A preliminary version of this paper appears in *Advances in Cryptology – CRYPTO 2026*.
This is the full version.

Post-Quantum Internet Key Exchange via Authenticated Forward-Secure KEM

Yunlei Zhao[1⋆], Biming Zhou[1⋆], Zhixiang Zhao[1⋆], Yifan Dong[1],
Cheng Huang[1], and Haodong Jiang[2]

[1] Fudan University, Shanghai 200433, China
ylzhao@fudan.edu.cn, bmzhou22@m.fudan.edu.cn, 24110240206@m.fudan.edu.cn, yfdong24@m.fudan.edu.cn,
chuang@fudan.edu.cn
[2] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, 450001, Henan, China
hdjiang13@163.com

Abstract. In this work, we present a new framework for signature-free, post-quantum secure authenticated key exchange (AKE) that simultaneously satisfies: (1) exchanging at most two standard ciphertexts of a key encapsulation mechanism (KEM); (2) computational symmetry; (3) perfect forward secrecy (PFS); (4) strong resilience to secret-state exposure; (5) strong resistance to decryption-error attacks; (6) admitting instantiations based on the native structure of ML-KEM under the MLWE assumption; and (7) provable security in both the random oracle model (ROM) and the quantum-accessible random oracle model (QROM) under the post-id eCK-PFS framework. This resolves several fundamental open questions in the literature. The core technical building block is a new cryptographic primitive, called an *authenticated forward-secure* KEM (AFS-KEM), which unifies authentication and forward secrecy within a single KEM abstraction and may be of independent interest.

Keywords: Post-quantum · Internet key exchange · AFS-KEM · ML-KEM · QROM

⋆ These authors contributed equally to this work.

Table of Contents

1 Introduction . . . 3
1.1 Our Contributions . . . 4
1.2 Technical Overview . . . 5
2 Preliminaries . . . 10
3 Authenticated Forward-Secure KEM . . . 10
4 The post-eCK-PFS Security Model . . . 13
4.1 Model syntax . . . 13
4.2 Adversarial Capabilities . . . 14
4.3 Match Soundness, Key Secrecy Security, and Freshness Predicates . . . 15
4.4 Explicit Authentication . . . 16
5 Authenticated Key Exchanges based on AFS-KEMs . . . 17
5.1 Key Secrecy . . . 17
5.2 Match Soundness. . . . 20
5.3 Explicit Authentication. . . . 20
6 Generic Construction of AFS-KEM . . . 20
6.1 Authenticated Forward-Secure PKE . . . 21
6.2 A General Construction of CCA-Secure AFS-KEM from CPA-Secure AFS-PKE . . . 22
7 AFS-KEM-ML: AFS-KEM Instantiations from ML-KEM . . . 23
7.1 Notations. . . . 23
7.2 AFS-PKE-ML . . . 24
7.3 AFS-KEM-ML . . . 25
7.4 Discussion and Future Work. . . . 26
A Appendix: Background . . . 32
A.1 Cryptographic Primitives . . . 32
A.2 (Quantum-Accessible) Random Oracle Model . . . 33
B Appendix: Security Analysis for $\mathsf{FO}_{\mathsf{AFS}}$ Transform . . . 35
B.1 Security Analysis in the Random Oracle Model. . . . 35
B.2 Security Analysis in the Quantum-Accessible Random Oracle Model. . . . 39
C Appendix: Additional Details Regarding AFS-PKE-ML and AFS-KEM-ML . . . 42
C.1 Correctness Analysis for AFS-PKE-ML . . . 42
C.2 Security Analysis for AFS-PKE-ML . . . 43
C.3 SPK-Bind, RSPK-Bind, Key-Coll Security Analysis for AFS-KEM-ML . . . 49
D Appendix: Security Analysis for pUAKE . . . 52
D.1 The pUAKE Protocol and Freshness Predicate $\mathsf{F}_{\mathsf{pUAKE}}$. . . . 52
D.2 Key Secrecy of the pUAKE protocols . . . 52
D.3 Full Analysis for Lemmas D.3, D.4, D.5, and D.7 . . . 54
D.4 Init-Explicit Authentication Analysis for pUAKE . . . 64
E Appendix: Security Analysis for pMAKE . . . 66
E.1 Key Secrecy of the pMAKE protocols. . . . 66
E.2 Full Analysis for Lemmas E.1, E.2, E.3, and E.4 . . . 68
E.3 Explicit Authentication Analysis for pMAKE . . . 83

1 Introduction

Post-Quantum Cryptography. Classical public-key cryptographic systems based on RSA [RSA78] and Diffie–Hellman [DH76] face a fundamental threat from quantum computing due to Shor's algorithm [Sho94]. To address this challenge, the U.S. National Institute of Standards and Technology (NIST) initiated a post-quantum cryptography standardization process in 2016, aiming to standardize public-key encryption, key encapsulation mechanisms (KEMs), and digital signatures resistant to quantum adversaries. In 2024, NIST published its first PQC standards, including ML-KEM, ML-DSA, and SLH-DSA [Nat24b, Nat24a, Nat24c]. In 2025, NIST additionally selected the code-based KEM HQC [MAB+25] as a complementary alternative to lattice-based ML-KEM, reflecting an emphasis on cryptographic diversity.

Signature-Free Post-Quantum Authenticated Key Exchange. Authenticated key exchange (AKE) is a fundamental primitive in cryptography that enables two parties to establish a shared session key [BR94]. With the rapid advancement of post-quantum cryptography, designing post-quantum (PQ) secure AKEs has become a central research focus [Nat17, SSW20, SFG25]. Authentication in PQ AKEs can be realized using PQ digital signatures. However, PQ signature schemes such as ML-DSA incur significantly higher communication and computational overhead than PQ KEMs such as ML-KEM at comparable security levels. As a result, KEM-based authentication approaches have emerged as an attractive alternative for constructing PQ AKE protocols and have been widely studied in the literature [BCGP08, FSXY12, FSXY13, HKSU20, LL23, PWZ23, PZ23, BCL+26]. Moreover, such approaches have also been explored in practical protocol designs, including PQ WireGuard [HNS+21, HKNW25a], KEMTLS [SSW20], and PQ IKE [WS25].

To achieve (perfect) forward secrecy and mutual (resp. unilateral) authentication, KEM-based AKE constructions [FSXY12, HKSU20] typically require **three** (resp. **two**) KEM instances[3] in parallel: one CPA-secure KEM w.r.t. an ephemeral public key and two (resp. one) CCA-secure KEMs w.r.t. the long-term public keys. As a result, such constructions entail the transmission of three (resp. two) PQ KEM ciphertexts, leading to substantial communication overhead in the post-quantum setting, since PQ KEMs have much larger public keys and ciphertexts than their classical counterparts. For example, ML-KEM-512 employs public keys of 800 bytes and ciphertexts of 768 bytes for 128-bit security, whereas the classical Curve25519-based DH-KEM [LHT16] uses public keys and ciphertexts of only 32 bytes. To mitigate this cost, a line of work [XLL+18, HKNW25a, BCL+26] has studied how to reduce the overall length of two ciphertexts arising in KEM-based AKEs. Recall that each ML-KEM ciphertext consists of a pair $(\mathbf{u}, \mathsf{v})$, where $\mathbf{u}$ is the dominant polynomial vector and v is a polynomial element. The basic idea in prior work is to let two KEM ciphertexts share a common component, thereby reducing the communication bandwidth. In particular, the work of [XLL+18] proposes Twin-Kyber, derived from ML-KEM/Kyber [Nat24b, BDK+18], which reduces ciphertext size by combining the two v polynomial components into a single polynomial. Subsequent independent works propose more compact constructions derived from ML-KEM, including Maul proposed in [BCL+26] and Rebar proposed in [HKNW25a] [4]. Both constructions achieve further ciphertext savings by reusing the dominant polynomial vector $\mathbf{u}$ [KKPP20]. The security of Maul and Rebar is proven under the Hint-MLWE assumption [KLSS23], which is relatively new and not extensively studied compared to the standard MLWE assumption [LS15]. Moreover, their concrete parameter choices in practical instantiations additionally rely on some heuristic assumptions [DJK+25, HKNW25a, BCL+26]. In particular, all these works [XLL+18, HKNW25a, BCL+26] require modifying the underlying structure of ML-KEM in accordance with two dependent ciphertexts. Notably, the frameworks of [FSXY12, HKSU20] are also not directly compatible with the ML-KEM standard: the former requires changing how randomness is derived in the Encaps algorithm, while the latter relies on a PKE-level formulation rather than a direct KEM interface.

Limitations and Open Challenges in Signature-Free PQ AKE. We summarize the limitations and open challenges in existing signature-free PQ AKE protocols as follows:

[3] The protocols in [HKSU20] are based on PKEs rather than KEMs, and apply the Fujisaki–Okamoto transform at the protocol level to reduce computational overhead, while the communication costs remain essentially unchanged.

[4] The work of [HKNW25a] formulates Rebar in the Reinforced-KEM setting and demonstrates its application to PQ WireGuard.

– To the best of our knowledge, no PFS-secure AKE protocol achieves mutual (resp. unilateral) authentication while exchanging only two (resp. one) standard KEM ciphertext(s)[5]. Here, "standard" refers to native ciphertexts (e.g., those of ML-KEM). It could be a long-standing open question in the literature.
– For an *efficient* construction of AKE from KEMs, to the best of our knowledge, there is still no lattice-based PFS-secure AKE protocol that achieves computational symmetry, which constitutes another long-standing open problem in lattice-based PQC.[6] This limitation stems from the asymmetric nature of lattice-based key encapsulation mechanisms. Nevertheless, computational symmetry remains desirable for practical AKE deployment [BMS19], particularly when both roles may run on constrained platforms such as smart cards or sensor nodes, as it can simplify implementation, facilitate dataflow optimization, and reduce code size, circuit area, and reconfiguration overhead.
– Existing analyses of PQ AKEs assume the peer's identity is fixed and known at session initiation. However, as illustrated by Internet Key Exchange (IKE) protocols and their formal analyses [CH98, CK01, CK02, Kra03, KHN+14], this assumption does not always hold in practice. A party may instead be activated with only partial information about its peer (e.g., a URL or IP address) and learn the peer's identity during the execution of the protocol. The same phenomenon appears in TLS-style deployments: a client is often invoked with a service name, URL, or IP address, while the authenticated identity and the corresponding long-term public key are obtained and validated from the certificate chain during the handshake [Res18]. This setting is commonly referred to as the *post-specified identity* (post-id) setting [CK02].[7] To the best of our knowledge, no general PQ AKE protocol has been explicitly designed and formally analyzed in the post-id setting.

Central Question for Signature-Free PQ AKE. Motivated by the challenges outlined above, we ask the following two central questions:

Motivating Question 1. Can one design a PFS-secure mutually authenticated AKE in the post-id setting that simultaneously: (1) transmits only **two** standard KEM public keys and **two** standard KEM ciphertexts[8], (2) is computationally symmetric, (3) admits instantiations based on the unmodified structure of ML-KEM under the standard MLWE assumption.

Motivating Question 2. Can one design a PFS-secure unilaterally authenticated AKE in the post-id setting that simultaneously: (1) transmits only **one** standard KEM public key and **one** standard KEM ciphertext, (2) admits instantiations based on the unmodified structure of ML-KEM under the standard MLWE assumption.

1.1 Our Contributions

In this work, we systematically address the two motivating questions by constructing new authenticated key exchange frameworks built upon a novel cryptographic primitive, the *authenticated forward-secure KEM (AFS-KEM)*. Our main contributions are summarized below and illustrated in Fig. 1.

1. We introduce authenticated forward-secure KEM (AFS-KEM), which unifies authentication and forward secrecy within a single KEM interface by combining ephemeral and static key pairs into a single composite key pair. We formalize its associated security notions, including IND-ECPA,

[5] We focus on KEM components, which dominate the overall communication cost in the post-quantum setting, and therefore omit symmetric-key components such as MAC tags.
[6] A trivial yet *inefficient* way to achieve symmetry in lattice-based AKE is to let both parties run two CPA-secure KEMs in parallel. However, this redundant approach further increases both computational and communication overhead.
[7] In [CK02], parties are assumed to know all users' long-term public keys in advance, while the intended peer identity is not fixed prior to protocol execution. In contrast, our setting assumes that both the peer identity and all users' long-term public keys are unknown at session initiation, capturing more practical deployment scenarios.
[8] "standard" refers to public keys and ciphertexts in the native style of ML-KEM, and we omit symmetric-key components such as OTP here.

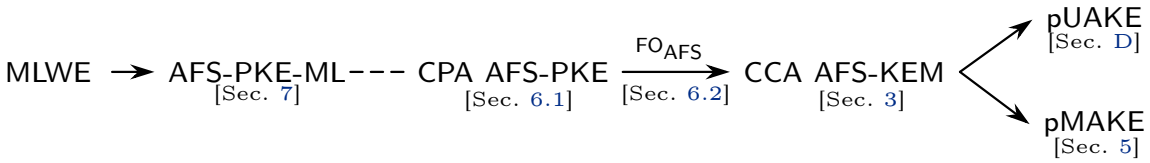


Fig. 1. Overview of our framework. Dashed edges indicate concrete instantiations.

IND-SCCA, and IND-LCK-SCCA. These notions capture confidentiality under the compromise of ephemeral or static key components, as well as security against legitimately adversarially chosen composite public keys.

2. Based on IND-ECPA, IND-SCCA and IND-LCK-SCCA secure AFS-KEM, we construct two compact PFS-secure AKE protocols, pMAKE and pUAKE, in the post-id setting, as illustrated in Fig. 2 and Fig. 4. In particular, pMAKE is a four-round, computationally symmetric, mutually authenticated AKE protocol that provides strong resilience to secret-state exposure, as both parties contribute ephemeral and static KEM key material. In contrast, in existing mutually authenticated protocols [DJK+25, HKNW25a, BCL+26], only one party contributes an ephemeral public key. pUAKE is a three-round, PFS-secure AKE with unilateral authentication. We extend the strong eCK-PFS security model [CF12, BCL+26] of eCK model [LLM07] to the post-id setting [CK02]. Within this framework, we prove that both protocols are secure in the random oracle model (ROM) [BR94] and in the quantum-accessible random oracle model (QROM) [BDF+11]. Moreover, we analyze explicit authentication for both protocols, following the methodology of [SSW20, BCL+26].
3. We introduce *authenticated forward-secure PKE* (AFS-PKE) together with its associated CPA security notions, including IND-ECPA, IND-SCPA, and IND-LCK-SCPA, analogous to those for AFS-KEM. We further propose $\mathsf{FO}_{\mathsf{AFS}}$, a variant of the Fujisaki–Okamoto transformation [FO99, FO13], which upgrades IND-ECPA and IND-SCPA/IND-LCK-SCPA AFS-PKE schemes to IND-ECPA and IND-SCCA/IND-LCK-SCCA AFS-KEM in both ROM and QROM.
4. As a concrete instantiation, we propose AFS-KEM-ML, a compact AFS-KEM derived from ML-KEM under the MLWE assumption. In particular, AFS-KEM-ML preserves the high computational efficiency of ML-KEM while retaining its native public-key and ciphertext structure. Consequently, the pMAKE (resp. pUAKE) protocol instantiated with AFS-KEM-ML transmits only **two** (resp. **one**) ML-KEM-style public keys and ciphertexts.[9] The adapted parameter sets of AFS-KEM-ML achieve communication savings of up to 21.9% (resp. 37.5%) over post-id DAKE_3 (resp. DUAKE_3) [BCL+26] and 30.4% (resp. 47.7%) over standard ML-KEM-based AKEs, while maintaining a decryption error rate comparable to ML-KEM. Furthermore, using the original ML-KEM parameters enables a direct instantiation from the ML-KEM standard and increases the communication savings to 26.3% (resp. 41.7%) and 33.1% (resp. 49.7%), respectively, as shown in Table 1.

Remark 1.1. Using the original ML-KEM parameters, the decryption error rate of AFS-KEM-ML ranges from 2^{-71} to 2^{-88}. Nevertheless, we believe this remains acceptable for practical deployments of AFS-KEM-ML-based pMAKE/pUAKE, as discussed in Section 7.3, since per-session re-randomization of long-term secret keys prohibitively limits key-recovery attacks against ML-KEM that exploit and amplify decryption failures (e.g., [DGJ+19, DRV20, BS20, DB22, AWK+25]). In other words, our AFS-KEM-ML-based pMAKE/pUAKE provide stronger resistance to decryption-error attacks.

1.2 Technical Overview

In this section, we focus on the unilateral authentication protocol pUAKE to illustrate our core techniques. The extension to the mutually authenticated protocol pMAKE follows naturally by composing two instances of unilateral authentication.

[9] We omit the one-time-pad ciphertexts and id, as the post-quantum KEM components dominate the overall communication cost.

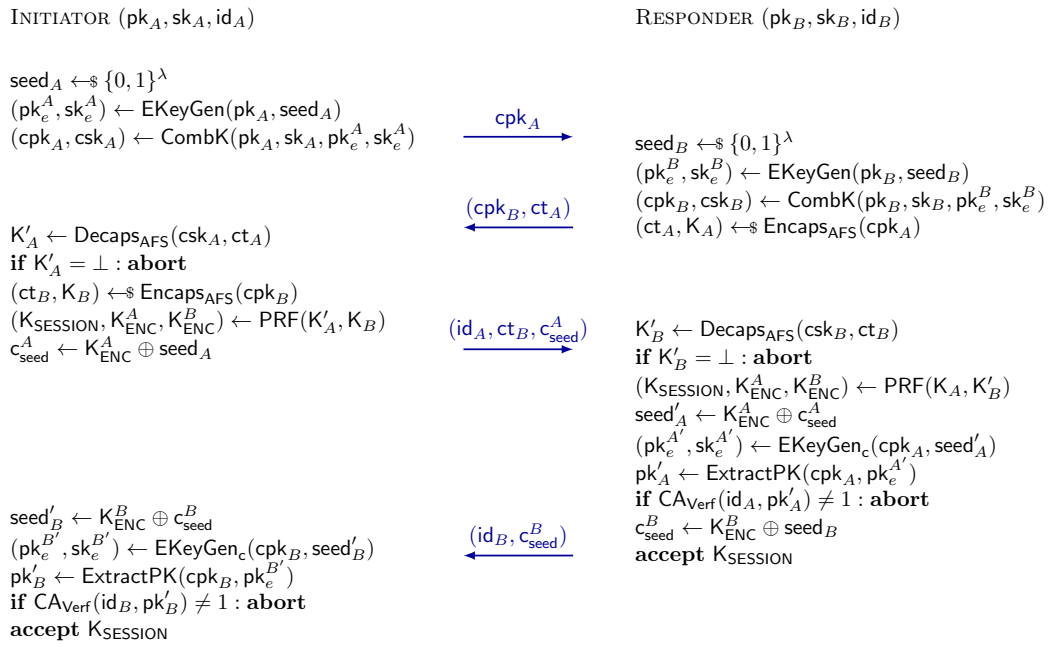


Fig. 2. The pMAKE protocol.

Table 1. Comparison of AKE protocols in terms of total communication cost (bytes) and underlying assumptions. All protocols are instantiated from ML-KEM [Nat24b]. Security levels follow the NIST PQC categories.

Metric	Sec.	Unilateral Authentication			Mutual Authentication		
		MLKEM-UAKE [Nat24b]	DUAKE [BCL+26]	pUAKE (this work)	MLKEM-AKE [Nat24b]	DAKE [BCL+26]	pMAKE (this work)
Assumption	-	MLWE	Hint-MLWE	MLWE	MLWE	Hint-MLWE	MLWE
Comm.(B)	1	3,168	2,580	1,728/1,600	4,768	4,180	3,456/3,200
	3	4,576	3,952	2,528/2,304	6,880	6,256	5,056/4,608
	5	6,304	5,270	3,296/3,168	9,472	8,438	6,592/6,336

[1] In the unilateral (resp. mutual) authentication setting, all protocols complete in three (resp. four) rounds to achieve PFS security. MLKEM-UAKE (resp. MLKEM-AKE) transmits two (resp. three) ML-KEM public keys, two (resp. three) ML-KEM ciphertexts, and one (resp. two) MAC tags, following the standard KEM-based AKE paradigm.

[2] For DUAKE, the communication cost accounts for two Maul public keys, one Maul ciphertext, and one MAC tag; for DAKE, it further includes one ML-KEM public key, one ML-KEM ciphertext, and one additional MAC tag. The Maul parameters are taken from Table 5 of [BCL+26].

[3] For pUAKE and pMAKE, the former values correspond to instantiations with adapted parameter sets that maintain a decryption-error rate comparable to that of ML-KEM, whereas the latter values correspond to instantiations with the original ML-KEM parameter sets.

[4] Each MAC tag is counted as 32 bytes, and each seed encryption $\mathsf{c}_{\mathsf{seed}}$ in pUAKE and pMAKE is counted as 32 bytes. We omit the length of identity strings for simplicity.

Extending Prior Unilaterally Authenticated AKEs to the Post-ID Setting. We briefly recall the standard AKE constructions based on the parallel-KEM paradigm [FSXY12, FSXY13, HKSU20] and explain how it can be adapted to construct unilaterally authenticated AKEs in the post-id setting. In line with most existing AKE constructions, we assume the existence of a trusted certification authority (CA) equipped with a verification algorithm $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_A, \mathsf{pk}_A)$, which returns 1 if and only if pk_A is a valid public key certified for identity id_A. Then, one can construct a PFS-secure, initiator-authenticated AKE, denoted by $\mathsf{UAKE}_{\mathsf{2KEM}}$, as illustrated in Fig. 3, in which unilateral authentication

and PFS[10] are achieved via two independent KEM instances, KEM_e and KEM_s, executed in parallel. In particular, the responder authenticates the initiator by verifying that $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_A, \mathsf{pk}_A) = 1$ and that the received MAC tag is valid. When both KEM instances are instantiated with ML-KEM, the techniques of [BCL+26, HKNW25b] allow reusing the dominant encapsulation component $\mathbf{u}$ in ct_A and ct_e, thereby further reducing the overall communication bandwidth under the Hint-MLWE assumption.

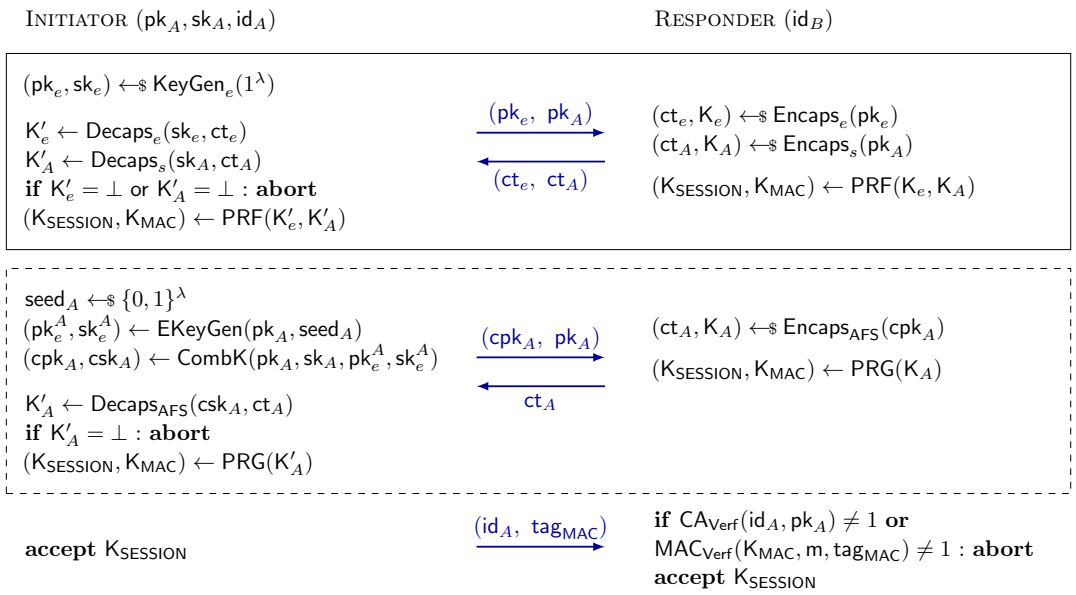


Fig. 3. The $\mathsf{UAKE}_{\mathsf{2KEM}}$ and $\mathsf{NaiveUAKE}_{\mathsf{AFS\text{-}KEM}}$ protocols. Elements shown in solid box appear only in $\mathsf{UAKE}_{\mathsf{2KEM}}$, while those in dashed box appear only in $\mathsf{NaiveUAKE}_{\mathsf{AFS\text{-}KEM}}$. For simplicity, the MAC computation is omitted. In particular, the authentication tag $\mathsf{tag}_{\mathsf{MAC}}$ is computed as $\mathsf{MAC}(\mathsf{K}_{\mathsf{MAC}}, \mathsf{m})$, where m is derived from the entire transcript up to this point.

Main Technique I: Authenticated Forward-Secure KEM. The central design principle of our AKE construction is to ensure that the KEM public key itself inherently provides authentication and forward secrecy. Motivated by this goal, we introduce a new cryptographic primitive called the *Authenticated Forward-Secure KEM* (AFS-KEM). An AFS-KEM combines a long-term static key pair with an ephemeral key pair into a single composite key pair $(\mathsf{spk}, \mathsf{csk})$ via CombK, which is then used directly by the $\mathsf{Encaps}_{\mathsf{AFS}}$ and $\mathsf{Decaps}_{\mathsf{AFS}}$ algorithms. Intuitively, the static component enforces authentication, while the ephemeral component guarantees forward secrecy of the derived KEM key.

We further define a family of security notions for AFS-KEM, including $\mathsf{IND\text{-}ECPA}$, $\mathsf{IND\text{-}SCCA}$, and $\mathsf{IND\text{-}LCK\text{-}SCCA}$. The prefix E (resp. S) denotes that security relies on the secrecy of the ephemeral (resp. static) key component, while allowing compromise of the static (resp. ephemeral) component. The LCK prefix captures whether the composite public key is honestly generated or in some special adversarially malformed, and $\mathsf{CPA}/\mathsf{CCA}$ specify the availability of a decapsulation oracle. To construct desired CCA-secure AFS-KEMs, we introduce Authenticated Forward-Secure Public-Key Encryption ($\mathsf{AFS\text{-}PKE}$), together with its associated CPA security notions, analogous to those for $\mathsf{AFS\text{-}KEM}$. We then propose a Fujisaki–Okamoto (FO) transform [FO99, FO13, HHK17] $\mathsf{FO}_{\mathsf{AFS}}$, which upgrades CPA-secure $\mathsf{AFS\text{-}PKE}$ into a desired CCA-secure $\mathsf{AFS\text{-}KEM}$. Once the appropriate security definition for $\mathsf{AFS\text{-}PKE}$ is established, the FO transform can be applied in a natural and modular way. The resulting security analysis for $\mathsf{FO}_{\mathsf{AFS}}$ closely follows that of standard FO-based KEM constructions [JZC+18, JZM19, BHH+19, AHU19, DHK+21].

[10] Perfect forward secrecy (PFS) ensures that a session key remains secure even if the long-term secrets used in its derivation are compromised after the session completes.

To illustrate the intuition underlying the AFS-KEM construction, consider a simple instantiation based on ML-KEM.[11] Let $\mathbf{u}_s = \mathbf{A}\mathbf{s}_s + \mathbf{e}_s$ and $\mathbf{u}_e = \mathbf{A}\mathbf{s}_e + \mathbf{e}_e$ denote the static and ephemeral ML-KEM public keys, respectively, with corresponding secret keys $\mathbf{s}_s, \mathbf{s}_e$ and error vectors $\mathbf{e}_s, \mathbf{e}_e$. The pairs $(\mathbf{s}_s, \mathbf{e}_s) = \mathsf{H}(\mathsf{seed}_s)$ and $(\mathbf{s}_e, \mathbf{e}_e) = \mathsf{H}(\mathsf{seed}_e)$ are derived from two independent uniformly random 32-byte seeds via a hash function H. We define the CombK algorithm as component-wise addition. Specifically, the composite public and secret keys are computed as $\mathsf{cpk} := \mathbf{u}_s + \mathbf{u}_e$[12] and $\mathsf{csk} := \mathbf{s}_s + \mathbf{s}_e$. Notably, the resulting composite key pair $(\mathsf{cpk}, \mathsf{csk})$ constitutes a valid ML-KEM-style key pair. Therefore, the encapsulation and decapsulation algorithms $\mathsf{Encaps}_{\mathsf{AFS}}$ and $\mathsf{Decaps}_{\mathsf{AFS}}$ are defined identically to the standard ML-KEM encapsulation and decapsulation procedures.

Throughout the following analysis, we assume for simplicity that AFS-KEM is instantiated from ML-KEM as described above.

Trivial Designs Based on AFS-KEM Are Insecure. A natural first attempt to construct a unilaterally authenticated AKE from AFS-KEM is illustrated in Fig. 3, which we denote by $\mathsf{NaiveUAKE}_{\mathsf{AFS\text{-}KEM}}$. In this design, the initiator sends a composite public key cpk_A, derived from its long-term public key pk_A and an ephemeral public key pk_e^A. Notably, pk_e^A is deterministically generated from an ephemeral seed seed_A via a hash function H. The responder then computes a ciphertext $(\mathsf{ct}_A, \mathsf{K}_A) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}_A)$ and derives the session key from the encapsulated key K_A. To enable verification of the initiator's identity, the initiator additionally transmits its long-term public key pk_A. Since $\mathsf{cpk}_A = \mathsf{pk}_A + \mathsf{pk}_e^A$, as illustrated by our ML-KEM–based instantiation, the initiator may equivalently transmit either $(\mathsf{cpk}_A, \mathsf{pk}_e^A)$ or $(\mathsf{pk}_A, \mathsf{pk}_e^A)$ to its peer.

However, the above construction is insecure. The fundamental issue is that the session key is derived solely from the composite public key cpk_A, while no mechanism enforces that cpk_A is honestly generated. In particular, the adversary may either set cpk_A to an adversarially chosen public key for which it knows the corresponding secret key csk_A, or define a malformed ephemeral public key of the form $\mathsf{pk}_e^A = -\mathsf{pk}_A + \mathsf{pk}'_e$, where $(\mathsf{pk}'_e, \mathsf{sk}'_e)$ is chosen by the adversary. In both cases, the adversary can directly compute the resulting session key and thereby successfully impersonate the initiator.

These attacks demonstrate that merely transmitting the components of cpk_A is insufficient. Instead, the initiator must additionally provide a *proof*, in some sense, that cpk_A is constructed honestly from its long-term and ephemeral public keys. A generic approach to achieving this is to employ zk-SNARKs [GMR89, BCCT12] to prove the correct formation of the composite public key cpk_A. However, such generic solutions incur prohibitive computational and communication overhead and are therefore unsuitable for post-quantum AKEs.

