settings [TLZ+21, LWM22].

- *Observation 1:* Most KOY-GL-style PAKEs provide security proofs in the CRS model, and do not require additional idealized assumptions.
- *Observation 2:* Additional supporting cryptographic building blocks (e.g., ASPHs and NIZKs) infer more design complexity and practical inefficiency.

Technical Description. The 3-pass KV-LWE-PAKE (Fig. 6) employs an SPHF similar to the KOY-PAKE [KOY01] (Fig. 14). First, the client runs a key generation algorithm to generate the verification key and secret (signing) key pair (VK, SK). They proceed by setting a label and encrypting the password w to obtain the ciphertext $C := \mathrm{Enc}_{pk}(\mathrm{label}, w; r)$. After receiving the message (Client|VK|C), the server computes its own encryption of the password to obtain the ciphertext C'. It continues by choosing a random hash key and computing the projection $s' := \alpha(k', \mathrm{label}, C)$, sending both C' and s' to the client. Upon receiving the server's ciphertext and projection, the client chooses it's own random hash key to compute the projection $s := \alpha(k, \mathrm{label'}, C')$. At this point, the client computes a temporary session key $tk := H_k(\mathrm{label'}, C'.w) \oplus H_{k'}(\mathrm{label}, C, w)$ and a random session key

Client		**Server**
w	CRS: pk	w
$(\mathrm{VK}, \mathrm{SK}) \leftarrow \mathcal{K}(1^\kappa)$		
$r \leftarrow \{0,1\}^*$		
$\mathrm{label} := \mathrm{VK}\mid\mathrm{Client}\mid\mathrm{Server}$		
$C := \mathrm{Enc}_{pk}(\mathrm{label}, w; r)$	$\xrightarrow{\mathrm{Client}\mid\mathrm{VK}\mid C}$	$r' \leftarrow \{0,1\}^*$
		$\mathrm{label}' := \varepsilon$
		$C' := \mathrm{Enc}_{pk}(\mathrm{label}', w; r')$
		$\mathrm{label} := \mathrm{VK}\mid\mathrm{Client}\mid\mathrm{Server}$
$\mathrm{label}' := \varepsilon$	$\xleftarrow{\mathrm{Server}\mid C'\mid s'}$	$k' \leftarrow K;\ s' := \alpha(k', \mathrm{label}, C)$
$k \leftarrow K;\ s := \alpha(k, \mathrm{label}', C')$		
$\mathrm{tk} := \mathrm{H}_k(\mathrm{label}', C', w) \oplus \mathrm{H}_{k'}(\mathrm{label}, C, w)$		
$\mathrm{sk} \leftarrow \{0,1\}^\ell; c := \mathrm{ECC}(\mathrm{sk})$		
$\Delta := \mathrm{tk} \oplus c$		
$\sigma \leftarrow \mathrm{Sign}_{\mathrm{SK}}(C\mid C'\mid s'\mid s\mid\Delta)$	$\xrightarrow{s\mid\Delta\mid\sigma}$	if $\mathrm{Vrfy}_{\mathrm{VK}}(C\mid C'\mid s'\mid s\mid\Delta, \sigma) = 1$:
		$\mathrm{tk}' := \mathrm{H}_k(\mathrm{label}', C', w) \oplus \mathrm{H}_{k'}(\mathrm{label}, C, w)$
		$\mathrm{sk} := \mathrm{ECC}^{-1}(\mathrm{tk}' \oplus \Delta)$

Figure 6: Katz and Vaikuntanathan [KV09]

sk. Using $\mathbf{ECC} : \{0,1\}^l \to \{0,1\}^n$, an error-correcting code that corrects 2ε-fraction of errors, the client computes $c := \mathbf{ECC}(sk)$ to finally set $\Delta := tk \oplus c$ and signs σ, sending s, Δ and σ to the server. To finalize the exchange, the server verifies σ and similarly computes a temporary session key tk' and the following final session key $sk := \mathbf{ECC}^{-1}(tk' \oplus \Delta)$.

7.3 Isogeny PAKEs

C1: EKE-Style.

Terada and Yoneyama [TY19] proposed the only EKE-style PAKE based on (C)SIDH. The authors refer to their constructions as SIDH-EKE and CSIDH-EKE respectively.

C2: PAK-Style.

Taraskin et al. [TSJL20] proposed a PAK-similar PAKE based on SIDH and group.

- *Observation 1:* Isogeny EKE is vulnerable to possible MitM and offline dictionary attacks, due to the distinguishability of valid isogeny public keys from invalid ones upon decrypting them with a wrong password [AJK+20].
- *Observation 2:* The modified public key message distribution is dependent of the used password, which is also the case in RLWE-PAK. [TSJL20].

C3: SPEKE-Style

Abdalla et al. [AEK+22a] build upon the H2G idea from the classical SPEKE protocol [Jab96], however with a CRS. The authors proposed two constructions, the 1-round (2-pass) X-GA-PAKE and the 3-pass Com-GA-PAKE. Both protocols are based on restricted effective group actions with the ability of computing the so-called quadratic twist, which implies the same hardness assumptions as in CSIDH. The CRS is used to fix two set elements and use the password to map a tuple of elements. The protocols can also be

User U		**Server** S
	$CRS := (x_0, x_1) \in \mathcal{X}^2$	
	$pw := (b_1, \dots, b_\ell) \in \{0,1\}^\ell$	
$(u_1, \dots, u_\ell) \xleftarrow{\$} \mathcal{G}^\ell$		$(s_1, \dots, s_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
for $i \in [\ell]$		**for** $i \in [\ell]$
$x_i^U := u_i \star x_{bi}$	$\xrightarrow{x_1^U, \dots, x_\ell^U}$	$x_i^S := s_i \star x_{bi}$
	$\xleftarrow{x_1^S, \dots, x_\ell^S}$	
for $i \in [\ell]$		**for** $i \in [\ell]$
$z_i := u_i \star x_i^S$		$z_i := s_i \star x_i^S$
	$K := H(U, S, x_1^U, \dots, x_\ell^U, x_i^S, \dots, x_\ell^S, pw, z_1, \dots, z_\ell)$	

Figure 7: The GA-PAKE Protocol [AEK+22b]

modified to construct other variants, e.g., by increasing the number of public parameters in the CRS, or by using quadratic twists in the setup phase.

Technical Description. The SPEKE protocol (Fig. 15) relies on DH, where the password is hashed into a group generator using the function f. The GA-PAKE (Fig. 7) resembles this construction in the group action setting by mapping the password to an ℓ-tuple of elements in $\mathcal{X}$, instead of one element. It utilizes a CRS with two elements $(x_0, x_1) \in \mathcal{X}^2$ as a trusted setup, and a password $pw := (b_1, \dots, b_\ell) \in \{0,1\}^\ell$ that is mapped to the tuple $(x_{b1}, \dots, x_{b\ell}) \in \mathcal{X}^\ell$. Afterwards, a DH key agreement is done using the basis x_{bi} for each $i \in [\ell]$. Both user and server generate random group elements and compute a set of new elements using the DH basis, which they can exchange simultaneously. Then both compute new elements using the exchanged sets to hash into a final session key, along with previously generated values, the password, and their IDs. We note that this version of the protocol is not secure, but serves illustration only. The secure protocol X-GA-PAKE doubles the message length in the first round and triples it in the key derivation, relying on a new assumption called SqInv-StCDH, where an adversary has to compute the square and the inverse of its input at the same time. The Com-GA-PAKE adds a commitment from the server before the first message, and security is reduced to standard isogeny assumptions.

7.4 Generic PAKEs

C1: EKE-Style.

Beguinet et al. [BCP+23] proposed the first construction to transform a black-box KEM into a PAKE in generic manner, i.e., relying on KEM properties rather than on PQC hardness assumptions. Inspired by EKE and OEKE [BM92], the authors proposed the CAKE and OCAKE protocols in the ROM and IC model. While CAKE encrypts both the the public key and the ciphertext, the ciphertext is authenticated with a key confirmation tag in OCAKE. Following Beguinet et al., Pan and Zeng [PZ23] and Alnahawi et al. [AHHR24] presented further security analysis for CAKE and OCAKE respectively. Hövelmanns et al. revisited the formal analysis of OCAKE in [HHKR25], addressing the need for a QIC and extended the proof to quantum setting with concrete bounds in the QROM.

- *Observation 1:* KEM-based PAKEs, especially EKE-style, appears to be of greater interest, as they enable utilizing black-box reductions to the security properties of (standardized) NIST PQC KEMs.

$$\begin{array}{lcr}
\textbf{Alice} & & \textbf{Bob} \\
\hline
A, \text{pw} & \text{ssid} & B, \text{pw} \\
\hline
(pk, sk) \leftarrow \text{KEM.KeyGen}(1^{K}) & & \\
\mathbf{Epk} \leftarrow E_1(\text{ssid}||\text{pw}, pw) & \xrightarrow{A, \mathbf{Epk}} & pk \leftarrow D_1(\text{ssid}||\text{pw}, \mathbf{Epk}) \\
 & & (c, K) \leftarrow \text{KEM.Encaps}(pk) \\
c' \leftarrow D_2(\text{ssid}||\text{pw}, \mathbf{Ec}) & \xleftarrow{B, \mathbf{Ec}} & \mathbf{Ec} \leftarrow E_2(\text{ssid}||\text{pw}, c) \\
K' \leftarrow \text{KEM.Decaps}(sk, c') & & \\
\text{SK} \leftarrow \text{H}(\text{ssid}, A, B, \mathbf{Epk}, \mathbf{Ec}, K') & & \text{SK} \leftarrow \text{H}(\text{ssid}, A, B, \mathbf{Epk}, \mathbf{Ec}, K')
\end{array}$$

Figure 8: The CAKE Protocol [BCP+23]

- *Observation 2:* KEM black-box reductions are very useful, but need to be programmed carefully to achieve sound simulation-based proofs [HHKR25].

- *Observation 3:* (O)CAKE is yet to be shown quantum-safe due to the lack of the QIC model. Considering the difficulty of reprogramming in the QROM, it is unclear if a formal analysis in UC can be achieved [MX23b, LLH25].

Technical Description. CAKE relies on two pairs of ICs, denoted by (E_1, D_1) and (E_2, D_2), which resemble ε_w in EKE. Both parties share knowledge of the password pw. Alice starts by generating a key-pair (pk, sk) and encrypts the public key using the ideal cipher $\mathbf{Epk} \leftarrow E_1(\text{ssid}||\text{pw}, pk)$. After receiving **Epk**, Bob can decrypt Alice's message to obtain her public key, plugging it into the encapsulation function to receive (c, K). Bob uses E_2 to encrypt the ciphertext, sending it back to Alice. After decrypting the message E_c to receive c', Alice decapsulates it to get $K' \leftarrow \text{KEM.Decaps}(sk, c')$. If Alice and Bob use matching passwords, the pre-keys match too, i.e., $K = K'$, and both parties obtain a matching session key SK via hashing.

C1: EKE-Style with HIC.

Dos Santos et al. [DGJ23] proposed a similar generic PAKE in the ROM, however utilizing the HIC, as opposed to the IC used in CAKE. This construction also relies on abstract KEM security properties but mainly addresses the lack of the QIC and issues with the unconditional uniformity of KEM public keys. Building upon the HIC idea of Dos Santos et al. [DGJ23], Arriaga et al. [ABJS24] proposed the Compact Half-Ideal Cipher (CHIC) protocol. The authors utilize said m2f construction (Fig. 9) in white-box manner and use a randomized value taken from the public key as a randomness seed. Their main contribution is a compact m2F and bandwidth-minimal KEM-to-PAKE compiler, where they also establish security requirements for KEMs with splittable public keys (not to be confused with Split-KEM as in hybrid key encapsulation [BBF+19]).

- *Observation 4:* KEMs with structured public keys (e.g., ML-KEM) do not satisfy statistical public key uniformity, and are thus vulnerable to detecting invalid keys upon decrypting with a wrong password [DGJ23, ABJS24, AASA+24, ABJ25].

- *Observation 5:* Relying on IND-CPA might not be sufficient for tight security proofs in generic PAKE constructions [ABJS24].

```
$\mathsf{m2F}_\pi(r, M)$
$R \leftarrow \mathcal{H}_1(\pi, r)$
$T \leftarrow M \odot R$
$t \leftarrow \mathcal{H}_2(\pi, T)$
$s \leftarrow \mathsf{IC.Enc}(t, r)$
return $(s, T)$

$\mathsf{m2F}_\pi^{-1}(s, T)$
$t \leftarrow \mathcal{H}_2(\pi, T)$
$r \leftarrow \mathsf{IC.Dec}(t, s)$
$R \leftarrow \mathcal{H}_1(\pi, r)$
$M \leftarrow T \odot R^{-1}$
return $(r, M)$
```

Figure 9: The modified 2-Round Fiestel, where $\odot$ is a group operation in G, and $(.)^{-1}$ is an inverse in G [ABJS24].

```
$\mathsf{2F}_k(r, M)$
$R \leftarrow \mathcal{H}_1(k, r)$
$T \leftarrow M \odot R$
$t \leftarrow \mathcal{H}_2(k, T)$
$s \leftarrow r \oplus t$
return $(s, T)$

$\mathsf{2F}_k^{-1}(s, T)$
$t \leftarrow \mathcal{H}_2(k, T)$
$r \leftarrow s \oplus t$
$R \leftarrow \mathcal{H}_{(}k, r)$
$M \leftarrow T \odot R^{-1}$
return $(r, M)$
```

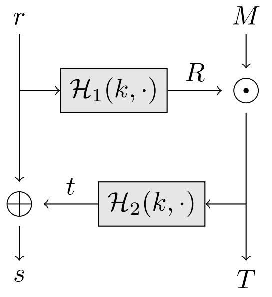


Figure 10: The 2-Round Fiestel, where $\oplus$ is an XOR operation on bit strings, $\odot$ is a group operation in G, and $(.)^{-1}$ computes the inverse of a group element [ABJ25].