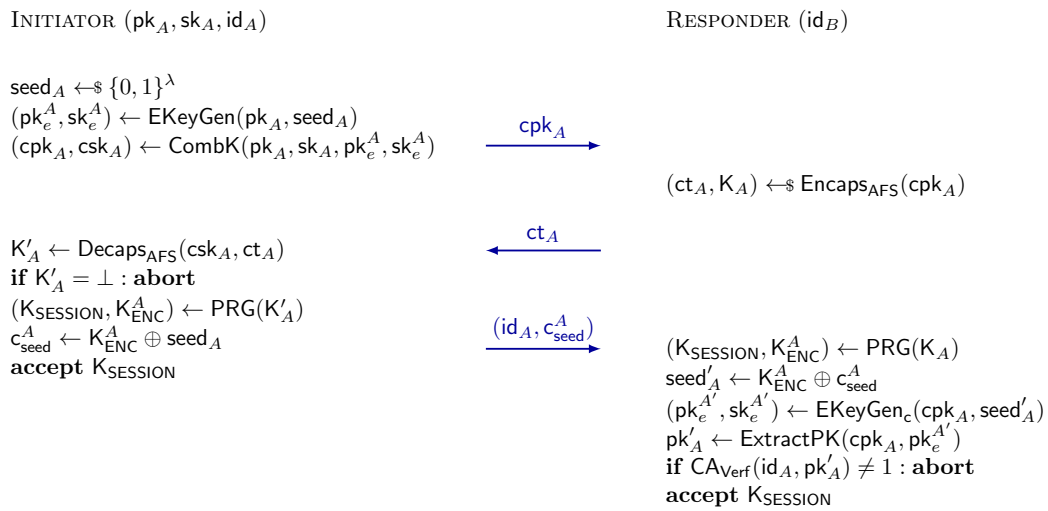


Fig. 4. The pUAKE protocol achieves *initiator-only authentication*.

[11] We omit inessential details such as the auxiliary secret used for implicit rejection in the secret key.

[12] For completeness, $\mathsf{seed}_{\mathsf{pk}}$ that is used to generate Matrix $\mathbf{A}$ is included as part of the static, ephemeral, and composite public keys. We also omit it here for simplicity.

Main Technique II: Proving cpk via Encrypted seed. Our main observation is that this *proof* need not hide the ephemeral secret seed_A from the honest responder, but only from the adversary. Accordingly, we adopt a significantly simpler approach: the initiator directly sends the one-time-pad ciphertext $\mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{seed}_A$ in the third round, as illustrated by our pUAKE construction in Fig. 4.[13] Then, the responder can locally reconstruct pk'_A from the composite public key cpk_A and the decrypted seed seed_A, and then verify the initiator's identity by checking $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_A, \mathsf{pk}'_A)$. If this verification succeeds, the responder is assured that $\mathsf{cpk}_A = \mathsf{pk}'_A + \mathsf{pk}^A_e$, where $\mathsf{pk}^A_e = \mathsf{EKeyGen}(\mathsf{cpk}_A, \mathsf{seed}_A)$. Moreover, this mechanism allows the initiator to omit explicitly transmitting its long-term public key pk_A, while the resulting one-time-pad ciphertext acts like a MAC: any modification causes the seed_A verification to fail except with negligible probability, thereby providing explicit authentication.

One can readily verify that the above attacks no longer apply to pUAKE. Specifically, when modeling the hash function H as a random oracle, any adversary that does not know the long-term secret key sk_A and outputs a malicious composite public key cpk_A (for which it knows the corresponding csk_A) cannot find a seed seed_e satisfying $(\mathbf{s}_e, \mathbf{e}_e) = \mathsf{H}(\mathsf{seed}_e)$ and $\mathbf{A}\mathbf{s}_e + \mathbf{e}_e = \mathsf{cpk}_A - \mathsf{pk}_A$ except with negligible probability.

We next analyze the security of the pUAKE protocol by focusing on two key attack scenarios that capture the main technical challenges.

Forward Secrecy against a Passive Adversary. We consider a passive adversary that later corrupts the initiator A's long-term secret key and attempts to compromise a previously established session π of initiator A. In this setting, security relies on the secrecy of the ephemeral seed seed_A in π, and a first attempt is to directly show that the derived KEM key K_A in π (and hence the session key by the security of PRG) remains computationally indistinguishable from uniform random for the adversary. However, we cannot directly replace K_A with a uniformly random key by relying on the IND-ECPA security of AFS-KEM. The reason is that the adversary observes a ciphertext encrypting seed_A under a symmetric key derived from K_A. Consequently, the reduction in attacking IND-ECPA must simulate the encryption without knowing seed_A, thereby introducing a key dependency[14].

To circumvent this key dependency obstacle, we model the pseudorandom generator PRG as a random oracle G and derive the session and encryption keys as $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}) = \mathsf{G}(\mathsf{K}_A)$. We first replace $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}})$ with uniformly random keys, which is indistinguishable to the adversary unless it queries the random oracle at the point K_A. We denote this bad event as QUERY. Conditioned on $\neg$QUERY, the session key is already uniformly random from the adversary's view. It therefore suffices to bound the probability of QUERY. To this end, we rely on the perfect secrecy of the one-time-pad to replace the ciphertext $\mathsf{c}^A_{\mathsf{seed}}$ with an encryption of an independent random seed $\hat{\mathsf{c}}^A_{\mathsf{seed}} = \mathsf{K}^A_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_A$, where $\hat{\mathsf{seed}}_A \leftarrow\!\$\ \{0,1\}^\lambda$. This step eliminates the circular dependency, since the encrypted message is now independent of seed_A. We can then invoke the IND-ECPA security of AFS-KEM to replace K_A with a uniformly random value, which implies that the probability of QUERY is negligible. Moreover, we lift the above ROM into the QROM using the semi-classical oracle techniques of [AHU19].

Impersonation Resistance against an Adaptive Adversary. In this case, the adversary attempts to impersonate a party A without knowing its long-term secret key sk_A. Let π denote the responder session targeted by the adversary and no honest peer session of A exists. Our analysis proceeds as follows. We first rely on the IND-LCK-SCCA security of AFS-KEM to replace the KEM key K_A in session π with a uniformly random value. Consequently, by the pseudorandomness of the PRG, the derived encryption key $\mathsf{K}^A_{\mathsf{ENC}}$ is also indistinguishable from a uniformly random key to the adversary. As a result, the recovered seed $\mathsf{seed}'_A = \mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{c}^A_{\mathsf{seed}}$ computed by π is uniformly random from the adversary's view, where $\mathsf{c}^A_{\mathsf{seed}}$ is chosen by the adversary. Consequently, the public key $\mathsf{pk}^{A'}_e$ derived

[13] This idea is conceptually inspired by identity-concealing authenticated key-exchange protocols [YZ10, YZ13, YZ14, Zha16, Zha20, LPWZ21]. In particular, CAKE [Zha16] protects identity privacy by combining static and ephemeral public keys in a non-malleable manner, while subsequent concealed identity-based AKE protocols [Zha20, LPWZ21] hide the identity-bound ephemeral seed and reveal the corresponding binding only through encrypted seed ciphertexts. We observe that encrypting only a short ephemeral seed via the more compact one-time pad is sufficient for the AFS-KEX construction, which is particularly valuable in the post-quantum setting.

[14] Key-dependent message (KDM) security, also known as circular security, was first formalized in [BRS02] and is a standard assumption in most fully homomorphic encryption (FHE) constructions. As shown in [BRS02], KDM security can be achieved for public-key encryption schemes in the ROM.

from the uniformly random seed'_A satisfies $\mathsf{pk}_e^{A'} = \mathsf{cpk}_A - \mathsf{pk}_A$ only with negligible probability, and thus the adversary cannot successfully impersonate party A.

In particular, the security notion IND-LCK-SCCA for AFS-KEM defined in Fig. 5 is tailored to the analysis setting considered here. In the IND-LCK-SCCA game, the adversary first outputs a composite public key cpk given an honestly generated static public key pk_s and access to a decapsulation oracle. The challenger computes $(\mathsf{ct}, \mathsf{K}_0) \leftarrow\$ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk})$, chooses an independent key $\mathsf{K}_1 \leftarrow\$ \mathcal{K}$ and a challenge bit $b \leftarrow\$ \{0,1\}$, and returns $(\mathsf{ct}, \mathsf{K}_b)$ to the adversary. The adversary succeeds if it outputs a seed seed_A such that $\mathsf{cpk} = \mathsf{pk}_s + \mathsf{pk}_e$, where $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{cpk}, \mathsf{seed}_A)$, and correctly guesses the challenge bit b. Therefore, the IND-LCK-SCCA security for AFS-KEM allows us to replace the session key K_A in π with a uniformly random KEM key.

To show that AFS-KEM-ML achieves IND-LCK-SCCA security, it suffices to prove that the underlying AFS-PKE-ML scheme satisfies IND-LCK-SCPA security, depicted in Fig. 7, which is the CPA analogue of IND-LCK-SCCA in the PKE setting. In the IND-LCK-SCPA security analysis for AFS-PKE-ML, we model H as a random oracle and assume that the adversary queries $\mathsf{H}(\mathsf{seed})$ just before outputting seed, since this does not change its advantage. We first replace pk_s with a uniformly random value under the MLWE assumption. We then aim to replace the challenge ciphertext ct_b^* with a uniformly random value via an MLWE hybrid argument, where ct_b^* constitutes an MLWE sample relative to $(\mathbf{A} \,\|\, \mathsf{cpk})$. However, performing this reduction requires embedding the MLWE instance into cpk, which is chosen by the adversary. In particular, we have $\mathsf{cpk} = \mathsf{pk}_s + \mathbf{A}\mathbf{s}_e + \mathbf{e}_e$ where $\mathsf{H}(\hat{\mathsf{seed}}) = (\mathbf{s}_e, \mathbf{e}_e)$ for some $\hat{\mathsf{seed}}$ queried by the adversary. To overcome this, the reduction guesses the corresponding $\hat{\mathsf{seed}}$ among the adversary's random-oracle queries and assumes that this is the first query on $\hat{\mathsf{seed}}$. Conditioned on this guess being correct (which occurs with probability $1/q_G$), the reduction can correctly embed the MLWE challenge, and the standard MLWE hybrid argument applies. In the QROM, this guessing argument can be justified using the measure-and-reprogram technique [DFMS19, DFM20].

Analysis in the Post-eCK-PFS Security Model. We provide only a high-level intuition for the security of pUAKE, while the formal analysis additionally relies on carefully defined binding properties of AFS-KEM [CDM24]. The analysis of pMAKE follows the same overall structure but achieves stronger guarantees due to mutual authentication, in which both parties contribute static and ephemeral key material. We analyze both pUAKE and pMAKE in the post-eCK-PFS security model, which extends the classical eCK-PFS model [CF12] to the post-id setting. We further consider additional security properties, including match soundness and explicit authentication, following the frameworks of [SSW20, BCL+26].

2 Preliminaries

Notation. The security parameter is denoted by λ. The set $\{0, \ldots, q-1\}$ is denoted by $[q]$. PPT refers to probabilistic polynomial-time algorithms. QPT refers to quantum polynomial-time algorithms. We use standard asymptotic notation, such as $O(\cdot)$ and $\mathsf{negl}(\cdot)$. All sequences are indexed from 0. For a sequence X, we write $X[i]$ to denote its $i+1$-th element. For a finite set $\mathcal{X}$, the notation $x \leftarrow\$ \mathcal{X}$ denotes uniform sampling from $\mathcal{X}$, while sampling from a distribution $\mathcal{D}$ is written as $x \leftarrow\$ \mathcal{D}$. The notations $\Pr[P : G]$ and $\Pr[P \mid G]$ denote the probability that the predicate P holds when its variables are instantiated by the execution of program G. The notation $x \triangleq y$ denotes that x is defined to be equal to y. For a randomized algorithm A, we write $y \leftarrow\$ A(x)$; for a deterministic algorithm A, we write $y \leftarrow A(x)$. The expression $x =?y$ evaluates to 1 if $x = y$, and to 0 otherwise. For a Boolean predicate P, we write $[P]$ for its indicator value, i.e., $[P] = 1$ if P holds and $[P] = 0$ otherwise. The size of a set X is denoted by $|X|$. The notation A^{H} (resp. $A^{|\mathsf{H}\rangle}$) denotes that algorithm A is given classical (resp. quantum) access to the oracle H.

Note. We use the standard security definitions for PKE, KEM, PRG, and PRF. The formal definitions of these basic cryptographic primitives and their corresponding security notions are presented in Appendix A.1. The (quantum-accessible) random oracle model is recalled in Appendix A.2.

3 Authenticated Forward-Secure KEM

In this section, we introduce the *Authenticated Forward-Secure Key Encapsulation Mechanism* (AFS-KEM) together with its associated security requirements.

Definition 3.1 (Authenticated Forward-Secure KEM). *An authenticated forward-secure key encapsulation mechanism (AFS-KEM) over $\mathcal{K}$ is a tuple of algorithms $\Pi_{\mathsf{AFS\text{-}KEM}} = (\mathsf{SKeyGen}, \mathsf{EKeyGen}, \mathsf{EKeyGen_c}, \mathsf{CombPK}, \mathsf{CombSK}, \mathsf{ExtractPK}, \mathsf{Encaps_{AFS}}, \mathsf{Decaps_{AFS}})$.*

- $(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow\!\!\$\ \mathsf{SKeyGen}(1^\lambda)$*: the long-term key-generation algorithm outputs a static key pair* $(\mathsf{pk}_s, \mathsf{sk}_s) \in \mathcal{PK}_s \times \mathcal{SK}_s$.
- $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})$*: the ephemeral key-generation algorithm takes as input the static public key* pk_s *and an explicit seed* $\mathsf{seed} \leftarrow\!\!\$\ \{0,1\}^\lambda$*, and deterministically outputs an ephemeral key pair* $(\mathsf{pk}_e, \mathsf{sk}_e) \in \mathcal{PK}_e \times \mathcal{SK}_e$.
- $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen_c}(\mathsf{cpk}, \mathsf{seed})$*: the second ephemeral key-generation algorithm takes as input the composite public key* cpk *and an explicit seed* $\mathsf{seed} \leftarrow\!\!\$\ \{0,1\}^\lambda$*, and deterministically outputs an ephemeral key pair* $(\mathsf{pk}_e, \mathsf{sk}_e) \in \mathcal{PK}_e \times \mathcal{SK}_e$.
- $\mathsf{cpk} \leftarrow \mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e)$*: the public-key combination algorithm takes a static public key* pk_s *and an ephemeral public key* pk_e *as input, and outputs a composite public key* cpk.
- $\mathsf{csk} \leftarrow \mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e)$*: the secret-key combination algorithm takes a static secret key* sk_s *and an ephemeral secret key* sk_e *as input, and outputs a composite secret key* csk. *For notational convenience, we write* $\mathsf{CombK}(\mathsf{pk}_s, \mathsf{sk}_s, \mathsf{pk}_e, \mathsf{sk}_e) \coloneqq (\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e), \mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e))$.
- $\mathsf{pk}_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}, \mathsf{pk}_e)$*: the public-key extraction algorithm takes as input the composite public key* cpk *and the ephemeral public key* pk_e*, and deterministically outputs the static public key* pk_s.
- $(\mathsf{ct}, K) \leftarrow\!\!\$\ \mathsf{Encaps_{AFS}}(\mathsf{cpk})$*: the encapsulation algorithm takes as input the composite public key and outputs a ciphertext–key pair* (ct, K)*, where* $\mathsf{ct} \in \mathcal{CT}$ *and* $K \in \mathcal{K}$.
- $K' \leftarrow \mathsf{Decaps_{AFS}}(\mathsf{csk}, \mathsf{ct})$*: the decapsulation algorithm takes as input the composite secret key and a ciphertext, and outputs either the corresponding key* $K' \in \mathcal{K}$ *or the rejection symbol* $\perp$.

Correctness for AFS-KEM. An AFS-KEM scheme $\Pi_{\mathsf{AFS\text{-}KEM}}$ is δ-correct if $\Pr[\mathsf{Decaps_{AFS}}(\mathsf{csk}, \mathsf{ct}) \neq K] \leq \delta$, where $(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow\!\!\$\ \mathsf{SKeyGen}(1^\lambda)$, $\mathsf{seed} \leftarrow\!\!\$\ \{0,1\}^\lambda$, $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})$, $\mathsf{cpk} \leftarrow \mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e)$, $\mathsf{csk} \leftarrow \mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e)$, and $(\mathsf{ct}, K) \leftarrow\!\!\$\ \mathsf{Encaps_{AFS}}(\mathsf{cpk})$.

Equivalently, we can define δ-correct as a correctness-failure experiment $\mathsf{Exp}^{\mathsf{corr}}_{\mathsf{AFS\text{-}KEM}}(1^\lambda)$ that honestly samples all values as above and outputs 1 if and only if $\mathsf{Decaps_{AFS}}(\mathsf{csk}, \mathsf{ct}) \neq K$. Thus, $\Pi_{\mathsf{AFS\text{-}KEM}}$ is δ-correct if and only if $\Pr\left[\mathsf{Exp}^{\mathsf{corr}}_{\mathsf{AFS\text{-}KEM}}(1^\lambda) = 1\right] \leq \delta$.

We further require AFS-KEM to satisfy the following two perfect correctness properties.

Perfect ExtractPK-Correctness. An authenticated forward-secure KEM $\Pi_{\mathsf{AFS\text{-}KEM}}$ is said to satisfy *perfect ExtractPK-correctness* if, for all $\mathsf{pk}_s \in \mathcal{PK}_s$ and $\mathsf{pk}_e \in \mathcal{PK}_e$, it holds that $\mathsf{ExtractPK}\big(\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e), \mathsf{pk}_e\big) = \mathsf{pk}_s$.

Ephemeral Key Generation Equivalence. An authenticated forward-secure KEM $\Pi_{\mathsf{AFS\text{-}KEM}}$ is said to satisfy *Ephemeral Key Generation Equivalence* if, for all $\mathsf{pk}_s \in \mathcal{PK}_s$ and $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})$ from some seed, it holds that $\mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed}) = \mathsf{EKeyGen_c}(\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e), \mathsf{seed})$.

We next introduce the following entropy-related notions, which characterize the min-entropy of the combined public key and ciphertext, respectively.

Min-Entropy of the Combined Public Key. An authenticated forward-secure KEM $\Pi_{\mathsf{AFS\text{-}KEM}}$ is said to have combined public keys with min-entropy at least k if, for every fixed long-term public key $\mathsf{pk}_s \in \mathcal{PK}_s$ and every $\mathsf{cpk}' \in \mathcal{CPK}$, it holds that $\Pr_{\mathsf{seed} \leftarrow\!\!\$ \{0,1\}^\lambda}\left[\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e) = \mathsf{cpk}'\right] \leq 2^{-k}$, where $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})$, and the probability is taken over the randomness of seed.

γ-Spreadness of Ciphertexts. An authenticated forward-secure KEM is said to satisfy γ-spreadness if, for any combined public key $\mathsf{cpk} \in \mathcal{CPK}$ and any ciphertext $\mathsf{ct}' \in \mathcal{CT}$, $\Pr[\mathsf{ct} = \mathsf{ct}'] \leq 2^{-\gamma}$, where $(\mathsf{ct}, K) \leftarrow\!\!\$\ \mathsf{Encaps_{AFS}}(\mathsf{cpk})$.

Next, we define the security notions for our AFS-KEM. Specifically, we consider the notions of IND-SCCA, IND-LCK-SCCA, and IND-ECPA security, as shown in Fig. 5. Here, the prefix S (resp. E) indicates that the notion is defined with respect to the static (resp. ephemeral) public key. The LCK qualifier specifies whether the composite public key is formed from an honestly generated ephemeral key or from an adversarially chosen seed. Finally, CPA and CCA indicate whether the adversary is granted access to a decapsulation oracle.

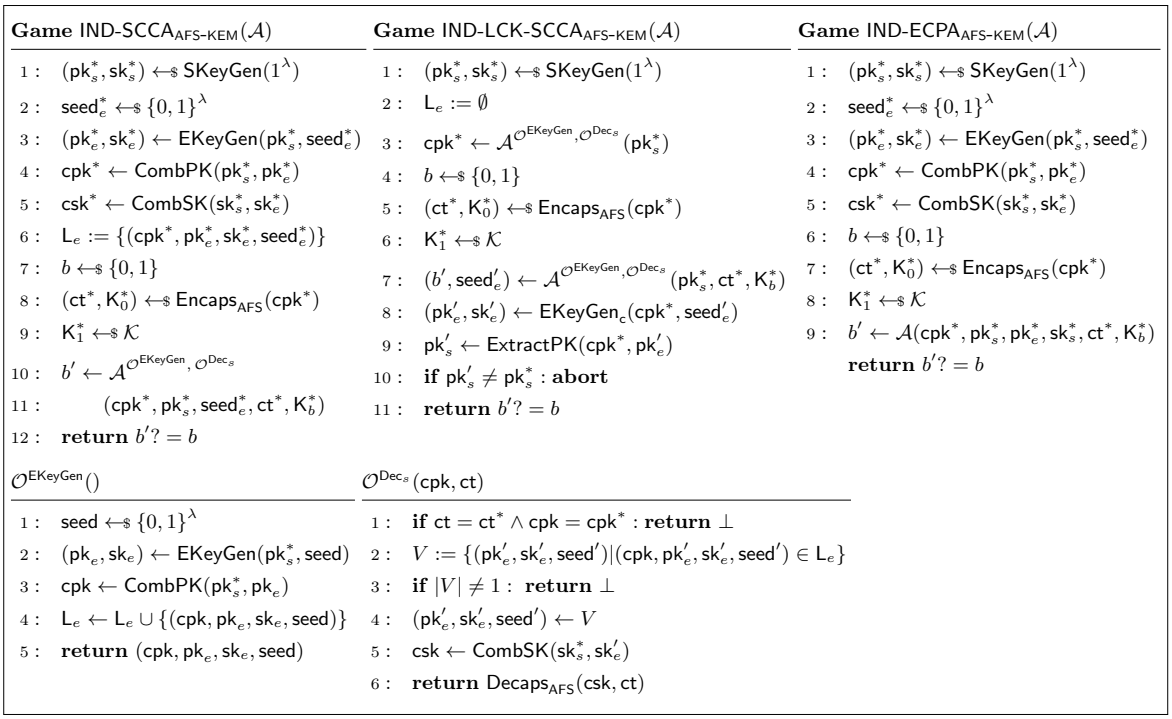


Fig. 5. IND-SCCA, IND-LCK-SCCA, and IND-ECPA security games for an AFS-KEM scheme

In particular, the *legitimate-chosen-key static chosen-ciphertext* security (IND-LCK-SCCA) captures the following guarantee. Even when a ciphertext is generated under a maliciously chosen composite public key cpk, which is combined from the honest static public key pk_s (which remains hidden from the adversary) and an adversarially generated ephemeral public key $\mathsf{pk}_e' = \mathsf{EKeyGen}(\mathsf{cpk}, \mathsf{seed}')$ for some seed' chosen by the adversary, the encapsulated KEM key K remains computationally indistinguishable from a uniformly random key, given the corresponding ciphertext ct.

Definition 3.2 (IND-SCCA Security of AFS-KEM). *For an AFS-KEM scheme, consider the security game* IND-SCCA$_{\mathsf{AFS\text{-}KEM}}$ *defined in Fig. 5. The scheme is said to achieve* IND-SCCA *security if for any polynomial time adversary $\mathcal{A}$ the following advantage is negligible in the security parameter λ:* $\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) := \left|\Pr[\mathsf{IND\text{-}SCCA}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}\right|$.

Definition 3.3 (IND-LCK-SCCA Security of AFS-KEM). *For an AFS-KEM scheme, consider the security game* IND-LCK-SCCA$_{\mathsf{AFS\text{-}KEM}}$ *defined in Fig. 5. The scheme is said to achieve* IND-LCK-SCCA *security if for any polynomial time adversary $\mathcal{A}$ the following advantage is negligible in the security parameter λ:* $\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) := \left|\Pr[\mathsf{IND\text{-}LCK\text{-}SCCA}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}\right|$.

Definition 3.4 (IND-ECPA Security of AFS-KEM). *For an AFS-KEM scheme, consider the security game* IND-ECPA$_{\mathsf{AFS\text{-}KEM}}$ *defined in Fig. 5. The scheme is said to achieve* IND-ECPA *security if for any polynomial time adversary the following advantage is negligible in the security parameter λ:* $\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) := \left|\Pr[\mathsf{IND\text{-}ECPA}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}\right|$.

We now formalize three intrinsic binding properties of AFS-KEM required for our AKE analysis. These notions follow the general binding property for KEM introduced in [CDM24] and the C2PRI property defined in [BCD+24]. The first property, static public key binding (SPK-Bind), ensures that a composite public key uniquely determines its underlying static public key, and that adversarially chosen ephemeral seeds cannot be used to extract the original static key. The second property, randomized static public key binding (RSPK-Bind), guarantees that no adversary can use a randomly chosen ephemeral seed to extract the original static public key from a composite key. The third property, key-collision resistance (Key-Coll), ensures that it is computationally infeasible for an adversary to produce key collisions in any of the three attack scenarios captured by the Key-Coll game.

Game $\mathsf{Key\text{-}Coll}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A})$

$g \leftarrow \mathcal{A}()$
$(\mathsf{pk}_s^*, \mathsf{sk}_s^*) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$
$\mathsf{seed}_e^* \leftarrow\!\$\ \{0,1\}^\lambda$
$(\mathsf{pk}_e^*, \mathsf{sk}_e^*) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s^*, \mathsf{seed}_e^*)$
$(\mathsf{cpk}^*, \mathsf{csk}^*) \leftarrow \mathsf{CombK}(\mathsf{pk}_s^*, \mathsf{sk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_e^*)$
if $g = 1$:
  $\mathsf{cpk}' \leftarrow \mathcal{A}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \mathsf{seed}_e^*)$
  $(\mathsf{ct}', \mathsf{K}') \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}')$
  $\mathsf{ct}^* \leftarrow \mathcal{A}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \mathsf{seed}_e^*, \mathsf{ct}', \mathsf{K}')$
  return $[\mathsf{cpk}' \neq \mathsf{cpk}^*] \wedge [\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}^*) = \mathsf{K}']$
if $g = 2$:
  $(\mathsf{ct}^*, \mathsf{K}^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$
  $\mathsf{cpk}' \leftarrow \mathcal{A}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \mathsf{seed}_e^*, \mathsf{ct}^*, \mathsf{K}^*)$
  $(\mathsf{ct}', \mathsf{K}') \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}')$
  $\mathsf{dummy} \leftarrow \mathcal{A}(\mathsf{ct}', \mathsf{K}')$
  return $[\mathsf{K}' = \mathsf{K}^*]$
if $g = 3$:
  $(\mathsf{ct}^*, \mathsf{K}^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$
  $\mathsf{ct}' \leftarrow \mathcal{A}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \mathsf{seed}_e^*, \mathsf{ct}^*, \mathsf{K}^*)$
  return $[\mathsf{ct}' \neq \mathsf{ct}^*] \wedge [\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}') = \mathsf{K}^*]$

Game $\mathsf{SPK\text{-}Bind}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A})$

$(\mathsf{pk}_s^*, \mathsf{sk}_s^*) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$
$(\overline{\mathsf{pk}_s^*}, \overline{\mathsf{sk}_s^*}) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$
$\mathsf{seed}_e^* \leftarrow\!\$\ \{0,1\}^\lambda$
$(\mathsf{pk}_e^*, \mathsf{sk}_e^*) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s^*, \mathsf{seed}_e^*)$
$\mathsf{cpk}^* \leftarrow \mathsf{CombPK}(\mathsf{pk}_s^*, \mathsf{pk}_e^*)$
$\mathsf{seed}' \leftarrow \mathcal{A}(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}_s^*}, \overline{\mathsf{sk}_s^*})$
$(\mathsf{pk}_e', \mathsf{sk}_e') \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}')$
$\mathsf{pk}_s' \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e')$
return $[\mathsf{pk}_s' = \overline{\mathsf{pk}_s^*}] \vee ([\mathsf{seed}' \neq \mathsf{seed}_e^*] \wedge [\mathsf{pk}_s' = \mathsf{pk}_s^*])$

Game $\mathsf{RSPK\text{-}Bind}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A})$

$(\mathsf{pk}_s^*, \mathsf{sk}_s^*) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$
$\mathsf{cpk} \leftarrow \mathcal{A}(\mathsf{pk}_s^*, \mathsf{sk}_s^*)$
$\mathsf{seed}_e \leftarrow\!\$\ \{0,1\}^\lambda$
$\mathsf{dummy} \leftarrow \mathcal{A}(\mathsf{seed}_e)$
$(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}, \mathsf{seed}_e)$
$\mathsf{pk}_s' \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}, \mathsf{pk}_e)$
return $[\mathsf{pk}_s' = \mathsf{pk}_s^*]$

Fig. 6. Security games defining the SPK-Bind, RSPK-Bind, and Key-Coll properties for an AFS-KEM scheme. In particular, in the Key-Coll game, the adversary can choose $g \in \{1,2,3\}$. For $g = 2$ in Key-Coll and RSPK-Bind, dummy denotes an output of $\mathcal{A}$ that is discarded by the experiment, as it does not affect the game advantage. This step is included solely to support the simulation in the subsequent AKE security proof.

Definition 3.5 (Static Public Key Binding of AFS-KEM). *For an AFS-KEM scheme, consider the security game* $\mathsf{SPK\text{-}Bind}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A})$ *defined in Fig. 6. The scheme is said to achieve* SPK-Bind *security if for any polynomial time adversary* $\mathcal{A}$*, the following advantage is negligible in the security parameter* λ*:* $\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) := \Pr[\mathsf{SPK\text{-}Bind}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) \Rightarrow 1]$.

Definition 3.6 (RSPK-Bind of AFS-KEM). *For an AFS-KEM scheme, consider the security game* $\mathsf{RSPK\text{-}Bind}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A})$ *defined in Fig. 6. The scheme is said to achieve* RSPK-Bind *security if for any polynomial-time adversary* $\mathcal{A}$*, the following advantage is negligible in the security parameter* λ*:* $\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) := \Pr[\mathsf{RSPK\text{-}Bind}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) \Rightarrow 1]$.

Definition 3.7 (Key Collision Security of AFS-KEM). *For an AFS-KEM scheme, consider the security game* $\mathsf{Key\text{-}Coll}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A})$ *defined in Fig. 6. We say that the scheme achieves* key-collision resistance *if for any polynomial time adversary* $\mathcal{A}$*, the following advantage is negligible in the security parameter* λ*:* $\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) := \Pr[\mathsf{Key\text{-}Coll}_{\mathsf{AFS\text{-}KEM}}(\mathcal{A}) \Rightarrow 1]$.

4 The post-eCK-PFS Security Model

The eCK-PFS variant [CF12] of the eCK model [LLM07] captures a strictly broader class of attack scenarios than the CK and CK^+ models [CK01,CK02,Kra05]. In this section, we define the post-eCK-PFS model, which extends the eCK-PFS framework [CF12,BCL+26] to the *post-specified peer* setting [CK02]. In contrast to the classical eCK-PFS framework, our model does not assume that the peer identity and the peer static public key of a session are fixed or known at the time of session activation. Instead, the peer identity and static public key may be learned as the protocol execution evolves.

4.1 Model syntax

The set U denotes the identities of honest participants in the system. Each identity $S \in U$ is associated with a certified static KEM public key pk_S and the corresponding private key sk_S, which form a static

key pair $(\mathsf{pk}_S, \mathsf{sk}_S) \leftarrow\!\!\$\ \mathsf{SKeyGen}(1^\lambda)$ generated during the setup phase. Each party may participate in multiple, possibly concurrent, instances of the protocol, each of which is referred to as a session. Each session is uniquely identified by a session identifier, $\pi \in U \times \mathbb{N}$, represented as a pair (U, n) that denotes the n-th local session of participant U. Each session π maintains the following collection of session-specific information.

- $\mathsf{actor} \in U$ denotes the party running the protocol instance.
- $\mathsf{peer} \in U \cup \{*\}$ denotes the identity of the intended communication partner. The value $*$ indicates that the peer is unknown and may be determined during protocol execution. This value is set upon receiving the peer's identity.
- $\mathsf{role} \in \{\mathsf{Init}, \mathsf{Resp}\}$ denotes whether the initiator or the responder owns the session.
- $\mathsf{key} \in (\mathcal{K} \cup \{\perp\})$ denotes the session key computed by the actor. It is initially set to $\perp$ and updated upon session acceptance.
- tr denotes the session transcript, consisting of a finite, time-ordered sequence of messages sent and received by the actor. Here, $\mathsf{tr}[i]$ denotes the set of messages exchanged in the $(i+1)$-th round of the protocol execution. We let sent and rec denote the subsequences of messages sent and received by the actor, respectively.[15]
- eph[16]denotes the set of ephemeral values associated with a session that may be leaked to the adversary.
- status denotes the current state of a session, which is either $\mathsf{running}$, $\mathsf{accepted}$, or $\mathsf{rejected}$.

Session Relations. For a session identified by π, we write $\pi.X$ as shorthand for the value of the session variable X. Two accepted sessions π and π' are said to be Match if $\pi.\mathsf{role} \neq \pi'.\mathsf{role}$ and $\pi.\mathsf{tr} = \pi'.\mathsf{tr}$. That is, two matching sessions have opposite roles and identical message transcripts.

We next introduce two auxiliary notions that capture partial agreement between sessions based on the first three protocol messages. A session π is said to be a $\mathsf{FirstMatch}$ session of π' if $\pi.\mathsf{role} \neq \pi'.\mathsf{role}$ and $\pi.\mathsf{tr}[i] = \pi'.\mathsf{tr}[i]$ for all $i \in \{0, 1\}$. A session π is said to be a $\mathsf{PartialMatch}$ session of π' if $\pi.\mathsf{role} \neq \pi'.\mathsf{role}$ and the two sessions agree on all KEM-related messages exchanged during the first three rounds of the protocol. Formally, for every $i \in \{0, 1, 2\}$, the KEM-related components (i.e., the KEM public key and ciphertext) of $\pi.\mathsf{tr}[i]$ and $\pi'.\mathsf{tr}[i]$ are identical.

4.2 Adversarial Capabilities

Following [CF12, BCL+26], we consider a (quantum) polynomial-time adversary with full control over the communication network. In particular, the adversary may arbitrarily intercept, modify, inject, delay, or drop messages. Consequently, all communication between protocol participants is mediated by the adversary, and even two honest parties can establish a session only via adversarially controlled message delivery.

To formalize the adversary's ability in a game-based security experiment, the adversary is given a set of oracles defined as follows. The first two queries, $\mathsf{NewSession}$ and Send, model the adversary's complete control over the communication network. The queries $\mathsf{RevealSK}$, $\mathsf{RevealEph}$, and $\mathsf{Corrupt}$ capture the adversary's ability to compromise participants and obtain secret information. Finally, the Test query models the adversary's challenge of distinguishing a real session key from a random one.

- $\mathsf{NewSession}(U, V, \mathsf{role})$: Creates a new session π with owner $\pi.\mathsf{actor} \leftarrow U$, intended peer $\pi.\mathsf{peer} \leftarrow V$, and role $\pi.\mathsf{Role} \leftarrow \mathsf{role}$. The peer identity may be unspecified, in which case $V = *$.
- $\mathsf{Send}(\pi, m)$: Sends a message m to session π. If the session π has not been created via $\mathsf{NewSession}$, the query returns $\perp$. Otherwise, the Send oracle executes the protocol on behalf of $\pi.\mathsf{actor}$ which receives m, updates the session state accordingly, and returns the response message and the updated session status $\pi.\mathsf{status}$.

[15] Note that rec is determined by the adversary via the send oracle defined later, whereas sent is generated according to the protocol specification.

[16] The precise contents of eph may vary across protocols and capture the specific ephemeral secret values that are allowed to be revealed to the adversary.

– $\mathsf{RevealSK}(\pi)$: If the session has not accepted (i.e., $\pi.\mathsf{status} \neq \mathsf{accepted}$), the query returns $\perp$. Otherwise, it reveals the session key $\pi.\mathsf{key}$ to the adversary.

– $\mathsf{RevealEph}(\pi)$: If $\pi.\mathsf{status} \in \{\mathsf{accepted}, \mathsf{rejected}\}$, the query returns $\perp$, reflecting the assumption that all ephemeral secret information is securely erased upon completion of the protocol.[17] Otherwise, the query reveals the ephemeral secret state $\pi.\mathsf{eph}$ to the adversary.

– $\mathsf{Corrupt}(U)$: Reveals the static secret key sk_U of party U. The time at which party U is corrupted is recorded.

– $\mathsf{Test}(\pi)$: This query may be issued at most once. A session queried by Test is called a test session and is denoted by π^{Test}. If π has not accepted (i.e., $\pi.\mathsf{status} \neq \mathsf{accepted}$), the query returns $\perp$. Otherwise, the Test oracle has a uniformly random bit b, which is fixed throughout the game. If $b = 0$, it returns the real session key $\pi.\mathsf{key}$. If $b = 1$, it samples a uniformly random key $\mathsf{key} \leftarrow\!\!\$\ \mathcal{K}$ and returns key.

4.3 Match Soundness, Key Secrecy Security, and Freshness Predicates

Match Soundness. Match soundness is defined via a security game in which the adversary faces no restrictions on its queries, and wins if it produces either (i) two matching completed sessions that derived different session keys, or (ii) three distinct completed sessions sharing the same message transcript.

Key Secrecy Security. To define key secrecy security, we consider an indistinguishability game in which the adversary attempts to distinguish a real session key from a uniformly random key, given access to the oracle queries defined above. To prevent trivial wins, the adversary's capabilities must be restricted, and such restrictions may vary across different protocols. Following [BCL+26], we abstract these restrictions by a freshness predicate F, which takes as input the adversary's query transcript and outputs true or false. For example, a simple freshness predicate F requires that the adversary has not issued any $\mathsf{RevealSK}$ query to the tested session π_{test} or to any matching session of π_{test}. Formally, F outputs true if this condition holds, and false otherwise. We now give a formal definition of F-Key Secrecy (F-KS) security. In the F-key secrecy game, the tested session is required to satisfy the freshness predicate F.

Definition 4.1 (F-KS Security). *Let Π be a key-exchange protocol among a set of parties $\mathcal{U}$, and let F be a freshness predicate defined over the adversary's query transcript. For any polynomial-time adversary $\mathcal{A}$, consider the experiment $G^{\mathsf{F\text{-}KS}}_{\Pi,\mathcal{A}}$ defined as follows:*

1. ***Setup.*** *The challenger generates all static key pairs $(\mathsf{pk}_U, \mathsf{sk}_U)$ for all identities $U \in \mathcal{U}$ and samples a uniformly random bit b (which will be used to answer Test queries). The public keys are provided to $\mathcal{A}$.*
2. ***Query.*** *The adversary has access to the oracles $\mathsf{NewSession}$, Send, $\mathsf{RevealSK}$, $\mathsf{RevealEph}$, $\mathsf{Corrupt}$, and Test.*
3. ***Stop.*** *At some point, $\mathcal{A}$ halts and outputs a guess b' for b.*
4. ***Finalize.*** *The adversary wins the game if the tested session π_{test} satisfies the predicate F and $b' = b$. Otherwise, the adversary loses the game, and the game $G^{\mathsf{F\text{-}KS}}_{\Pi,\mathcal{A}}$ outputs a random bit.*

We say Π satisfies $\mathsf{F\text{-}KS}$ security if, for all PPT adversaries $\mathcal{A}$ the advantage $\mathrm{Adv}^{\mathsf{F\text{-}KS}}_{\Pi}(\mathcal{A})$ is negligible in the security parameter λ:

$$\mathrm{Adv}^{\mathsf{F\text{-}KS}}_{\Pi}(\mathcal{A}) = \left| \Pr\left[\mathsf{G}^{\mathsf{F\text{-}KS}}_{\Pi}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2} \right|.$$

Notation for Freshness Predicates Following [BCL+26], we introduce auxiliary notation to simplify the formalization of the freshness predicates for our protocols pUAKE and pMAKE in later sections. All predicates defined in this subsection are Boolean-valued and are evaluated with respect to the adversary's query transcript.

[17] We assume that all randomness used by the protocol is securely erased upon completion of the protocol, regardless of whether the session accepts or rejects.

Let $\pi.\mathsf{Init}$ (resp. $\pi.\mathsf{Resp}$) denote the initiator (responder) party of session π. More formally, $\pi.\mathsf{Init}$ (resp. $\pi.\mathsf{Resp}$) denotes the party $\pi.\mathsf{actor}$ if $\pi.\mathsf{role} = \mathsf{Init}$ (resp. $\pi.\mathsf{role} = \mathsf{Resp}$), and $\pi.\mathsf{peer}$ otherwise. For a tested session π^{Test} and $\mathsf{X} \in \mathsf{A}, \mathsf{P}, \mathsf{I}, \mathsf{R}$, Let $\mathsf{party}(\mathsf{X})$ denote the parties $\pi^{\mathsf{Test}}.\mathsf{actor}$, $\pi^{\mathsf{Test}}.\mathsf{peer}$, $\pi^{\mathsf{Test}}.\mathsf{Init}$, and $\pi^{\mathsf{Test}}.\mathsf{Resp}$, respectively. For $\mathsf{Y} \in \{\mathsf{Match}, \mathsf{FirstMatch}, \mathsf{PartialMatch}\}$, we write $\mathsf{Set}(\mathsf{Y})$ for the set of sessions that stand in relation Y to the test session π^{Test}. Using these notations, we define the following freshness predicates:

- Y is true if and only if $\mathsf{Set}(\mathsf{Y})$ is non-empty.
- CorX is true if and only if party X has been corrupted, that is, if the adversary has issued a $\mathsf{Corrupt}(X)$ query.
- $\mathsf{CorXbeforeT}$ (resp. $\mathsf{CorXafterT}$) is true if and only if party X is corrupted before (resp. after) the test session π^{Test} accepts.
- $\mathsf{AisInit}$ is true if and only if the role of party $\mathsf{party}(\mathsf{A})$ is Init.
- $\mathsf{AisResp}$ is true if and only if the role of party $\mathsf{party}(\mathsf{A})$ is Resp.
- SkT is true if and only if the adversary has issued a $\mathsf{RevealSK}(\pi^{\mathsf{Test}})$ query.
- SkM is true if and only if there exists a session $s \in \mathsf{Set}(\mathsf{Match})$ such that the adversary has issued a $\mathsf{RevealSK}(s)$ query.
- EphT is true if and only if the adversary has issued a $\mathsf{RevealEph}(\pi^{\mathsf{Test}})$ query.
- EphY is true if and only if there exists a session $s \in \mathsf{Set}(Y)$ such that the adversary has issued a $\mathsf{RevealEph}(s)$ query.

4.4 Explicit Authentication

The Explicit Authentication property captures the requirement that whenever a session π accepts and its intended peer is not corrupted, there exists a unique honest matching session for π. Note that if π sends the final message of the exchange, then even if the intended peer is not corrupted, this does not guarantee that the final message sent by π is actually received by an honest matching session.

Following [BCL+26], we consider two variants of this notion, namely *Init-explicit authentication* and *Resp-explicit authentication*, which are defined based on the notion of malicious acceptance.

Definition 4.2 (Z-Malicious Acceptance). *Let $Z \in \{\mathsf{Init}, \mathsf{Resp}\}$. For a session π, we define the predicate MalAcc_Z to be true if and only if all of the following conditions hold:*

1. *$\pi.\mathsf{role} = Z$ and $\pi.\mathsf{status} = \mathsf{accepted}$;*
2. *$\mathsf{Corrupt}(\pi.\mathsf{peer})$ was not queried before π accepts;*
3. *there does not exist an honest session π' with $\pi'.\mathsf{actor} = \pi.\mathsf{peer}$ such that*
   - *if sessions with role Z send the final message of the protocol, then π and π' share the same message transcript except possibly for the final message sent by π and received by π', and*
   - *otherwise, π and π' share the same message transcript and π' is completed. Note that in this case, the sessions are matching.*

Definition 4.3 (X-Explicit Authentication). *Let $X \in \{\mathsf{Init}, \mathsf{Resp}\}$, and let $\overline{X}$ denote the unique role in $\{\mathsf{Init}, \mathsf{Resp}\}$ such that $\overline{X} \neq X$. Let Π be a key-exchange protocol among a set of parties $\mathcal{U}$. For any probabilistic polynomial-time adversary $\mathcal{A}$, consider the experiment $G^{X\text{-}\mathsf{EA}}_{\Pi,\mathcal{A}}$ defined as follows:*

1. ***Setup.*** *The challenger generates all static key pairs $(\mathsf{pk}_U, \mathsf{sk}_U)$ for all identities $U \in \mathcal{U}$ and provides the public keys to $\mathcal{A}$.*
2. ***Query.*** *The adversary has access to the oracles $\mathsf{NewSession}$, Send, $\mathsf{RevealSK}$, $\mathsf{RevealEph}$, and $\mathsf{Corrupt}$.*
3. ***Stop.*** *At some point, $\mathcal{A}$ halts and outputs a session π.*
4. ***Finalize.*** *The adversary wins the experiment if $\mathsf{MalAcc}_{\overline{X}}(\pi) = \mathsf{true}$.*

*We say Π satisfies X-*explicit authentication *if for all polynomial time adversaries $\mathcal{A}$, the probability of the adversary winning the experiment is negligible in the security parameter λ:*

$$\mathrm{Adv}^{X\text{-}\mathsf{EA}}_{\Pi}(\mathcal{A}) = \Pr\left[\mathsf{G}^{X\text{-}\mathsf{EA}}_{\Pi}(\mathcal{A}) \Rightarrow 1\right].$$

5 Authenticated Key Exchanges based on AFS-KEMs

In this section, we formalize and analyze our mutually authenticated protocol pMAKE (Fig. 2) and unilaterally authenticated protocol pUAKE (Fig. 4). Both protocols are constructed from AFS-KEM schemes that satisfy IND-ECPA, IND-SCCA and IND-LCK-SCCA security, and are analyzed in the post-eCK-PFS model. Moreover, we require that AFS-KEM satisfies perfect ExtractPK-correctness and ephemeral-key-generation equivalence. In particular, we show that both protocols achieve match soundness, key secrecy, and explicit authentication. We focus primarily on the security of pMAKE here, while the full analysis of pUAKE is deferred to Appendix D.

Post-id Setting and Certification Authority. Both AKE protocols are analyzed in the post-id setting, where parties neither possess each other's static public keys nor know the peer identity before protocol execution. We further assume that all party identities are unique and that a public certification authority (CA)[18] binds identities to their corresponding static public keys. Formally, the CA is modeled as a trusted verification predicate $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}, \mathsf{pk}) \in \{0, 1\}$, which outputs 1 if and only if pk is the certified static public key of id.

Leakage through RevealEph. Our analysis of the pMAKE protocol incorporates a RevealEph query to capture ephemeral-state leakage. In particular, the security model considers leakage only to ephemeral secrets that can be precomputed offline. For an initiator session, the query reveals the randomness seed_A used in EKeyGen, which uniquely determines the ephemeral secret key sk_e^A. Analogously, for a responder session, the query reveals the randomness seed_B used in EKeyGen and hence the corresponding ephemeral secret key sk_e^B. We assume the randomness used for encapsulation is securely erased after computation.

Remark 5.1 (Precomputation and State Leakage). In many existing KEM-based AKE protocols in the pre-specified-peer setting, the peer's long-term public key is known before session activation. Hence, an implementation may be tempted to precompute encapsulation outputs offline, e.g., $(\mathsf{ct}_A, \mathsf{K}_A) \leftarrow \mathsf{Encaps}(\mathsf{pk}_A)$, and store K_A for later use in an actual protocol execution, thereby reducing online computation. This implementation practice should be avoided: storing the precomputed key K_A for later use effectively turns it into local ephemeral state that may be exposed by RevealEph. If this state is leaked, an adversary may use it to complete a session while impersonating the intended peer, leading to a KCI-style state-leakage attack.

Our construction avoids this pitfall by tying the key computation to a fresh, session-specific cpk. Since cpk is generated during the protocol execution and is unavailable before session activation, the encapsulation output cannot be precomputed offline. Thus, the encapsulated key is bound to the current session, ruling out KCI-style attacks based on the leakage of retained precomputed keys.

This protection can be further strengthened by deriving the ephemeral secret key from a session-specific auxiliary string aux, e.g., $\mathsf{sk}_e \leftarrow H(\mathsf{seed}_e, \mathsf{aux})$, where aux may include the current transcript prefix, role labels, nonces, and timestamps. In this case, the corresponding ephemeral public key and combined public key are also bound to aux and therefore cannot be precomputed before the current session context is fixed. Moreover, after recovering seed_e, the peer can recompute sk_e using the same aux and verify that the received cpk is consistent with the session-bound ephemeral key material. Hence, any accepted cpk is tied to the current session context rather than to retained precomputed state.

5.1 Key Secrecy

Key secrecy with perfect forward secrecy is achieved for both pMAKE and pUAKE in the ROM and QROM, where the PRG and PRF are modeled as (quantum-accessible) random oracles. The analysis mainly relies on the IND-ECPA, IND-SCCA, and IND-LCK-SCCA security of AFS-KEM, together with its Key-Coll, SPK-Bind, and RSPK-Bind security.

We next define the freshness predicate $\mathsf{F}_{\mathsf{pMAKE}}$ for pMAKE and prove that pMAKE achieves $\mathsf{F}_{\mathsf{pMAKE}}$-key secrecy in Theorem 5.3. The corresponding freshness predicate $\mathsf{F}_{\mathsf{pUAKE}}$ and the analysis for pUAKE are deferred to Appendix D (Definition D.1, Theorems D.2, D.6).

[18] Such a certification authority can be instantiated via standard mechanisms, including public-key infrastructures (PKI), certificate validation, or trusted public directories.

PartialMatch. As defined in Sec. 4.1, two sessions π and π' are said to be PartialMatch if they agree on the KEM-related components of the first three protocol messages. Concretely, in the pMAKE protocol, sessions π and π' are PartialMatch if $(\pi.\mathsf{cpk}_A, \pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_A, \pi.\mathsf{ct}_B) = (\pi'.\mathsf{cpk}_A, \pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_A, \pi'.\mathsf{ct}_B)$, where $\pi.X$ denotes the value X in session π.

Definition 5.2 (Freshness Predicate for pMAKE). *Using the notation from Sec. 4.3, we define the freshness predicate* $\mathsf{F}_{\mathsf{pMAKE}}$ *as a disjunction of the cases* $\mathsf{FM}_1, \ldots, \mathsf{FM}_6$*, each capturing a admissible pattern of adversarial capabilities. In the subsequent security analysis, for the cases* FM_1–FM_4*, we show that the adversary's advantage in winning the key-secrecy game is negligible. The cases* FM_5 *and* FM_6 *are handled by showing that these events occur only with negligible probability.*

Specifically, the analyses of cases FM_1–FM_4 *denote the scenarios where a* PartialMatch *session of* π^{Test} *exists, and as long as at least one of the secret materials* $(\mathsf{sk}_A, \mathsf{sk}_B, \mathsf{seed}_A, \mathsf{seed}_B)$ *in the tested session* π^{Test} *and its* PartialMatch *session is not revealed to the adversary, the session key remains secure. The analyses of cases* FM_5 *and* FM_6 *indicate that if the adversary does not corrupt the peer of* π^{Test} *before* π^{Test} *accepts, then a* PartialMatch *session of* π^{Test} *must exist.*

$$
\begin{aligned}
\mathsf{FM}_1 &\triangleq \overline{\mathsf{CorA}} \wedge \mathsf{PartialMatch} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FM}_2 &\triangleq \overline{\mathsf{CorP}} \wedge \mathsf{PartialMatch} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FM}_3 &\triangleq \overline{\mathsf{EphT}} \wedge \mathsf{PartialMatch} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FM}_4 &\triangleq \overline{\mathsf{EphPartialMatch}} \wedge \mathsf{PartialMatch} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FM}_5 &\triangleq \mathsf{CorPafterT} \wedge \overline{\mathsf{PartialMatch}} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FM}_6 &\triangleq \overline{\mathsf{CorP}} \wedge \overline{\mathsf{PartialMatch}} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}}.
\end{aligned}
$$

We define

$$
\mathsf{F}_{\mathsf{pMAKE}} \triangleq \bigvee_{i=1}^{6} \mathsf{FM}_i\,.
$$

Theorem 5.3 ($\mathsf{F}_{\mathsf{pMAKE}}$-Key Secrecy of pMAKE in the ROM and QROM). *Let there be at most N parties and at most S sessions activated in the* $\mathsf{F}_{\mathsf{pMAKE}}$-KS *game. Suppose that the underlying* AFS-KEM *is δ-correct, satisfies γ-spreadness, and satisfies perfect* ExtractPK*-correctness and ephemeral-key-generation equivalence. We model the* PRF *as a random oracle* G*, to which the adversary makes at most q_{G} queries. Then, for any PPT adversary $\mathcal{A}$ attacking the* $\mathsf{F}_{\mathsf{pMAKE}}$-KS *security of* pMAKE*, there exist PPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{C}, \mathcal{D}, \mathcal{F}$, and $\mathcal{Z}$, constructed below and running in time comparable to that of $\mathcal{A}$, such that*

$$
\begin{aligned}
\mathsf{Adv}^{\mathsf{F}_{\mathsf{pMAKE}}\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le\ & \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}}\\
& + NS\left(\begin{array}{l}
4\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{LCK}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_3) + 2(S+1)\,\mathsf{Adv}^{\mathsf{dual}\text{-}\mathsf{PRF}}_{\mathsf{PRF}}(\mathcal{Z})\\
\quad + 4(2NS+1)\,\mathsf{Adv}^{\mathsf{RSPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{D})\\
\quad + S\left(\begin{array}{l}
4\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_1) + 8\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{ECPA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_2)\\
\quad + 8(S+1)\,\mathsf{Adv}^{\mathsf{Key}\text{-}\mathsf{Coll}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{F}) + \dfrac{4q_{\mathsf{G}}}{|\mathcal{K}|} + 16\delta\\
\quad + (8N^2 + 29N + 20)\,\mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C})
\end{array}\right)
\end{array}\right).
\end{aligned}
$$

If we model PRF *as a quantum-accessible random oracle* G*, to which the adversary makes at most q_{G} quantum queries, then, for any QPT adversary $\mathcal{A}$ attacking the* $\mathsf{F}_{\mathsf{pMAKE}}$-KS *security of* pMAKE*, there*

exist QPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{C}, \mathcal{D}, \mathcal{F}$, *and* $\mathcal{Z}$, *running in time comparable to that of* $\mathcal{A}$, *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{F}_{\mathsf{pMAKE}}\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq{}& \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} \\ &+ NS\left(\begin{array}{l} 4\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + 2(S+1)\,\mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 4\,\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) \\ + S\left(\begin{array}{l} 4\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + 8(S+1)\,\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ + (5N+4)\,\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + 16\delta \end{array}\right) \end{array}\right) \\ &+ 4NS^2\left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|} + \beta\right)} + \beta\right),\end{aligned}$$

where $\beta = N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$. *Here* k_1 *and* k_2 *are the min-entropy bounds on the honest static and combined public keys of* AFS-KEM, *respectively. Furthermore,* $|\mathcal{K}|$ *denotes the cardinality of the* AFS-KEM *encapsulated key space.*

Proof sketch. The full proof is given in Appendix E.1. We present only a proof sketch here. We assume that all honest users hold distinct static public keys, that all honest sessions derive distinct combined public keys cpk, including cpk_A and cpk_B, and that AFS-KEM is perfectly correct. These assumptions follow from the high min-entropy of the static public keys and the combined public keys, together with the δ-correctness of AFS-KEM. Under these assumptions, a session cannot have more than one PartialMatch or Match session.

According to the definition of $\mathsf{F}_{\mathsf{pMAKE}}$ (Definition 5.2), we analyze the six disjoint freshness cases FM_1–FM_6. Let π denote the tested session, and let π' denote its unique PartialMatch session, if it exists. By symmetry of the protocol, we assume without loss of generality that $\pi.\mathsf{role} = \mathsf{Resp}$. The case $\pi.\mathsf{role} = \mathsf{Init}$ follows by a similar argument.

Analysis for Cases FM_1 and FM_2 in the standard model. The analyses of Cases FM_1 and FM_2 are similar. We therefore focus on Case FM_2. Accordingly, we consider the setting where $\pi.\mathsf{role} = \mathsf{Resp}$ and $\pi.\mathsf{peer} = P^*$ is uncorrupted. First, by the SPK-Bind security of AFS-KEM (Definition 3.5), we have $\pi'.\mathsf{actor} = P^*$ and if π' accepts, then π' must be the unique Match session of π. At a high level, cpk provides a binding to both its static public key pk_s and its ephemeral seed seed_e. Any attempt by the adversary to change the identity id (hence the corresponding static public key) or c_{seed} (which changes the recovered seed_e since the OTP encryption is deterministic under a fixed key) will make the cpk verification fail. Then we guess the sessions π and π', as well as the peer identity $\pi.\mathsf{peer}$, denoted by P^*, which succeeds with probability $1/(NS^2)$. We then rely on the IND-SCCA security of AFS-KEM to replace the KEM key K_A in sessions π and π' with a uniformly random value, provided that the adversary does not know the static secret key sk_{P^*}. Notably, the decapsulation oracle is used to simulate all other sessions $\hat{\pi} \notin \{\pi, \pi'\}$ with $\hat{\pi}.\mathsf{actor} = P^*$, without knowing sk_{P^*}. Next, by the security of the PRF, we replace $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, and $\mathsf{K}^B_{\mathsf{ENC}}$ in sessions π and π' with independent uniformly random keys. Therefore, now at most one accepted session derives the same session key as π, namely the matching session, while any other accepted session derives an independent session key. By the definition of FM_2, the session key of the matching session of π cannot be revealed to the adversary. Thus, the adversary's advantage in this case is negligible.

Analysis for Cases FM_3 and FM_4 in the ROM. In these two cases, we model the PRF as a random oracle G. The analyses of Cases FM_3 and FM_4 are similar, so we focus on Case FM_4 here. That is, we consider the setting where the seed_A in π' is not leaked to the adversary and $\pi.\mathsf{role} = \mathsf{Resp}$. We first guess the sessions π and π', as well as the identity of $\pi'.\mathsf{actor}$, which succeeds with probability $1/(NS^2)$. First, by the key-collision security of AFS-KEM (Definition 3.7), except with negligible probability there exists no other honestly generated session $\hat{\pi} \notin \{\pi, \pi'\}$ that derives the same KEM key $\mathsf{K}^{\mathsf{Test}}_A$, where $\mathsf{K}^{\mathsf{Test}}_A$ denotes the value of K_A in π and π'. We then replace $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, and $\mathsf{K}^B_{\mathsf{ENC}}$ in sessions π and π' with the same uniformly random values. We define the event QUERY as the adversary querying G with input $(\mathsf{K}^{\mathsf{Test}}_A, \cdot)$. Conditioned on $\neg\mathsf{QUERY}$, this replacement is indistinguishable to the adversary. Then, the SPK-Bind security of AFS-KEM implies that at most one accepted session derives the same session key as π, namely the matching session, whose key is unrevealed by definition. Consequently, the adversary's advantage in this case is negligible.

It therefore suffices to bound the probability of the event QUERY. Since the OTP key $\mathsf{K}^A_{\mathsf{ENC}}$ is uniform from the adversary's view, perfect secrecy of the OTP allows us to replace c^A_{seed} in π' with an OTP encryption of an independent random seed. Then, by the IND-ECPA security of AFS-KEM, we replace the KEM key $\mathsf{K}^{\mathsf{Test}}_A$ in π and π' with a uniformly random value. After these hybrids, the value $\mathsf{K}^{\mathsf{Test}}_A$ is uniformly random and independent of the adversary's view, and hence Pr[QUERY] is negligible.

Analysis for Cases FM_3 and FM_4 in the QROM. We lift the ROM argument for Cases FM_3,FM_4 to the QROM using the semi-classical oracle technique of [AHU19].

Analysis for Cases FM_5 and FM_6 in the standard model. Since the arguments for Cases FM_5 and FM_6 are analogous, we focus on Case FM_6. We show that the event FM_6 occurs only with negligible probability. We first guess the tested session π and its peer identity $\pi.\mathsf{peer}$, denoted by P^*, which succeeds with probability $1/(NS)$. Based on the IND-LCK-SCCA security of AFS-KEM, we can replace the KEM key K_A in session π and in the (unique, if exists) session $\overline{\pi}$ satisfying $\pi.\mathsf{cpk}_A = \overline{\pi}.\mathsf{cpk}_A$ and $\pi.\mathsf{ct}_A = \overline{\pi}.\mathsf{ct}_A$ with the same uniformly random value. The reduction proceeds as follows. It receives a challenge public key pk_s, which is embedded as pk_{P^*}. When session π receives cpk_A, the reduction forwards it to its challenger and obtains a challenge ciphertext ct together with a key K_b, which are then embedded as $\mathsf{K}_A = K_b$ and $\mathsf{ct}_A = \mathsf{ct}$ in π and $\overline{\pi}$ (if exists). All other sessions with actor P^* are simulated using the decapsulation oracle.

Notably, if such a session $\overline{\pi}$ exists and $\overline{\pi}$ is not the PartialMatch session of π, then by the key-collision security of AFS-KEM we have $\pi.\mathsf{K}_B \neq \overline{\pi}.\mathsf{K}_B$. We then replace $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, and $\mathsf{K}^B_{\mathsf{ENC}}$ in session π and in its PartialMatch session (if exists) with the same uniformly random values, while assigning an independent uniformly random key to the session $\overline{\pi}$ (if exists) that is not the PartialMatch session of π. Since only the PartialMatch session shares the same encryption keys as π, if such a PartialMatch session does not exist, then these encryption keys are never used in any other session. Therefore, the seed $\mathsf{seed} = \mathsf{K}_{\mathsf{ENC}} \oplus \mathsf{c}_{\mathsf{seed}}$ recovered by π is uniform random from the adversary's view by the perfect secrecy of the OTP. We can then rely on the RSPK-Bind security of the underlying AFS-KEM scheme (Definition 3.6) to show that right now the session π accepts only with negligible probability, Consequently, Case FM_6 can occur only with negligible probability.

5.2 Match Soundness.

Match soundness holds for both pUAKE and pMAKE. The first condition follows directly from the correctness of AFS-KEM. For the second condition, the min-entropy of the combined public key ensures that two distinct sessions derive the same composite public key cpk only with negligible probability. Moreover, the γ-spreadness property of AFS-KEM ensures that two distinct sessions that receive valid cpk (resp. cpk_A, cpk_B) from their peer produce the same encapsulation ciphertext only with negligible probability.