C1: EKE-Style without IC.

Although McQuoid et al. [MRR20] did not provide a concrete PQC instantiation, they introduced a generalization of EKE using Programmable-Once Public Functions (POPF), or a 1-out-of-N Oblivious Transfer (OT) that is referred to as 1-OPRF. These PAKEs do not rely on an IC, and can also accommodate PQC KEMs with certain properties. The POPF construction is a 2-round Feistel (2f), which can be viewed as keyed randomized function, as opposed to a keyed permutation [ABJ25] (i.e., IC). To encrypt a PAKE initiator's public key, a password-derived key is used to hash the public key into an algebraic group, and thus shifting its value within the the same group (cf. Fig. 10). Since the 2f can also be queried in the reverse direction, it enables programming in simulation-based proofs. Januzelli et al. [JRX24] fixed an MitM vulnerability in said POPF, and provided a thorough analysis on the UC security of EKE-style PAKEs, especially regarding the security properties required from the underlying KEX or KEM. Arriaga et al. [ABJ25] addressed issues regarding specific proof techniques and provided an extensive analysis for the POPF-based UC PAKE and also specifying concrete KEM security properties.

- *Observation 6:* Although abstract KEM security properties are well studied, there is little consensus on their correct usage in PAKE formal analysis.
- *Observation 7:* Beside key security (IND), EKE requires KEMs with (strong) PR for public keys (uniformity) and ciphertexts (anonymity). OEKE additionally requires session key Collision-Resistance (CRF) [JRX24].
- *Observation 8:* Programmability in PAKE proofs can be achieved without an IC, yet it needs to be handled carefully to thwart guessing attacks [ABJ25].

```
Splittable Public Key of (R)(M)LWE
$(pk, sk) \leftarrow \mathsf{KEM.KGen}(1^\kappa)$
$(\mathbf{A}, \mathbf{b}) \leftarrow pk : (\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$
$\rho \leftarrow pk$ // Extract seed bit string
.......... Knowing $\mathbf{b}$ reuse $\rho$ ..........
$\mathbf{A} \leftarrow \rho$ // Deterministic expansion
$pk \leftarrow (\mathbf{A}, \mathbf{b})$
return $pk$
```

Figure 11: Splitting the public key of an (R)(M)LWE KEM to obtain the fixed-length seed ρ appended (or prepended) to the public key pk [AASA$^+$24].

C1: Semi-Generic EKE-Style without IC.

Inspired by the idea of splittable KEMs, Alnahawi et al. [AASA$^+$24] propoesed a new approach to construct a secure PAKE without IC. This PAKE relies on a RO-modeled hash function used to extend the password to the same length as the uniform string part of the public key (the seed used to sample a lattice base) in LWE, RLWE, and MLWE KEMs. The authors also make use of similar KEM properties and introduce new notions for LWE KEMs with splittable public keys (Fig. 11).

- *Observation 10:* Additional assumptions from primitives (e.g., LWE) requires parameter changes to remedy decreased security or complete breaks [AASA$^+$24].

- *Observation 11:* Exploring non-standard usage of KEMs with abstract properties may enable new PAKE designs with a reasonable security trade-offs [AASA$^+$24].

C3: Semi-Generic (No Specific Style).

Lyu et al. [LLH24] proposed the first UC PAKEs in the QROM, with three passes and MA in the CRS model. Two ROM construction rely on Lossy Public Key Encryption (LPKE) on IND-CCA KEMs, and QROM constructions uses a variant called extractable LPKE (eLPKE) and utilizes hash functions as PRFs. Here, the KEM FO transformation is used directly to lift the PKE security from IND-CPA to IND-CCA in white-box manner.

7.5 aPAKE Compilers (Symmetric to Asymmetric Transformation)

Basically, an aPAKE compiler provides a (generic) recipe to obtain a secure asymmetric PAKE from another secure symmetric PAKE, in order to protect against server compromise attacks. This is done through composition, where the symmetric PAKE is augmeented with some additional steps, and hence it is usually analyzed in UC. The aPAKE compiler of Gentry et al. [GMR06] is the first one instantiated form quantum-safe primitives (lattices) and signatures. Nearly a decade later, McQuoid and Xu [MX23b] proposed a UC aPAKE compiler from isogeny-based group action. Recently, Lyu et al. [LLH25] also proposed a generic compiler based on KEMs and Authenticated Encryption (AE) and compared their construction to the afore mentioned ones. Hence we refer to their work for further reading, as this specific type of PAKE is out of the scope of our paper.

7.6 Security Overview

Tab. 2 offers an overview of the properties found in KEM-based PAKEs. While, IND- and ANO- are generally present in almost all PAKEs, CHIC opts for OW-CPA for key security and for ANO-PCA for anonymity, the latter of which is also used in the CAKE and OCAKE versions of Pan and Zeng, and Alnahawi et al. respectively. The CHIC construction also defines UNI-PK for splittable KEMs similar to HIC-EKE, which is defined as Fuzzy-KEM in CAKE and the follow-up work by Pan and Zeng. The OCAKE follow-up of Alnahawi et al. and NICE-PAKE refer to this property as PKU and SPLIT-PKU respectively. McQuoid et al. and Januzelli et al. use different naming conventions and define (Strong) Pseudo-Randomness ((S)PR) and Non-Malleability (NM), as well as Collision-Resistance (CFR) (cf. Sec. 4). NICE-PAKE derives properties specific to splittable KEMs w.r.t. the A-part of a public key as anonymity (A-SEC) and Collision-Freeness (A-CFR).

Table 2: Security Properties of KEM-Based Generic PQC PAKEs

Protocol	Session Key	Public Key	Ciphertext	Other
POPF-EKE [MRR20]	IND-CPA	PR	PR / ROB	-
(O)CAKE [BCP+23]	IND-CPA	Fuzziness	ANO-CPA	-
HIC-EKE [DGJ23]	IND-CPA	UNI-PK	ANO-CCA	-
Pan and Zeng (CAKE) [PZ23]	IND-CPA	Fuzziness	ANO-PCA	Multi-User
Alnahawi et al. (OCAKE) [AHHR24]	IND-CPA	PKU	ANO-PCA	Multi-User
Januzelli et al. [JRX24] (EKE)	IND-CPA	PR	SPR	PR-NM
Januzelli et al. [JRX24] (OEKE)	IND-CPA	PR	PR/CFR	PR-NM
CHIC [ABJS24]	OW-CPA	UNI-PK	ANO-PCA	-
NICE-PAKE [AASA+24]	IND-CCA	SPLIT-PKU	ANO-CCA	A-SEC / A-CFR
NoIC-PAKE [ABJ25]	OW-CPA	UNI-PK	ANO-PCA	-
Hövelmanns et al. [HHKR25] (CAKE)	IND-CPA	PKU	ANO-PCA	Multi-User
Hövelmanns et al. [HHKR25] (OCAKE)	IND-CPA	PKU	ANO-PCA	Multi-User

Tab. 3 presents an overview of basic security properties for all surveyed PAKEs. These are the number of flows, security of the session key (IND-), forward secrecy (FS), and mutual authentication (MA). The session key security is denoted by either IND-CPA or IND-CCA depending on the underlying key agreement scheme. Forward secrecy can either be weak (FS) or perfect (PFS), as no authors state that their constructions do not provide any at all. Mutual authentication is either checked or not, indicating explicit or implicit authentication. The number of flows indicates how many messages are sent from one protocol participant to another (i.e., one flow equals one message). We note that one round in a cryptographic protocol indicates one back and forth message exchange between two parties. That is, one round consists of two messages (or passes).

7.7 Performance Overview

In Tab. 4 and Tab. 5 we compile information on bit-security, communication cost and computation cost for two-party and three-party PAKEs respectively. Any values for communication and computation cost given are rounded to three decimal places where appropriate, otherwise values are adopted exactly as provided in their original publications. Some publications yield multiple table rows for specific security or implementation variants of a PAKE. Such cases may include the variant of an underlying KEM, the variant of a security parameter set (where the values n, p and q denote the security parameter of the underlying hardness assumption), a slight variation in the implementation of a protocol or two different protocols within the same publication. Communication cost describes the total size of outgoing messages of a given party and is either given in bits (b), bytes (B) or kilobytes (kB) while computation costs are either given in microseconds (μs), milliseconds (ms), seconds (s) or cycles (c).

Table 3: Security Properties of PQC PAKE Protocols

Class	Protocol	Flows	IND-	FS	MA
Balanced					
C1	Terada and Yoneyama ((C)SIDH-EKE) [TY19]	2	CPA	PFS	✗
	McQuoid et al. (POPF-EKE) [MRR20]	2	CPA	PFS	✗
	Dos Santos et al. (HIC-EKE) [DGJ23]	2	CPA	PFS	✗
	Beguinet et al. (CAKE) [BCP+23][1]	2	CPA	PFS	✗
	Beguinet et al. (OCAKE) [BCP+23]	2	CPA	PFS	✓
	Pan and Zeng (CAKE) [PZ23]	2	CPA	PFS	✗
	Alnahawi et al. (OCAKE) [AHHR24]	3	CPA	PFS	✓
	Januzelli et al. (POPF-EKE) [JRX24]	2	CPA	PFS	✗
	Januzelli et al. (POPF-OEKE) [JRX24]	2	CPA	PFS	✓
	Arriaga et al. (CHIC) [ABJS24]	2	OW-CPA	PFS	✗
	Alnahawi et al. (NICE-PAKE) [AASA+24][1]	2	CCA	PFS	✗
	Arriaga et al. (NoIC-PAKE) [ABJ25]	2	OW-CPA	PFS	✗
	Hövelmanns et al. (OCAKE) [HHKR25]	3	CPA	PFS	✓
C2	Zhu, Geng [ZG15]	2	CPA	PFS	✗
	Ding et al. (RLWE-PAK) [DAL+17]	3	CPA	FS	✓
	Ding et al. (RLWE-PPK) [DAL+17]	2	CPA	FS	✗
	Gao et al. (RLWE-PAK) [GDL+17]	3	CPA	FS	✓
	Gao et al. (RLWE-PPK) [GDL+17]	2	CPA	FS	✗
	Taraskin et al. (SIDH-PAK) [TSJL20]	3	CPA	FS	✓
	Yang et al. (RLWE-PAK) [YGWX19]	3	CPA	FS	✓
	Jiang et al. (PAKEs) [JGH+20]	3	CPA	PFS	✓
	Ren et al. (MLWE-PAK)[RGW23] ([RG22])	3	CPA	FS	✓
	Seyhan, Akleylek [SA23]	3	CCA	PFS	✓
	Basu et al. (MLWR-2PAKA) [BSIA23][2]	4	CPA	PFS	✓
C3	Katz, Vaikuntanathan [KV09]	3	CCA	PFS	✗
	Xu et al. (RLWE-3PAKE) [XHCC17]	6	CPA	PFS	✓
	Zhang, Yu [ZY17]	2	CCA	PFS	✗
	Choi et al. (AtLast) [CAK+18][3]	5	CPA	FS	✓
	Li, Wang [LW18]	2	CPA / CCA	FS	✗
	Li, Wang [LW19]	2	CCA	FS	✗
	Karbasi et al. (Ring-PAKE) [KAA19]	3	CCA	PFS	✗
	Yin et al. [YGS+20][4]	2	CCA	FS	✓
	Lyu et al. [LLH24]	3	CCA	?	✓
Augmented					
C2	Gao et al. [GDLL17]	2	CPA	FS	✓
C3	Zhu et al. [ZHS14]	5	?	PFS	✓
	Feng et al. [FHZ+18][†5]	3	CPA	FS	✓
	Liu et al. [LZJY19]	?	CPA	FS	✓
	Dabra et al. (LBA-PAKE) [DBK20][†5]	3	CPA	FS	✓
	Tang et al. [TLZ+21]	3	CCA	FS	✓
	Li et al. [LWM22][7]	3	CCA	PFS	✓
	Islam, Basu (BP-3PAKA) [IB21][†5]	4	CPA	PFS	✓
	Abdalla et al. (X-GA-PAKE) [AEK+22a]	2	CPA	PFS	✓
	Abdalla et al. (Com-GA-PAKE) [AEK+22a]	3	CPA	PFS	✓
	Wang et al. (LB-ID-2PAKA) [WCL+23][5]	2	CCA	PFS	✓
	Guo et al. [GSG+23]	2	CCA	FS	✓
	Chaudhary et al. [CKS23][†5]	4	?	PFS	✓
	Yang et al. [YZYW25] (K-PAKE)	1	CCA	PFS	✓

[1] Mutual authentication requires an additional key confirmation round.
[2] Number of rounds excluding the initialization phase.
[3] No explicit authentication with the server, only between users.
[4] Applies to both 2PAKE and 3PAKE variants.
[5] Excluding the registration phase.
[†] Anonymous PAKEs.