5.3 Explicit Authentication.

Init-explicit authentication is achieved by both pMAKE and pUAKE, while pMAKE additionally provides Resp-explicit authentication. The arguments establishing explicit authentication for pMAKE closely follow the analysis of Cases FM_5 and FM_6. The only difference lies in the final step, where we further show that if a session π accepts, then the unique existing PartialMatch session must satisfy all requirements of the explicit-authentication property. For example, to prove Init-explicit authentication for pMAKE, consider any responder session π that accepts and whose peer has not been corrupted before π accepts. In the final step, we show that the PartialMatch session of π must additionally have the same third-round message as π.

The formal analyses of Explicit Authentication are deferred to Section E.3 for pMAKE and Section D.4 for pUAKE.

6 Generic Construction of AFS-KEM

In this section, we first introduce *Authenticated Forward-Secure Public-Key Encryption* (AFS-PKE) together with its associated security requirements. We then show how to obtain a CCA-secure AFS-KEM

from a CPA-secure AFS-PKE via the Fujisaki–Okamoto (FO) transform [FO99, FO13, HHK17]. Once the appropriate definition of AFS-PKE is established, the FO transform can be applied in a natural manner, and the resulting security proof closely follows those for standard KEM constructions [JZC+18, JZM19, BHH+19, AHU19, DHK+21].

6.1 Authenticated Forward-Secure PKE

Similar to AFS-KEM, an AFS-PKE combines a static key pair with an ephemeral key pair into a single *composite* public/secret key pair. The composite key material is then used by the encryption and decryption algorithms.

Definition 6.1 (Authenticated Forward-Secure PKE). *An authenticated forward-secure public-key encryption scheme (*AFS-PKE*) over message space $\mathcal{M}$ is a tuple of algorithms $\Pi_{\mathsf{AFS\text{-}PKE}} = (\mathsf{SKeyGen}, \mathsf{EKeyGen}, \mathsf{EKeyGen_c}, \mathsf{CombPK}, \mathsf{CombSK}, \mathsf{ExtractPK}, \mathsf{Enc_{AFS}}, \mathsf{Dec_{AFS}})$. where all algorithms are defined analogously to those of* AFS-KEM*, except that encryption and decryption operate over an explicit message space $\mathcal{M}$. Specifically, the encryption algorithm $\mathsf{Enc_{AFS}}(\mathsf{cpk}, m)$ (when necessary, we make the encryption randomness explicit by writing $\mathsf{Enc_{AFS}}(\mathsf{cpk}, m; r)$, where $r \leftarrow\!\!\$\ \mathcal{R}$) takes a composite public key cpk and a message $m \in \mathcal{M}$, and outputs a ciphertext $\mathsf{ct} \in \mathcal{CT}$. Meanwhile, the decryption algorithm $\mathsf{Dec_{AFS}}(\mathsf{csk}, \mathsf{ct})$ takes the corresponding composite secret key csk and a ciphertext $\mathsf{ct} \in \mathcal{CT}$, and outputs either a message $m' \in \mathcal{M}$ or $\perp$. The complete definition of* AFS-PKE *is given in Definition A.5.*

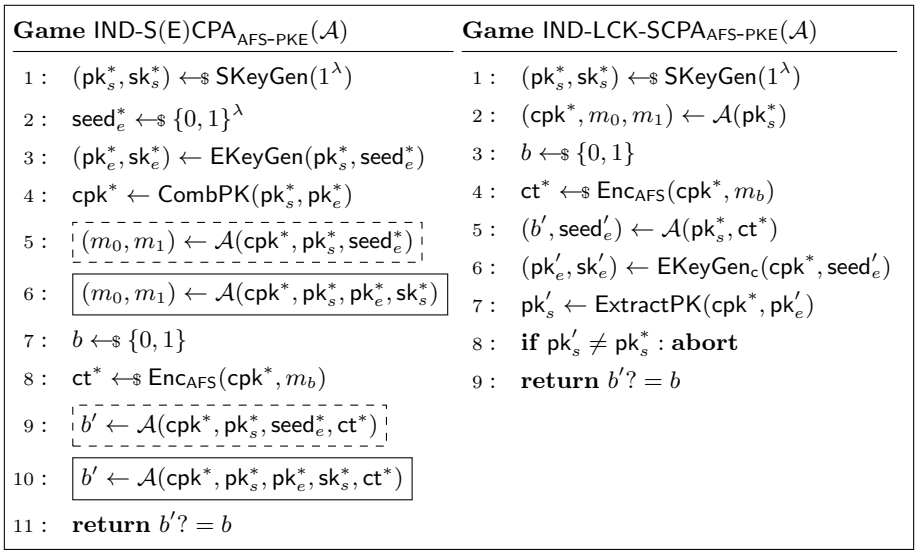


Fig. 7. The IND-ECPA, IND-SCPA, and IND-LCK-SCPA security games for AFS-PKE. Dashed boxes denote IND-SCPA, and solid boxes denote IND-ECPA.

Correctness for AFS-PKE. An AFS-PKE scheme $\Pi_{\mathsf{AFS\text{-}PKE}}$ with message space $\mathcal{M}$ is δ-correct if $\mathbb{E}\big[\max_{m \in \mathcal{M}} \Pr\big[\mathsf{Dec_{AFS}}\big(\mathsf{csk}, \mathsf{Enc_{AFS}}(\mathsf{cpk}, m)\big) \neq m\big]\big] \leq \delta$, where the expectation is taken over the randomness of $(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow\!\!\$\ \mathsf{SKeyGen}(1^\lambda)$, $\mathsf{seed} \leftarrow\!\!\$\ \{0,1\}^\lambda$, $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})$, $\mathsf{cpk} \leftarrow \mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e)$, and $\mathsf{csk} \leftarrow \mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e)$.

Min-Entropy of the Combined Public Key. An AFS-PKE has combined public keys with min-entropy at least k if, for all $\mathsf{pk}_s \in \mathcal{PK}_s$ and $\mathsf{cpk}' \in \mathcal{CPK}$, $\Pr_{\mathsf{seed} \leftarrow\!\!\$ \{0,1\}^\lambda}[\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e) = \mathsf{cpk}'] \leq 2^{-k}$, where pk_e is generated as $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})$.

We define the security notions IND-ECPA, IND-SCPA, and IND-LCK-SCPA for AFS-PKE, as shown in Fig. 7. These notions parallel the corresponding security definitions of AFS-KEM, but are formulated in the PKE setting without decryption oracle access.

6.2 A General Construction of CCA-Secure AFS-KEM from CPA-Secure AFS-PKE

Based on a CPA-secure AFS-PKE scheme, a CCA-secure AFS-KEM can be constructed via the $\mathsf{FO}_{\mathsf{AFS}}$ transformation defined below, following the general FO paradigm of [FO99, HHK17, DHK+21]. Our $\mathsf{FO}_{\mathsf{AFS}}$ transformation takes as input an AFS-PKE scheme together with hash functions $\mathsf{H} : \{0,1\}^* \to \mathcal{K}$, $\mathsf{G} : \{0,1\}^* \to \mathcal{R}$, and $\mathsf{H}_{\mathsf{rej}} : \{0,1\}^* \to \mathcal{K}$, and outputs an AFS-KEM as described in Fig. 8. In our FO-transformed construction, the decapsulation algorithm additionally takes the combined public key cpk as input, and cpk is used to derive both the encapsulated key and the implicit-rejection keys.

In particular, the $\mathsf{FO}_{\mathsf{AFS}}$ transformation enhances the security of the underlying PKE scheme $\Pi_{\mathsf{AFS\text{-}PKE}}$ by lifting its IND-LCK-SCPA/IND-SCPA and IND-ECPA security to IND-LCK-SCCA/IND-SCCA and IND-ECPA security, respectively. Theorem 6.2 summarizes the resulting security guarantees of the $\mathsf{FO}_{\mathsf{AFS}}$ transformation in both the ROM and the QROM, with full proofs deferred to Appendix B. Notably, the proof follows the standard techniques used in prior analyses of FO-based KEM constructions [HHK17, JZC+18, JZM19, BHH+19, DHK+21].

$\underline{\mathsf{SKeyGen}(1^\lambda)}$
1 : $s \leftarrow\!\!\$\ \{0,1\}^\ell$
2 : $(\mathsf{pk}'_s, \mathsf{sk}'_s) \leftarrow\!\!\$\ \mathsf{SKeyGen}'(1^\lambda)$
3 : $\mathsf{sk}_s \leftarrow (\mathsf{sk}'_s, s),\ \mathsf{pk}_s \leftarrow \mathsf{pk}'_s$
4 : **return** $(\mathsf{pk}_s, \mathsf{sk}_s)$

$\underline{\mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})}$
1 : $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}'(\mathsf{pk}_s, \mathsf{seed})$
2 : **return** $(\mathsf{pk}_e, \mathsf{sk}_e)$

$\underline{\mathsf{EKeyGen}_{\mathsf{c}}(\mathsf{cpk}, \mathsf{seed})}$
1 : $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}_{\mathsf{c}}{}'(\mathsf{cpk}, \mathsf{seed})$
2 : **return** $(\mathsf{pk}_e, \mathsf{sk}_e)$

$\underline{\mathsf{ExtractPK}(\mathsf{cpk}, \mathsf{pk}_e)}$
1 : $\mathsf{pk}_s \leftarrow \mathsf{ExtractPK}'(\mathsf{cpk}, \mathsf{pk}_e)$
2 : **return** pk_s

$\underline{\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e)}$
1 : $\mathsf{cpk} \leftarrow \mathsf{CombPK}'(\mathsf{pk}_s, \mathsf{pk}_e)$
2 : **return** cpk

$\underline{\mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e)}$
1 : **parse** $(\mathsf{sk}'_s, s) \leftarrow \mathsf{sk}_s$
2 : $\mathsf{csk}' \leftarrow \mathsf{CombSK}'(\mathsf{sk}'_s, \mathsf{sk}_e)$
3 : $\mathsf{csk} \leftarrow (\mathsf{csk}', s)$
4 : **return** csk

$\underline{\mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk})}$
1 : $m \leftarrow\!\!\$\ \mathcal{M}$
2 : $K \leftarrow \mathsf{H}(\mathsf{cpk}, m)$
3 : $r \leftarrow \mathsf{G}(\mathsf{cpk}, m)$
4 : $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m; r)$
5 : **return** (ct, K)

$\underline{\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{cpk}, \mathsf{csk}, \mathsf{ct})}$
1 : **parse** $(\mathsf{csk}', s) \leftarrow \mathsf{csk}$
2 : $m' \leftarrow \mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}', \mathsf{ct})$
3 : $K \leftarrow \mathsf{H}(\mathsf{cpk}, m')$
4 : $r \leftarrow \mathsf{G}(\mathsf{cpk}, m')$
5 : $\tilde{K} \leftarrow \mathsf{H}_{\mathsf{rej}}(\mathsf{cpk}, s, \mathsf{ct})$
6 : **if** $m' = \perp$ **or** $\mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m'; r) \neq \mathsf{ct}$:
7 : **return** $\tilde{K}$
8 : **return** K

Fig. 8. Our $\mathsf{FO}_{\mathsf{AFS}}$ transform for constructing an AFS-KEM $\Pi_{\mathsf{AFS\text{-}KEM}} = (\mathsf{SKeyGen}, \mathsf{EKeyGen}, \mathsf{EKeyGen}_{\mathsf{c}}, \mathsf{CombPK}, \mathsf{CombSK}, \mathsf{ExtractPK}, \mathsf{Encaps}_{\mathsf{AFS}}, \mathsf{Decaps}_{\mathsf{AFS}})$ from an AFS-PKE scheme $\Pi_{\mathsf{AFS\text{-}PKE}} = (\mathsf{SKeyGen}', \mathsf{EKeyGen}', \mathsf{EKeyGen}'_{\mathsf{c}}, \mathsf{CombPK}', \mathsf{CombSK}', \mathsf{ExtractPK}', \mathsf{Enc}_{\mathsf{AFS}}, \mathsf{Dec}_{\mathsf{AFS}})$.

Theorem 6.2 ($\mathsf{FO}_{\mathsf{AFS}}$ transform in the ROM and QROM). *Let $\Pi_{\mathsf{AFS\text{-}KEM}}$ denote the* AFS-KEM *scheme obtained by applying the $\mathsf{FO}_{\mathsf{AFS}}$ transformation of Fig. 8 to an underlying δ-correct* AFS-PKE *scheme $\Pi_{\mathsf{AFS\text{-}PKE}}$ over message space $\mathcal{M}$. Let H, G, and $\mathsf{H}_{\mathsf{rej}}$ be modeled as random oracles. Assume that the composite public keys generated in $\Pi_{\mathsf{AFS\text{-}PKE}}$ have min-entropy at least k. For any adversary $\mathcal{A}$ against the* IND-LCK-SCCA *security of $\Pi_{\mathsf{AFS\text{-}KEM}}$ (resp. its* IND-SCCA *or* IND-ECPA *security) that makes at most q_{D} queries to $\mathcal{O}^{\mathsf{Dec}_s}$, at most q_{E} queries to $\mathcal{O}^{\mathsf{EKeyGen}}$, and at most q_{H}, q_{G}, and $q_{\mathsf{H}_{\mathsf{rej}}}$ queries to H, G, and $\mathsf{H}_{\mathsf{rej}}$, respectively, there exists an adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \alpha,$$
$$\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \alpha,$$
$$\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \frac{2\,(q_{\mathsf{H}} + q_{\mathsf{G}})}{|\mathcal{M}|}.$$

If H, G, *and* $\mathsf{H}_{\mathsf{rej}}$ *are modeled as quantum-accessible random oracles and* $\mathcal{A}$ *makes at most* $q_{\mathsf{H}}, q_{\mathsf{G}}, q_{\mathsf{H}_{\mathsf{rej}}}$ *(quantum) queries to* H, G, *and* $\mathsf{H}_{\mathsf{rej}}$, *respectively, then there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) \leq 2\sqrt{\hat{q}\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + 4\,(q_{\mathsf{G}}+q_{\mathsf{H}}+q_{\mathsf{D}})/|\mathcal{M}|\Big)} + \beta,$$

$$\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) \leq 2\sqrt{\hat{q}\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + 4\,(q_{\mathsf{G}}+q_{\mathsf{H}}+q_{\mathsf{D}})/|\mathcal{M}|\Big)} + \beta,$$

$$\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) \leq 2\sqrt{\hat{q}_{\mathsf{ECPA}}\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + 4\,(q_{\mathsf{G}}+q_{\mathsf{H}})/|\mathcal{M}|\Big)}.$$

Here $\hat{q} = q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}} + 1$, $\hat{q}_{\mathsf{ECPA}} = q_{\mathsf{H}} + q_{\mathsf{G}} + 1$, $\alpha = 2\,(q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}})\,|\mathcal{M}|^{-1} + q_{\mathsf{H}_{\mathsf{rej}}}\,2^{-\lambda} + (q_{\mathsf{E}}+1)^2\,2^{-k} + (q_{\mathsf{D}} + q_{\mathsf{G}} + q_{\mathsf{H}})\,\delta$, *and* $\beta = (q_{\mathsf{E}}+1)^2 2^{-k} + 2q_{\mathsf{H}_{\mathsf{rej}}}2^{-\lambda/2} + 16(q_{\mathsf{E}}+1)\delta\,(q_{\mathsf{G}}+q_{\mathsf{H}}+q_{\mathsf{D}}+1)^2$.

7 AFS-KEM-ML: AFS-KEM Instantiations from ML-KEM

In this section, we present the construction AFS-KEM-ML derived from ML-KEM [Nat24b] under the standard MLWE assumption. The scheme is obtained by applying the $\mathsf{FO}_{\mathsf{AFS}}$ transformation to the ML-KEM–based instantiation of authenticated forward-secure PKE, denoted AFS-PKE-ML.

$\mathsf{SKeyGen}(1^\lambda)$

$\mathsf{seed}_{\mathsf{pk}} \leftarrow\!\$\ \{0,1\}^{256}$
$\mathsf{seed}_s \leftarrow\!\$\ \{0,1\}^{\lambda}$
$\mathbf{A} \leftarrow \mathsf{G}(\mathsf{seed}_{\mathsf{pk}}) \in \mathcal{R}_q^{k\times k}$
$(\mathbf{s}_s, \mathbf{e}_s) \leftarrow \mathsf{G}'(\mathsf{seed}_s) \in \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$
$\mathbf{t}_s = \mathbf{A}\mathbf{s}_s + \mathbf{e}_s \bmod q$
$\mathsf{pk}_s = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_s)$
$\mathsf{sk}_s = \mathbf{s}_s$
return $(\mathsf{pk}_s,\ \mathsf{sk}_s)$

$\mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e)$

$\mathbf{s}_c = \mathsf{sk}_s + \mathsf{sk}_e \bmod q$
$\mathsf{csk} = \mathbf{s}_c$
return csk

$\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e)$

parse $(\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_s) \leftarrow \mathsf{pk}_s$
parse $(\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_e) \leftarrow \mathsf{pk}_e$
$\mathbf{t}_c = \mathbf{t}_s + \mathbf{t}_e \bmod q$
$\mathsf{cpk} = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_c)$
return cpk

$\mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed}_e)$

parse $(\mathsf{seed}_{\mathsf{pk}}, \cdot) \leftarrow \mathsf{pk}_s$
$\mathbf{A} \leftarrow \mathsf{G}(\mathsf{seed}_{\mathsf{pk}})$
$(\mathbf{s}_e, \mathbf{e}_e) \leftarrow \mathsf{G}'(\mathsf{seed}_e) \in \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$
$\mathbf{t}_e = \mathbf{A}\mathbf{s}_e + \mathbf{e}_e \bmod q$
$\mathsf{pk}_e = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_e)$
$\mathsf{sk}_e = \mathbf{s}_e$
return $(\mathsf{pk}_e, \mathsf{sk}_e)$

$\mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m, r)$

parse $(\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_c) \leftarrow \mathsf{cpk}$
$\mathbf{A} \leftarrow \mathsf{G}(\mathsf{seed}_{\mathsf{pk}})$
$\mathbf{r} \leftarrow\!\$\ \mathcal{B}_{\eta_1}^k$
$(\mathbf{e}_1, e_2) \leftarrow\!\$\ \mathcal{B}_{\eta_2}^k \times \mathcal{B}_{\eta_2}$
$\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$
$v = \mathbf{t}_c^T\mathbf{r} + e_2 + \mathsf{Encode}(m)$
$\mathsf{ct}_1 = \mathsf{Compress}_q(\mathbf{u}, d_u)$
$\mathsf{ct}_2 = \mathsf{Compress}_q(v, d_v)$
return $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$

$\mathsf{EKeyGen}_{\mathsf{c}}(\mathsf{cpk}, \mathsf{seed}_e)$

parse $(\mathsf{seed}_{\mathsf{pk}}, \cdot) \leftarrow \mathsf{cpk}$
$\mathbf{A} \leftarrow \mathsf{G}(\mathsf{seed}_{\mathsf{pk}})$
$(\mathbf{s}_e, \mathbf{e}_e) \leftarrow \mathsf{G}'(\mathsf{seed}_e) \in \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$
$\mathbf{t}_e = \mathbf{A}\mathbf{s}_e + \mathbf{e}_e \bmod q$
$\mathsf{pk}_e = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_e)$
$\mathsf{sk}_e = \mathbf{s}_e$
return $(\mathsf{pk}_e, \mathsf{sk}_e)$

$\mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}, \mathsf{ct})$

$\mathbf{u}' = \mathsf{Decompress}_q(\mathsf{ct}_1, d_u)$
$v' = \mathsf{Decompress}_q(\mathsf{ct}_2, d_v)$
$m' = \mathsf{Decode}(v' - \mathbf{s}_c^T\mathbf{u}')$
return m'

$\mathsf{ExtractPK}(\mathsf{cpk}, \mathsf{pk}_e)$

parse $(\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_c) \leftarrow \mathsf{cpk}$
parse $(\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_e) \leftarrow \mathsf{pk}_e$
$\mathbf{t}_s = \mathbf{t}_c - \mathbf{t}_e \bmod q$
$\mathsf{pk}_s = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_s)$
return pk_s

Fig. 9. Algorithms for AFS-PKE-ML.

7.1 Notations.

Let n and q be positive integers, we define $\mathcal{R}_q = \mathbb{Z}_q[x]/\,(x^n+1)$ where $n = 2^{k-1}$ such that x^n+1 is the 2^k-th cyclotomic polynomial. By default, regular font letters denote elements in $\mathcal{R}_q$, bold lower-case letters are column vectors, and bold upper-case letters are matrices. For a column vector $\mathbf{v}$ and a matrix $\mathbf{A}$, we write $\mathbf{v}^T$ and $\mathbf{A}^T$ as their transpose, respectively.

Given any real number x, $\lfloor x \rceil$ denotes the rounding of x to the closest integer with ties being rounded up. For an even (resp. odd) positive integer α, let $r' = r \bmod^{\pm} \alpha$ denote the unique integer in $(-\alpha/2, \alpha/2]$ (resp. $(-(\alpha-1)/2, (\alpha-1)/2]$) such that $\alpha | (r-r')$. For any positive integer α, $r' = r \bmod^{+} \alpha$ denotes the unique integer in $[0, \alpha)$ such that $\alpha | (r - r')$, and $r' = r \mod \alpha$ means that there is no need to strictly distinguish between the above two situations.

For any $f \in \mathbb{Z}_q$, we write $\|f\|_\infty$ for $\left|f \bmod^{\pm} q\right|$, then we define the infinity norm for $w \in \mathcal{R}_q$ as $\|w\|_\infty = \max_i \|f_i\|_\infty$, where $w = \sum_{i=0}^{n-1} f_i x^i$. Similarly, for $\mathbf{w} = (w_1, \cdots, w_k) \in \mathcal{R}_q^k$, we have $\|\mathbf{w}\|_\infty = \max_i \|w_i\|_\infty$.

For a finite set S, $x \leftarrow\!\!\$\ S$ means that x is chosen uniformly at random from S. For a probabilistic distribution D, $x \leftarrow D$ is the operation of picking an element x according to D. Given a positive integer η, we define $\mathcal{B}_\eta$ as the centered binomial distribution with parameter η, and $x \leftarrow \mathcal{B}_\eta$ can be given by $x = \sum_{i=1}^{\eta} (a_i - b_i)$, where $(a_1, ..., a_\eta, b_1, ..., b_\eta) \leftarrow\!\!\$\ \{0,1\}^{2\eta}$. Moreover, the aforementioned operations can be trivially extended to the case of polynomials.

7.2 AFS-PKE-ML

Let k, d_u, d_v be positive integers, and let $\mathcal{M} = \{0,1\}^{256}$ denote the message space. Let $\mathsf{G} : \{0,1\}^* \to \mathcal{R}_q$ and $\mathsf{G}' : \{0,1\}^* \to \mathcal{B}_\eta$ be extendable-output functions (XOFs) that take bit strings as input and can produce outputs of arbitrary length. Fig. 9 illustrates the construction of AFS-PKE-ML derived from ML-KEM.PKE. We define $\mathsf{Compress}_q(x, d) = \lfloor 2^d x/q \rceil \bmod^{+} 2^d$ and $\mathsf{Decompress}_q(x, d) = \lfloor qx/2^d \rceil$ for $x \in \mathbb{Q}$, and apply these functions coefficient-wise for $x \in \mathcal{R}_q$. Moreover, $\mathsf{Decode}(m)$ maps each bit m_i of m to $\lfloor q/2 \rceil\, m_i$, and $\mathsf{Encode}(\cdot)$ outputs $m_i = \lfloor \frac{2}{q}(v' - \mathbf{s}^T \mathbf{u}')_i \rceil$.

In ML-KEM.PKE, a public key consists of a seed and a polynomial vector in $\mathcal{R}_q^k$. Accordingly, key combination reuses the seed and performs coefficient-wise addition modulo q on the polynomial vectors. Formally, the composite public key is defined as $\mathsf{cpk} = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_s + \mathbf{t}_e \bmod q)$, and the composite secret key as $\mathsf{csk} = \mathsf{sk}_s + \mathsf{sk}_e \bmod q$. The original static public key can be efficiently recovered by subtracting the ephemeral public key from the composite one, yielding an algebraic ExtractPK procedure with perfect correctness. Here, we critically rely on the additive homomorphism of MLWE to build AFS-PKE. Analogously, we note that AFS-PKE can also be built based on discrete logarithm by exploiting its multiplicative homomorphism, which is beyond the scope of this work.

Theorem 7.1 (Correctness of AFS-PKE-ML). *Using the notation in Fig. 9, the* AFS-PKE-ML *is δ-correct provided that*

$$\Pr\left[\left\|(\mathbf{e}_s + \mathbf{e}_e)^T \mathbf{r} + e_2 + e_v - (\mathbf{s}_s + \mathbf{s}_e)^T (\mathbf{e}_1 + \mathbf{e}_u)\right\|_\infty \geq \frac{q}{4}\right] = \delta,$$

where $\mathbf{e}_u = \mathsf{Decompress}_q(\mathsf{ct}_1, d_u) - \mathbf{A}^T \mathbf{r} + \mathbf{e}_1$, *and* $e_v = \mathsf{Decompress}_q(\mathsf{ct}_2, d_v) - \mathbf{t}^T \mathbf{r} + e_2$.

The proof of Theorem 7.1 is provided in Appendix C.1.

Theorem 7.2 (IND-LCK-SCPA Security of AFS-PKE-ML in the ROM/QROM). *We model the hash functions* G *and* G' *as random oracles. For any adversary $\mathcal{A}$ against the* IND-LCK-SCPA *security of* AFS-PKE-ML *making at most q_{G} and $q_{\mathsf{G}'}$ queries to* G *and* G', *respectively, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ running in time comparable to that of $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k,k,q}(\mathcal{B}) + q_{\mathsf{G}'}\, \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}) + q_{\mathsf{G}'}\, 2^{-\lambda}.$$

If the hash functions G *and* G' *are modeled as quantum-accessible random oracles and $\mathcal{A}$ makes at most q_{G} and $q_{\mathsf{G}'}$ (quantum) queries to* G *and* G', *respectively, then there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ running in time comparable to that of $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k,k,q}(\mathcal{B}) + (2q_{\mathsf{G}'} + 1)^2\, \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}) + 2q_{\mathsf{G}'}\, 2^{-\lambda/2}.$$

Proof sketch. We outline the ROM proof of IND-LCK-SCPA security and defer the full proof to Appendix C.2. We first assume that the adversary queries $\mathsf{G}'(\mathsf{seed}'_e)$ before outputting seed'_e. The proof then invokes the MLWE assumption to replace the static public key pk^*_s with a uniformly random element. Conditioned on non-abort, the adversary's output cpk^* and seed'_e must satisfies

$\mathsf{cpk}^* = (\mathsf{seed}^*_{\mathsf{pk}}, \mathbf{t}^*_s + \mathbf{A}\mathbf{s}^*_e + \mathbf{e}^*_e)$, where $(\mathbf{s}^*_e, \mathbf{e}^*_e) = \mathsf{G}'(\mathsf{seed}'_e)$. In the next game hop, the challenger guesses an index $i \in [q]$, where seed'_e is first queried in the $(i+1)$-th G' query. This guess succeeds with probability $1/q$. Upon a correct guess, the challenger samples $(\mathbf{s}^*_e, \mathbf{e}^*_e)$ in advance and embeds this value into the random oracle at the guessed query. It then samples $\mathbf{t}^*_c$ uniformly, sets $\mathbf{t}^*_s := \mathbf{t}^*_c - (\mathbf{A}\mathbf{s}^*_e + \mathbf{e}^*_e)$, and uses $\mathsf{cpk}^* = (\mathsf{seed}^*_{\mathsf{pk}}, \mathbf{t}^*_c)$ to compute the challenge ciphertext ct^*. Finally, since $\mathbf{t}^*_c$ is now uniformly random, the MLWE assumption allows us to replace $(\mathbf{u}, v)$ in ct^* with uniformly random values. In the QROM, the guessing step is justified via the measure-and-reprogram technique [DFMS19, DFM20].

Theorem 7.3 (IND-S(E)CPA Security of AFS-PKE-ML in the QROM). *We model the hash function* G *as a quantum-accessible random oracle. For any adversary* $\mathcal{A}$ *against either the* IND-ECPA *or the* IND-SCPA *security of AFS-PKE-ML that makes at most* q_G *(quantum) queries to the random oracle* G*, there exist adversaries* $\mathcal{B}, \mathcal{C}, \mathcal{D}$ *running in time comparable to that of* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}S(E)CPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k,k,q}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{G}'}(\mathcal{D}).$$

The proofs of IND-ECPA and IND-SCPA security for AFS-PKE-ML follow the standard IND-CPA analysis of ML-KEM and are deferred to Appendix C.2.

7.3 AFS-KEM-ML

Based on AFS-PKE-ML, the AFS-KEM-ML is obtained by applying our $\mathsf{FO}_{\mathsf{AFS}}$ transform (shown in Fig. 8). The transform upgrades the CPA security of AFS-PKE-ML to the corresponding CPA/CCA security of AFS-KEM-ML. For AFS-KEM-ML, the properties of perfect ExtractPK-correctness, ephemeral key generation equivalence, min-entropy of the combined public key, and γ-spreadness[19] hold directly. The analyses of the SPK-Bind, RSPK-Bind, and Key-Coll security of AFS-KEM-ML are deferred to Theorems C.2, C.3, and C.4 in Section C.3.

Table 2. Parameter sets for ML-KEM-based AFS-KEM.

	n	q	k	(η_1, η_2)	(d_u, d_v)	(C, Q)	δ	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{pk}\rvert + \lvert\mathsf{ct}\rvert$
AFS-MLKEM-1	256	7681	2	(5,3)	(11,4)	(112,102)	2^{-140}	864	832	1696
AFS-MLKEM-3	256	7681	3	(4,2)	(11,5)	(179,162)	2^{-167}	1280	1216	2496
AFS-MLKEM-5	256	7681	4	(3,2)	(11,5)	(243,220)	2^{-191}	1696	1568	3264
ML-KEM-1	256	3329	2	(3,2)	(10,4)	(117,108)	2^{-71}	800	768	1568
ML-KEM-3	256	3329	3	(2,2)	(10,4)	(183,166)	2^{-84}	1184	1088	2272
ML-KEM-5	256	3329	4	(2,2)	(11,5)	(256,232)	2^{-88}	1568	1568	3136

Parameter Selection. For security preservation, we adapt the ML-KEM parameters to construct AFS-MLKEM-1, AFS-MLKEM-3, and AFS-MLKEM-5, corresponding to the NIST security levels, while ensuring decryption failure probabilities and security levels comparable to those of standard ML-KEM. Table 2 reports the public key size $\lvert\mathsf{pk}\rvert$, ciphertext size $\lvert\mathsf{ct}\rvert$, and the total bandwidth $\lvert\mathsf{pk}\rvert + \lvert\mathsf{ct}\rvert$ (in bytes). Security is evaluated against primal attacks, where "C" and "Q" denote estimated classical and quantum security strengths (in bits), and "δ" represents the error probability computed via a Python script (see Thm. 7.1).

Alternatively, one may opt for the standard ML-KEM parameters to leverage standardized API support and high-performance libraries. In Table 2, ML-KEM-1, ML-KEM-3, and ML-KEM-5 denote the three standard ML-KEM parameter sets corresponding to NIST security categories 1, 3, and 5, respectively. While this significantly facilitates integration into existing cryptographic frameworks, it involves a trade-off in error rates. Nevertheless, we suggest that error rates ranging from 2^{-71} to 2^{-88} remain reasonable for real-world deployments of pMAKE/pUAKE based on AFS-KEM-ML.

[19] γ-spreadness reduces to that of ML-KEM, since the combined public key cpk follows the same structure as a valid ML-KEM public key.

Analysis of Error Rates in Practice. We suggest the error rates from 2^{-71} to 2^{-88} may be reasonable for deploying AFS-KEM-ML based pMAKE/pUAKE in practice, on the following grounds.

Practical considerations. From a functional and implementation perspective, occasional failures at extremely low rates, such as 2^{-71} per session, can be handled via standard session re-establishment or retransmission mechanisms. Moreover, as an engineering reference point, the IEEE P802.3dj project documents linked from the task-force page specify a bit-error-rate objective of at most 10^{-13}, i.e., approximately $2^{-43.19}$, at the MAC/PLS service interface, after the relevant physical-layer mechanisms such as forward-error correction [IEE]. Thus, 2^{-44} can be viewed as a conservative engineering reference level derived from this residual BER objective. We emphasize that this comparison is not a direct identification of the two error notions. The IEEE P802.3dj objective concerns residual transmission errors at a communication interface, whereas the failure probabilities analyzed here concern cryptographic decapsulation failures in ephemeral key-exchange sessions. Nevertheless, both quantities represent rare functional failures that can be handled by retrying the affected communication or re-establishing the affected session.

This viewpoint is also consistent with Frodo-AKE [BCD+16], where decryption failure probabilities ranging from $2^{-41.8}$ to $2^{-33.8}$ are considered acceptable for practical deployment. Therefore, the substantially lower failure probabilities of our protocols, ranging from 2^{-71} to 2^{-88}, provide a comfortable margin for practical use in authenticated key exchange.

Security considerations. Key-recovery attacks against ML-KEM exploiting decryption failures have been extensively studied [DGJ+19, DRV20, BS20, DB22, AWK+25]. In particular, finding one decryption failure with respect to a fixed long-term key significantly facilitates finding additional failures. Moreover, existing decryption-failure attacks (DFA) require the adversary to target the same secret key and collect thousands of ciphertexts that trigger decapsulation failures, thereby progressively narrowing the search space of the target key. This motivates the use of very low error rates in standard ML-KEM instantiations.

This attack pattern applies naturally to standard KEM-based AKE constructions, where an adversary can establish multiple sessions with the same user and use that user as an oracle revealing whether crafted ciphertexts induce failures. Importantly, all such failure-inducing ciphertexts correspond to the same long-term secret key, enabling the adversary to aggregate leakage across sessions.

In contrast, this attack paradigm appears substantially less applicable to our pMAKE and pUAKE. In our setting, the effective decapsulation key is a combined key $\mathsf{csk} = \mathsf{sk}_s + \mathsf{sk}_e$, where the ephemeral component sk_e is freshly sampled in each session and immediately discarded upon completion. Consequently, for any fixed combined key csk_i, with overwhelming probability the adversary can trigger and observe at most one decryption failure. Notably, in such a failing session, the adversary additionally obtains an OTP encryption of seed_e under a uniformly random implicit-rejection key, which leaks no information about seed_e by the perfect secrecy of the OTP. As a result, a single decryption failure leaks at most one inequality, yielding information that is fundamentally insufficient to recover the full high-dimensional key csk. Furthermore, since the fresh ephemeral component sk_e continuously changes the effective target key across sessions, the adversary is structurally prevented from aggregating leakage across multiple failures, which is a prerequisite for existing decryption-failure attacks.

Even under an extremely pessimistic scenario in which a single decryption failure leaks the entire combined key $\mathsf{csk} = \mathsf{sk}_s + \mathsf{sk}_e$ (which should not occur in practice), the security of the underlying static key remains within standard margins, as assessed in [DDGR20]. In this case, the adversary obtains noisy information on sk_s with noise variance σ_e^2. Using the leaky-LWE estimator from [DDGR20], we evaluate this setting for ML-KEM-768 and find that such leakage reduces quantum security by approximately 8.5 bits, which remains within standard post-quantum security margins. Notably, this does not contradict our proof, which does not rely on the Hint-MLWE assumption, as the combined secret key csk is never revealed in the security model.

7.4 Discussion and Future Work.

Our protocols pMAKE and pUAKE can be instantiated from ML-KEM while preserving its native public-key and ciphertext structure. They can therefore reuse most of the software implementations or hardware circuits of ML-KEM, substantially simplifying deployment. This interface-level compatibility

facilitates the integration of post-quantum KEMs into authenticated key-exchange protocols and supports incremental, low-friction deployment using existing cryptographic libraries.

Moreover, the pMAKE and pUAKE protocols can be extended to achieve identity hiding[20] in the common-reference-string (CRS) setting. In particular, assume that all users share the same public matrix $\mathbf{A}$, or equivalently the same public seed $\mathsf{seed}_{\mathbf{A}}$ used to generate $\mathbf{A}$. Then a combined public key such as $\mathsf{cpk}_A = \mathsf{pk}_A + \mathsf{pk}_e^A$ does not reveal the static public key pk_A associated with id_A, since the fresh ephemeral public-key component pk_e^A masks the static component inside the combined key.

The techniques underlying AFS-KEM-ML can be naturally extended to integer lattices, yielding a FrodoKEM-based AFS-KEM instantiation under the LWE assumption [LBES25]. Moreover, the NIST-selected HQC scheme exhibits similar public-key structural properties, making it a natural candidate for the AFS-KEM paradigm. As discussed above, classical Diffie–Hellman–type constructions also admit compatible key structures that may enable natural AFS-KEM instantiations.

We believe that the AFS-KEM framework can also reduce the communication overhead of AKE protocols in the pre-id setting, where the peer identity is fixed and known at session initiation. More generally, exploring the applications of AFS-KEM in practical AKE protocols, such as TLS and IKE, is an interesting direction. A detailed investigation of these instantiations and applications is left for future work.

Finally, our frameworks are currently analyzed primarily in the ROM and QROM. Reducing their reliance on random oracles and establishing security in the standard model remain interesting directions for future work.

Acknowledgments

We would like to thank the anonymous reviewers of CRYPTO 2026 for their insightful comments and suggestions. We thank Boyue Fang for helpful discussions on decryption-failure attacks (DFA). The work of Yunlei Zhao was supported by the Shanghai Collaborative Innovation Fund (No. XTCX-KJ-2023-54) and the Special Fund for Key Technologies in Blockchain of the Shanghai Science and Technology Committee (No. 23511100300). The work of Haodong Jiang was supported by the National Natural Science Foundation of China (No. 62572482).

The authors used large language models solely for basic language assistance, including grammar and spelling checks. The authors take full responsibility for the correctness and integrity of all content.

References

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 269–295. Springer, Cham, August 2019. https://doi.org/10.1007/978-3-030-26951-7_10.

AWK+25. Samy Amer, Yingchen Wang, Hunter Kippen, Thinh Dang, Daniel Genkin, Andrew Kwong, Alexander Nelson, and Arkady Yerukhimovich. PQ-hammer: End-to-end key recovery attacks on post-quantum cryptography using rowhammer. In Marina Blanton, William Enck, and Cristina Nita-Rotaru, editors, *2025 IEEE Symposium on Security and Privacy*, pages 3567–3582. IEEE Computer Society Press, May 2025. https://doi.org/10.1109/SP61157.2025.00048.

BBBV97. Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. https://doi.org/10.1137/S0097539796300933.

BCCT12. Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *ITCS 2012*, pages 326–349. ACM, January 2012. https://doi.org/10.1145/2090236.2090263.

BCD+16. Joppe W. Bos, Craig Costello, Léo 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 Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1006–1018. ACM Press, October 2016. https://doi.org/10.1145/2976749.2978425.

[20] Identity hiding means that the public protocol transcript does not reveal the parties' identities to passive adversaries during the handshake.

BCD+24. Manuel Barbosa, Deirdre Connolly, João Diogo Duarte, Aaron Kaiser, Peter Schwabe, Karoline Varner, and Bas Westerbaan. X-Wing. *CiC*, 1(1):21, 2024. https://doi.org/10.62056/a3qj89n4e.

BCGP08. Colin Boyd, Yvonne Cliff, Juan González Nieto, and Kenneth G. Paterson. Efficient one-round key exchange in the standard model. In Yi Mu, Willy Susilo, and Jennifer Seberry, editors, *ACISP 08*, volume 5107 of *LNCS*, pages 69–83. Springer, Berlin, Heidelberg, July 2008. https://doi.org/10.1007/978-3-540-70500-0_6.

BCL+26. Hugo Beguinet, Céline Chevalier, Guirec Lebrun, Thomas Legavre, Thomas Ricosset, Maxime Roméas, and Éric Sageloli. DAKE: Bandwidth-efficient (u)AKE from double-KEM. In Shi Bai and Edoardo Persichetti, editors, *Public-Key Cryptography – PKC 2026*, volume 16554 of *Lecture Notes in Computer Science*. Springer, Cham, 2026. https://doi.org/10.1007/978-3-032-26740-5_1.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Berlin, Heidelberg, December 2011. https://doi.org/10.1007/978-3-642-25385-0_3.

BDK+18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 353–367. IEEE, 2018. https://doi.org/10.1109/EUROSP.2018.00032.

Bel06. Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 602–619. Springer, Berlin, Heidelberg, August 2006. https://doi.org/10.1007/11818175_36.

BHH+19. Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 61–90. Springer, Cham, December 2019. https://doi.org/10.1007/978-3-030-36033-7_3.

BMS19. Colin Boyd, Anish Mathuria, and Douglas Stebila. *Protocols for Authentication and Key Establishment*. Information Security and Cryptography. Springer, Berlin, Heidelberg, 2 edition, 2019. https://doi.org/10.1007/978-3-662-58146-9.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. https://doi.org/10.1145/168588.168596.

BR94. Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 232–249. Springer, Berlin, Heidelberg, August 1994. https://doi.org/10.1007/3-540-48329-2_21.

BRS02. John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75. Springer, 2002. https://doi.org/10.1007/3-540-36492-7_6.

BS20. Nina Bindel and John M. Schanck. Decryption failure is more likely after success. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 206–225. Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-44223-1_12.

CDM24. Cas Cremers, Alexander Dax, and Niklas Medinger. Keeping up with the KEMs: Stronger security notions for KEMs and automated analysis of KEM-based protocols. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 1046–1060. ACM Press, October 2024. https://doi.org/10.1145/3658644.3670283.

CF12. Cas J. F. Cremers and Michele Feltz. Beyond eCK: Perfect forward secrecy under actor compromise and ephemeral-key reveal. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, *ESORICS 2012*, volume 7459 of *LNCS*, pages 734–751. Springer, Berlin, Heidelberg, September 2012. https://doi.org/10.1007/978-3-642-33167-1_42.

CH98. David Carrel and Dan Harkins. The Internet Key Exchange (IKE). RFC 2409, November 1998. https://doi.org/10.17487/RFC2409.

CK01. Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 453–474. Springer, Berlin, Heidelberg, May 2001. https://doi.org/10.1007/3-540-44987-6_28.

CK02. Ran Canetti and Hugo Krawczyk. Security analysis of IKE's signature-based key-exchange protocol. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 143–161. Springer, Berlin, Heidelberg, August 2002. URL: https://eprint.iacr.org/2002/120/, https://doi.org/10.1007/3-540-45708-9_10.

DB22. Jan-Pieter D'Anvers and Senne Batsleer. Multitarget decryption failure attacks and their application to Saber and Kyber. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 3–33. Springer, Cham, March 2022. https://doi.org/10.1007/978-3-030-97121-2_1.

DDGR20. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 329–358. Springer, Cham, August 2020. https://doi.org/10.1007/978-3-030-56880-1_12.

DFM20. Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 602–631. Springer, Cham, August 2020. https://doi.org/10.1007/978-3-030-56877-1_21.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Cham, August 2019. https://doi.org/10.1007/978-3-030-26951-7_13.

DGJ+19. Jan-Pieter D'Anvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede. Decryption failure attacks on IND-CCA secure lattice-based schemes. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 565–598. Springer, Cham, April 2019. https://doi.org/10.1007/978-3-030-17259-6_19.

DH76. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. https://doi.org/10.1109/TIT.1976.1055638.

DHK+21. Julien Duman, Kathrin Hövelmanns, Eike Kiltz, Vadim Lyubashevsky, and Gregor Seiler. Faster lattice-based KEMs via a generic Fujisaki-Okamoto transform using prefix hashing. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2722–2737. ACM Press, November 2021. https://doi.org/10.1145/3460120.3484819.

DJK+25. Yevgeniy Dodis, Daniel Jost, Shuichi Katsumata, Thomas Prest, and Rolfe Schmidt. Triple ratchet: A bandwidth efficient hybrid-secure signal protocol. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part VIII*, volume 15608 of *LNCS*, pages 302–331. Springer, Cham, May 2025. https://doi.org/10.1007/978-3-031-91101-9_11.

DRV20. Jan-Pieter D'Anvers, Mélissa Rossi, and Fernando Virdia. (One) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 3–33. Springer, Cham, May 2020. https://doi.org/10.1007/978-3-030-45727-3_1.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Berlin, Heidelberg, August 1999. https://doi.org/10.1007/3-540-48405-1_34.

FO13. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. https://doi.org/10.1007/s00145-011-9114-1.

FSXY12. Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 467–484. Springer, Berlin, Heidelberg, May 2012. https://doi.org/10.1007/978-3-642-30057-8_28.

FSXY13. Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism. In Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey Tzeng, editors, *ASIACCS 13*, pages 83–94. ACM Press, May 2013. https://doi.org/10.1145/2484313.2484323.

GHHM21. Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 637–667. Springer, Cham, December 2021. https://doi.org/10.1007/978-3-030-92062-3_22.

GMR89. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. https://doi.org/10.1137/0218012.

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Cham, November 2017. https://doi.org/10.1007/978-3-319-70500-2_12.

HKNW25a. Keitaro Hashimoto, Shuichi Katsumata, Guilhem Niot, and Thom Wiggers. Revisiting PQ WireGuard: A comprehensive security analysis with a new design using reinforced KEMs. Cryptology ePrint Archive, Paper 2025/1758, 2025. To appear at IEEE Symposium on Security and Privacy (S&P) 2026. URL: https://eprint.iacr.org/2025/1758.

HKNW25b. Keitaro Hashimoto, Shuichi Katsumata, Guilhem Niot, and Thom Wiggers. Revisiting PQ WireGuard: A comprehensive security analysis with a new design using reinforced KEMs. Cryptology ePrint Archive, Paper 2025/1758, 2025. URL: https://eprint.iacr.org/2025/1758.

HKSU20. Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 389–422. Springer, Cham, May 2020. https://doi.org/10.1007/978-3-030-45388-6_14.

HNS+21. Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Fiona Johanna Weber, and Philip R. Zimmermann. Post-quantum WireGuard. In *2021 IEEE Symposium on Security and Privacy*, pages 304–321. IEEE Computer Society Press, May 2021. https://doi.org/10.1109/SP40001.2021.00030.

HRS16. Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 387–416. Springer, Berlin, Heidelberg, March 2016. https://doi.org/10.1007/978-3-662-49384-7_15.

IEE. IEEE P802.3dj Task Force. IEEE P802.3dj 200 Gb/s, 400 Gb/s, 800 Gb/s, and 1.6 Tb/s Ethernet Task Force. https://www.ieee802.org/3/dj/index.html. Accessed: 2026-06-25.

JZC+18. Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 96–125. Springer, Cham, August 2018. https://doi.org/10.1007/978-3-319-96878-0_4.

JZM19. Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 227–248. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-25510-7_13.

KHN+14. Charlie Kaufman, Paul E. Hoffman, Yoav Nir, Pasi Eronen, and Tero Kivinen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296, October 2014. https://doi.org/10.17487/RFC7296.

KKPP20. Shuichi Katsumata, Kris Kwiatkowski, Federico Pintore, and Thomas Prest. Scalable ciphertext compression techniques for post-quantum KEMs and their applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 289–320. Springer, Cham, December 2020. https://doi.org/10.1007/978-3-030-64837-4_10.

KLSS23. Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 549–580. Springer, Cham, August 2023. https://doi.org/10.1007/978-3-031-38554-4_18.

Kra03. Hugo Krawczyk. SIGMA: The "SIGn-and-MAc" approach to authenticated Diffie-Hellman and its use in the IKE protocols. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 400–425. Springer, Berlin, Heidelberg, August 2003. https://doi.org/10.1007/978-3-540-45146-4_24.

Kra05. Hugo Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 546–566. Springer, Berlin, Heidelberg, August 2005. https://doi.org/10.1007/11535218_33.

LBES25. Patrick Longa, Joppe W. Bos, Stephan Ehlen, and Douglas Stebila. FrodoKEM: key encapsulation from learning with errors. Internet-Draft draft-longa-cfrg-frodokem-01, Internet Engineering Task Force, September 2025. Work in Progress. URL: https://datatracker.ietf.org/doc/draft-longa-cfrg-frodokem/01/.

LHT16. Adam Langley, Mike Hamburg, and Sean Turner. Elliptic Curves for Security. RFC 7748, January 2016. https://doi.org/10.17487/RFC7748.

LL23. You Lyu and Shengli Liu. Two-message authenticated key exchange from public-key encryption. In Gene Tsudik, Mauro Conti, Kaitai Liang, and Georgios Smaragdakis, editors, *ESORICS 2023, Part I*, volume 14344 of *LNCS*, pages 414–434. Springer, Cham, September 2023. https://doi.org/10.1007/978-3-031-50594-2_21.

LLM07. Brian A. LaMacchia, Kristin E. Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *Provable Security, First International Conference, ProvSec 2007, Wollongong, Australia, November 1-2, 2007, Proceedings*, volume 4784 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2007. https://doi.org/10.1007/978-3-540-75670-5_1.

LPWZ21. Huanhuan Lian, Tianyu Pan, Huige Wang, and Yunlei Zhao. Identity-based identity-concealed authenticated key exchange. In Elisa Bertino, Haya Shulman, and Michael Waidner, editors, *Computer Security – ESORICS 2021: 26th European Symposium on Research in Computer Security, ESORICS 2021, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part II*, volume 12973 of *Lecture Notes in Computer Science*, pages 651–675, Cham, 2021. Springer. https://doi.org/10.1007/978-3-030-88428-4_32.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015. https://doi.org/10.1007/S10623-014-9938-4.

MAB+25. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jurjen Bos, Jean-Christophe Deneuville, Arnaud Dion, Philippe Gaborit, Jérôme Lacan, Edoardo Persichetti, Jean-Marc Robert, Pascal Véron, and Gilles Zémor. HQC: Hamming Quasi-Cyclic, 2025. Selected by NIST for standardization on Mar. 11, 2025. https://pqc-hqc.org/.

Nat17. National Institute of Standards and Technology (NIST). Post-quantum cryptography standardization. https://www.nist.gov/pqcrypto, 2017. Created January 3, 2017; last updated December 11, 2025.

Nat24a. National Institute of Standards and Technology. Module-lattice-based digital signature standard (FIPS 204). Federal Information Processing Standards Publication 204, U.S. Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, USA, August 2024. https://doi.org/10.6028/NIST.FIPS.204.

Nat24b. National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard (FIPS 203). Federal Information Processing Standards Publication 203, U.S. Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, USA, August 2024. https://doi.org/10.6028/NIST.FIPS.203.

Nat24c. National Institute of Standards and Technology. Stateless hash-based digital signature standard (FIPS 205). Federal Information Processing Standards Publication 205, U.S. Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, USA, August 2024. https://doi.org/10.6028/NIST.FIPS.205.

NC00. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2 edition, 2000.

PWZ23. Jiaxin Pan, Benedikt Wagner, and Runzhi Zeng. Lattice-based authenticated key exchange with tight security. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 616–647. Springer, Cham, August 2023. https://doi.org/10.1007/978-3-031-38554-4_20.

PZ23. Jiaxin Pan and Runzhi Zeng. A generic construction of tightly secure password-based authenticated key exchange. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VIII*, volume 14445 of *LNCS*, pages 143–175. Springer, Singapore, December 2023. https://doi.org/10.1007/978-981-99-8742-9_5.

Res18. Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018. https://doi.org/10.17487/RFC8446.

RSA78. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978. https://doi.org/10.1145/359340.359342.

SFG25. Douglas Stebila, Scott Fluhrer, and Shay Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft draft-ietf-tls-hybrid-design-16, Internet Engineering Task Force, September 2025. Work in Progress. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/16/.

Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994. https://doi.org/10.1109/SFCS.1994.365700.

SSW20. Peter Schwabe, Douglas Stebila, and Thom Wiggers. Post-quantum TLS without handshake signatures. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1461–1480. ACM Press, November 2020. https://doi.org/10.1145/3372297.3423350.

WS25. Guilin Wang and Valery Smyslov. KEM-based Authentication for IKEv2 with Post-quantum Security. IETF Internet-Draft, `draft-wang-ipsecme-kem-auth-ikev2-02`, October 2025. Work in Progress, last updated 18 Oct 2025. URL: https://www.ietf.org/archive/id/draft-wang-ipsecme-kem-auth-ikev2-02.html.

XLL+18. Haiyang Xue, Xianhui Lu, Bao Li, Bei Liang, and Jingnan He. Understanding and constructing AKE via double-key key encapsulation mechanism. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 158–189. Springer, Cham, December 2018. https://doi.org/10.1007/978-3-030-03329-3_6.

YZ10. Andrew Chi-Chih Yao and Yunlei Zhao. Deniable internet key exchange. In Jianying Zhou and Moti Yung, editors, *Applied Cryptography and Network Security, 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings*, volume 6123 of *Lecture Notes in Computer Science*, pages 329–348, 2010. https://doi.org/10.1007/978-3-642-13708-2_20.

YZ13. Andrew Chi-Chih Yao and Yunlei Zhao. OAKE: a new family of implicitly authenticated diffie-hellman protocols. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 1113–1128. ACM, 2013. https://doi.org/10.1145/2508859.2516695.

YZ14. Andrew Chi-Chih Yao and Yunlei Zhao. Privacy-preserving authenticated key-exchange over internet. *IEEE Trans. Inf. Forensics Secur.*, 9(1):125–140, 2014. https://doi.org/10.1109/TIFS.2013.2293457.

Zha12. Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Berlin, Heidelberg, August 2012. https://doi.org/10.1007/978-3-642-32009-5_44.

Zha16. Yunlei Zhao. Identity-concealed authenticated encryption and key exchange. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1464–1479. ACM Press, October 2016. https://doi.org/10.1145/2976749.2978350.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Cham, August 2019. https://doi.org/10.1007/978-3-030-26951-7_9.

Zha20. Yunlei Zhao. Identity-based authenticated encryption with identity confidentiality. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve Schneider, editors, *Computer Security – ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part II*, volume 12309 of *Lecture Notes in Computer Science*, pages 633–653, Cham, 2020. Springer. https://doi.org/10.1007/978-3-030-59013-0_31.

A Appendix: Background

A.1 Cryptographic Primitives

Definition A.1 (Public-Key Encryption). *A public-key encryption (PKE) scheme over the message space $\mathcal{M}$ consists of three algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

- *The key-generation algorithm* Gen *takes as input the security parameter λ and outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$*, denoted* $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$*. Usually, we will omit the input of* Gen *for brevity.*
- *The encryption algorithm* Enc *takes as input a public key* pk *and a message $m \in \mathcal{M}$, and outputs a ciphertext* ct*, denoted* $\mathsf{ct} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, m)$.
- *The decryption algorithm* Dec *is deterministic and takes as input a secret key* sk *and a ciphertext* ct*, and outputs either a message $m' \in \mathcal{M}$ or the failure symbol $\perp$, denoted* $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.

A PKE scheme is called deterministic (DPKE) if Enc *is deterministic.*

Correctness. A PKE scheme is said to be δ-correct if

$$\mathbb{E}_{(\mathsf{pk},\mathsf{sk})\leftarrow\!\!\$\mathsf{Gen}}\left[\max_{m\in\mathcal{M}} \Pr\left[\mathsf{Dec}(\mathsf{sk},\mathsf{ct}) \neq m : \mathsf{ct} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, m)\right]\right] \leq \delta,$$

where the expectation is over $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}$. A scheme is perfectly correct if $\delta = 0$.

Definition A.2 (Key Encapsulation Mechanism). *A key encapsulation mechanism (KEM) over $\mathcal{K}$ is a tuple of algorithms* $(\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$*: the key-generation algorithm takes as input the security parameter λ and outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$*. Usually, we will omit the input of* Gen *for brevity.*
- $(\mathsf{ct}, K) \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})$*: the encapsulation algorithm takes as input a public key* pk *and outputs a pair* (ct, K)*, where $K \in \mathcal{K}$ and $\mathsf{ct} \in \mathcal{C}$.*
- $K' \leftarrow \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$*: the decapsulation algorithm takes as input a secret key* sk *and a ciphertext* ct*, and outputs either a key $K' \in \mathcal{K}$ or the rejection symbol $\perp$ (if explicit rejection is allowed).*

Correctness. A KEM scheme is said to be δ-correct if

$$\Pr\left[\,\mathsf{Decaps}(\mathsf{sk},\mathsf{ct}) \neq K \;\middle|\; (\mathsf{pk},\mathsf{sk}) \leftarrow\!\$\ \mathsf{Gen};\ (\mathsf{ct},K) \leftarrow\!\$\ \mathsf{Encaps}(\mathsf{pk})\right] \leq \delta.$$

Min-Entropy of KEM Public Keys. We say that honestly generated KEM public keys have min-entropy at least k if for any fixed $\hat{\mathsf{pk}} \in \mathcal{PK}$ it holds that $\Pr_{(\mathsf{pk},\mathsf{sk})\leftarrow\!\$\mathsf{Gen}(1^\lambda)}[\mathsf{pk} = \hat{\mathsf{pk}}] \leq 2^{-k}$.

Definition A.3 (PRG). *A pseudorandom generator (PRG) is a deterministic polynomial-time algorithm* $\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ *with* $\ell(n) > n$. *For an adversary* $\mathcal{A}$, *its IND-PRG advantage is defined as*

$$\mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG},\mathcal{A}} = \left|\Pr[s \leftarrow\!\$\ \{0,1\}^n : \mathcal{A}(\mathsf{PRG}(s)) \Rightarrow 1] - \Pr\left[r \leftarrow\!\$\ \{0,1\}^{\ell(n)} : \mathcal{A}(r) \Rightarrow 1\right]\right|.$$

Definition A.4 (Dual-PRF). *A pseudorandom function (PRF) is a keyed function* $\mathsf{PRF} : \mathcal{K} \times \mathcal{L} \to \{0,1\}^\lambda$. *For an adversary* $\mathcal{A}$, *its PRF advantage is defined as*

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}} = \left|\Pr\left[\mathsf{k} \leftarrow\!\$\ \mathcal{K} : \mathcal{A}^{\mathsf{PRF}(\mathsf{k},\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathcal{R}(\cdot)} \Rightarrow 1\right]\right|,$$

where $\mathcal{R} : \mathcal{L} \to \{0,1\}^\lambda$ *is a uniformly random function. A PRF satisfies* dual-PRF *security [Bel06] if both* PRF *and the swapped function* $\mathsf{PRF}'(x,y) = \mathsf{PRF}(y,x)$ *are secure PRFs. In our analysis, we assume dual-PRF security for* PRF.

Definition A.5 (Authenticated Forward-Secure PKE). *An authenticated forward-secure public-key encryption scheme (*AFS-PKE*) over message space* $\mathcal{M}$ *is a tuple of algorithms* $\Pi_{\mathsf{AFS\text{-}PKE}} = (\mathsf{SKeyGen}, \mathsf{EKeyGen}, \mathsf{EKeyGen_c}, \mathsf{CombPK}, \mathsf{CombSK}, \mathsf{ExtractPK}, \mathsf{Enc_{AFS}}, \mathsf{Dec_{AFS}})$.

- $(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$*: the long-term key-generation algorithm outputs a static key pair* $(\mathsf{pk}_s, \mathsf{sk}_s) \in \mathcal{PK}_s \times \mathcal{SK}_s$.
- $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}_s, \mathsf{seed})$*: the ephemeral key-generation algorithm takes as input the static public key* pk_s *and a seed* $\mathsf{seed} \in \{0,1\}^\lambda$, *and deterministically outputs an ephemeral key pair* $(\mathsf{pk}_e, \mathsf{sk}_e) \in \mathcal{PK}_e \times \mathcal{SK}_e$.
- $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen_c}(\mathsf{cpk}, \mathsf{seed})$*: the composite-key-based ephemeral key-generation algorithm takes as input the composite public key* cpk *and a seed* $\mathsf{seed} \in \{0,1\}^\lambda$, *and deterministically outputs an ephemeral key pair* $(\mathsf{pk}_e, \mathsf{sk}_e) \in \mathcal{PK}_e \times \mathcal{SK}_e$.
- $\mathsf{cpk} \leftarrow \mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e)$*: the public-key combination algorithm takes as input a static public key* pk_s *and an ephemeral public key* pk_e, *and outputs a composite public key* cpk.
- $\mathsf{csk} \leftarrow \mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e)$*: the secret-key combination algorithm takes as input a static secret key* sk_s *and an ephemeral secret key* sk_e, *and outputs a composite secret key* csk. *For notational convenience, we write* $\mathsf{CombK}(\mathsf{pk}_s, \mathsf{sk}_s, \mathsf{pk}_e, \mathsf{sk}_e) \coloneqq (\mathsf{CombPK}(\mathsf{pk}_s, \mathsf{pk}_e), \mathsf{CombSK}(\mathsf{sk}_s, \mathsf{sk}_e))$.
- $\mathsf{pk}_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}, \mathsf{pk}_e)$*: the public-key extraction algorithm takes as input a composite public key* cpk *and an ephemeral public key* pk_e, *and deterministically outputs the corresponding static public key* pk_s.
- $\mathsf{ct} \leftarrow\!\$\ \mathsf{Enc_{AFS}}(\mathsf{cpk}, m)$*: the encryption algorithm takes as input the composite public key* cpk *and a message* $m \in \mathcal{M}$, *and outputs a ciphertext* $\mathsf{ct} \in \mathcal{CT}$. *When necessary, we make the encryption randomness explicit by writing* $\mathsf{ct} \leftarrow \mathsf{Enc_{AFS}}(\mathsf{cpk}, m; r)$, *where* $r \leftarrow\!\$\ \mathcal{R}$ *for a randomness space* $\mathcal{R}$.
- $m' \leftarrow \mathsf{Dec_{AFS}}(\mathsf{csk}, \mathsf{ct})$*: the decryption algorithm takes as input the composite secret key* csk *and a ciphertext* $\mathsf{ct} \in \mathcal{CT}$, *and outputs either a message* $m' \in \mathcal{M}$ *or the rejection symbol* $\perp$.