Table 4: Comparison of Bit Security and Performance for 2-Party PAKEs

Class	PAKE		Security		Computational Cost			Communication Cost		
	Publication	Variant	Classic	Quantum	Client	Server	Total	Client	Server	Total
Balanced										
C1	Terada and Yoneyama [TY19]	SIDH	NIST I	NIST I	-	-	80.6ms	64B	64B	128B
		CSIDH	NIST I	NIST I	-	-	5.0ms	330B	330B	660B
	Beguinet et al. [BCP+23]	CAKE	-	102bit	-	-	-	-	-	-
		OCAKE	-	162bit	-	-	-	-	-	-
	Alnahawi [AHHR24]	Kyber512	-	-	-	-	0.995s	-	-	-
		Kyber1024	-	-	-	-	2.039s	-	-	-
		Frodo640Shake	-	-	-	-	73.057s	-	-	-
		bikel1	-	-	-	-	26.519s	-	-	-
	Arriaga et al. [ABJS24]	Kyber512	128bit	-	84μs	74μs	158μs	800B	800B	1600B
		Kyber768	128bit	-	168μs	85μs	253μs	1,184B	1,120B	2,304B
		Kyber1024	128bit	-	206μs	123μs	329μs	1,568B	1,600B	3,168B
C2	Ding et al. [DAL+17]	RLWE-PAK	-	76bit	2,981.251μs	2,884.243μs	6,702.656μs	4,136B	4,256B	8,392B
		RLWE-PPK	-	76bit	-	-	-	-	-	-
	Gao et al. [GDL+17]	RLWE-PAK	≥200bit	82bit	0.176ms	0.175ms	0.351ms	3,904B	4,000B	7,904B
		RLWE-PPK	≥200bit	82bit	0.203ms	0.203ms	0.406ms	3.75kB	3.875kB	7.625kB
	Taraskin et al. [TSJL20]	p434	-	-	-	-	142×10^6c	-	-	-
		p503	-	-	-	-	228×10^6c	-	-	-
	Yang et al. [YGWX19]	avx2	228bit	206bit	145,964c	137,313c	283,277c	1,864B	2,592B	4456B
		portable C	228bit	206bit	294,460c	270,227c	564,687c	1,864B	2,592B	4456B
	Jiang et al. [JGH+20]	-	-	-	0.2s	0.71s	0.91s	39,990B	167,090B	207,080B
	Ren et al. [RG22, RGW23]	Light	128bit	116bit	89.76ms	93.35ms	183.11ms	928B	1,056B	1,984B
		Recomm.	195bit	177bit	126.71ms	126.05ms	252.76ms	1,344B	1,472B	2,816B
		Paranoid	263bit	239bit	174.97ms	169.52ms	344.49ms	1,760B	1,888B	3.648B
	Seyhan, Akleylek [SA23]	Lightsaber	-	128bit	104,824μs	60,632μs	165,456μs	896B	1,600B	2,496B
		Saber	-	192bit	172,427μs	97,758μs	270,185μs	1,344B	2,368B	3,712B
		Firesaber	-	256bit	57,756μs	152,232μs	409,988μs	1,760B	3,168B	4,928B
	Basu et al. [BSIA23]	$p=2^{10}$, $q=2^{13}$	127bit	116bit	-	-	-	-	-	2816B
		$p=2^{9}$, $q=2^{15}$	140bit	127bit	-	-	-	-	-	2560B
Augmented										
C2	Gao et al. [GDLL17]	-	209bit	-	0.286ms	0.257ms	0.543ms	3,963B	4,032B	7,995B
C3	Feng et al. [FHZ+18]	w/out Precomp.	-	-	2.307μs	0.222μs	2.529μs	5,121b	4,609b	9,730b
		w/ Precomp.	-	-	1.184μs	0.075μs	1.259μs	5,121b	4,609b	9,730b
	Dabra et al. [DBK20]	n=128	-	-	6.501ms	33.298ms	39.799ms	3528b	3296b	6824b
		n=256	-	-	17.372ms	66.094ms	83.466ms	6600b	6368b	12,968b
		n=512	100bit	75bit	26.271ms	136.442ms	162.713ms	12,744b	12,512b	25,256b
	Li et al. [LWM22]	Classical	-	-	116ms	361ms	477ms	26,326b	32,950b	59,312b
		Quantum	-	-	116ms	473ms	589ms	29,602b	40,320b	69,922b
	Ding et al. [DCQ22]	n=128	-	-	-	-	-	4496b	4224b	8720b
		n=256	-	-	-	-		8,464b	8,192b	16,656b
		n=512	-	-	39.22ms	12.65ms	51.87ms	16,400b	16,128b	32,528b
	Dharminder et al. [DRD+23]	-	-	-	2.297μs	0.229μs	2.526μs	-	-	9,790b
	Dadsena et al. [DJRD23]	-	-	-	-	-	2.826μs	-	-	9,725b
	Kumar et al. [KGKD23]	-	-	-	2.297μs	0.229μs	2.526μs	-	-	9,726b
	Yang et al. [YZYW25]	Kyber	-	-	-	-	-	459,5B	459B	918,5B

Table 5: Comparison of Bit Security and Performance for 3-Party PAKEs

Class	PAKE		Security		Runtime				Message Sizes			
	Publication	Implementation	Classic	Quantum	Client$_A$	Client$_B$	Server	Total	Client$_A$	Client$_B$	Server	Total
Balanced												
C3	Xu et al. [XHCC17]	-	-	-	0.067ms	0.071ms	0.122ms	0.259ms	-	-	-	-
Augmented												
C3	Liu et al. [LZJY19]	-	-	-	3.267ms	4.155ms	2.195ms	9.617ms	-	-	-	-
	Tang et al. [TLZ+21]	n=32	-	-	2.27ms	0.38ms	20.52ms	23.17ms	-	-	-	7kB
		n=64	-	-	4.44ms	0.63ms	63.1ms	68.17ms	-	-	-	13kB
		n=128	-	-	10.38ms	1.07ms	353.49ms	364.91ms	-	-	-	24kB
		n=256	-	-	24.31ms	3.39ms	3,373.06ms	3,400.76ms	-	-	-	48kB
		n=512	-	-	82.11ms	4.62ms	45,802.87ms	45,889.6ms	-	-	-	79kB
	Islam, Basu [IB21]	-	-	-	-	-	-	-	5,249b	5,249b	9,408b	19,906b
	Guo et al. [GSG+23]	-	-	222bit	-	-	-	0.788ms	-	-	-	10,080B
	Chaudhary et al. [CKS23]	-	-	-	4.9μs	-	0.215μs	5.214μs	-	-	-	19,226b

7.8 Real World Use Cases

In the following, we establish a mapping between current PQC PAKEs and existing real world applications, where classical PAKEs are already used. We adopt the PAKE use cases from [HvO22], in addition to one new use case [DFG+23]. That being said, we disregard a discussion on non-PAKE alternatives, i.e., cryptographic solutions without using PAKEs, since it is sufficiently covered in [HvO22].

Credential Recovery

Several well-known commercial IT systems including Apple iCloud, 1Password manager, ProtonMail, and Blizzard incorporate classical PAKE protocols such as the Secure Remote Password protocol (SRP-6a) [Wu98, Wu02] for general account access, user authentication, or credential recovery [HvO22].

Discussion. We recall the previous observations related to the risks of pre-computation attacks and server compromise. Thus, an augmented PAKE is especially preferable, as it is highly recommended to register users without storing the actual password on the server. However, most of the proposed C3 augmented PQC PAKEs are 3-party PAKEs, whereas most C3 2-party PAKEs are balanced. Therefore, it is difficult to name specific candidates for this use case. Gao et al. [GDLL17] explicitly state that their PQC PAKE is an SRP-similar protocol and provide reasonable security margins and benchamrking values for the well-studied RLWE problem. Hence, this PAKE could be a candidate for replacing SRP-6a, should a full proof be provided for the construction.

Device Pairing

The most common applications of PAKE-based device pairing are found in situations where it is not possible to rely on a Public Key Infrastructure (PKI). This especially the case for relatively small devices and e-cards equipped with smart chips (i.e., embedded and IoT), or in the context of Wi-Fi connection[ASWZ24]. For instance PACE [BFK09] is used in electronic IDs and eMRTDs (Machine Readable Travel Document) for secure connection establishment with terminals and e-card readers; and Dragonfly [Har08] is used in WPA3 for establishing a Wi-Fi connection between an access point and a client [HvO22].

Discussion. Several PQC PAKE proposals claim constructions tailored for embedded and IoT devices (e.g., [FHZ+18, DBK21, LWM22, DRD+23, RGW23, RSM+23]). On the design level, we believe that balanced 2-party PQC PAKEs in classes C1 and C2 are the most suitable for ad-hoc device pairing (e.g., [TY19, BCP+23, ABJS24, JRX24, ABJ25]), where no registration is possible nor needed (e.g., Wi-Fi and eMRTDs). However, in other use cases (e.g., smart-gadgets, smart-meters, e-mobility etc.), light weight C3 PAKEs with prior registration may also be a valid option. Further, the low computational power and resource constraints of such small devices inherently infer restrictions on the choice of a suitable PAKE. Embedded and IoT devices are additionally more prone to physical attacks such as Side-Channel Analysis (SCA) and Fault Injections (FI), the mitigation of which comes with additional costs both in memory and run time [ASWZ24]. Considering the already existing extra costs of PQC schemes, light weight PAKEs with fast execution times are recommended.

End-to-End (E2E) Secure Channel

E2E encryption is usually found in chat applications that mostly utilize an authenticated key exchange (AKE) using static keys from a connected PKI (e.g., Signal's X3DH). Nevertheless, some applications involving embedded and IoT devices cannot rely on a PKI. For instance, the J-PAKE protocol is used in Thread IoT products, Palemoon web browser, and the Smoke Chat application for android; whereas EC-SPEKE is used in Blackberry Messager (BBM) [HvO22][7].

[7] Note that Blackberry Messenger was discontinued in 2019. And its successor BBMe has also been discontinued.

Discussion. Since such applications may rule out a trusted setup in the client-client case, we believe that balanced PQC PAKEs in C1 and C2 with PFS and mutual authentication (e.g., [JGH+20, SA23, BCP+23, BSIA23]) are most suitable. That being said, use cases with a possible trusted third-party would enable the use of augmented 3-party PAKEs (e.g., [LZJY19, IB21, RSM+23, GSG+23]). Similar to device pairing, small devices require PAKEs with light weight implementations and resistance to physical attacks. Considering the ever increasing computational power of personal portable devices, this requirement does not necessarily apply to smart-phones and tablets etc.. Still, E2E session key establishment in both the client-server and the client-client models imposes the requirement of PFS.

End-to-End (E2E) Encrypted Backups

To the best of our knowledge, the only E2E encrypted backup application relying on a PAKE is found in the WhatsApp Backup Protocol (WBP), which was released in 2021 [DFG+23]. WBP allows users to recover their backup keys using a password and ultimately retrieve their chat histories from lost or broken devices. Under the hood, OPAQUE serves as the main cryptographic protocol related to password usage. Being an aPAKE, it enables a key exchange between a user and a server[8] without revealing the actual password to the server. Apparently, directly elevating the security of WBP to PQ-settings requires the employment of a PQ-version of OPAQUE, or an augmented C3 PQC 2-party PAKE (e.g., Gao et al [GDL+17]). Similar to credential recovery, we were unable to identify other suitable candidates for this use case.

8 Honorable Mentions

Symmetric (Fuzzy) PAKEs

This type of PAKEs focuses on use cases where the shared password (or credentials) of two parties in a protocol instance are not necessarily identical, yet close enough up to a predefined threshold. This method is mainly utilized to amend the shortcomings of balanced and augmented PAKEs in dealing with frequent typing errors in passwords, but more importantly when using biometric data as authentication credentials [Ott24] (e.g., iris scans and fingerprints). Since Fuzzy PAKEs are mostly built from inherently quantum-safe primitives (e.g. garbled circuits), they can be considered relevant in quantum-resilient PAKE research. A recent work by Ottenhues [Ott24] presents and overview of such symmetric Fuzzy PAKE protocols and compares their security both in theory and practice. This overview includes one protocol built from garbled circuits [DHP+18] and two from ECC [DHP+18, BFH+23]. Other Fuzzy PAKEs include the work of Erwig et al. [EHOR20], which relies on OT and robust secret sharing to obtain two asymmetric Fuzzy PAKEs in the UC model. Further, Bauspieß et al. [BSP+24] present a modification of unlinkable fuzzy vault schemes combined with OPRFs, which can be instantiated with lattices, to construct a protocol for biometrics-based AKE. They also compare to similar constructions including the previously mentioned PAKEs, and to the works of Wang et al. [WHC+21], Han et al. [HXL+23] (ttPAKE), and Zhang et al. [ZYL+23] (BAKA).

Quantum-Annoying PAKEs

A PAKE is considered quantum-annoying, if a quantum adversary can break the PAKE, but only through solving a specific type of mathematical problem (e.g., discrete logarithm) for each password they try [ES21]. Thus, quantum-annoying PAKEs do not promise full

[8] Precisely, the server only relays user messages to a HSM (Hardware Security Module), with which a user establishes a secure session.

quantum-resilience, but they do provide a certain degree of resistance against quantum computers by making some classical operations more expensive for quantum adversaries. The main idea is to make offline dictionary attacks more expensive by hiding group elements, so that an adversary needs to compute the discrete logarithm for each offline password guess [TES23]. That is, even if they succeeds in computing the discrete logarithm (i.e., essentially break DH) in an online protocol session, they cannot directly relate the computed value to a certain single password. To the best of our knowledge, the notion of quantum annoying PAKEs is mainly found in the generic group model for discrete logarithm-based PAKEs, and was formalized by Eaton and Stebila [ES21] based on the classical symmetric CPace protocol, which was shown to inherently satisfy this property. Following that, Tiepelt et al. [TES23] presented a simple modification to the classical asymmetric KHAPE-HMQV [GJK21] PAKE protocol by adding an IC-based encryption to one protocol message, which also makes KHAPE quantum annoying.

9 Unexplored Territory

We observe that a few designs are still unrepresented or not fully explored in the PQC realm, which can still be adapted to PQC, and had not yet been addressed in the literature. Such constructions include, PAKEs that make use of supporting building blocks that have PQC replacements (e.g., OPRFs and NIZKs), and where the password usage is not dependent on a mathematical properties inherent to classical cryptography (e.g., password-derived generators as in PACE or password-derived exponents as in SPR-6).

Augmented OPAQUE-Style with OPRFs

The augmented PAKE protocol OPAQUE was proposed by Jarecki et al. [JKX18] defining the strong aPAKE functionality, primarily addressing the vulnerabilities of pre-computation attacks. While there is still no PQ OPAQUE proposal in the literature, the original authors did provide two general frameworks to obtain a strong aPAKE in the UC model. The first framework is based on an AKE scheme and an OPRF, whereas the second requires an authenticated encryption scheme (AE) and a KEX in addition to the OPRF. Both versions of OPAQUE consist of a registration and a login phase, where the OPRF is meant to hide the values associated with the password. In other words, a user runs an OPRF on their password with a server to obtain an ephemeral random secret, which they later on use as a private key for a key exchange in the login phase. In the AKE variant, both user and server already provide their (static) public keys during registration for later usage in the login phase. An instantiation of a PQC OPAQUE from isogeny-based OPRFs was given by Heimberger et al. in [HHM+24]. However, it is drastically inefficient compared to classical instantiations, and no formal proof was provided. In [BDFH24], Beullens et al. presented a secure OPRF from PQ multi-party computation in UC, and suggested their OPRF can be, composed, or plugged-in an OPAQUE instantiation. Since an AKE with either unilateral or bilateral authentication can easily be constructed using a PQC KEM (as shown by Bos et al. with CRYSTALS-Kyber [BDK+18]), and there exists a number of PQ OPRFs (cf. Sec. 5), we suggest realizing a KEM-based PQC OPAQUE adaptation For a user U and a server S as follows:

Registration (with bilateral static keys):

- S chooses a fresh OPRF key $k \xleftarrow{\$} \{0,1\}^n$, a static KEM key pair (sk_s, pk_s), and sends pk_s to U.
- U executes the OPRF $\mathcal{F}$ with S using the password π as input to obtain a secret $s = \mathcal{F}_k(\pi)$. U derives a KEM key pair (sk_u, pk_u) from s, computes an authentication

key $K_{MAC} = HMAC_s(pk_s)$, and sends K_{MAC} to S.

- S stores the values (sk_s, pk_s, pk_u), the OPRF key k, and K_{MAC}.

Login (with session key establishment):

- U executes the OPRF $\mathcal{F}$ with S using the password π as input to obtain a secret $s = \mathcal{F}_k(\pi)$. U derives an ephemeral KEM key pair (sk, pk) from s then receives pk_s and K_{MAC} from S and verifies K_{MAC}.
- U encapsulates pk_s into (K_s, C_s) and sends thei pk along with C_s to S.
- S decapsulates C_s with their secret key sk_s to obtain K_s and encapsulates pk, pk_u into $(K, C), (K_u, C_u)$ respectively, and then send C, C_u to U.
- U decapsulates C, C_u to obtain K, K_u respectively.
- Both U and S set their final keys (session key) to $\overline{K} = \mathcal{H}(K, K_u, K_s)$.

Remarks: We note, and stress, that the PQC OPAQUE outline is neither verified nor formally analyzed, and thus serves only as a rough sketch for a possible instantiation of OPAQUE using a PQC KEM and a PQC OPRF. That being said, other variants without static public keys (i.e., with an AE), or using a PQC KEX (e.g., Ding's RLWE scheme) are very likely to be possible as well. Further, and as noted in the original OPAQUE paper, the UC aPAKE functionality requires a non-black-box hardness assumption on the ROM in order to extract password guesses, which automatically implies the need for programming in the QROM. Last but not least, the used KEX or KEM must provide PFS and key-compromise impersonation (KCI) security. Nevertheless, an open question remains in finding a mechanism to to derive a valid key pair from the random secret obtained via the OPRF execution between the user and the server. Since OPAQUE originally relies on a H2C function to map a password to an EC base generator, it seems intuitive to resort to its adaptation to H2G settings, which can be directly obtained from isogenies (cf. [HHM+24]) or C2 PQC PAKE constructions (e.g., RLWE-PAK), and rightfully so, since OPAQUE belongs to class C2 according to [HvO22]. While this might answer the question of instantiating OPAQUE directly with a PQC primitive, it is not clear how to realize this in combination with generic KEMs. So far, the only existing method to construct generic KEM-based PQC PAKEs relies mainly on an IC encryption (or similar), which differs from the public key modification (PAK-style) method. As the key generation routine in a KEM does not usually allow for any input values, a rather uncomfortable approach would be to use the OPRF obtained secret directly as the secret key, and apply it to the key generation base (e.g., a lattice base matrix) to generate a public key. Another approach might be found in using a PQC Non-Interactive Key Exchange (NIKE), e.g., SWOOSH [GdKQ+24], that enables a white-box key generation routine, while also maintaining the non-interactive characteristics of a KEM. We consider finding a solution an open question for future work.

J-PAKE-Style with NIZKs

The J-PAKE protocol proposed by Hao and Ryan [HR10] can be considered the first PAKE to make use NIZKs (namely the Schnorr protocol [Sch91]) and achieve a relatively efficient construction. The balanced J-PAKE essentially aims at providing a proof of knowledge of the password without actually revealing it. It relies on a the juggling technique using NIZKs, where random public keys are combined in order to achieve a vanishing effect when both parties supply exactly the same passwords. Other variants and adaptations of J-PAKE include RO-J-PAKE and CRS-J-PAKE proposed by Lencrenon et al. [LST16], and sJ-PAKE proposed by Abdalla et at. [ABR+21] (sJ-PAKE replaces two NIZK proofs by two exponentiations). Recently, Hao et al. [HBCvO23] proposed the augmented Owl protocol addressing the limitations of SPR-6a and OPAQUE, however inspired by J-PAKE and elevating it to an augmented PAKE. Considering the ongoing progress on realizing secure and efficient PQ ZKP systems (including variants such as NIZKs, SNARKs etc.), it is quite reasonable to assume that a PQC version of J-PAKE can be realized while maintaining the original design. Intuitively, replacing the underlying KEX in J-PAKE by a PQC scheme, and providing PQC NIZK proofs for the secrets of the used scheme might very well yield a PQC instantiations of J-PAKE and similarly of Owl. An open question remains as to finding a suitable mechanisms for replacing the password exponentiations used to derive a shared key in J-PAKE, which is very likely to be coupled to the used PQC-based NIZK system and KEX. For a generic KEM, a similar issue to OPAQUE arises considering how to integrate said mechanism in black-box routines.

Threshold PAKEs

In order to mitigate the risks of offline dictionary attacks following server compromise, *Multi-Party Computation* (MPC) can be utilized to distribute a stored password among multiple servers. By doing so, an attacker needs to compromise more than one server instance bound by some threshold t to reconstruct server data, and hence the name Threshold PAKE, aka tPAKE [GJK$^+$25]. There already exist many approaches to realize symmetric tPAKEs from (t, n)-MPC protocols as in [Jab01, DRG03, ACFP05]. This idea underwent further development and became known as *Password-Protected Secret Sharing* (PPSS) or *Password-Authenticated Secret Sharing* (PASS) as in [BJSL11, CLN12, JKK14, JKKX16]. Recently, Gu et al. [GJK$^+$25] extended this approach to asymmetric settings and formalized the notion of (augmented) atPAKEs, where the password hash value is distributed among multiple servers using a secret-sharing scheme. Whereas tPAKEs are still non-present in PQ-settings, it is worth considering for a straight-forward adaptation as in the case of OPAQUE. This is mainly due to the fact that the construction in [GJK$^+$25] can be considered a threshold version of OPAQUE, as it mainly relies on a threshold OPRF (tOPRF). Nevertheless, we suffice with this short overview, as this specific type of PAKEs is clearly beyond the scope of this paper.

Decoy, Honey and Oblivious PAKEs

In line with the previously mentioned PAKE designs, another approach to reducing the threats of server compromise and credential leakage can be found in utilizing decoy accounts or decoy passwords [ARS24]. Interestingly, this type of PAKEs addresses the issues of password leakage in different manner than the previously mentioned designs (e.g., OPAQUE, J-PAKE, and tPAKEs). The original idea of Honey PAKEs, introduced by Becerra et al. [BRRS18], introduces decoy passwords (honeywords) that are utilized to detect password guesses. The real password is hence called the sugarword, and the set of all decoy passwords with the real one are the sweetwords [ARS24]. Similar approaches can also be found in the oblivious O-PAKE proposed by Kiefer and Manulis [KM15], as well as the HPAKE (as in honey) proposed by Li et al. [LWL22]. In a very recent publication, Arriaga et al. [ARS24] propose the SweetPAKE and BeePAKE (as in a honey producing buzzing bee) protocols building upon the afore mentioned works and comparing their results. Nonetheless, and again considering the scope of this paper, we suffice to mentioning the prior constructions as possible candidates for PQC adaptation, as they mostly provide generalized protocol frameworks in black-box manner.

10 Observations and Discussion

Design Paradigms and Schemes

Save for a few exceptions, we are yet to witness the birth of a new paradigm in PQC PAKE design that is more than a mere adoption or adaptation of classical ones. Further, other prominent classical PAKE designs such as in SPEKE (except for isogeny-based GA-PAKE), SPR-6, SPAKE, (Au)CPace, OPAQUE, and J-PAKE are currently non-present in the PQC realm, some of which we address in App. 9.

Design Paradigms. There are three mainstream trends in PQC PAKE design with strong focus on LWE, RLWE and MLWE schemes (cf. Tab. 1):

- PAKEs following the (O)EKE paradigm with focus on generic NIST PQC KEMs, and especially ML-KEM (C1).
- PAKEs following the PAK-suite paradigm using LWE and its variants directly and relying on different reconciliation mechanisms (C2 and C3).
- PAKEs following the KOY-GL paradigm using SPHF, APSH, and NIZK constructions from LWE and its variants (C2 and C3).

PQC Schemes. As observed in classes C2 and C3, There exists a plethora of RLWE-PAK and LWE ASPH PAKEs, but the main contribution is more or less restricted to performance or security improvements. Ultimately, it appears that the prevailing tendency is set on the lattices, especially because of the standardization of CRYSTALS-Kyber as ML-KEM. On the other hand, isogeny-based PAKEs are remarkably under-represented and there are no code-based PAKEs at all. Thus, it is worth investigating, whether more isogeny PAKEs are attainable; and whether code-based primitives are also a viable option.

Discussion. The lack of diversity in PQC PAKE designs can be traced back to the fact that many classical PAKE designs are tightly coupled to password usage within the DH paradigm. This is clearly seen in the absence of approaches relying on secret key exponentiation or base generator modification (H2G and H2C) using the password or a password derived value. Here, we differentiate between multiple cases w.r.t the chosen PQC scheme in a PAKE construction.

KEMs. Regardless of the underlying primitive, generic KEMs provide limited support to designs operating (mathematically speaking) on the PKE level. Such designs include base mapping subroutines, password-derived generators and secrets. This is due to the fact that the KEM algorithms are used as a black-box interface within the PAKE protocol, and should preferably not be used as a white-box. For instance, using the password within the ML-KEM key generation to obtain a new lattice base defeats the purpose of a generic design, and ultimately restricts the PAKE to concrete hardness assumptions.

Splittable KEMs. This desing had only been explored for KEMs based on lattice LWE primitives and variants (e.g., RLWE and MLWE). Still, splittable KEMs do maintain the provided algorithm interface in black-box manner, and may not directly support operations within the underlying PKE. Designs of splittable lattice-based KEM-PAKEs could also apply to KEMs from other primitives.

Lattices. Using PQC primitives directly offers more flexibility in, e.g., PAK-style LWE and RLWE PAKEs, where the H2G is used to shift the public key. Nevertheless, other approaches remain difficult to obtain from LWE and variants, as they do not support the exact operations as in discrete logarithm DH or ECDH. Another obstacle is that most RLWE schemes still require a reconciliation mechanism. Thus, it is worth investigating, whether PQC schemes like SWOOSH [GdKQ+24] can be used for a DH-like PAKE design.

Isogenies. While isogenies strongly resemble ECDH, classical approaches cannot be directly translated to SIDH and CSIDH, as sufficiently discussed in [AJK+20]. According to Azarderakhsh et al. [AJK+20], and unlike their classical DH-based counterparts, EKE-style, PAK-style, and J-PAKE style isogeny-based PAKEs are not secure. Further, the secuirty of SPEKE-style and Dragonfly-like isogeny PAKEs is questionable, since they are difficult to realize when hashing into public keys.

Design Simplicity (or Complexity)

We argue that generic constructions relying on proven abstract security properties rather than direct hardness assumptions (as discussed in Barbosa et al. [BGHJ24] and Januzelli et al. [JRX24]) may reduce the overall complexity, and even make the peer-reviewing process easier and more reliable. That is especially the case for PQC schemes and protocols, as their security has not matured to the level of classical cryptography yet. Further, generic designs enable realizing frameworks that support interchangeability and adaptability in the case of sudden breaks and the emergence of new threats.

Hybrid Schemes and Crypto-Agility

Several governmental bodies and institutions (e.g., NIST, BSI, and ETSI) recommend a transition to PQC in the near future. Nevertheless, they also recommend applying hybrid schemes (PQC combined with classical cryptography) due to the skepticism still surrounding the new PQC KEMs and digital signatures. In line with this recommendation, several hybrid KEMs were proposed (e.g., KEM Combiners [GHP18] and X-Wing [BCD+24]).

Generic Hybrids. There is still no clear answer to the question of finding a generic approach to construct a hybrid PAKE [KR24]. Hesse and Rosenberg [HR24], Lyu and Liu [LL24] (almost simultaneously), and Günther et al. [GRSV25] addressed this issue and proposed PQC PAKE combiners and hybrid PAKE frameworks.