A.2 (Quantum-Accessible) Random Oracle Model

We refer the reader to [NC00] for foundational concepts in quantum computation and quantum information. The Random Oracle Model (ROM) [BR93] provides an idealized framework for cryptographic analysis, where a hash function is treated as a publicly accessible random oracle. In this model, an adversary can only obtain the value of a hash function $\mathsf{H}(x)$ by querying the random oracle on x, effectively treating H as a truly random function. In the quantum setting, an adversary may evaluate

this function on superpositions of inputs. Consequently, the Quantum-Accessible Random Oracle Model (QROM) extends the ROM by allowing quantum adversaries to query the random oracle using quantum states [BDF+11]. In the following, we introduce key technical lemmas in the QROM that will be used throughout this paper.

Lemma A.6 (Simulating the Random Oracle [Zha12, Theorem 6.1]). *Let* H *be an oracle drawn uniformly at random from the set of* $2q$*-wise independent functions. Then, any quantum algorithm making at most* q *queries to* H *has zero advantage in distinguishing* H *from a truly random function.*

Lemma A.7 (Hash Function as a PRF in the QROM [BHH+19, Corollary 1]). *Let* $\mathsf{H} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *be a quantum-accessible random oracle. Define the keyed function family* $\mathsf{F} = \{\mathsf{F}_k\}_{k\in\mathcal{K}}$ *by* $\mathsf{F}_k(x) := \mathsf{H}(k, x)$*. Let* $\mathcal{A}$ *be a quantum* $\mathsf{IND\text{-}PRF}$ *adversary that makes* q *quantum queries to* H *and any number of queries to a challenge oracle* $\mathcal{O}_b$*, where* $b \leftarrow\!\$\, \{0,1\}$ *and* $\mathcal{O}_0(\cdot) = \mathsf{H}(k,\cdot)$ *for* $k \leftarrow\!\$\, \mathcal{K}$*, and* $\mathcal{O}_1 \leftarrow\!\$\, \mathcal{Y}^{\mathcal{X}}$ *is a uniformly random function. Then*

$$\left|\Pr[b = b' : b' \leftarrow \mathcal{A}^{\mathcal{O}_b,\mathsf{H}}] - \frac{1}{2}\right| \leq \frac{2q}{\sqrt{|\mathcal{K}|}}.$$

Lemma A.8 (Generic Search Problem [HRS16, Lemma 3]). *Let* $\gamma \in [0,1]$ *and let* Z *be a finite set. Define the function* $N_1 : Z \to \{0,1\}$ *such that for each* $z \in Z$*,* $N_1(z) = 1$ *with probability* $p_z \leq \gamma$*, and* $N_1(z) = 0$ *otherwise. Let* N_2 *be the constant-zero function:* $N_2(z) = 0$ *for all* z*. If a quantum oracle algorithm* $\mathcal{A}$ *makes at most* q *queries to* N_1 *(or* N_2*), then*

$$\left|\Pr[b = 1 : b \leftarrow \mathcal{A}^{N_1}] - \Pr[b = 1 : b \leftarrow \mathcal{A}^{N_2}]\right| \leq 8(q+1)^2\gamma.$$

Lemma A.9 (Single-Point Reprogramming in the QROM, adapted from [GHHM21, Proposition 2]). *Let* X *and* Y *be finite sets. Consider the following two experiments* Repro_0 *and* Repro_1*. First, sample a random oracle* $\mathsf{O}_0 \leftarrow\!\$\, Y^X$ *and set* $\mathsf{O}_1 := \mathsf{O}_0$*. A quantum distinguisher* $\mathcal{D}$ *is given quantum access to* O_b*, where* $b \in \{0,1\}$*, and may issue one adaptive reprogramming oracle* $\mathsf{Reprogram}$*. In* $\mathsf{Reprogram}$*, the experiment samples* $x^* \leftarrow\!\$\, X$ *and* $y^* \leftarrow\!\$\, Y$*, sets* $\mathsf{O}_1(x^*) := y^*$*, and returns* x^* *to* $\mathcal{D}$*. If* $\mathcal{D}$ *makes at most* q *quantum queries to the random oracle, then*

$$\left|\Pr[\mathsf{Repro}_1^{\mathcal{D}} \Rightarrow 1] - \Pr[\mathsf{Repro}_0^{\mathcal{D}} \Rightarrow 1]\right| \leq \frac{3}{2}\sqrt{\frac{q}{|X|}}.$$

Ambainis et al. [AHU19] propose a variant of OW2H called the semi-classical oracle $\mathcal{O}_S^{SC}$, which performs a semi-classical measurement only on the output $|f_S(x)\rangle$ but not on the input $|x\rangle$. Here, f_S is the indicator function such that $f_S(x) = 1$ if $x \in S$, and 0 otherwise.

Semi-classical oracle. Roughly speaking, the semi-classical oracle $\mathcal{O}_S^{SC}$ measures only the output $|f_S(x)\rangle$ but not the input $|x\rangle$, where f_S is the indicator function such that $f_S(x) = 1$ if $x \in S$ and 0 otherwise. Formally, for a query to $\mathcal{O}_S^{SC}$ with $\sum_{x,z} a_{x,z}|x\rangle|z\rangle$, $\mathcal{O}_S^{SC}$ performs the following steps:

1. Initialize a single qubit L with $|0\rangle$,
2. Transform $\sum_{x,z} a_{x,z}|x\rangle|z\rangle|0\rangle$ into $\sum_{x,z} a_{x,z}|x\rangle|z\rangle|f_S(x)\rangle$,
3. Measure the qubit L.

After performing this semi-classical measurement, the query state becomes $\sum_{x,z:f_S(x)=y} a_{x,z}|x\rangle|z\rangle$ (non-normalized), where the measurement output is $y \in \{0,1\}$.

Lemma A.10 (Semi-classical OW2H [AHU19, Theorem 1]). *Let* $S \subseteq X$ *be random. Let* $\mathcal{O}_1$*,* $\mathcal{O}_2$ *be oracles with domain* X *and codomain* Y *such that* $\mathcal{O}_1(x) = \mathcal{O}_2(x)$ *for any* $x \notin S$*. Let* z *be a random bitstring. (The oracles* $\mathcal{O}_1$*,* $\mathcal{O}_2$*,* S*, and* z *may have an arbitrary joint distribution* D*.) Let* $\mathcal{O}_S^{SC}$ *be an oracle that performs the semi-classical measurements corresponding to the projectors* M_y*, where* $M_y := \sum_{x\in X:f_S(x)=y} |x\rangle\langle x|$ *(*$y \in \{0,1\}$*). Let* $\mathcal{O}_2 \setminus S$ *("*$\mathcal{O}_2$ *punctured on* S*") be an oracle that first queries* $\mathcal{O}_S^{SC}$ *and then* $\mathcal{O}_2$*. Let* $A^{\mathcal{O}_1}(z)$ *be an oracle algorithm with query depth* q*. Let* Find *denote the event that during the execution of* $A^{\mathcal{O}_2\setminus S}(z)$*,* $\mathcal{O}_S^{SC}$ *outputs 1 during the semi-classical measurements. Then:*

$$\begin{aligned} P_{left} &:= \Pr[b = 1 : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, b \leftarrow A^{\mathcal{O}_1}(z)] \\ P_{right} &:= \Pr[b = 1 : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, b \leftarrow A^{\mathcal{O}_2}(z)] \\ P_{find} &:= \Pr[\mathsf{Find} : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, A^{\mathcal{O}_2\setminus S}(z)] \end{aligned}$$

Then: $|P_{left} - P_{right}| \leq 2\sqrt{(q+1)P_{find}}$, *and* $|\sqrt{P_{left}} - \sqrt{P_{right}}| \leq 2\sqrt{(q+1)P_{find}}$. *The lemma also holds with the bound* $\sqrt{(q+1)P_{find}}$ *for an alternative definition of* $P_{right} = \Pr[b = 1 \wedge \neg \mathsf{Find} : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, b \leftarrow A^{\mathcal{O}_2 \setminus S}(z)]$.

Lemma A.11 (Search in semi-classical oracle [AHU19, Corollary 1]). *Suppose that* S *and* z *are independent, and that* A *is a* q*-query algorithm. Let* $P_{max} := \max_{x \in X} \Pr[x \in S]$. *Then* $\Pr[\mathsf{Find} : A^{O_S^{SC}}(z)] \leq 4q \cdot P_{max}$.

Lemma A.12 ((Single-input) Measure-and-Reprogram, adapted from [DFM20, Theorem 2]). *Let* $\mathcal{X}$ *and* $\mathcal{Y}$ *be finite non-empty sets. Let* A *be an arbitrary oracle quantum algorithm that makes at most* q *queries to a uniformly random* $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$ *and outputs some* $x \in \mathcal{X}$ *together with a (possibly quantum) output* z. *Let* $S^A(\Theta)$ *be an oracle algorithm that, on input a uniformly random* $\Theta \in \mathcal{Y}$, *randomly samples a pair* $(i, b) \in ([q] \times \{0,1\}) \cup \{(q, 0)\}$, *runs* $A^{|\mathsf{H}_i\rangle}$ *to obtain a (possibly quantum) output* z. *The oracle* H_i *is specified as follows:* S *measures* A*'s* $(i+1)$*-th query input to obtain* x, *answers* A*'s* ℓ*-th* H*-query using* H *if* $\ell < i + 1 + b$, *and answers it using* $\mathsf{H}_{x\Theta}$ *(*$\mathsf{H}_{x\Theta}(x) = \Theta$ *and* $\mathsf{H}_{x\Theta}(x') = \mathsf{H}(x')$ *for all* $x' \neq x$*) for* $\ell \geq i + 1 + b$. *Finally,* $S^A(\Theta)$ *outputs* (x, z). *Then for any fixed* $x_0 \in \mathcal{X}$ *and any (possibly quantum) predicate* V, *it holds that*

$$\Pr_{\Theta}\Big[x = x_0 \wedge V(x, \Theta, z) \ : \ (x, z) \leftarrow S^A(\Theta)\Big] \geq \frac{1}{(2q+1)^2} \Pr_{\mathsf{H}}\Big[x = x_0 \wedge V\big(x, \mathsf{H}(x), z\big) \ : \ (x, z) \leftarrow A^{\mathsf{H}}\Big].$$

B Appendix: Security Analysis for $\mathsf{FO}_{\mathsf{AFS}}$ Transform

Theorem 6.2 ($\mathsf{FO}_{\mathsf{AFS}}$ transform in the ROM and QROM). *Let* $\Pi_{\mathsf{AFS\text{-}KEM}}$ *denote the* AFS-KEM *scheme obtained by applying the* $\mathsf{FO}_{\mathsf{AFS}}$ *transformation of Fig. 8 to an underlying* δ*-correct* AFS-PKE *scheme* $\Pi_{\mathsf{AFS\text{-}PKE}}$ *over message space* $\mathcal{M}$. *Let* H, G, *and* $\mathsf{H}_{\mathsf{rej}}$ *be modeled as random oracles. Assume that the composite public keys generated in* $\Pi_{\mathsf{AFS\text{-}PKE}}$ *have min-entropy at least* k. *For any adversary* $\mathcal{A}$ *against the* IND-LCK-SCCA *security of* $\Pi_{\mathsf{AFS\text{-}KEM}}$ *(resp. its* IND-SCCA *or* IND-ECPA *security) that makes at most* q_{D} *queries to* $\mathcal{O}^{\mathsf{Dec}_s}$, *at most* q_{E} *queries to* $\mathcal{O}^{\mathsf{EKeyGen}}$, *and at most* q_{H}, q_{G}, *and* $q_{\mathsf{H}_{\mathsf{rej}}}$ *queries to* H, G, *and* $\mathsf{H}_{\mathsf{rej}}$, *respectively, there exists an adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) &\leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \alpha,\\
\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) &\leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \alpha,\\
\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) &\leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \frac{2\,(q_{\mathsf{H}} + q_{\mathsf{G}})}{|\mathcal{M}|}.
\end{aligned}$$

If H, G, *and* $\mathsf{H}_{\mathsf{rej}}$ *are modeled as quantum-accessible random oracles and* $\mathcal{A}$ *makes at most* $q_{\mathsf{H}}, q_{\mathsf{G}}, q_{\mathsf{H}_{\mathsf{rej}}}$ *(quantum) queries to* H, G, *and* $\mathsf{H}_{\mathsf{rej}}$, *respectively, then there exists an adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) &\leq 2\sqrt{\hat{q}\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + 4\,(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}})/|\mathcal{M}|\Big)} + \beta,\\
\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) &\leq 2\sqrt{\hat{q}\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + 4\,(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}})/|\mathcal{M}|\Big)} + \beta,\\
\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) &\leq 2\sqrt{\hat{q}_{\mathsf{ECPA}}\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + 4\,(q_{\mathsf{G}} + q_{\mathsf{H}})/|\mathcal{M}|\Big)}.
\end{aligned}$$

Here $\hat{q} = q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}} + 1$, $\hat{q}_{\mathsf{ECPA}} = q_{\mathsf{H}} + q_{\mathsf{G}} + 1$, $\alpha = 2\,(q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}})\,|\mathcal{M}|^{-1} + q_{\mathsf{H}_{\mathsf{rej}}}\,2^{-\lambda} + (q_{\mathsf{E}} + 1)^2\,2^{-k} + (q_{\mathsf{D}} + q_{\mathsf{G}} + q_{\mathsf{H}})\,\delta$, *and* $\beta = (q_{\mathsf{E}} + 1)^2 2^{-k} + 2q_{\mathsf{H}_{\mathsf{rej}}} 2^{-\lambda/2} + 16(q_{\mathsf{E}} + 1)\delta\,(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}} + 1)^2$.

B.1 Security Analysis in the Random Oracle Model.

Security Analysis for IND-LCK-SCCA in the ROM.

IND-LCK-SCCA Game G_0-G_3

$(\mathsf{pk}_s^*, \mathsf{sk}_s^*) \leftarrow\!\$\ \mathsf{SKeyGen}'(1^\lambda)$
$s^* \leftarrow\!\$\ \{0,1\}^\lambda,\ \mathsf{L}_e := \emptyset,\ b \leftarrow\!\$\ \{0,1\}$
$\mathsf{Oracles} = \{\mathcal{O}^{\mathsf{EKeyGen}}, \mathcal{O}^{\mathsf{Dec}_s}, \mathsf{H}, \mathsf{H}_{\mathsf{rej}}, \mathsf{G}\}$
$\mathsf{cpk}^* \leftarrow \mathcal{A}^{\mathsf{Oracles}}(\mathsf{pk}_s^*)$
$m^* \leftarrow\!\$\ \mathcal{M}$
$\mathsf{K}_0^* \leftarrow \mathsf{H}(\mathsf{cpk}^*, m^*)$
$r^* \leftarrow \mathsf{G}(\mathsf{cpk}^*, m^*)$
$\mathsf{ct}^* \leftarrow \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, m^*; r^*)$
$\mathsf{K}_1^* \leftarrow\!\$\ \mathcal{K}$
$(b', \mathsf{seed}'_e) \leftarrow \mathcal{A}^{\mathsf{Oracles}}(\mathsf{pk}_s^*, \mathsf{ct}^*, \mathsf{K}_b^*)$
$(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}'_{\mathsf{c}}(\mathsf{cpk}^*, \mathsf{seed}'_e)$
$\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}'(\mathsf{cpk}^*, \mathsf{pk}'_e)$
if $\mathsf{pk}'_s \neq \mathsf{pk}_s^*$: **abort**
return $b' ? = b$

$\mathcal{O}^{\mathsf{EKeyGen}}()$

$\mathsf{seed} \leftarrow\!\$\ \{0,1\}^\lambda$
$(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{EKeyGen}'(\mathsf{pk}_s^*, \mathsf{seed})$
$\mathsf{cpk} \leftarrow \mathsf{CombPK}'(\mathsf{pk}_s^*, \mathsf{pk}_e)$
if $\exists\, (\mathsf{cpk}, \cdot, \cdot, \cdot) \in \mathsf{L}_e$: // $//G_1$
  $\mathsf{COLL} \leftarrow$ **true** $//G_1$
  abort// $//G_1$
$\mathsf{L}_e \leftarrow \mathsf{L}_e \cup \{(\mathsf{cpk}, \mathsf{pk}_e, \mathsf{sk}_e, \mathsf{seed})\}$
return $(\mathsf{cpk}, \mathsf{pk}_e, \mathsf{sk}_e, \mathsf{seed})$

$\mathsf{H}_{\mathsf{rej}}, \mathsf{G}$

Random oracles

$\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct})$

if $\mathsf{ct} = \mathsf{ct}^* \wedge \mathsf{cpk} = \mathsf{cpk}^*$: **return** $\perp$
$V := \{(\mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}') \mid (\mathsf{cpk}, \mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}') \in \mathsf{L}_e\}$
if $|V| \neq 1$: **return** $\perp$
$(\mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}') \leftarrow V$
$\tilde{K} \leftarrow \mathsf{H}_{\mathsf{rej}}(\mathsf{cpk}, s^*, \mathsf{ct})$ $//G_0, G_1$
$\tilde{K} \leftarrow \overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}, \mathsf{ct})$ $//G_2, G_3$
return $\tilde{K}$ $//G_3$
$\mathsf{csk} \leftarrow \mathsf{CombSK}'(\mathsf{sk}_s^*, \mathsf{sk}'_e)$
$m' \leftarrow \mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}, \mathsf{ct})$
$K \leftarrow \mathsf{H}(\mathsf{cpk}, m')$
$r \leftarrow \mathsf{G}(\mathsf{cpk}, m')$
if $m' = \perp$ **or** $\mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m'; r) \neq \mathsf{ct}$:
  return $\tilde{K}$
return K

$\mathsf{H}(\mathsf{cpk}, m)$

if $\exists\, K : (\mathsf{cpk}, m, K) \in \mathsf{L}_{\mathsf{H}}$: **return** K
$K \leftarrow\!\$\ \mathcal{K}$
if $\exists\, \mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}' : (\mathsf{cpk}, \mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}') \in \mathsf{L}_e$: $//G_3$
  $\mathsf{ct}' \leftarrow \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m; \mathsf{G}(\mathsf{cpk}, m))$ $//G_3$
  $K \leftarrow \overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}, \mathsf{ct}')$ $//G_3$
$\mathsf{L}_{\mathsf{H}} \leftarrow \mathsf{L}_{\mathsf{H}} \cup \{(\mathsf{cpk}, m, K)\}$
return K

Fig. 10. Games G_0-G_3 for the proof of IND-LCK-SCCA security in the ROM

Proof. We proceed with a sequence of games, which are given in detail in Fig. 10.

Game G_0. Game G_0 is the original IND-LCK-SCCA security game. Hence,

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\Pi_{\mathsf{KEM}}}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|.$$

Game G_1. In this game, we abort if a collision occurs among the combined public keys cpk in L_e. Conditioned on no abort, each cpk corresponds to at most one entry in the list L_e, and hence we have $|V| \leq 1$ in $\mathcal{O}^{\mathsf{Dec}_s}$. By the k-min-entropy of the combined public key cpk, the collision probability is at most $q_E^2/2^k$. Games G_0 and G_1 are identical until such a collision occurs, therefore,

$$\left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]\right| \leq \frac{q_E^2}{2^k}.$$

Game G_2. We replace $\mathsf{H}_{\mathsf{rej}}$ inside $\mathcal{O}^{\mathsf{Dec}_s}$ by a new internal random oracle $\overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}, \mathsf{ct})$ that does not take the secret s^* as input. This modification is indistinguishable to $\mathcal{A}$ unless $\mathcal{A}$ queries the random oracle $\mathsf{H}_{\mathsf{rej}}$ on an input containing s^*. Since $s^* \leftarrow\!\$\ \{0,1\}^\lambda$ is hidden from $\mathcal{A}$, we obtain

$$\left|\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]\right| \leq \frac{q_{\mathsf{H}_{\mathsf{rej}}}}{2^\lambda}.$$

Game G_3. In this game, we simulate the decapsulation oracle $\mathcal{O}^{\mathsf{Dec}_s}$ without using the secret key sk_s^*. Specifically, we only rely on $\overline{\mathsf{H}}_{\mathsf{rej}}$ in $\mathcal{O}^{\mathsf{Dec}_s}$ and patch the random oracle H as shown in Fig. 10.

Note that if AFS-PKE were perfectly correct, then the random oracle patching would also be perfectly correct, and therefore the two games would look identical in $\mathcal{A}$'s view.

We define the event CORR that $\mathcal{A}$ queries $\mathsf{G}(\mathsf{cpk}, m)$ for some tuple $(\mathsf{cpk}, \mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}') \in \mathsf{L}_e$ such that a correctness error is induced. That is, we have $m \neq \mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}, \mathsf{ct})$. where $\mathsf{ct} = \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m; \mathsf{G}(\mathsf{cpk}, m))$ and $\mathsf{csk} \leftarrow \mathsf{CombSK}'(\mathsf{sk}^*_s, \mathsf{sk}'_e)$, Since there are at most $(q_D + q_G + q_H)$ explicit and implicit queries to G, and AFS-PKE is δ-correct, we have $\Pr[\mathsf{CORR}] \leq (q_D + q_G + q_H)\delta$.

Next, we show that when the CORR event does not occur, G_2 and G_3 are identical. That is $\left|\Pr[G^{\mathcal{A}}_2 \Rightarrow 1] - \Pr[G^{\mathcal{A}}_3 \Rightarrow 1]\right| \leq \Pr[\mathsf{CORR}]$. Consider a query $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct})$ such that there exist $\mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}'$ with $(\mathsf{cpk}, \mathsf{pk}'_e, \mathsf{sk}'_e, \mathsf{seed}') \in \mathsf{L}_e$. Notably, if such a tuple does not exist, then $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct})$ return $\perp$ in both G_2 and G_3. Moreover, let $\mathsf{csk} \leftarrow \mathsf{CombSK}'(\mathsf{sk}^*_s, \mathsf{sk}'_e)$ and define $m' := \mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}, \mathsf{ct})$ and $\mathsf{ct}' = \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m'; \mathsf{G}(\mathsf{cpk}, m'))$. Consider the following cases.

- Case 1: $m' = \perp$. H is never called on $m' = \perp$, and hence $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct}) = \overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}, \mathsf{ct})$ has the same distribution in both G_2 and G_3.
- Case 2: $m' \neq \perp \wedge \mathsf{ct} \neq \mathsf{ct}'$. Both games return $\overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}, \mathsf{ct})$. The only way for $\mathcal{A}$ to detect a difference between the two games is to make a query $\mathsf{H}(\mathsf{cpk}, m)$ such that $\mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}, m; \mathsf{G}(\mathsf{cpk}, m)) = \mathsf{ct}$. Notably, in G_3 $\mathsf{H}(\mathsf{cpk}, m)$ would return the same key as in $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct})$, whereas in G_2 the two keys would be independent. Since $\mathsf{ct} \neq \mathsf{ct}'$, we also have $m \neq m'$. Therefore, such a query implies that the CORR event occurs.
- Case 3: $m' \neq \perp \ \wedge \ \mathsf{ct} = \mathsf{ct}'$. In Game G_2 we have $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct}) = \mathsf{H}(\mathsf{cpk}, m')$. In Game G_3 we have $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct}) = \mathsf{H}(\mathsf{cpk}, m)$ for all m that deterministically encrypt to the same ct. The only way for $\mathcal{A}$ to detect a difference between the two games is to query H on some value $m \neq m'$ that also deterministically encrypts to the same ct. But this also implies that the CORR event occurs.

Combining all the analyses we have,

$$\left|\Pr[G^{\mathcal{A}}_2 \Rightarrow 1] - \Pr[G^{\mathcal{A}}_3 \Rightarrow 1]\right| \leq (q_{\mathsf{D}} + q_{\mathsf{G}} + q_{\mathsf{H}})\delta.$$

Game G_4. We replace K^*_0 and r^* with uniformly random values and define the event QUERY that $\mathcal{A}$ queries either $\mathsf{H}(\mathsf{cpk}^*, m^*)$ or $\mathsf{G}(\mathsf{cpk}^*, m^*)$. If QUERY does not occur, then Games G_3 and G_4 are identical. Hence,

$$\left|\Pr[G^{\mathcal{A}}_3 \Rightarrow 1] - \Pr[G^{\mathcal{A}}_4 \Rightarrow 1]\right| \leq \Pr[\mathsf{QUERY} : G_4].$$

In Game G_4, the bit b is independent of the adversary's view, and thus

$$\Pr[G^{\mathcal{A}}_4 \Rightarrow 1] = 1/2.$$

To bound $\Pr[\mathsf{QUERY} : G_4]$, we construct an adversary $\mathcal{B}$ against the IND-LCK-SCPA security of $\Pi_{\mathsf{AFS\text{-}PKE}}$. Adversary $\mathcal{B}$ receives pk^*_s from its challenger and forwards it to $\mathcal{A}$. When $\mathcal{A}$ outputs cpk^*, adversary $\mathcal{B}$ submits it together with two random messages $m^*_0, m^*_1 \leftarrow\!\!\$\ \mathcal{M}$ to its challenger and receives a challenge ciphertext ct^*_b. Then adversary $\mathcal{B}$ perfectly simulates Game G_4 for $\mathcal{A}$ by embedding $\mathsf{ct}^* = \mathsf{ct}^*_b$. When $\mathcal{B}$ receives seed'_e from $\mathcal{A}$, $\mathcal{B}$ records that value for later output.

If $\mathcal{A}$ queries $\mathsf{H}(\mathsf{cpk}^*, m^*_{b'})$ or $\mathsf{G}(\mathsf{cpk}^*, m^*_{b'})$, adversary $\mathcal{B}$ eventually outputs (b', seed'_e). Otherwise, $\mathcal{B}$ outputs a random bit and seed'_e. Notably, if Game G_4 does not abort, then seed'_e is a valid output for $\mathcal{B}$. Since m^*_{1-b} is information-theoretically hidden, the probability that $\mathcal{A}$ queries m^*_{1-b} is at most $(q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}})/|\mathcal{M}|$. Assume that this is not the case. We have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \frac{q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}}}{|\mathcal{M}|} &\geq \left|\Pr[b = b'] - \frac{1}{2}\right| \\
&= \left|\Pr[\mathsf{QUERY} : G_4] + \frac{1}{2}\Pr[\neg\,\mathsf{QUERY} : G_4] - \frac{1}{2}\right| \\
&= \frac{1}{2}\Pr[\mathsf{QUERY} : G_4].
\end{aligned}$$

Therefore, $\Pr[\mathsf{QUERY} : G_4] \leq 2\left(\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \frac{q_{\mathsf{H}}+q_{\mathsf{G}}+q_{\mathsf{D}}}{|\mathcal{M}|}\right)$.

Combining all the bounds completes the proof. □

Security Analysis for IND-SCCA in the ROM.

Proof. Since the proof closely follows the analysis of IND-LCK-SCCA in the ROM, we only highlight the differences and omit repeated arguments.

Game G_0 is the original IND-SCCA security game, i.e., $\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|$. Games G_1–G_4 follow the same sequence of transformations as in the IND-LCK-SCCA proof in the ROM. Specifically, in G_1 we abort if a collision for cpk occurs in L_e. Notably, we may have $q_E + 1$ elements in L_e. In Games G_2 and G_3, we transform the experiment to simulate the oracle $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct})$ without knowledge of sk_s^*. Finally, in Game G_4, we replace $\mathsf{K}_0^* = \mathsf{H}(\mathsf{cpk}^*, m^*)$ and $r^* = \mathsf{G}(\mathsf{cpk}^*, m^*)$ with uniformly random values $\mathsf{K}_0^* \leftarrow\!\!\$\ \mathcal{K}$ and $r^* \leftarrow\!\!\$\ \mathcal{R}$. Recall that the event QUERY denotes that the adversary queries $\mathsf{H}(\mathsf{cpk}^*, m^*)$ or $\mathsf{G}(\mathsf{cpk}^*, m^*)$.

It remains to bound $\Pr[\mathsf{QUERY} : G_4]$. We construct a reduction $\mathcal{B}$ against the IND-SCPA security of AFS-PKE as follows. Adversary $\mathcal{B}$ receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{seed}_e^*)$ from its challenger, chooses random messages $m_0^*, m_1^* \leftarrow\!\!\$\ \mathcal{M}$, and submits them to obtain a challenge ciphertext ct_b^*. It then forwards $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{seed}_e^*, \mathsf{ct}_b^*, K^*)$ to $\mathcal{A}$ and perfectly simulate the game as G_4, where K^* is sampled uniformly at random. If $\mathcal{A}$ queries $\mathsf{H}(\mathsf{cpk}^*, m_{b'}^*)$ or $\mathsf{G}(\mathsf{cpk}^*, m_{b'}^*)$ in the game, $\mathcal{B}$ outputs b' and aborts. Otherwise it outputs a random bit. Since m_{1-b}^* is information-theoretically hidden, the probability that it is queried is at most $(q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}})/|\mathcal{M}|$. Assume that this is not the case. Similar to the analysis in IND-LCK-SCCA, we have

$$\Pr[\mathsf{QUERY} : G_4] \leq 2\left(\mathsf{Adv}^{\mathsf{IND\text{-}SCPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \frac{q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}}}{|\mathcal{M}|}\right).$$

This completes the proof. □

Security Analysis for IND-ECPA in the ROM.