Discussion. Said works provide generic recipes for hybrid PAKEs using parallel and sequential (or serial) combiners in UC. However, Hesse and Rosenberg show the impossibility of achieving a parallel combiner with minimal overhead using the existing PQC PAKEs. Similarly, Lyu and Liu claim that a parallel combiner requires both used PAKEs to satisfy the properties of a full DH-type PAKE, which so far can only by obtained from group action isogenies in the PQC realm. Optimally, hybrid PAKEs should also allow for an interchangeable KEM usage in plug-and-play manner and consequentially enable crypto-agility in sophisticated manner [ASW+22]. Nevertheless, Günther et al. [GRSV25] suggest relying on Obfuscated KEMs [GRSV25] (OKEM) to realize hybrid IND-CCA security and unconditional public key obfuscation for LWE-based OKEMs, which they utilize to construct a hybrid PAKE secure agaisnt adaptive corruption in UC.

Semantic Security

Some questions still surround the properties needed from PQC schemes to realize secure generic PAKEs. Most efforts, although not exclusively, address generic designs.

IND-CPA vs. IND-CCA. It was usually believed that CPA security is sufficient for building secure PAKEs [Jar22]. However, it was recently argued that CCA security is required for tighter proofs [ABJS24]. Further, Recent works suggested the notions of PR and SPR encompassing session key security w.r.t. to PAKE message flows invloving public keys and ciphertexts as well [MRR20, JRX24].

KEM Security Properties (for PAKEs). Generic KEM constructions need to additionally address novel notions such as public key uniformity, anonymity, and robustness. Although these notions are increasingly reaching a rather stable state in the literature (e.g., [Xag22, GMP22, MX23a, CDM23, Sch24, BCD+24, JRX24]).

Proof Models. Almost all PAKE designs still need to address the issues arising from classical IC and ROM usage. Moreover, efficient designs and proofs in the standard model are strongly wished for. This might however be extremely difficult to achieve with many schemes relying mainly on hashing and permutations.

Discussion. There is little consensus on the minimum core security of KEX or KEM in PAKEs. Further, it is yet to be seen how the potential of additional properties can be fully leveraged in generic PQC PAKE design. Moreover, based on the attained level of maturity in PQC PAKE design, it is not audacious to suggest that future constructions should not rely on non-quantum-safe models or assumptions. Surprisingly, the overwhelming majority of the PAKEs reviewed in this paper rely on the classical ROM, and only a few make use of abstract security notions (cf. Tab. 1 and Tab. 2). Whereas QROM proofs are starting to emerge (both for collision bounds and online extra tion), it is still not foreseeable when, and if the QIC is something that can be considered in the future.

Public Keys and Passwords

C1 EKE-like constructions encrypting the public key with an IC may suffer from vulnerability to offline dictionary attacks [AJK+20, AASA+24]. In C2 PAK-like constructions, the modified public key message distribution is directly dependent of the password due to the H2G password usage [AJK+20].

Discussion. Essentially, public keys need to be indistinguishable from random bit strings, which may not be the case for PQC keys yielding a certain structure [AJK+20, GRSV25] such as, ML-KEM keys [AASA+24]. The public key uniformity notion is meant to address this issue in theory, however practical instantiations and implementations need to either work around this problem, or find a way to make the keys unstructured in some sense (e.g., obfuscation [AASA+24]). Further, extracting information about the password from the public key message is assumed to be infeasible, yet it may still be possible to obtain partial information about the used keys, as discussed in the following for leakage attacks.

Tight Bounds and Reductions

Reducing the need for idealized objects, or providing proofs in QROM and QICM is needed for almost all PQC PAKEs (cf. Tab. 1). In the absence of an IC, it is difficult to prove knowledge of any information about an adversary's interactions with public keys in security proofs [AASA+24, ABJ25, HHKR25]. Thus, it might be hard to enumerate passwords that can be ruled-out through active password guessing. For honest key generation, it is rather easy to deal with guesses targeting key derivations or hashing on the final key. However, password guesses under unknown secret keys are difficult to handle, where the initiator's key pair was generated maliciously. Hence, it is hard to formulate a hardness assumption bound to the number of password guessing queries for an unknown number of malicious key pairs [AASA$^+$24, HHKR25].

Discussion. PQC KEMs mainly differ from bare primitives in the fact that they cannot yet apply self-reducible [KTAT20] properties to remove multiplicative factors in multiple-queries for certain security properties [ACH$^+$24]. Random self-reducibility allows for independent instances of a property to be reduced to one, thus leading to tighter reductions. For instance, the multiplicative factor for the number of queries on an IC directly affects the bounds on public key encryptions [PZ23, AHHR24]. A PQC KEM with a tight proof of multi-instance security would solve this problem. For a RO, one should consider how to deal with queries on public key modifying or masking operations that allow for offline dictionary attacks. As for the QROM, allowing adversaries to perform superposition queries makes it hard for a simulator to extract pre-images or reprogram ROs [LLH25]. Still, RO reprogramming is probably not necessary (at least not in UC proofs [Hes20, MX23b, LLH25]), but the QROM still needs to simulate real session keys using a decision oracle. Some works argue that classical output transfer is acceptable if the extraction is possible using recent online extractability techniques [HHKR25]. An open question is however if quantum rewinding is applicable in PAKE proofs that rely on rewinding or back-patching (e.g., relaxed security models [ABB$^+$20, HTTY24]).

Sufficient Analysis and Unknown Attack Surfaces

As observed in LBA-PAKE [DBK20], the authors attempted fixing vulnerabilities in a previous construction [FHZ$^+$18], yet introduced a signal leakage attack on the PWE assumption that lead to full key recovery [DCQ22]. This is also seen in BP-3PAKA [IB21], which was addressed in follow-up works by Chaudhary et al. [CKS23], Kumar et al. [KGKD23], Dadsena et al. [DJRD23], and Dharminder et al. [DRD$^+$23]. The common denominator among these examples is that they all suffered from signal leakage attacks.

Known Attacks. Most anonymous PQC PAKEs suffer from vulnerabilities against impersonation, stolen smart-card, and password guessing attacks; and sometimes even fail to provide user anonymity or non-traceability (cf. Sec. 7.2). Furthermore, many augmented PAKEs do not explicitly address the possibility of pre-computation attacks, where an attacker can leverage the password dictionary for an offline pre-computation attack before compromising the server (cf. Sec. 7.2).

New Attacks. New attacks are not restricted to the used security models, but can also target the underlying schemes cryptographically and physically (cf. Sec. 7.2).

Discussion. Assuming the soundness of a certain hardness assumption and its usage in proof reductions (e.g., PWE to LWE via H2G), works building upon it need to pay special attention to practical threats, and not only theoretical assumptions. Moreover, since anonymous communication is required in real world use cases such as eIDs and e-healthcare [ASWZ24], we believe that the work on constructions with more rigorous formal and physical analysis is required.

Implementations, Performance and Standardization

Since research surrounding PQC PAKEs is still in its early stages, there are not sufficient driving factors (e.g., NIST PQC standardization etc.). Thus, comparing performance and finding a common benchmarking baseline is an open problem.

Thorough Evaluation. Testing and benchmarking projects are missing in PQC PAKE research. Also, many works do not provide clear experimental results and do not directly address the security guarantees and assumptions of their constructions (cf. Sec. 7.7). Further, some experimental results and benchmarks, especially for LWE and RLWE seem unreasonable or rather unconvincing.

Evaluation Comparison. Different papers providing performance indicators widely differ in the measurement setup and used units (e.g., clock cycles, milliseconds or computational cost etc.), and the benchmarking environments also differ in their computational capabilities (cf. Sec. 7.7). Hence, it is extremely difficult to make any accurate statements regarding which constructions is more secure or can perform better under which circumstances.

Discussion. Similar to projects aimed at PQC KEMs such as the Open Quantum Safe (OQS) project [SM16] and pqm4 [KRSS19], there is an urgent need to make an effort to provide reproducible performance benchmarks for different PQC PAKEs on various platforms, making them also comparable and comprehensive.

11 Conclusion and Future Work

In this paper, we extensively reviewed and systematized nearly 50 PAKEs based on PQC hardness assumptions and KEMs. Conclusively, only a small percentage of these works offer ground-breaking novelty in terms of design paradigms, as the major contributions are focused on optimizations of existing ones. Further, most PQC PAKEs heavily rely on the RO and the IC models, and need to either address these models in quantum settings, or eliminate their usage completely to achieve real post-quantum security. Nevertheless, some works were able to shed some light on undiscovered security issues or unknown attack surfaces. Compared to direct approaches, generic PAKEs utilizing PQC KEMs seem more attractive, due to their simpler designs and manageable interfaces. However, they often pose more requirements w.r.t. the security notions of the used KEMs.

Based on our review, a pressing order of business for future work would be an official project providing a unified hardware and software framework for PQC PAKE implementation and benchmarking. Further, a such project can actively integrate and test PQC PAKEs in real world applications in order to evaluate their claimed suitability for certain use cases, and investigate their physical security. Moreover, hybrid schemes can also be considered in such evaluations to assess their applicability and feasibility for future cryptographic migrations. On the theoretical side, future PQC PAKEs have to consciously aim at designs with crypto-agility, where building blocks are at least interchangeable, if not update-able. That being said, the most urgent future work concerns these very building blocks. As previously observed, we believe that KEM security properties and idealized objects should get special attention. That is on the one hand to actively involve abstract security notions in PAKEs for tighter reductions and simpler designs. On the other hand, to follow up on quantum lifting techniques for the IC and the ROM. A slightly different approach could be found in designing PAKEs that are non-reliant on such models, which could be the more difficult path, yet the one bearing the most fruit.

References

[AASA$^+$24] Nouri Alnahawi, Jacob Alperin-Sheriff, Daniel Apon, Gareth T. Davies, and Alexander Wiesmaier. NICE-PAKE: On the security of KEM-based PAKE constructions without ideal ciphers. Cryptology ePrint Archive, Paper 2024/1957, 2024.

[ABB$^+$20] Michel Abdalla, Manuel Barbosa, Tatiana Bradley, Stanisław Jarecki, Jonathan Katz, and Jiayu Xu. Universally composable relaxed password authenticated key exchange. In *Annual International Cryptology Conference*, pages 278–307. Springer, 2020.

[Abd14] Michel Abdalla. Password-based authenticated key exchange: An overview. In *Provable Security: 8th International Conference, ProvSec 2014*, 2014.

[ABJ25] Afonso Arriaga, Manuel Barbosa, and Stanislaw Jarecki. NoIC: PAKE from KEM without ideal ciphers. Cryptology ePrint Archive, Paper 2025/231, 2025.

[ABJS24] Afonso Arriaga, Manuel Barbosa, Stanislaw Jarecki, and Marjan Skrobot. C'est très chic: A compact password-authenticated key exchange from lattice-based kem, 2024.

[ABP15] Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Public-key encryption indistinguishable under plaintext-checkable attacks. In *Public-Key Cryptography – PKC 2015*, 2015.

[ABPS23] Gorjan Alagic, Chen Bai, Alexander Poremba, and Kaiyan Shi. On the two-sided permutation inversion problem. *arXiv preprint arXiv:2306.13729*, 2023.

[ABR$^+$21] Michel Abdalla, Manuel Barbosa, Peter B. Rønne, Peter Y. A. Ryan, and Petra Šala. Security Characterization of J-PAKE and its Variants, 2021.

[ACFP05] Michel Abdalla, Olivier Chevassut, Pierre-Alain Fouque, and David Pointcheval. A simple threshold authenticated key exchange from short secrets. In *Advances in Cryptology-ASIACRYPT 2005: 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005. Proceedings 11*, pages 566–584. Springer, 2005.

[ACH$^+$24] Joël Alwen, Matthew Campagna, Dominik Hartmann, Shuichi Katsumata, Eike Kiltz, Jake Massimo, Marta Mularczyk, Guillermo Pascual-Perez, Thomas Prest, and Peter Schwabe. How multi-recipient kems can help the deployment of post-quantum cryptography. *Fifth PQC Standardization Conference*, 2024.

[ADDG24] Martin R Albrecht, Alex Davidson, Amit Deo, and Daniel Gardham. Crypto dark matter on the torus: Oblivious prfs from shallow prfs and tfhe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2024.

[ADDS21] Martin R Albrecht, Alex Davidson, Amit Deo, and Nigel P Smart. Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. In *IACR International Conference on Public-Key Cryptography*, 2021.

[AEK$^+$22a] Michel Abdalla, Thorsten Eisenhofer, Eike Kiltz, Sabrina Kunzweiler, and Doreen Riepel. Password-authenticated key exchange from group actions. In *Annual international cryptology conference*, pages 699–728. Springer, 2022.

[AEK$^+$22b] Michel Abdalla, Thorsten Eisenhofer, Eike Kiltz, Sabrina Kunzweiler, and Doreen Riepel. Password-Authenticated Key Exchange from Group Actions. In *Advances in Cryptology – CRYPTO 2022*, 2022.

[AFP05] Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-Based Authenticated Key Exchange in the Three-Party Setting. In *Public Key Cryptography - PKC 2005*, 2005.

[AHHR24] Nouri Alnahawi, Kathrin Hövelmanns, Andreas Hülsing, and Silvia Ritsch. Towards post-quantum secure pake-a tight security proof for ocake in the bpr model. In *International Conference on Cryptology and Network Security*, pages 191–212. Springer, 2024.

[AJK$^+$20] Reza Azarderakhsh, David Jao, Brian Koziel, Jason T. LeGrow, Vladimir Soukharev, and Oleg Taraskin. How Not to Create an Isogeny-Based PAKE. In *Applied Cryptography and Network Security*, 2020.

[Als16] Saed A Alsayigh. *New Password Authenticated Key Exchange Based on the Ring Learning with Errors.* PhD thesis, University of Cincinnati, 2016.

[AR17] Gorjan Alagic and Alexander Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. In *Annual international conference on the theory and applications of cryptographic techniques*, 2017.

[ARS24] Afonso Arriaga, Peter YA Ryan, and Marjan Skrobot. Sweetpake: Key exchange with decoy passwords. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 1017–1033, 2024.

[ASW$^+$22] N. Alnahawi, N. Schmitt, A. Wiesmaier, A. Heinemann, and T. Graßmeyer. On the State of Crypto-Agility. In *18. Deutscher IT-Sicherheitskongress*. SecuMedia Verlags-GmbH, February 2022. Preprint: https://ia.cr/2023/487.

[ASWZ24] Nouri Alnahawi, Nicolai Schmitt, Alexander Wiesmaier, and Chiara-Marie Zok. Toward next generation quantum-safe eids and emrtds: A survey. *ACM Trans. Embed. Comput. Syst.*, 23, 2024.

[Bas23] Andrea Basso. A post-quantum round-optimal oblivious PRF from isogenies. Cryptology ePrint Archive, Paper 2023/225, 2023.

[BBDQ18] Fabrice Benhamouda, Olivier Blazy, Léo Ducas, and Willy Quach. Hash proof systems over lattices revisited. In *IACR International Workshop on Public Key Cryptography*, 2018.

[BBF$^+$19] Nina Bindel, Jacqueline Brendel, Marc Fischlin, Brian Goncalves, and Douglas Stebila. Hybrid key encapsulation mechanisms and authenticated key exchange. In *Post-Quantum Cryptography: 10th International Conference, PQCrypto 2019*, 2019.

[BCD$^+$16] Joppe Bos, Craig Costello, Leo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[BCD$^+$24] Manuel Barbosa, Deirdre Connolly, João Diogo Duarte, Aaron Kaiser, Peter Schwabe, Karolin Varner, and Bas Westerbaan. X-wing. *IACR Communications in Cryptology*, 1(1), 2024.

[BCMR19] Erica Blum, Makana Castillo-Martin, and Michael Rosenberg. Survey on the security of the quantum rom. 2019.

[BCP$^+$23] Hugo Beguinet, Céline Chevalier, David Pointcheval, Thomas Ricosset, and Mélissa Rossi. Get a cake: Generic transformations from key encaspulation mechanisms to password authenticated key exchanges. In *International Conference on Applied Cryptography and Network Security*, 2023.

[BDF$^+$11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Advances in Cryptology–ASIACRYPT 2011*, 2011.

[BDFH24] Ward Beullens, Lucas Dodgson, Sebastian Faller, and Julia Hesse. The 2hash oprf framework and efficient post-quantum instantiations. *Cryptology ePrint Archive*, 2024.

[BDK$^+$18] Joppe Bos, Leo Ducas, Eike Kiltz, T Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO '98*, 1998.

[Ber09] Daniel J. Bernstein. Introduction to post-quantum cryptography. In *Post-Quantum Cryptography*, 2009.

[BFH$^+$23] Jonathan Bootle, Sebastian Faller, Julia Hesse, Kristina Hostáková, and Johannes Ottenhues. Generalized fuzzy password-authenticated key exchange from error correcting codes. In *Asiacrypt*, 2023.

[BFK09] Jens Bender, Marc Fischlin, and Dennis Kügler. Security Analysis of the PACE Key-Agreement Protocol. In *Information Security*, 2009.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, New York, NY, USA, 1988. Association for Computing Machinery.

[BGHJ24] Manuel Barbosa, Kai Gellert, Julia Hesse, and Stanislaw Jarecki. Bare pake: universally composable key exchange from just passwords. In *Annual International Cryptology Conference*, pages 183–217. Springer, 2024.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

[BJSL11] Ali Bagherzandi, Stanislaw Jarecki, Nitesh Saxena, and Yanbin Lu. Password-protected secret sharing. In *Proceedings of the 18th ACM conference on Computer and Communications Security*, pages 433–444, 2011.

[BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pages 520–550. Springer, 2020.

[BL17] Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549, 2017.

[Bla06] John Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. In *Fast Software Encryption: 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers 13*, 2006.

[BM92] S.M. Bellovin and M. Merritt. Encrypted key exchange: password-based protocols secure against dictionary attacks. In *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, 1992.

[BM25] Andrea Basso and Luciano Maino. Poke: A compact and efficient pke from higher-dimensional isogenies. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2025.

[BMP00] Victor Boyko, Philip MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using diffie-hellman. In *Advances in Cryptology—EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques Bruges, Belgium, May 14–18, 2000 Proceedings 19*, pages 156–171. Springer, 2000.

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated Key Exchange Secure against Dictionary Attacks. In *Advances in Cryptology — EUROCRYPT 2000*, 2000.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993.

[BRRS18] José Becerra, Peter B Rønne, Peter YA Ryan, and Petra Sala. Honeypakes. In *Security Protocols XXVI: 26th International Workshop, Cambridge, UK, March 19–21, 2018, Revised Selected Papers 26*, pages 63–77. Springer, 2018.

[BSIA23] Swagatam Basu, Kübra Seyhan, SK Hafizul Islam, and Sedat Akleylek. Mlwr-2paka: A hybrid module learning with rounding-based authenticated key agreement protocol for two-party communication. *IEEE Systems Journal*, 2023.

[BSP$^+$24] Pia Bauspieß, Tjerand Silde, Matej Poljuha, Alexandre Tullot, Anamaria Costache, Christian Rathgeb, Jascha Kolberg, and Christoph Busch. Brake: Biometric resilient authenticated key exchange. *IEEE Access*, 2024.

[CAK$^+$18] Rakyong Choi, Hyeongcheol An, Kwangjo Kim, et al. AtLast: another three-party lattice-based PAKE scheme. In *Proceedings of the 2018 Symposium on Cryptography and Information Security (SCIS 2018)*, 2018.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, 2001.

[CCH$^+$19] Matthew Campagna, Craig Costello, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, David Urbanik, et al. Supersingular isogeny key encapsulation. https://www.sike.org/files/SIDH-spec.pdf, 2019.

[CCJ$^+$16] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on Post-Quantum Cryptography. US Department of Commerce, National Institute of Standards and Technology, 2016.

[CDM23] Cas Cremers, Alexander Dax, and Niklas Medinger. Keeping up with the kems: Stronger security notions for kems and automated analysis of kem-based protocols. 2023.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4), 2004.

[CHL22] Sílvia Casacuberta, Julia Hesse, and Anja Lehmann. Sok: oblivious pseudorandom functions. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 625–646. IEEE, 2022.

[CK01] Ran Canetti and Hugo Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *Advances in Cryptology – EUROCRYPT 2001*, 2001.

[CKS23] Dharminder Chaudhary, Uddeshaya Kumar, and Kashif Saleem. A Construction of Three Party Post Quantum Secure Authenticated Key Exchange Using Ring Learning with Errors and ECC Cryptography. *IEEE Access*, 2023.

[CL24] Zhengjun Cao and Lihua Liu. A note on "a new password-authenticated module learning with rounding-based key exchange protocol: Saber.PAKE". Cryptology ePrint Archive, Paper 2024/683, 2024.

[CLM$^+$18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *Advances in Cryptology–ASIACRYPT 2018*, 2018.

[CLN12] Jan Camenisch, Anna Lysyanskaya, and Gregory Neven. Practical yet universally composable two-server password-authenticated secret sharing. In *Proceedings of the 2012 ACM conference on Computer and Communications Security*, pages 525–536, 2012.

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Paper 2019/428, 2019.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*, 2002.

[DAL$^+$17] Jintai Ding, Saed Alsayigh, Jean Lancrenon, Saraswathy RV, and Michael Snook. Provably Secure Password Authenticated Key Exchange Based on RLWE for the Post-Quantum World. In *Topics in Cryptology – CT-RSA 2017*, 2017.

[DBK20] Vivek Dabra, Anju Bala, and Saru Kumari. LBA-PAKE: Lattice-based anonymous password authenticated key exchange for mobile devices. *IEEE Systems Journal*, 2020.

[DBK21] Vivek Dabra, Anju Bala, and Saru Kumari. LBA-PAKE: Lattice-Based Anonymous Password Authenticated Key Exchange for Mobile Devices. *IEEE Systems Journal*, 2021.

[DCQ22] Ruoyu Ding, Chi Cheng, and Yue Qin. Further analysis and improvements of a lattice-based anonymous PAKE scheme. *IEEE Systems Journal*, 2022.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *Symposium on the Theory of Computing*, 1991.

[DdSGP24] Cyprien Delpech de Saint Guilhem and Robi Pedersen. New proof systems and an oprf from csidh. In *IACR International Conference on Public-Key Cryptography*, pages 217–251. Springer, 2024.

[DF11] Yi Ding and Lei Fan. Efficient password-based authenticated key exchange from lattices. In *2011 Seventh International Conference on Computational Intelligence and Security*, 2011.

[DFG$^+$23] Gareth T Davies, Sebastian Faller, Kai Gellert, Tobias Handirk, Julia Hesse, Máté Horváth, and Tibor Jager. Security analysis of the whatsapp end-to-end encrypted backup protocol. In *Annual International Cryptology Conference*, pages 330–361. Springer, 2023.

[DFMS22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 677–706. Springer, 2022.

[DGJ23] Bruno F. Dos Santos, Yanqi Gu, and Stanislaw Jarecki. Randomized half-ideal cipher on groups with applications to uc (a)pake. In *Advances in Cryptology – EUROCRYPT 2023*. Springer Nature Switzerland, 2023.

[DGJK22] Bruno F. Dos Santos, Yanqi Gu, Stanislaw Jarecki, and Hugo Krawczyk. Asymmetric PAKE with Low Computation and communication. In *Advances in Cryptology – EUROCRYPT 2022*, 2022.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 1976.

[DHP$^+$18] Pierre-Alain Dupont, Julia Hesse, David Pointcheval, Leonid Reyzin, and Sophia Yakoubov. Fuzzy password-authenticated key exchange. In *Eurocrypt*, 2018.

[Din17] Jintai Ding. Password based key exchange from ring learning with errors, 2017.

[DJRD23] Pradeep Kumar Dadsena, Jainendra Jain, Saurabh Rana, and Dharminder Dharminder. A construction of post quantum secure authenticated key agreement design for mobile digital rights management system. *Multimedia Tools and Applications*, 82, 2023.

[DKSRV18] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure kem. In *Progress in Cryptology–AFRICACRYPT 2018*, 2018.

[DRD$^+$23] Dharminder Dharminder, Challa Bhageeratha Reddy, Ashok Kumar Das, Youngho Park, and Sajjad Shaukat Jamal. Post-quantum lattice-based secure reconciliation enabled key agreement protocol for iot. *IEEE Internet of Things Journal*, 2023.

[DRG03] Mario Di Raimondo and Rosario Gennaro. Provably secure threshold password-authenticated key exchange. In *Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22*, pages 507–523. Springer, 2003.

[DXL12] Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. *IACR Cryptol. ePrint Arch. 2012/688*, 2012.

[DY83] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 1983.

[EHOR20] Andreas Erwig, Julia Hesse, Maximilian Orlt, and Siavash Riahi. Fuzzy asymmetric password-authenticated key exchange. In *Asiacrypt*, 2020.

[ES21] Edward Eaton and Douglas Stebila. The "quantum annoying" property of password-authenticated key exchange protocols. In *Post-Quantum Cryptography: 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, Proceedings 12*, pages 154–173. Springer, 2021.

[FHZ$^+$18] Qi Feng, Debiao He, Sherali Zeadally, Neeraj Kumar, and Kaitai Liang. Ideal lattice-based anonymous authentication protocol for mobile devices. *IEEE Systems Journal*, 2018.

[FIPR05] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *Proceedings of the Second International Conference on Theory of Cryptography*, TCC'05, page 303–324, Berlin, Heidelberg, 2005. Springer-Verlag.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, Lecture Notes in Computer Science, 1999.

[FOO23] Sebastian Faller, Astrid Ottenhues, and Johannes Ottenhues. Composable oblivious pseudo-random functions via garbled circuits. In *International Conference on Cryptology and Information Security in Latin America*, 2023.

[For19] Internet Engineering Task Force. PAKE Selection Process. IETF, 2019.

[FTTY19] Atsushi Fujioka, Katsuyuki Takashima, Shintaro Terada, and Kazuki Yoneyama. Supersingular isogeny diffie–hellman authenticated key exchange. In *Information Security and Cryptology–ICISC 2018*, 2019.

[GdKQ+24] Phillip Gajland, Bor de Kock, Miguel Quaresma, Giulio Malavolta, and Peter Schwabe. {SWOOSH}: Efficient {Lattice-Based}{Non-Interactive} key exchange. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 487–504, 2024.

[GDL+17] Xinwei Gao, Jintai Ding, Lin Li, RV Saraswathy, and Jiqiang Liu. Efficient implementation of password-based authenticated key exchange from RLWE and post-quantum TLS. *IACR Cryptol. ePrint Arch. 2017/1192*, 2017.

[GDLL17] Xinwei Gao, Jintai Ding, Jiqiang Liu, and Lin Li. Post-Quantum Secure Remote Password Protocol from RLWE Problem. In *Information Security and Cryptology*, 2017.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GHP18] Federico Giacon, Felix Heuer, and Bertram Poettering. Kem combiners. In *Public-Key Cryptography–PKC 2018: 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I 21*, pages 190–218. Springer, 2018.