Proof. Game G_0 is the original IND-ECPA security game, that is,

$$\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{KEM}}}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|.$$

Game G_1. In Game G_1, we replace $\mathsf{K}_0^* = \mathsf{H}(\mathsf{cpk}^*, m^*)$ and $r^* = \mathsf{G}(\mathsf{cpk}^*, m^*)$ with uniformly random values $\mathsf{K}_0^* \leftarrow\!\!\$\ \mathcal{K}$ and $r^* \leftarrow\!\!\$\ \mathcal{R}$. Notably, r^* is the randomness used to compute the challenge ciphertext ct^*. Let QUERY denote the event that the adversary queries $\mathsf{H}(\mathsf{cpk}^*, m^*)$ or $\mathsf{G}(\mathsf{cpk}^*, m^*)$. When QUERY does not occur, Games G_0 and G_1 are identical. Therefore,

$$\left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\mathsf{QUERY} : G_1].$$

In Game G_1, the bit b is independent of the adversary's view, and thus

$$\Pr[G_1^{\mathcal{A}} \Rightarrow 1] = 1/2.$$

It remains to bound $\Pr[\mathsf{QUERY} : G_1]$. We construct a reduction $\mathcal{B}$ against the IND-ECPA security of AFS-PKE as follows. Adversary $\mathcal{B}$ receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*)$ from its challenger, chooses random messages $m_0^*, m_1^* \leftarrow\!\!\$\ \mathcal{M}$, and submits them to obtain a challenge ciphertext ct_b^*. It then forwards $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*, \mathsf{ct}_b^*, K^*)$ to $\mathcal{A}$ and perfectly simulates Game G_1, where K^* is sampled uniformly at random. If $\mathcal{A}$ queries $\mathsf{H}(\mathsf{cpk}^*, m_{b'}^*)$ or $\mathsf{G}(\mathsf{cpk}^*, m_{b'}^*)$, $\mathcal{B}$ outputs b' and aborts; otherwise it outputs a random bit. Since m_{1-b}^* is information-theoretically hidden, the probability that it is queried is at most $(q_{\mathsf{H}} + q_{\mathsf{G}})/|\mathcal{M}|$. Assume that this is not the case. We have

$$\Pr[\mathsf{QUERY} : G_1] \leq 2\left(\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{AFS\text{-}PKE}}}(\mathcal{B}) + \frac{q_{\mathsf{H}} + q_{\mathsf{G}}}{|\mathcal{M}|}\right).$$

This completes the proof. □

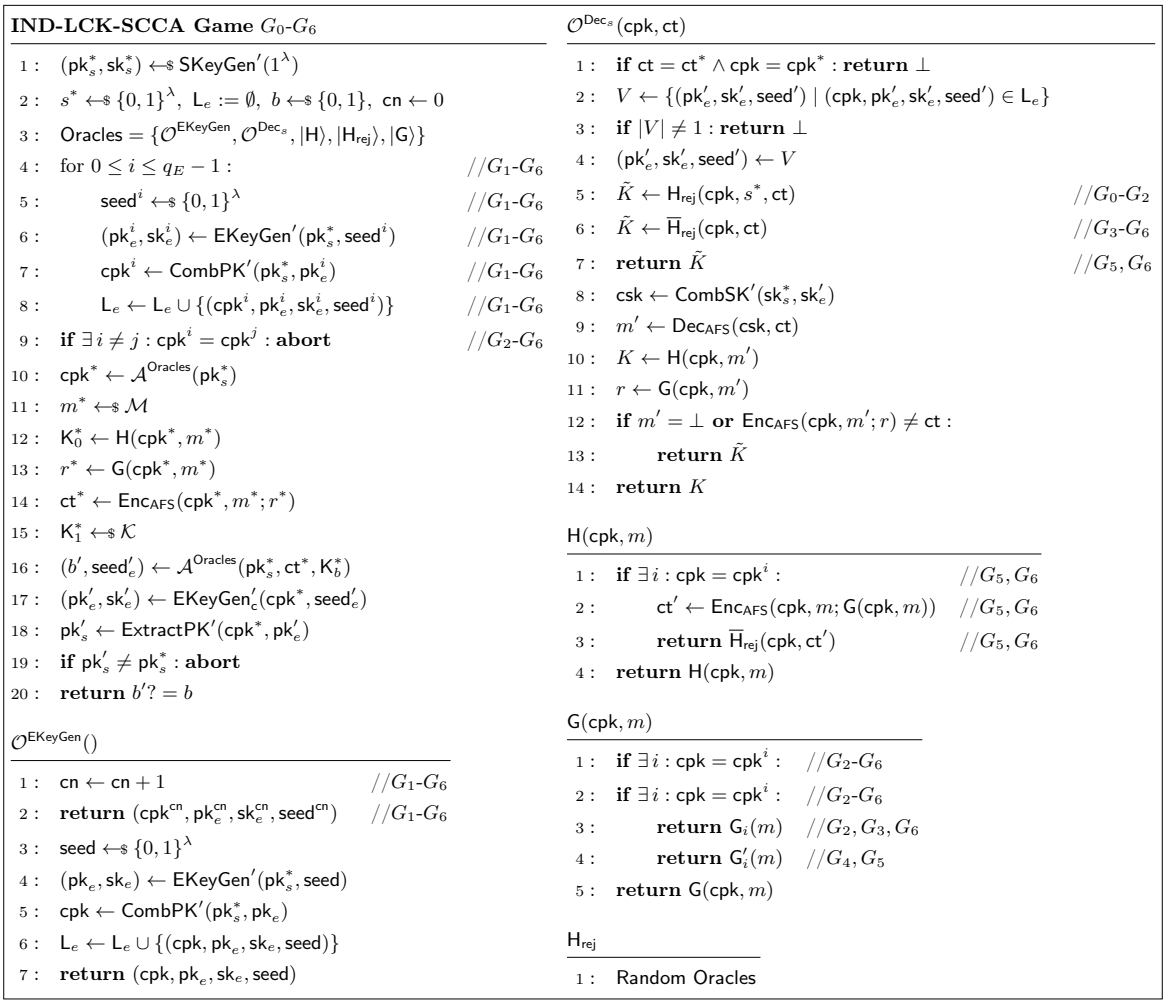


Fig. 11. Games G_0-G_6 for the proof of IND-LCK-SCCA security in the QROM

B.2 Security Analysis in the Quantum-Accessible Random Oracle Model.

Security Analysis for IND-LCK-SCCA in the QROM.

Proof. The detailed descriptions of the games are given below, and an overview of Games G_0 to G_6 is provided in Figure 11.

Game G_0. Game G_0 is the original IND-LCK-SCCA security game. Hence,

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\Pi_{\mathsf{KEM}}}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|.$$

Game G_1. In this game, we precompute all tuples $(\mathsf{cpk}, \mathsf{pk}_e, \mathsf{sk}_e, \mathsf{seed})$ that will be stored in L_e. Recall that there are at most q_E queries to $\mathcal{O}^{\mathsf{EKeyGen}}()$, we denote the resulting tuples by $(\mathsf{cpk}^i, \mathsf{pk}_e^i, \mathsf{sk}_e^i, \mathsf{seed}^i)$ for $0 \le i \le q_E - 1$. Moreover, we define $\mathsf{csk}^i = \mathsf{CombSK}(\mathsf{sk}_s^*, \mathsf{sk}_e^i)$. This change is purely conceptual and does not affect the adversary's view. That is,

$$\Pr[G_0^{\mathcal{A}} \Rightarrow 1] = \Pr[G_1^{\mathcal{A}} \Rightarrow 1].$$

Game G_2. In this game, we abort if a collision occurs among the precomputed combined public keys, that is, if there exist indices $i \ne j$ such that $\mathsf{cpk}^i = \mathsf{cpk}^j$. Conditioned on no abort, each cpk corresponds to at most one entry in L_e, and hence we have $|V| \le 1$ in $\mathcal{O}^{\mathsf{Dec}_s}$. By the k-min-entropy of the combined public key, the collision probability is at most $q_E^2/2^k$. Moreover, we use an internal

random oracle $\mathsf{G}_i(m)$ to simulate $\mathsf{G}(\mathsf{cpk}^i, m)$. Games G_1 and G_2 are identical until such a collision occurs; therefore,

$$\left|\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]\right| \leq \frac{q_E^2}{2^k}.$$

Game G_3. In this game, we replace $\mathsf{H}_{\mathsf{rej}}$ inside $\mathcal{O}^{\mathsf{Dec}_s}$ by a new internal random oracle $\overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}, \mathsf{ct})$ that does not take the secret s^* as input. Since $s^* \leftarrow\!\!\$\ \{0,1\}^\lambda$ is hidden from $\mathcal{A}$, Lemma A.7 implies

$$\left|\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \Pr[G_3^{\mathcal{A}} \Rightarrow 1]\right| \leq \frac{2q_{\mathsf{H}_{\mathsf{rej}}}}{\sqrt{2^\lambda}}.$$

Game G_4. In this game, for $0 \leq i \leq q_E - 1$, we replace the internal random oracle G_i by an oracle G'_i which samples only "good" randomness for any combined public key cpk^i, meaning that no decryption failures can occur when using $\mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}^i, \cdot)$ on proper ciphertexts of the form $\mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^i, \cdot)$. Specifically, for each $0 \leq i \leq q_E - 1$ and $m \in \mathcal{M}$, we define $\mathcal{R}^i_{\mathsf{BAD}}(m) = \{r \in \mathcal{R} : \mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}^i, \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^i, m)) \neq m\}$. In G_4, we replace each internal oracle G_i, where $0 \leq i \leq q_E - 1$, by an oracle G'_i that uniformly samples an output in $\mathcal{R} \setminus \mathcal{R}^i_{\mathsf{BAD}}$. Following the same analysis in the proof of [DHK+21, Theorem 3.2, Game 4] and a hybrid argument, we have

$$\left|\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]\right| \leq 8q_E\delta(q_G + q_H + q_D + 1)^2.$$

Game G_5. In this game, we modify the behavior of $\mathsf{H}(\mathsf{cpk}, m)$ as follows. For every input such that $\mathsf{cpk} = \mathsf{cpk}^i$ for some $0 \leq i \leq q_E - 1$, we set $\mathsf{H}(\mathsf{cpk}, m) := \overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}^i, \mathsf{ct})$, where $\mathsf{ct} := \mathsf{Enc}_{\mathsf{AFS}}\big(\mathsf{cpk}^i, m; \mathsf{G}'_i(\mathsf{cpk}^i, m)\big)$. Under this modification of H, the oracle $\mathcal{O}^{\mathsf{Dec}_s}$ can be simulated without relying on the secret key sk^*_s. Concretely, upon any query $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}^i, \mathsf{ct})$ with $0 \leq i \leq q_E - 1$, the oracle responds with $K = \overline{\mathsf{H}}_{\mathsf{rej}}(\mathsf{cpk}^i, \mathsf{ct})$.

Due to the modification introduced in G_4, for every tuple (cpk^i, m), it holds that if $\mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^i, m; \mathsf{G}(\mathsf{cpk}^i, m)) = \mathsf{ct}$, then we have $\mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk}^i, \mathsf{ct}) = m$. Consequently, the mapping from (cpk^i, m) to $(\mathsf{cpk}^i, \mathsf{ct})$ is injective. Moreover, since $\overline{\mathsf{H}}_{\mathsf{rej}}$ is never evaluated on such inputs in G_4, the adversary's view is identical in Games G_4 and G_5. That is,

$$\Pr[G_4^{\mathcal{A}} \Rightarrow 1] = \Pr[G_5^{\mathcal{A}} \Rightarrow 1].$$

Game G_6. In this game, we undo the changes in G_4, that is, we replace each G'_i by G_i for $0 \leq i \leq q_E - 1$. Using the same argument as in G_4, we have

$$\left|\Pr[G_5^{\mathcal{A}} \Rightarrow 1] - \Pr[G_6^{\mathcal{A}} \Rightarrow 1]\right| \leq 8q_E\delta(q_G + q_H + q_D + 1)^2.$$

Game G_7. In this game, we introduce a single random oracle $\mathsf{H} \times \mathsf{G} : \mathcal{CPK} \times \mathcal{M} \to \mathcal{K} \times \mathcal{R}$ to jointly simulate the random oracles H and G. That is, we define $(\mathsf{H}(\mathsf{cpk}, m), \mathsf{G}(\mathsf{cpk}, m)) := \mathsf{H} \times \mathsf{G}(\mathsf{cpk}, m)$. Since this change is purely conceptual, we have

$$\Pr[G_6^{\mathcal{A}} \Rightarrow 1] = \Pr[G_7^{\mathcal{A}} \Rightarrow 1].$$

Game G_8. In this game, we replace $\mathsf{H} \times \mathsf{G}$ with another oracle $\overline{\mathsf{H} \times \mathsf{G}}$. In particular, $\overline{\mathsf{H} \times \mathsf{G}}(x) = (K, r)$ if $x = (\mathsf{cpk}^*, m^*)$, where K, r are randomly chosen from $\mathcal{K}, \mathcal{R}$, and $\overline{\mathsf{H} \times \mathsf{G}}(x) = \mathsf{H} \times \mathsf{G}(x)$ for all $x \neq (\mathsf{cpk}^*, m^*)$. Notably, G_8 still uses $\mathsf{H} \times \mathsf{G}(\mathsf{cpk}^*, m^*)$ to compute K^*_0 and r^*. Therefore in Game G_8, the adversary's view is independent of b, and

$$\Pr[G_8^{\mathcal{A}} \Rightarrow 1] = 1/2.$$

Since we have reprogrammed $\mathsf{H} \times \mathsf{G}$ in G_8 on (cpk^*, m^*), by Lemma A.10 we have

$$|\Pr[G_7^{\mathcal{A}} \Rightarrow 1] - \Pr[G_8^{\mathcal{A}} \Rightarrow 1]| \leq 2\sqrt{(q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}} + 1)\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*, m^*)}(m^*)]}.$$

Here $\mathcal{C}$ runs Game G_8 with adversary $\mathcal{A}$ using the oracle $\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)$ instead of $\mathsf{H} \times \mathsf{G}$. In particular, $m^* \leftarrow\!\!\$\ \mathcal{M}$ is treated as an input to $\mathcal{C}$. Moreover, $\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*, m^*)}]$ denotes the probability that the oracle $\mathcal{O}^{SC}_{(\mathsf{cpk}^*, m^*)}$ ever outputs 1 during the execution of $\mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*, m^*)}(m^*)$.

Game G_9. In this game, we change $c^* = \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, m^*; r^*)$ to $c^* = \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, \hat{m}^*; r^*)$, where $\hat{m}^*$ is sampled independently from $\mathcal{M}$. Let $\mathcal{D}$ denote the algorithm with no input that runs Game G_9 with adversary $\mathcal{A}$ and oracle $\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)$.

To bound $\left|\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}(m^*)] - \Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()]\right|$, we construct an adversary $\mathcal{B}$ against the IND-LCK-SCPA security of AFS-PKE as follows. Adversary $\mathcal{B}$ receives pk_s^* from its challenger and forwards it to $\mathcal{A}$. When $\mathcal{A}$ outputs cpk^*, adversary $\mathcal{B}$ submits it together with two random messages $m_0^*, m_1^* \leftarrow\!\!\$\ \mathcal{M}$ to its challenger and receives a challenge ciphertext ct_b^*. Then $\mathcal{B}$ perfectly simulates Game G_9 for $\mathcal{A}$, except that it embeds $\mathsf{ct}^* = \mathsf{ct}_b^*$ and uses $\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m_0^*)$ to simulate the random oracle instead of $\mathsf{H} \times \mathsf{G}$. Finally, if FIND occurs, $\mathcal{B}$ outputs 1; otherwise it outputs 0.

It is clear that if $b = 0$, then $\mathcal{B}$ provides a perfect simulation of $\mathcal{C}$, and if $b = 1$, it provides a perfect simulation of $\mathcal{D}$. That is, $\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}(m^*)] = \Pr[\mathcal{B} \Rightarrow 1 : b = 0]$ and $\Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()] = \Pr[\mathcal{B} \Rightarrow 1 : b = 1]$. Therefore, we have

$$\left|\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}(m^*)] - \Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()]\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE}}(\mathcal{B}).$$

Analysis of $\Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()]$. There is clear independence between the (empty) input of $\mathcal{D}$ and the set S. That is, m^* is independent of the adversary's view. Using Lemma A.11, we have

$$\Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()] \leq 4(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}})/|\mathcal{M}|.$$

Combining all the bounds completes the proof. □

Security Analysis for IND-SCCA in the QROM.

Proof. Since the proof closely follows the analysis of IND-LCK-SCCA in the QROM, we only highlight the differences here. Game G_0 is the original IND-SCCA security game, i.e.,

$$\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\Pi_{\mathsf{AFS\text{-}KEM}}}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|.$$

Games G_1–G_9 follow the same sequence of transformations as in the IND-LCK-SCCA proof in the QROM. In particular, in this case we have $q_E + 1$ elements in L_e. Thus, we replace the term q_E in the IND-LCK-SCCA proof with $q_E + 1$. The final difference from the IND-LCK-SCCA analysis in this proof is the bound $\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}(m^*)] - \Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()]$.

To bound this term, we construct an adversary $\mathcal{B}$ against the IND-SCPA security of AFS-PKE as follows. Adversary $\mathcal{B}$ receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{seed}_e^*)$ from its challenger, chooses random messages $m_0^*, m_1^* \leftarrow\!\!\$\ \mathcal{M}$, and submits them to obtain a challenge ciphertext ct_b^*. It then forwards $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{seed}_e^*, \mathsf{ct}_b^*, K^*)$ to $\mathcal{A}$ and perfectly simulates the game as in G_9, where K^* is sampled uniformly at random. Moreover, $\mathcal{B}$ uses $\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m_0^*)$ to simulate the random oracle instead of $\mathsf{H} \times \mathsf{G}$. Finally, if FIND occurs, $\mathcal{B}$ outputs 1; otherwise, it outputs 0.

It is clear that if $b = 0$, then $\mathcal{B}$ provides a perfect simulation of $\mathcal{C}$, and if $b = 1$, it provides a perfect simulation of $\mathcal{D}$. That is, $\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}(m^*)] = \Pr[\mathcal{B} \Rightarrow 1 : b = 0]$ and $\Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()] = \Pr[\mathcal{B} \Rightarrow 1 : b = 1]$. Therefore, we have

$$\left|\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}(m^*)] - \Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H}\times\mathsf{G}}\setminus(\mathsf{cpk}^*,m^*)}()]\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE}}.$$

This completes the proof. □

Security Analysis for IND-ECPA in the QROM.

Proof. Game G_0 is the original IND-ECPA security game, that is,

$$\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\Pi_{\mathsf{KEM}}}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|.$$

Game G_1. In this game, we introduce a single random oracle $\mathsf{H} \times \mathsf{G} : \mathcal{CPK} \times \mathcal{M} \to \mathcal{K} \times \mathcal{R}$ to jointly simulate the random oracles H and G. That is, we define $(\mathsf{H}(\mathsf{cpk}, m), \mathsf{G}(\mathsf{cpk}, m)) := \mathsf{H} \times \mathsf{G}(\mathsf{cpk}, m)$. The total number of oracle queries made by the adversary to $\mathsf{H} \times \mathsf{G}$ is at most $q_{\mathsf{H}} + q_{\mathsf{G}}$. Since this change is purely conceptual, we have

$$\Pr[G_0^{\mathcal{A}} \Rightarrow 1] = \Pr[G_1^{\mathcal{A}} \Rightarrow 1].$$

Game G_2. In this game, we replace $\mathsf{H} \times \mathsf{G}$ with another oracle $\overline{\mathsf{H} \times \mathsf{G}}$. In particular, $\overline{\mathsf{H} \times \mathsf{G}}(x) = (K, r)$ if $x = (\mathsf{cpk}^*, m^*)$, where K, r are sampled uniformly from $\mathcal{K}, \mathcal{R}$, and $\overline{\mathsf{H} \times \mathsf{G}}(x) = \mathsf{H} \times \mathsf{G}(x)$ for all $x \neq (\mathsf{cpk}^*, m^*)$. Notably, Game G_2 still uses $\mathsf{H} \times \mathsf{G}(\mathsf{cpk}^*, m^*)$ to compute K_0^* and r^*. Therefore, in Game G_2, the adversary's view is independent of b, and thus

$$\Pr[G_2^{\mathcal{A}} \Rightarrow 1] = 1/2.$$

Since we have reprogrammed $\mathsf{H} \times \mathsf{G}$ in G_2 on (cpk^*, m^*), Lemma A.10 implies $|\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]| \leq 2\sqrt{(q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{D}} + 1)\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}(m^*)]}$. Here $\mathcal{C}$ runs Game G_2 with adversary $\mathcal{A}$ using the oracle $\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)$ instead of $\mathsf{H} \times \mathsf{G}$. In particular, $m^* \leftarrow\!\!\$\ \mathcal{M}$ is treated as an input to $\mathcal{C}$. Moreover, $\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}]$ denotes the probability that the oracle $\mathcal{O}^{SC}_{(\mathsf{cpk}^*, m^*)}$ ever outputs 1 during the execution of $\mathcal{C}^{\overline{\mathsf{H} \times \mathsf{G} \setminus (\mathsf{cpk}^*, m^*)}}(m^*)$.

Game G_3. In this game, we change $c^* = \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, m^*; r^*)$ to $c^* = \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, \hat{m}^*; r^*)$, where $\hat{m}^*$ is sampled independently from $\mathcal{M}$. Let $\mathcal{D}$ denote the algorithm with no input that runs Game G_3 with adversary $\mathcal{A}$ and oracle $\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)$.

To bound $\left|\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}(m^*)] - \Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}()]\right|$, we construct a reduction $\mathcal{B}$ against the IND-ECPA security of AFS-PKE. Adversary $\mathcal{B}$ receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*)$ from its challenger, chooses random messages $m_0^*, m_1^* \leftarrow\!\!\$\ \mathcal{M}$, and submits them to obtain a challenge ciphertext ct_b^*. It then forwards $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*, \mathsf{ct}_b^*, K^*)$ to $\mathcal{A}$ and perfectly simulates Game G_3, where K^* is sampled uniformly at random. Moreover, $\mathcal{B}$ uses $\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m_0^*)$ to simulate the random oracle instead of $\mathsf{H} \times \mathsf{G}$. Finally, if FIND occurs, $\mathcal{B}$ outputs 1; otherwise it outputs 0. It is clear that if $b = 0$, then $\mathcal{B}$ provides a perfect simulation of $\mathcal{C}$, and if $b = 1$, it provides a perfect simulation of $\mathcal{D}$. That is, $\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}(m^*)] = \Pr[\mathcal{B} \Rightarrow 1 : b = 0]$ and $\Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}()] = \Pr[\mathcal{B} \Rightarrow 1 : b = 1]$. Therefore, we have

$$\left|\Pr[\mathsf{FIND} : \mathcal{C}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}(m^*)] - \Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}()]\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}PKE}}.$$

Analysis of $\Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}()]$. There is clear independence between the (empty) input of $\mathcal{D}$ and the set S, and thus m^* is independent of the adversary's view. Using Lemma A.11, we obtain

$$\Pr[\mathsf{FIND} : \mathcal{D}^{\overline{\mathsf{H} \times \mathsf{G}} \setminus (\mathsf{cpk}^*, m^*)}()] \leq 4(q_{\mathsf{G}} + q_{\mathsf{H}})/|\mathcal{M}|.$$

This completes the proof. □

C Appendix: Additional Details Regarding AFS-PKE-ML and AFS-KEM-ML

C.1 Correctness Analysis for AFS-PKE-ML

Theorem 7.1 (Correctness of AFS-PKE-ML). *Using the notation in Fig. 9, the* AFS-PKE-ML *is δ-correct provided that*

$$\Pr\left[\left\|(\mathbf{e}_s + \mathbf{e}_e)^T\mathbf{r} + e_2 + e_v - (\mathbf{s}_s + \mathbf{s}_e)^T(\mathbf{e}_1 + \mathbf{e}_u)\right\|_\infty \geq \frac{q}{4}\right] = \delta,$$

where $\mathbf{e}_u = \mathsf{Decompress}_q(\mathsf{ct}_1, d_u) - \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$, *and* $e_v = \mathsf{Decompress}_q(\mathsf{ct}_2, d_v) - \mathbf{t}^T\mathbf{r} + e_2$.

Proof. From Algorithm $\mathsf{Dec}_{\mathsf{AFS}}(\mathsf{csk},\mathsf{ct})$, the value of $\mathbf{u}'$ and v' is

$$\mathbf{u}' = \mathsf{Decompress}_q(\mathsf{Compress}_q(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1, d_u), d_u) = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1 + \mathbf{e}_u$$

and

$$\begin{aligned} v' &= \mathsf{Decompress}_q(\mathsf{Compress}_q(\mathbf{t}_c^T\mathbf{r} + e_2 + \mathsf{Encode}(m), d_v), d_v) \\ &= \mathbf{t}_c^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m \\ &= (\mathbf{A}(\mathbf{s}_s + \mathbf{s}_e) + \mathbf{e}_s + \mathbf{e}_e)^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m \\ &= (\mathbf{A}\mathbf{s}_s + \mathbf{A}\mathbf{s}_e)^T\mathbf{r} + \mathbf{e}_s^T\mathbf{r} + \mathbf{e}_e^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m \\ &= (\mathbf{s}_s + \mathbf{s}_e)^T\mathbf{A}^T\mathbf{r} + \mathbf{e}_s^T\mathbf{r} + \mathbf{e}_e^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m, \end{aligned}$$

respectively. Then we have

$$\begin{aligned} & v' - \mathbf{s}_c^T\mathbf{u}' \\ &= (\mathbf{s}_s + \mathbf{s}_e)^T\mathbf{A}^T\mathbf{r} + \mathbf{e}_s^T\mathbf{r} + \mathbf{e}_e^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m - \mathbf{s}_c^T(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1 + \mathbf{e}_u) \\ &= (\mathbf{s}_s + \mathbf{s}_e)^T\mathbf{A}^T\mathbf{r} + \mathbf{e}_s^T\mathbf{r} + \mathbf{e}_e^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m - (\mathbf{s}_s + \mathbf{s}_e)^T(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1 + \mathbf{e}_u) \\ &= (\mathbf{e}_s + \mathbf{e}_e)^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m - (\mathbf{s}_s + \mathbf{s}_e)^T(\mathbf{e}_1 + \mathbf{e}_u). \end{aligned}$$

Since $m' = \mathsf{Decode}(v' - \mathbf{s}_c^T\mathbf{u}')$, we have

$$\begin{aligned} m' &= \left\lfloor \frac{2}{q}((\mathbf{e}_s + \mathbf{e}_e)^T\mathbf{r} + e_2 + e_v + \left\lfloor \frac{q}{2} \right\rceil m - (\mathbf{s}_s + \mathbf{s}_e)^T(\mathbf{e}_1 + \mathbf{e}_u)) \right\rceil \\ &= \left\lfloor m + \frac{2}{q}((\mathbf{e}_s + \mathbf{e}_e)^T\mathbf{r} + e_2 + e_v - (\mathbf{s}_s + \mathbf{s}_e)^T(\mathbf{e}_1 + \mathbf{e}_u)) \right\rceil \end{aligned}$$

If $\left\|(\mathbf{e}_s + \mathbf{e}_e)^T\mathbf{r} + e_2 + e_v - (\mathbf{s}_s + \mathbf{s}_e)^T(\mathbf{e}_1 + \mathbf{e}_u)\right\|_\infty < \frac{q}{4}$, we can write $m' = \lfloor m + err \rceil$ for $err < \frac{1}{2}$, which implies $m' = m$. This completes the proof. □

C.2 Security Analysis for AFS-PKE-ML

In AFS-KEM-ML, we rely on the Module Learning with Errors (MLWE) assumption, as in ML-KEM.

Definition C.1 (Module-LWE Problem [LS15]). *Let χ be a probability distribution over the ring $\mathcal{R}_q$, and let k, l, q be positive integers. For an adversary $\mathcal{A}$, its advantage against the decisional module learning with errors problem $\mathsf{MLWE}_{\chi,k,l,q}$ is defined as*

$$\mathsf{Adv}_{\mathsf{MLWE}_{\chi,k,l,q}}(\mathcal{A}) := \left|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}) = 1]\right|,$$

where $\mathbf{A} \leftarrow\!\!\$\ \mathcal{R}_q^{k\times l}$, $\mathbf{s} \leftarrow\!\!\$\ \chi^l$, $\mathbf{e} \leftarrow\!\!\$\ \chi^k$, and $\mathbf{u} \leftarrow\!\!\$\ \mathcal{R}_q^k$.

Theorem 7.2 (IND-LCK-SCPA Security of AFS-PKE-ML in the ROM/QROM). *We model the hash functions G and G' as random oracles. For any adversary $\mathcal{A}$ against the* IND-LCK-SCPA *security of* AFS-PKE-ML *making at most q_{G} and $q_{\mathsf{G}'}$ queries to G and G', respectively, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ running in time comparable to that of $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k,k,q}(\mathcal{B}) + q_{\mathsf{G}'}\,\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}) + q_{\mathsf{G}'}\,2^{-\lambda}.$$

If the hash functions G and G' are modeled as quantum-accessible random oracles and $\mathcal{A}$ makes at most q_{G} and $q_{\mathsf{G}'}$ (quantum) queries to G and G', respectively, then there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ running in time comparable to that of $\mathcal{A}$ such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k,k,q}(\mathcal{B}) + (2q_{\mathsf{G}'}+1)^2\,\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}) + 2q_{\mathsf{G}'}\,2^{-\lambda/2}.$$

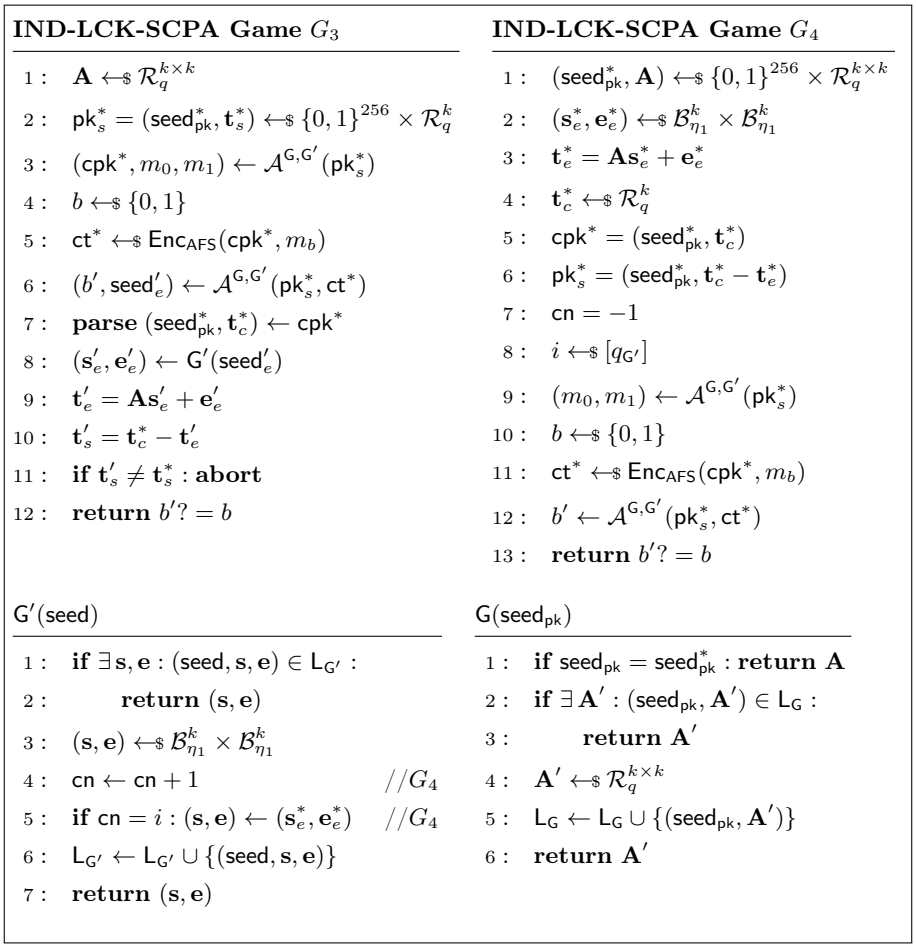


Fig. 12. Games G_3, G_4 for the IND-LCK-SCPA security of AFS-PKE-ML in the ROM

Security Analysis in the ROM.

Proof. We proceed with a sequence of games, where G_3 and G_4 are given in detail in Fig. 12.