[GJK21] Yanqi Gu, Stanislaw Jarecki, and Hugo Krawczyk. KHAPE: Asymmetric PAKE from Key-Hiding Key Exchange. In *Advances in Cryptology – CRYPTO 2021*, 2021.

[GJK+25] Yanqi Gu, Stanislaw Jarecki, Pawel Kedzior, Phillip Nazarian, and Jiayu Xu. Threshold pake with security against compromise of all servers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 66–100. Springer, 2025.

[Gjö24] Kristian Gjösteen. Password-authenticated key exchange and applications. Cryptology ePrint Archive, Paper 2024/1057, 2024.

[GL03] Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. In *International Conference on the Theory and Applications of Cryptographic Techniques*, 2003.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 1984.

[GMP22] Paul Grubbs, Varun Maram, and Kenneth G Paterson. Anonymous, robust post-quantum public key encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2022.

[GMR85] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, New York, NY, USA, 1985. Association for Computing Machinery.

[GMR06] Craig Gentry, Philip MacKenzie, and Zulfikar Ramzan. A method for making password-based key exchange resilient to server compromise. In *Annual International Cryptology Conference*, pages 142–159. Springer, 2006.

[GO14] Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. *Journal of cryptology*, 27(3):506–543, 2014.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, Philadelphia, Pennsylvania, United States, 1996.

[GRSV25] Felix Günther, Michael Rosenberg, Douglas Stebila, and Shannon Veitch. Hybrid obfuscated key exchange and KEMs. Cryptology ePrint Archive, Paper 2025/408, 2025.

[GSG$^+$23] Songhui Guo, Yunfan Song, Song Guo, Yeming Yang, and Shuaichao Song. Three-Party Password Authentication and Key Exchange Protocol Based on MLWE. *Symmetry*, 2023.

[Har08] Dan Harkins. Simultaneous authentication of equals: A secure, password-based key exchange for mesh networks. In *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, 2008.

[HBCvO23] Feng Hao, Samiran Bag, Liqun Chen, and Paul C. van Oorschot. Owl: An augmented password-authenticated key exchange scheme. Cryptology ePrint Archive, Paper 2023/768, 2023.

[Hes20] Julia Hesse. Separating symmetric and asymmetric password-authenticated key exchange. In *International Conference on Security and Cryptography for Networks*, pages 579–599. Springer, 2020.

[HHKR25] Kathrin Hövelmanns, Andreas Hülsing, Mikhail Kudinov, and Silvia Ritsch. CAKE requires programming - on the provable post-quantum security of (o)CAKE. Cryptology ePrint Archive, Paper 2025/458, 2025.

[HHM$^+$24] Lena Heimberger, Tobias Hennerbichler, Fredrik Meisingseth, Sebastian Ramacher, and Christian Rechberger. Oprfs from isogenies: designs and analysis. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 575–588, 2024.

[HM24] Kathrin Hövelmanns and Christian Majenz. A note on failing gracefully: completing the picture for explicitly rejecting fujisaki-okamoto transforms using worst-case correctness. In *International Conference on Post-Quantum Cryptography*, pages 245–265. Springer, 2024.

[HPA21a] James Howe, Thomas Prest, and Daniel Apon. Sok: How (not) to design and implement post-quantum cryptography. In *Cryptographers' Track at the RSA Conference*, pages 444–477. Springer, 2021.

[HPA21b] James Howe, Thomas Prest, and Daniel Apon. Sok: How (not) to design and implement post-quantum cryptography. In *Cryptographers' Track at the RSA Conference*, 2021.

[HR10] Feng Hao and Peter Ryan. J-pake: authenticated key exchange without pki. *Transactions on Computational Science XI: Special Issue on Security in Computing, Part II*, 2010.

[HR24] Julia Hesse and Michael Rosenberg. PAKE combiners and efficient post-quantum instantiations. Cryptology ePrint Archive, Paper 2024/1621, 2024.

[HS14] Feng Hao and Siamak F Shahandashti. The speke protocol revisited. In *International Conference on Research in Security Standardisation*, 2014.

[HTTY24] Shuya Hanai, Keisuke Tanaka, Masayuki Tezuka, and Yusuke Yoshida. Universally composable relaxed asymmetric password-authenticated key exchange. In *International Conference on Security and Cryptography for Networks*, pages 272–293. Springer, 2024.

[HvO22] Feng Hao and Paul C van Oorschot. Sok: Password-authenticated key exchange–theory, practice, standardization and real-world lessons. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 2022.

[HXL$^+$23] Yunxia Han, Chunxiang Xu, Shanshan Li, Changsong Jiang, and Kefei Chen. ttpake: Typo tolerance password-authenticated key exchange. *Journal of Information Security and Applications*, 79:103658, 2023.

[HY18] Akinori Hosoyamada and Kan Yasuda. Building quantum-one-way functions from block ciphers: Davies-meyer and merkle-damgård constructions. In *Advances in Cryptology–ASIACRYPT 2018*, 2018.

[IB21] SK Hafizul Islam and Swagatam Basu. PB-3PAKA: Password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environments. *Journal of Information Security and Applications*, 2021.

[IEE09] IEEE. IEEE Standard Specification for Password-Based Public-Key Cryptographic Techniques. *IEEE Std 1363.2-2008*, 2009.

[ISO17] ISO/IEC. Part 4: Mechanisms based on weak secrets. volume 11770-4. Standard, International Organization for Standardization, Geneva, CH, 2017.

[Jab96] David P. Jablon. Strong password-only authenticated key exchange. *ACM SIGCOMM Computer Communication Review*, 1996.

[Jab01] David P Jablon. Password authentication using multiple servers. In *Cryptographers' Track at the RSA Conference*, pages 344–360. Springer, 2001.

[Jar22] Stanislaw Jarecki. Password Authenticated Key Exchange: Protocols and Security Models. *Asymmetric Cryptography: Primitives and Protocols*, 2022.

[JGH$^+$20] Shaoquan Jiang, Guang Gong, Jingnan He, Khoa Nguyen, and Huaxiong Wang. PAKEs: new framework, new techniques and more efficient lattice-based constructions in the standard model. In *IACR International Conference on Public-Key Cryptography*, 2020.

[JKK14] Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk. Round-optimal password-protected secret sharing and t-pake in the password-only model. In *Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014, Proceedings, Part II 20*, pages 233–253. Springer, 2014.

[JKKX16] Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online). In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 276–291. IEEE, 2016.

[JKX18] Stanislaw Jarecki, Hugo Krawczyk, and Jiayu Xu. OPAQUE: An Asymmetric PAKE Protocol Secure Against Pre-computation Attacks. In *Advances in Cryptology – EUROCRYPT 2018*, 2018.

[JMV01] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1:36–63, 2001.

[JRX24] Jake Januzelli, Lawrence Roy, and Jiayu Xu. Under what conditions is encrypted key exchange actually secure? Cryptology ePrint Archive, Paper 2024/324, 2024.

[JTCW18] Yi-Siou Jheng, Raylin Tso, Chien-Ming Chen, and Mu-En Wu. Password-Based Authenticated Key Exchange from Lattices for Client/Server Model. In *Advances in Computer Science and Ubiquitous Computing*, 2018.

[JZ16] Zhengzhong Jin and Yunlei Zhao. Optimal key consensus in presence of noise. *arXiv preprint arXiv:1611.06150*, 2016.

[KAA19] Amir Hassani Karbasi, Reza Ebrahimi Atani, and Shahabaddin Ebrahimi Atani. A New Ring-Based SPHF and PAKE Protocol on Ideal Lattices. *ISeCure*, 2019.

[KCM24] Novak Kaluderovic, Nan Cheng, and Katerina Mitrokotsa. A post-quantum distributed OPRF from the legendre PRF. Cryptology ePrint Archive, Paper 2024/544, 2024.

[KGKD23] Uddeshaya Kumar, Manish Garg, Saru Kumari, and Dharminder Dharminder. A construction of post quantum secure and signal leakage resistant authenticated key agreement protocol for mobile communication. *Transactions on Emerging Telecommunications Technologies*, 34(1), 2023.

[KKZZ14] Jonathan Katz, Aggelos Kiayias, Hong-Sheng Zhou, and Vassilis Zikas. Distributing the setup in universally composable multi-party computation. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, PODC '14, page 20–29, New York, NY, USA, 2014. Association for Computing Machinery.

[KM15] Franziskus Kiefer and Mark Manulis. Oblivious pake: Efficient handling of password trials. In *International Conference on Information Security*, pages 191–208. Springer, 2015.

[KMP$^+$17] Vladimir Kolesnikov, Naor Matania, Benny Pinkas, Mike Rosulek, and Ni Trieu. Practical multi-party private set intersection from symmetric-key techniques. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1257–1272, 2017.

[KNW18] Michael Kreutzer, Ruben Niederhagen, and Michael Waidner. Eberbacher Gespräch on Next Generation Crypto. *Fraunhofer SIT*, 2018.

[KOY01] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient password-authenticated key exchange using human-memorable passwords. In *International Conference on the Theory and Applications of Cryptographic Techniques*, 2001.

[KR24] Jonathan Katz and Michael Rosenberg. Latke: a framework for constructing identity-binding pakes. In *Annual International Cryptology Conference*, pages 218–250. Springer, 2024.

[KRSS19] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. pqm4: Testing and benchmarking NIST PQC on ARM cortex-m4. Cryptology ePrint Archive, Paper 2019/844, 2019.

[KTAT20] Tomoki Kawashima, Katsuyuki Takashima, Yusuke Aikawa, and Tsuyoshi Takagi. An efficient authenticated key exchange from random self-reducibility on CSIDH. Cryptology ePrint Archive, Paper 2020/1178, 2020.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange from lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, 2009.

[LL24] You Lyu and Shengli Liu. Hybrid password authentication key exchange in the UC framework. Cryptology ePrint Archive, Paper 2024/1630, 2024.

[LLH24] You Lyu, Shengli Liu, and Shuai Han. Universal composable password authenticated key exchange for the post-quantum world. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2024.

[LLH25] You Lyu, Shengli Liu, and Shuai Han. Efficient asymmetric pake compiler from kem and ae. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 34–65. Springer, 2025.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 1–23. Springer, 2010.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.

[LST16] Jean Lancrenon, Marjan Skrobot, and Qiang Tang. Two more efficient variants of the J-PAKE protocol. In *Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings*, volume 9696 of *Lecture Notes in Computer Science*, pages 58–76. Springer, 2016.

[LW18] Zengpeng Li and Ding Wang. Two-Round PAKE Protocol over Lattices Without NIZK. In *International Conference on Information Security and Cryptology*. Springer, 2018.

[LW19] Zengpeng Li and Ding Wang. Achieving One-Round Password-based Authenticated Key Exchange over Lattices. *IEEE Transactions on Services Computing*, 2019.

[LWL22] Wenting Li, Ping Wang, and Kaitai Liang. Hpake: Honey password-authenticated key exchange for fast and safer online authentication. *IEEE Transactions on Information Forensics and Security*, 18:1596–1609, 2022.

[LWM22] Zengpeng Li, Ding Wang, and Eduardo Morais. Quantum-Safe Round-Optimal Password Authentication for Mobile Devices. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[LZJY19] Chao Liu, Zhongxiang Zheng, Keting Jia, and Qidi You. Provably secure three-party password-based authenticated key exchange from rlwe. In *Information Security Practice and Experience: 15th International Conference, ISPEC 2019*, 2019.

[Mac02] Philip MacKenzie. The PAK suite: Protocols for Password-Authenticated Key Exchange. In *Ieee P1363.2*, 2002.

[MMW24] Christian Majenz, Giulio Malavolta, and Michael Walter. Permutation superposition oracles for quantum query lower bounds. *arXiv preprint arXiv:2407.09655*, 2024.

[MRR20] Ian McQuoid, Mike Rosulek, and Lawrence Roy. Minimal symmetric pake and 1-out-of-n ot from programmable-once public functions. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 425–442, 2020.

[MX23a] Varun Maram and Keita Xagawa. Post-quantum anonymity of kyber. In *IACR International Conference on Public-Key Cryptography*. Springer, 2023.

[MX23b] Ian McQuoid and Jiayu Xu. An efficient strong asymmetric pake compiler instantiable from group actions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 176–207. Springer, 2023.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudorandom functions. *J. ACM*, 51(2):231–262, 2004.

[NY90] M Naor and M Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing - STOC '90*, 1990.

[OP01] Tatsuaki Okamoto and David Pointcheval. React: Rapid enhanced-security asymmetric cryptosystem transform. In *Topics in Cryptology — CT-RSA 2001*, 2001.

[Ott24] Johannes Ottenhues. An overview of symmetric fuzzy pake protocols. *Sicherheit 2024*, 2024.

[Pei14] Chris Peikert. Lattice Cryptography for the Internet. In *Post-Quantum Cryptography*, 2014.

[Poi22] David Pointcheval. *Asymmetric Cryptography: Primitives and Protocols.* John Wiley & Sons, Inc. Hoboken, NJ, USA, 2022.

[PZ23] Jiaxin Pan and Runzhi Zeng. A generic construction of tightly secure password-based authenticated key exchange. In *International Conference on the Theory and Application of Cryptology and Information Security*, 2023.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA, 2005. Association for Computing Machinery.

[Reg06] Oded Regev. Lattice-based cryptography. In *Annual International Cryptology Conference*. Springer, 2006.

[Reg10] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 7(30):11, 2010.

[RG22] Peixin Ren and Xiaozhuo Gu. Practical Post-quantum Password-Authenticated Key Exchange Based-on Module-Lattice. In *Information Security and Cryptology – ICISC 2021*, 2022.

[RGW23] Peixin Ren, Xiaozhuo Gu, and Ziliang Wang. Efficient module learning with errors-based post-quantum password-authenticated key exchange. *IET Information Security*, 2023.

[RHCB21] Prasanna Ravi, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. Lattice-Based Key-Sharing Schemes: A Survey. *ACM Comput. Surv.*, 2021.

[Ros21] Ansis Rosmanis. Tight bounds for inverting permutations via compressed oracle arguments. *arXiv preprint arXiv:2103.08975*, 2021.

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Annual international cryptology conference*, 1991.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RSM$^+$23] Purva Rewal, Mrityunjay Singh, Dheerendra Mishra, Komal Pursharthi, and Ankita Mishra. Quantum-safe three-party lattice based authenticated key agreement protocol for mobile devices. *Journal of Information Security and Applications*, 2023.

[SA23] Kübra Seyhan and Sedat Akleylek. A new password-authenticated module learning with rounding-based key exchange protocol: Saber.PAKE. *The Journal of Supercomputing*, 2023.

[SA24] Kübra Seyhan and Sedat Akleylek. A new lattice-based password authenticated key exchange scheme with anonymity and reusable key. *PeerJ Computer Science*, 2024.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. J. Cryptology 4, 1991.

[Sch24] Sophie Schmieg. Unbindable kemmy schmidt: Ml-kem is neither mal-bind-k-ct nor mal-bind-k-pk. 2024.

[SHB21] István András Seres, Máté Horváth, and Péter Burcsi. The legendre pseudorandom function as a multivariate quadratic cryptosystem: Security and applications. Cryptology ePrint Archive, Paper 2021/182, 2021.

[Sho97] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 1997.

[SM16] Douglas Stebila and Michele Mosca. Post-quantum key exchange for the internet and the open quantum safe project. In *International Conference on Selected Areas in Cryptography*, pages 14–37. Springer, 2016.

[Smi18] Benjamin Smith. Pre-and post-quantum diffie–hellman from groups, actions, and isogenies. In *Arithmetic of Finite Fields: 7th International Workshop, WAIFI 2018*, 2018.

[SS19] Shingo Sato and Junji Shikata. So-cca secure pke in the quantum random oracle model or the quantum ideal cipher model. In *Cryptography and Coding: 17th IMA International Conference*, 2019.

[SWL$^+$21] Qin SHU, Shengbao WANG, Fanyi LU, Lidong HAN, and Xiao TAN. Universally Composable Two-Party Password-Based Authenticated Key Exchange from Ideal Lattices. *Journal of Electronics & Information Technology*, 43, 2021.

[TES23] Marcel Tiepelt, Edward Eaton, and Douglas Stebila. Making an asymmetric pake quantum-annoying by hiding group elements. In *European Symposium on Research in Computer Security*, pages 168–188. Springer, 2023.

[TLZ$^+$21] Yongli Tang, Ying Li, Zongqu Zhao, Jing Zhang, Lina Ren, and Yuanhong Li. Improved verifier-based three-party password-authenticated key exchange protocol from ideal lattices. *Security and Communication Networks*, 2021.

[TSJL20] Oleg Taraskin, Vladimir Soukharev, David Jao, and Jason T. LeGrow. Towards Isogeny-Based Password-Authenticated Key Establishment. *Journal of Mathematical Cryptology*, 2020.

[Tue13] Max Tuengerthal. *Analysis of Real-World Security Protocols in a Universal Composability Framework.* Logos Verlag Berlin GmbH, 2013.

[TY19] Shintaro Terada and Kazuki Yoneyama. Password-Based Authenticated Key Exchange from Standard Isogeny Assumptions. In *Provable Security*, 2019.

[Unr20] Dominique Unruh. Post-quantum verification of fujisaki-okamoto. In *Advances in Cryptology – ASIACRYPT 2020*, 2020.

[Unr21] Dominique Unruh. Compressed permutation oracles (and the collision-resistance of sponge/SHA3). Cryptology ePrint Archive, Paper 2021/062, 2021.

[Unr23] Dominique Unruh. Towards compressed permutation oracles. In *International Conference on the Theory and Application of Cryptology and Information Security*, 2023.

[WCL$^+$23] Jinhua Wang, Ting Chen, Yanyan Liu, Yu Zhou, and XinFeng Dong. Efficient Two-Party Authentication Key Agreement Protocol Using Reconciliation Mechanism from Lattice. In *International Conference on Security and Privacy in New Computing Environments*, 2023.

[WHC$^+$21] Mei Wang, Kun He, Jing Chen, Zengpeng Li, Wei Zhao, and Ruiying Du. Biometrics-authenticated key exchange for secure messaging. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2618–2631, 2021.

[Wu98] Thomas D. Wu. The secure remote password protocol. In *Network and Distributed System Security Symposium*, 1998.

[Wu02] Thomas D. Wu. Srp-6: Improvements and refinements to the secure remote password protocol, 2002.

[WW14] Huixin Wu and Feng Wang. A survey of noninteractive zero knowledge proof system and its applications. *The scientific world journal*, 2014(1), 2014.

[Xag22] Keita Xagawa. Anonymity of nist pqc round 3 kems. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2022.

[XHCC17] Dongqing Xu, Debiao He, Kim-Kwang Raymond Choo, and Jianhua Chen. Provably Secure Three-party Password Authenticated Key Exchange Protocol Based On Ring Learning With Error. *IACR Cryptol. ePrint Arch. 2017/360*, 2017.

[XZZ24] Yuting Xiao, Rui Zhang, and Hong-Sheng Zhou. Maximizing the utility of cryptographic setups: Secure PAKEs, with either functional RO or CRS. Cryptology ePrint Archive, Paper 2024/1640, 2024.

[YBH$^+$24] Yibin Yang, Fabrice Benhamouda, Shai Halevi, Hugo Krawczyk, and Tal Rabin. Gold OPRF: Post-quantum oblivious power residue PRF. Cryptology ePrint Archive, Paper 2024/1955, 2024.

[YGS$^+$20] Anqi Yin, Yuanbo Guo, Yuanming Song, Tongzhou Qu, and Chen Fang. Two-Round Password-Based Authenticated Key Exchange from Lattices. *Wireless Communications and Mobile Computing*, 2020.

[YGWX19] Yingshan Yang, Xiaozhuo Gu, Bin Wang, and Taizhong Xu. Efficient password-authenticated key exchange from RLWE based on asymmetric key consensus. In *International Conference on Information Security and Cryptology*, 2019.

[YHL13] Mao Ye, Xue-xian Hu, and Wen-fen Liu. Password authenticated key exchange protocol in the three party setting based on lattices. *Journal of Electronics & Information Technology*, 2013.

[YLZ$^+$21] Jinxia Yu, Huanhuan Lian, Zongqu Zhao, Yongli Tang, and Xiaojun Wang. Chapter Four - Provably secure verifier-based password authenticated key exchange based on lattices. In *Advances in Computers*, 2021.

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2021.

[YZYW25] Yatao Yang, Ruoyan Zhao, Fangrui Yin, and Ke Wang. K-pake: post quantum password authentication key exchange protocol for satellite networks. *Cluster Computing*, 28(4):223, 2025.

[ZG15] Hongfeng Zhu and Shuai Geng. Simple and Universal Construction for Round-Optimal Password Authenticated Key Exchange towards Quantum-Resistant. *Journal of Information Hiding and Multimedia Signal Processing*, 2015.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pages 239–268. Springer, 2019.

[ZHS14] Hongfeng Zhu, Xin Hao, and Yang Sun. Elliptic Curve Isogenies-Based Three-party Password Authenticated Key Agreement Scheme towards Quantum-Resistant. *J. Inf. Hiding Multim. Signal Process.*, 2014.

[ZTJ21] Li Zi-chen, Xie Ting, and Zhang Juan-mei. Post Quantum Password-Based Authentication Key Exchange Protocol Based on Ring Learning with Errors Problem. *ACTA ELECTONICA SINICA*, 2021.

[ZY17] Jiang Zhang and Yu Yu. Two-round PAKE from approximate SPH and instantiations from lattices. In *Advances in Cryptology–ASIACRYPT 2017*, 2017.

[ZYL$^+$23] Shiwen Zhang, Ziwei Yan, Wei Liang, Kuan-Ching Li, and Ciprian Dobre. Baka: Biometric authentication and key agreement scheme based on fuzzy extractor for wireless body area networks. *IEEE Internet of Things Journal*, 2023.

A Corresponding Classical Design Class Representatives

Table 6: PQC PAKE Representatives and Corresponding Classical PAKEs.

Class	PQC PAKE(s)	Classical PAKE(s)
C1	CAKE [BCP+23]	EKE2 [BPR00]
C2	RLWE-PAK [DAL+17]	PAK [Mac02]
C3	KV-PAKE [KV09] / GA-PAKE [AEK+22b]	KOY-PAKE [KOY01] / SPEKE [Jab96]

C1: EKE

Alice		**Bob**
A		B
$x \in_R [0, p-1]$		
	$\xrightarrow{A, \varepsilon_w(g^x \bmod p)}$	
		$y \in_R [0, p-1]$
	$\xleftarrow{B, \varepsilon_w(g^x \bmod p)}$	
$K = H(A\|\|B\|\|g^x\|\|g^y\|\|g^xy)$		$K = H(A\|\|B\|\|g^x\|\|g^y\|\|g^xy)$

Figure 12: The EKE2 Protocol [BPR00] - adopted from [HvO22]

C2: PAK

Client		**Server**
Input: S, π		$\pi_S[C] = \langle (H_1(\pi_C))^{-1} \rangle$
$x \overset{R}{\leftarrow} \mathbb{Z}_q$		
$\alpha \leftarrow g^x$		
$\gamma \leftarrow H_1(\pi)$		
$m \leftarrow \alpha \cdot \gamma$		
	$\xrightarrow{\langle C, m \rangle}$	
		Abort if $\neg ACCEPTABLE(m)$
		$y \overset{R}{\leftarrow} \mathbb{Z}_q$
		$\mu \leftarrow g^y$
		$\gamma' \leftarrow \pi_S[C]$
		$\alpha \leftarrow m \cdot \gamma'$
		$\sigma \leftarrow \alpha^y$
		$k \leftarrow H_2(\langle C, S, m, \mu, \sigma, \gamma' \rangle)$
$\sigma \leftarrow \mu^x$	$\xleftarrow{\langle \mu, k \rangle}$	$k'' \leftarrow H_3(\langle C, S, m, \mu, \sigma, \gamma' \rangle)$
$\gamma' \leftarrow (\gamma)^{-1}$		
Abort if $k \neq H_2(\langle C, S, m, \mu, \sigma, \gamma' \rangle)$		
$k' \leftarrow H_3(\langle C, S, m, \mu, \sigma, \gamma' \rangle)$	$\xrightarrow{k'}$	Abort if $k' \neq k''$

Figure 13: The PAK Protocol [Mac02]

C3: KOY-PAKE

$$
\begin{array}{lcr}
\textbf{Client} & p, q, g_1, g_2, h, c, d, \mathcal{H} & \textbf{Server} \\
\hline
(\mathrm{VK},\mathrm{SK}) \leftarrow \mathrm{SigGen}(1^\kappa) & & \\
r_1 \leftarrow \mathbb{Z}_q & & \\
A = g_1^{r_1}; B = g_2^{r_1} & & \\
C = h^{r_1} g_1^{pw_C} & & \\
\alpha = \mathcal{H}(Client|\mathrm{VK}|A|B|C) & & \\
D = (cd^\alpha)^{r_1} & \xrightarrow{Client|\mathrm{VK}|A|B|C|D} & x_2, y_2, z_2, w_2, r_2 \leftarrow \mathbb{Z}_q \\
 & & \alpha' = \mathcal{H}(Client|\mathrm{VK}|A|B|C) \\
 & & E = g_1^{x_2} g_2^{y_2} h^{z_2} (cd^{\alpha'})^{w_2} \\
 & & F = g_1^{r_2}; G = g_2^{r_2} \\
 & & I = h^{r_2} g_1^{pw_C} \\
 & & \beta = \mathcal{H}(Server|E|F|G|I) \\
x_1, y_1, z_1, w_1 \leftarrow \mathbb{Z}_q & \xleftarrow{Server|E|F|G|I|J} & J = (cd^\beta)^{r_2} \\
\beta' = \mathcal{H}(Server|E|F|G|I) & & \\
K = g_1^{x_1} g_2^{y_1} h^{z_1} (cd^{\beta'})^{w_1} & & \\
\mathrm{Sig} = \mathrm{Sign}_{\mathrm{SK}}(\beta'|K) & \xrightarrow{K|\mathrm{Sig}} & \text{if } \mathrm{Verify}_{\mathrm{VK}}((\beta|K), \mathrm{Sig}) = 1 \\
 & & C' = C/g_1^{pw_C} \\
I' = I/g_1^{pw_C} & & sk_S = K^{r_2} A^{x_2} B^{y_2} (C')^{z_2} D^{w_2} \\
sk_C = E^{r_1} F^{x_1} G^{y_1} (I')^{z_2} J^{w_1} & & \text{else } sk_S \leftarrow \mathcal{G}
\end{array}
$$

Figure 14: The KOY PAKE [KOY01]

C3: SPEKE

$$
\begin{array}{lcr}
\textbf{Alice} & & \textbf{Bob} \\
\hline
A & & B \\
\hline
x \in_R [1, q-1] & & \\
 & \xrightarrow{A, f(w)^x \bmod p} & \\
 & & \text{Validate Key} \\
 & & y \in_R [1, q-1] \\
 & \xleftarrow{B, f(\mathbf{w})^y \bmod p} & \\
\text{Validate Key} & & \\
K = H(sID||f(w)^{xy}) & & K = H(sID||f(w)^{xy})
\end{array}
$$

Figure 15: The Patched SPEKE protocol [HS14] - adopted from [HvO22]