Game G_0. Game G_0 is the initial IND-LCK-SCPA security game. Moreover, we assume that the adversary queries $\mathsf{G}'(\mathsf{seed}_e')$ before outputting seed_e'. Notably, this does not change the adversary's advantage. Indeed, for any adversary $\mathcal{A}$ making at most $q_{\mathsf{G}'}$ queries to G', we can construct a random oracle middleman adversary $\mathcal{A}'$ that always queries $\mathsf{G}'(\mathsf{seed}_e')$ before outputting the value seed_e' received from $\mathcal{A}$. Moreover, the combined public key cpk^* output by the adversary must be well-formed, namely $\mathsf{cpk}^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^*)$, where $\mathsf{seed}_{\mathsf{pk}}^*$ is the seed in pk_s^* used to generate the matrix $\mathbf{A}$. Otherwise, the game aborts, since this would imply $\mathsf{pk}_s' \neq \mathsf{pk}_s^*$. That is,

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1. In game G_1, we change how to simulate the random oracle G. Specifically, at the beginning of the game, we first sample $(\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{A}) \leftarrow\$ \{0,1\}^{256} \times \mathcal{R}_q^{k\times k}$. Then we define G such that $\mathsf{G}(\mathsf{seed}_{\mathsf{pk}}^*) = \mathbf{A}$. This game is the same as G_0 for $\mathcal{A}$, thus

$$\mathsf{Adv}^{G_0}(\mathcal{A}) = \mathsf{Adv}^{G_1}(\mathcal{A}).$$

Game G_2. In this game, we replace $(\mathbf{s}_s^*, \mathbf{e}_s^*) \leftarrow \mathsf{G}'(\mathsf{seed}_s^*)$ with uniformly random $(\mathbf{s}_s^*, \mathbf{e}_s^*) \leftarrow\$ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$. If the adversary never queries $\mathsf{G}'(\mathsf{seed}_s^*)$, then Games G_1 and G_2 are identical for the adversary. Since $\mathsf{seed}_s^* \leftarrow\$ \{0,1\}^\lambda$ is uniformly random from the adversary's perspective, the probability that the adversary queries it is at most $q_{\mathsf{G}'}/2^\lambda$. Therefore, we have

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \leq \frac{q_{\mathsf{G}'}}{2^\lambda}.$$

Game G_3. In game G_3, $\mathbf{t}_s^* = \mathbf{A}\mathbf{s}_s^* + \mathbf{e}_s^*$ is sampled uniformly at random. For any adversary that can distinguish between G_2 and G_3, there exists an adversary $\mathcal{B}$ that can solve the $\mathsf{MLWE}(\mathcal{B}_{\eta_1}, k, k, q)$ problem. Therefore, we have

$$\left|\mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k,k,q}(\mathcal{B}).$$

Game G_4. The full description of G_4 is given in Fig. 12. In particular, in G_4 the challenger samples a uniformly random $\mathbf{t}_c^* \leftarrow\!\$\ \mathcal{R}_q^k$ and defines the challenge composite public key as $\mathsf{cpk}^* = (\mathsf{seed}^*_{\mathsf{pk}}, \mathbf{t}_c^*)$, which is used to compute ct^*. It further samples independent ephemeral secrets $(\mathbf{s}_e^*, \mathbf{e}_e^*)$ and an index $i \leftarrow\!\$\ [q_{\mathsf{G}'}]$, and sets $\mathbf{t}_e^* := \mathbf{A}\mathbf{s}_e^* + \mathbf{e}_e^*$ and $\mathsf{pk}_s^* = (\mathsf{seed}^*_{\mathsf{pk}}, \mathbf{t}_c^* - \mathbf{t}_e^*)$. When the adversary makes the $(i+1)$-th query to G', the challenger returns $(\mathbf{s}_e^*, \mathbf{e}_e^*)$.

For any adversary $\mathcal{A}$ that wins in Game G_3, we construct an adversary $\hat{\mathcal{A}}$ attacking the Game G_4 as follows. The adversary $\hat{\mathcal{A}}$ forwards pk_s^* to $\mathcal{A}$. Upon receiving $(\mathsf{cpk}', m_0, m_1)$ from $\mathcal{A}$, it forwards (m_0, m_1) to its challenger and obtains the challenge ciphertext ct^*, which is then given to $\mathcal{A}$. All random-oracle queries of $\mathcal{A}$ are answered using $\hat{\mathcal{A}}$'s own oracle access. Finally, when $\mathcal{A}$ outputs (b', seed'_e), $\hat{\mathcal{A}}$ outputs the bit b'. In particular, when the sampled index $i \leftarrow\!\$\ [q_{\mathsf{G}'}]$ in G_4 coincides with the first query of the form $\mathsf{G}'(\mathsf{seed}'_e)$ made by $\mathcal{A}$ (which occurs with probability $1/q_{\mathsf{G}'}$), the composite public key cpk output by $\mathcal{A}$ is identical to the challenge value cpk^* in the game of $\hat{\mathcal{A}}$, provided that the game of $\mathcal{A}$ does not abort. Therefore, the simulation for $\mathcal{A}$ is perfect, and $\hat{\mathcal{A}}$ succeeds whenever $\mathcal{A}$ succeeds. Consequently, we have

$$\mathsf{Adv}^{G_3}(\mathcal{A}) \leq q_{\mathsf{G}'} \cdot \mathsf{Adv}^{G_4}(\hat{\mathcal{A}}).$$

Game G_5. In Game G_5, the values $\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$ and $v = \mathbf{t}_c^{*T}\mathbf{r} + e_2$ in the challenge ciphertext ct^* are replaced by uniformly random values $\mathbf{u} \leftarrow\!\$\ \mathcal{R}_q^k$ and $v \leftarrow\!\$\ \mathcal{R}_q$. Note that adding any fixed value $\lfloor \frac{q}{2} \rceil\, m$ to v preserves the uniform distribution over $\mathcal{R}_q$. For any adversary that can distinguish Game G_4 from Game G_5, we can construct an adversary $\mathcal{C}$ against $\mathsf{MLWE}(\mathcal{B}_{\eta_1}, k+1, k, q)$ such that

$$\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}).$$

In game G_5, ct^* and m_b are independent, therefore

$$\mathsf{Adv}^{G_5}(\mathcal{A}) = 0.$$

Combining all bounds above completes the proof. □

Security Analysis in the QROM.

Proof. We proceed with a sequence of games, where G_3, G_4, G_5 are given in detail in Fig. 13.

Game G_0. Game G_0 is the initial $\mathsf{IND\text{-}LCK\text{-}SCPA}$ security game. Notably, the combined public key cpk^* output by the adversary must be of the form $\mathsf{cpk}^* = (\mathsf{seed}^*_{\mathsf{pk}}, \mathsf{t}_c^*)$, where $\mathsf{seed}^*_{\mathsf{pk}}$ is the seed in pk_s^* used to generate the matrix $\mathbf{A}$. Otherwise the game aborts, as this would imply $\mathsf{pk}'_s \neq \mathsf{pk}_s^*$. That is,

$$\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1. In this game, we modify how the random oracle G is simulated. That is, we first sample $(\mathsf{seed}_{\mathsf{pk}}, \mathbf{A}) \leftarrow\!\$\ \{0,1\}^{256} \times \mathcal{R}_q^{k\times k}$ and a $2q_G$-wise independent function $\hat{\mathsf{G}} : \{0,1\}^{256} \times \mathcal{R}_q^{k\times k}$. Then we define $\mathsf{G}(x) = \hat{\mathsf{G}}(x) + \mathbf{A} - \mathbf{M}$ for every $x \in \{0,1\}^{256}$, where $\mathbf{M} = \mathsf{G}'(\mathsf{seed}_A)$. Using Lemma A.6, a $2(q_G+1)$-wise independent function $\hat{\mathsf{G}}$ is perfectly indistinguishable from a random oracle to any adversary $\mathcal{A}$ making at most (q_G+1) oracle queries (including the implicit query $\hat{\mathsf{G}}(\mathsf{seed}_A)$ made by the challenger). Therefore, G is also perfectly indistinguishable from a random oracle for the adversary $\mathcal{A}$. Notably, we have $\mathsf{G}(\mathsf{seed}_{\mathsf{pk}}) = \mathbf{A}$ and $\mathbf{A}$ is uniformly random sampled in this game and

$$\Pr[G_0^{\mathcal{A}} \Rightarrow 1] = \Pr[G_1^{\mathcal{A}} \Rightarrow 1].$$

IND-LCK-SCPA Game G_3, G_4

$\mathsf{cn} = -1$ //G_4
$(i, b) \leftarrow\$ ([q_{\mathsf{G}'}] \times \{0,1\}) \cup \{(q, 0)\}$ //G_4
$\mathbf{A} \leftarrow\$ \mathcal{R}_q^{k \times k}$
$\mathsf{pk}_s^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_s^*) \leftarrow\$ \{0,1\}^{256} \times \mathcal{R}_q^k$
$(\mathbf{s}_e^*, \mathbf{e}_e^*) \leftarrow\$ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$ //G_4
$\mathbf{t}_e^* = \mathbf{A}\mathbf{s}_e^* + \mathbf{e}_e^*$ //G_4
$(\mathsf{cpk}^*, m_0, m_1) \leftarrow \mathcal{A}^{\lvert \mathsf{G}' \rangle, \lvert \mathsf{G} \rangle}(\mathsf{pk}_s^*)$ //G_3
$(\mathsf{cpk}^*, m_0, m_1) \leftarrow \mathcal{A}^{\lvert \mathsf{G}_i' \rangle, \lvert \mathsf{G} \rangle}(\mathsf{pk}_s^*)$ //G_4
$b \leftarrow\$ \{0,1\}$
$\mathsf{ct}^* \leftarrow\$ \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, m_b)$
$(b', \mathsf{seed}_e') \leftarrow \mathcal{A}^{\lvert \mathsf{G}' \rangle, \lvert \mathsf{G} \rangle}(\mathsf{pk}_s^*, \mathsf{ct}^*)$ //G_3
$(b', \mathsf{seed}_e') \leftarrow \mathcal{A}^{\lvert \mathsf{G}_i' \rangle, \lvert \mathsf{G} \rangle}(\mathsf{pk}_s^*, \mathsf{ct}^*)$ //G_4
parse $(\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^*) \leftarrow \mathsf{cpk}^*$
if $\mathbf{t}_c^* - \mathbf{t}_e^* \neq \mathbf{t}_s^*$: **abort** //G_4
return $b' ? = b$ //G_4
$(\mathbf{s}_e', \mathbf{e}_e') \leftarrow \mathsf{G}'(\mathsf{seed}_e')$
$\mathbf{t}_e' \leftarrow \mathbf{A}\mathbf{s}_e' + \mathbf{e}_e'$
if $\mathbf{t}_c^* - \mathbf{t}_e' \neq \mathbf{t}_s^*$: **abort**
return $b' ? = b$

IND-LCK-SCPA Game G_5

$\mathsf{cn} = -1$
$(i, b) \leftarrow\$ ([q_{\mathsf{G}'}] \times \{0,1\}) \cup \{(q, 0)\}$
$(\mathbf{A}, \mathsf{seed}_{\mathsf{pk}}^*) \leftarrow\$ \mathcal{R}_q^{k \times k} \times \{0,1\}^{256}$
$\mathbf{t}_c^* \leftarrow\$ \mathcal{R}_q^k$
$(\mathbf{s}_e^*, \mathbf{e}_e^*) \leftarrow\$ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$
$\mathbf{t}_e^* = \mathbf{A}\mathbf{s}_e^* + \mathbf{e}_e^*$
$\mathsf{pk}_s^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^* - \mathbf{t}_e^*)$
$(m_0, m_1) \leftarrow \mathcal{A}^{\lvert \mathsf{G}_i' \rangle, \lvert \mathsf{G} \rangle}(\mathsf{pk}_s^*)$
$b \leftarrow\$ \{0,1\}$
$\mathsf{cpk}^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^*)$
$\mathsf{ct}^* \leftarrow\$ \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, m_b)$
$(b', \mathsf{seed}_e') \leftarrow \mathcal{A}^{\lvert \mathsf{G}_i' \rangle, \lvert \mathsf{G} \rangle}(\mathsf{pk}_s^*, \mathsf{ct}^*)$
return $b' ? = b$

$\mathsf{G}'(\mathsf{seed})$

random oracle

$\mathsf{G}(\mathsf{seed}_{\mathsf{pk}})$

simulates as G_1

$\mathsf{G}_i'(\mathsf{seed})$

$\mathsf{cn} \leftarrow \mathsf{cn} + 1$
if $\mathsf{cn} \geq i + b \ \wedge \ \mathsf{seed} = \mathsf{seed}_{i+1}$:
    seed_{i+1} is the measurement outcome
    of $\mathcal{A}$'s $(i+1)$-th query input register
    return $(\mathbf{s}_e^*, \mathbf{e}_e^*)$
return $\mathsf{G}'(\mathsf{seed})$

Fig. 13. G_3-G_5 for the IND-LCK-SCPA security of AFS-PKE-ML in the QROM

Game G_2. In this game, we replace $(\mathbf{s}_s^*, \mathbf{e}_s^*) \leftarrow \mathsf{G}'(\mathsf{seed}_s^*)$ with uniformly random $(\mathbf{s}_s^*, \mathbf{e}_s^*) \leftarrow\$ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$. Since $\mathsf{seed}_s^* \leftarrow\$ \{0,1\}^\lambda$ is uniformly random from the adversary's perspective, by Lemma A.7, we have

$$\left| \mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A}) \right| \leq \frac{q_{\mathsf{G}'}}{\sqrt{2^\lambda}}.$$

Game G_3. In game G_3, $\mathbf{t}_s^* = \mathbf{A}\mathbf{s}_s^* + \mathbf{e}_s^*$ is sampled uniformly at random. For any adversary that can distinguish between G_2 and G_3, there exists an adversary $\mathcal{B}$ that can solve the $\mathsf{MLWE}(\mathcal{B}_{\eta_1}, k, k, q)$ problem. Therefore, we have

$$\left| \mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A}) \right| \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1}, k, k, q}(\mathcal{B}).$$

Let $C^{\lvert \mathsf{G}' \rangle}$ be an oracle algorithm that perfectly simulates Game G_3 for the adversary $\mathcal{A}$. Internally, it samples a random oracle G', a bit b, a matrix $\mathbf{A}$, and a static public key $\mathsf{pk}_s^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_s^*)$, and computes the corresponding challenge ciphertext ct^* as G_3. Let $\mathsf{cpk}^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^*)$, (seed_e', b') denote the adversary's output. Let $x = \mathsf{seed}_e'$, $y = (y_0, y_1) = \mathsf{G}'(\mathsf{seed}_e')$, and $z = (z_1, z_2, z_3, z_4) = (\mathbf{A}, \mathbf{t}_c^* - \mathbf{t}_s^*, b, b')$. C finally outputs (x, z). Define the predicates $V_1(x, y, z) = [\, z_1 y_0 + y_1 =? z_2 \,]$ and $V_2(x, y, z) = [\, z_3 =? z_4 \,]$, and let $V := V_1 \wedge V_2$. This predicate V captures the adversary's success condition. Hence, $\mathsf{Adv}^{G_3}(\mathcal{A}) = \sum_{x_0} \Pr_{\mathsf{G}'}\big[x = x_0 \wedge V(x, \mathsf{G}'(x), z) = 1 : (x, z) \leftarrow C^{\mathsf{G}'}\big] \cdot \big|1 - 1/2\big|$.

Let $S^C(\Theta)$ be an oracle algorithm that randomly samples a pair $(i,b) \in ([q] \times \{0,1\}) \cup \{(q,0)\}$, runs $C^{|\mathsf{G}'\rangle}$ until the $(i+1)$-th query, measures the query input registers to obtain x, continues to run $C^{|\mathsf{G}'\rangle}$ until the $(i+b+1)$-th G'-query, reprograms G' to $\mathsf{G}'_{x\Theta}$ (where $\mathsf{G}'_{x\Theta}(x) = \Theta$ and $\mathsf{G}'_{x\Theta}(x') = \mathsf{G}'(x')$ for all $x' \neq x$), and runs $C^{|\mathsf{G}'_{x\Theta}\rangle}$ until the end to output z. Let $y = (y_0, y_1) = \mathsf{G}'(x) = \Theta$, and $z = (z_1, z_2, z_3, z_4) = (\mathbf{A}, \mathbf{t}_c^* - \mathbf{t}_s^*, b, b')$. C finally outputs (x,z). Sampling $\Theta = (\mathbf{s}_e^*, \mathbf{e}_e^*) \leftarrow\!\$\ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$ and a random oracle G', $S^C(\Theta)$ perfectly simulates Game G_4 (shown in Fig. 13). And we have $\mathsf{Adv}^{G_4}(\mathcal{A}) = \sum_{x_0} \Pr\big[x = x_0 \wedge V(x,\Theta,z) = 1 : (x,z) \leftarrow S^C(\Theta)\big] \cdot \big|1 - 1/2\big|$.

According to Lemma A.12, we have $\sum_{x_0} \Pr_{\mathsf{G}'}\big[x = x_0 \wedge V(x,\mathsf{G}'(x),z) = 1 : (x,z) \leftarrow C^{\mathsf{G}'}\big] \leq (2q_{\mathsf{G}'}+1)^2 \sum_{x_0} \Pr_{\Theta}\big[x = x_0 \wedge V(x,\Theta,z) = 1 : (x,z) \leftarrow S^C(\Theta)\big]$. Therefore, we have

$$\mathsf{Adv}^{G_3}(\mathcal{A}) \leq (2q_{\mathsf{G}'}+1)^2 \mathsf{Adv}^{G_4}(\mathcal{A}).$$

Game G_5. In this game, we modify the challenger's behavior as illustrated in Fig. 13. Notably, if Game G_4 does not abort, we have $\mathsf{cpk}^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_e^* + \mathbf{t}_s^*)$, therefore the challenger can directly uses cpk^* to compute the challenge ciphertext ct^*. In this game, we first sample $\mathbf{t}_c^* \leftarrow\!\$\ \mathcal{R}_q^k$ and then set $\mathbf{t}_s^* := \mathbf{t}_c^* - \mathbf{t}_e^*$. We then use $\mathsf{cpk}^* := (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^*)$ to compute the challenge ciphertext ct_b.

Any adversary $\mathcal{A}$ that wins Game G_4 can be used to build an adversary $\mathcal{A}'$ that wins Game G_5 with the same advantage. Concretely, the adversary $\mathcal{A}'$ uses $(\mathsf{pk}_s^*, \mathsf{ct}^*)$ and its oracles to simulate Game G_4 for $\mathcal{A}$ and outputs the bit returned by $\mathcal{A}$. The simulation is perfect whenever Game G_4 does not abort, and thus a win of $\mathcal{A}$ in G_4 implies a win of $\mathcal{A}'$ in G_5. Therefore,

$$\mathsf{Adv}^{G_4}(\mathcal{A}) \leq \mathsf{Adv}^{G_5}(\mathcal{A}).$$

Game G_6. In Game G_6, the values $\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$ and $v = \mathbf{t}_c^{*T}\mathbf{r} + e_2$ in the challenge ciphertext ct^* are replaced by uniformly random values $\mathbf{u} \leftarrow\!\$\ \mathcal{R}_q^k$ and $v \leftarrow\!\$\ \mathcal{R}_q$. Note that adding any fixed value $\lfloor \frac{q}{2} \rceil\, m$ to v preserves the uniform distribution over $\mathcal{R}_q$. For any adversary that can distinguish Game G_5 from Game G_6, we can construct an adversary $\mathcal{C}$ against $\mathsf{MLWE}(\mathcal{B}_{\eta_1}, k+1, k, q)$ such that

$$\Big|\mathsf{Adv}^{G_5}(\mathcal{A}) - \mathsf{Adv}^{G_6}(\mathcal{A})\Big| \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}).$$

In game G_6, ct^* and m_b are independent, therefore

$$\mathsf{Adv}^{G_6}(\mathcal{A}) = 0.$$

Combining all bounds above completes the proof. □

Theorem 7.3 (IND-S(E)CPA Security of AFS-PKE-ML in the QROM). *We model the hash function* G *as a quantum-accessible random oracle. For any adversary* $\mathcal{A}$ *against either the* IND-ECPA *or the* IND-SCPA *security of AFS-PKE-ML that makes at most* q_G *(quantum) queries to the random oracle* G*, there exist adversaries* $\mathcal{B}, \mathcal{C}, \mathcal{D}$ *running in time comparable to that of* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}S(E)CPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k,k,q}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_{\eta_1},k+1,k,q}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{G}'}(\mathcal{D}).$$

Proof. Since the proofs of IND-ECPA and IND-SCPA security are similar, we briefly outline the argument for IND-ECPA here. We proceed with a sequence of games, where G_3-G_5 are given in detail in Fig. 14.

Game G_0. Game G_0 is the initial IND-ECPA security game. Therefore, we have

$$\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}PKE\text{-}ML}}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|.$$

Game G_1. In this game, we modify how the random oracle G is simulated. That is, we first sample $(\mathsf{seed}_{\mathsf{pk}}, \mathbf{A}) \leftarrow\!\$\ \{0,1\}^{256} \times \mathcal{R}_q^{k\times k}$ and a $2q_G$-wise independent function $\mathsf{G}' : \{0,1\}^{256} \times \mathcal{R}_q^{k\times k}$. Then we define $\mathsf{G}(x) = \mathsf{G}'(x) + \mathbf{A} - \mathbf{M}$ for every $x \in \{0,1\}^{256}$, where $\mathbf{M} = \mathsf{G}'(\mathsf{seed}_A)$. Using Lemma A.6, a $2(q_G+1)$-wise independent function G' is perfectly indistinguishable from a random oracle to any adversary $\mathcal{A}$ making at most (q_G+1) oracle queries (including the implicit query $\mathsf{G}'(\mathsf{seed}_A)$ made by

IND-ECPA Game G_3

$(\mathsf{pk}_s^*, \mathsf{sk}_s^*) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$
parse $(\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_s^*) \leftarrow \mathsf{pk}_s^*$
$\mathbf{t}_e^* \leftarrow\!\$\ \mathcal{R}_q^k$
$\mathsf{pk}_e^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_e^*)$
$\mathsf{cpk}^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_s^* + \mathbf{t}_e^*)$
$(m_0, m_1) \leftarrow \mathcal{A}^{|\mathsf{G}\rangle}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*)$
$b \leftarrow\!\$\ \{0,1\}$
$\mathsf{ct}^* \leftarrow\!\$\ \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, m_b)$
$b' \leftarrow \mathcal{A}^{|\mathsf{G}\rangle}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*, \mathsf{ct}^*)$
return $b' ? = b$

$\mathsf{G}(\mathsf{seed}_{\mathsf{pk}})$

simulates as G_1

IND-ECPA Game G_4-G_5

$(\mathsf{pk}_s^*, \mathsf{sk}_s^*) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$
parse $(\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_s^*) \leftarrow \mathsf{pk}_s^*$
$\mathbf{t}_c^* \leftarrow\!\$\ \mathcal{R}_q^k$
$\mathsf{cpk}^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^*)$
$\mathsf{pk}_e^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^* - \mathbf{t}_s^*)$
$(m_0, m_1) \leftarrow \mathcal{A}^{|\mathsf{G}\rangle}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*)$
$b \leftarrow\!\$\ \{0,1\}$
$\mathsf{ct}^* \leftarrow\!\$\ \mathsf{Enc}_{\mathsf{AFS}}(\mathsf{cpk}^*, m_b)$ $//G_4$
$\mathbf{u} \leftarrow\!\$\ \mathcal{R}_q^k,\ \mathsf{ct}_1 = \mathsf{Compress}_q(\mathbf{u}, d_u)$ $//G_5$
$v \leftarrow\!\$\ \mathcal{R}_q,\ \mathsf{ct}_2 = \mathsf{Compress}_q(v, d_v)$ $//G_5$
$\mathsf{ct}^* = (\mathsf{ct}_1, \mathsf{ct}_2)$ $//G_5$
$b' \leftarrow \mathcal{A}^{|\mathsf{G}\rangle}(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{sk}_s^*, \mathsf{ct}^*)$
return $b' ? = b$

Fig. 14. G_3-G_5 for the IND-ECPA security proofs of AFS-PKE-ML.

the challenger). Therefore, G is also perfectly indistinguishable from a random oracle for the adversary $\mathcal{A}$. Notably, we have $\mathsf{G}(\mathsf{seed}_{\mathsf{pk}}) = \mathbf{A}$ in this game and

$$\Pr[G_0^{\mathcal{A}} \Rightarrow 1] = \Pr[G_1^{\mathcal{A}} \Rightarrow 1].$$

Game G_2. In this game, we replace $(\mathbf{s}_e^*, \mathbf{e}_e^*) \leftarrow \mathsf{G}'(\mathsf{seed}_e^*)$ with uniformly random $(\mathbf{s}_e^*, \mathbf{e}_e^*) \leftarrow\!\$\ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$. For any adversary $\mathcal{A}$ that can distinguish between Games G_1 and G_2, we construct an adversary $\mathcal{D}$ against the IND-PRG security of G'. Adversary $\mathcal{D}$ receives y_b, where y_0 is the real PRG output computed from a random seed, and y_1 is a truly random value from the output space of G'. Adversary $\mathcal{D}$ perfectly simulates Game G_2 except that it uses y_b as $(\mathbf{s}_e^*, \mathbf{e}_e^*)$ to compute $(\mathsf{pk}_e^*, \mathsf{sk}_e^*)$. When $b = 0$, $\mathcal{D}$ perfectly simulates G_1 and when $b = 1$, $\mathcal{D}$ perfectly simulates G_2. That is, $\Pr[G_1^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathcal{D} \Rightarrow 1 \mid b = 0]$ and $\Pr[G_2^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathcal{D} \Rightarrow 1 \mid b = 1]$. Therefore,

$$\left|\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]\right| \leq \mathsf{Adv}_{\mathsf{G}'}^{\mathsf{IND\text{-}PRG}}(\mathcal{D}).$$

Game G_3. In this game, we replace $\mathbf{t}_e^* = \mathbf{A}\mathbf{s}_e^* + \mathbf{e}_e^*$ in pk_e^* with a uniformly random value $\mathbf{t}_e^* \leftarrow\!\$\ \mathcal{R}_q^k$. For any adversary $\mathcal{A}$ that can distinguish between G_2 and G_3, we can construct an adversary $\mathcal{B}$ that solves the $\mathsf{MLWE}(\mathcal{B}_{\eta_1}, k, k, q)$ problem. Therefore, we have

$$\left|\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \Pr[G_3^{\mathcal{A}} \Rightarrow 1]\right| \leq \mathsf{Adv}_{\mathcal{B}_{\eta_1},k,k,q}^{\mathsf{MLWE}}(\mathcal{B}).$$

Game G_4. In this game, we re-express several elements of G_3 in an equivalent manner. Specifically, the polynomial vector $\mathbf{t}_c^*$ in cpk^* is sampled uniformly at random, and the ephemeral public key is defined as $\mathsf{pk}_e^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^* - \mathbf{t}_s^*)$. The distribution of $\mathbf{t}_e^*$ and $\mathbf{t}_c^*$ in G_3 is identical to that in G_4. Thus,

$$\Pr[G_3^{\mathcal{A}} \Rightarrow 1] = \Pr[G_4^{\mathcal{A}} \Rightarrow 1].$$

Game G_5. In this game, the values $\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$ and $v = \mathbf{t}_c^{*T}\mathbf{r} + e_2$ in the challenge ciphertext ct^* are replaced by uniformly random $\mathbf{u} \leftarrow\!\$\ \mathcal{R}_q^k$ and $v \leftarrow\!\$\ \mathcal{R}_q$ (adding any fixed $\lfloor \frac{q}{2} \rceil\, m$ to v preserves uniformity). Similarly, for any adversary that can distinguish between G_4 and G_5, we can construct an adversary $\mathcal{C}$ against the $\mathsf{MLWE}(\mathcal{B}_{\eta_1}, k+1, k, q)$ problem. That is, we have

$$\left|\Pr[G_4^{\mathcal{A}} \Rightarrow 1] - \Pr[G_5^{\mathcal{A}} \Rightarrow 1]\right| \leq \mathsf{Adv}_{\mathcal{B}_{\eta_1},k+1,k,q}^{\mathsf{MLWE}}(\mathcal{C}).$$

In this game, ct^* and m_b are independent. Therefore,

$$\Pr[G_5^{\mathcal{A}} \Rightarrow 1] = \frac{1}{2}.$$

Combining all bounds above completes the proof. □

C.3 SPK-Bind, RSPK-Bind, Key-Coll Security Analysis for AFS-KEM-ML

Theorem C.2 (SPK-Bind **Security of** AFS-KEM-ML **in the ROM/QROM).** *Let $q_{\mathsf{G}'}$ be the number of queries made by the adversary to the random oracle G', and let $p_{\eta_1} := \max_x \Pr[\mathrm{CBD}_{\eta_1} = x] = \frac{\binom{2\eta_1}{\eta_1}}{2^{2\eta_1}}$. For any PPT adversary $\mathcal{A}$ in the ROM, we have*

$$\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \le 2^{-256} + (q_{\mathsf{G}'} + 1)p_{\eta_1}^{kn}.$$

For any QPT adversary $\mathcal{A}$ making at most $q_{\mathsf{G}'}$ quantum queries to G' in the QROM, we have

$$\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \le 2^{-256} + O((q_{\mathsf{G}'} + 1)^2 p_{\eta_1}^{kn}).$$

Proof. Recall that in the SPK-Bind security game, the adversary receives $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and outputs a seed seed'. Then, the challenger computes $(\mathsf{pk}_e', \mathsf{sk}_e') \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}')$ and $\mathsf{pk}_s' \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e')$. The adversary wins if either $\mathsf{pk}_s' = \overline{\mathsf{pk}}_s^*$, or $\mathsf{seed}' \neq \mathsf{seed}_e^*$ and $\mathsf{pk}_s' = \mathsf{pk}_s^*$.

First, consider the case where $\mathsf{pk}_s' = \overline{\mathsf{pk}}_s^*$. By the construction of AFS-KEM-ML, the static public key contains the seed component $\mathsf{seed}_{\mathsf{pk}}$ used to generate the matrix $\mathbf{A}$. Moreover, the seed component $\mathsf{seed}_{\mathsf{pk}}'$ in pk_s' is the same as the $\mathsf{seed}_{\mathsf{pk}}^*$ in the static public key pk_s^*. Therefore, for pk_s' to equal $\overline{\mathsf{pk}}_s^*$, it is necessary that $\mathsf{seed}_{\mathsf{pk}}^* = \overline{\mathsf{seed}}_{\mathsf{pk}}^*$, where $\overline{\mathsf{seed}}_{\mathsf{pk}}^*$ denotes the corresponding seed component contained in $\overline{\mathsf{pk}}_s^*$. Since $\mathsf{seed}_{\mathsf{pk}}^*$ and $\overline{\mathsf{seed}}_{\mathsf{pk}}^*$ are independently sampled uniform 256-bits, this event occurs with probability at most 2^{-256}.

Second, consider the case where $\mathsf{seed}' \neq \mathsf{seed}_e^*$ and $\mathsf{pk}_s' = \mathsf{pk}_s^*$. Write $(\mathbf{s}_e^*, \mathbf{e}_e^*) = \mathsf{G}'(\mathsf{seed}_e^*)$, $(\mathbf{s}_e', \mathbf{e}_e') = \mathsf{G}'(\mathsf{seed}')$, and $\mathbf{A} = \mathsf{G}(\mathsf{seed}_{\mathsf{pk}}^*)$. By construction, $\mathsf{cpk}^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_c^*)$ with $\mathbf{t}_c^* = \mathbf{t}_s^* + \mathbf{t}_e^* \bmod q$, where $\mathbf{t}_e^* = \mathbf{A}\mathbf{s}_e^* + \mathbf{e}_e^* \bmod q$. On input seed', the challenger computes $\mathbf{t}_e' = \mathbf{A}\mathbf{s}_e' + \mathbf{e}_e' \bmod q$ and extracts $\mathsf{pk}_s' = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_s')$ with $\mathbf{t}_s' = \mathbf{t}_c^* - \mathbf{t}_e' \bmod q$. Thus, $\mathsf{pk}_s' = \mathsf{pk}_s^*$ implies $\mathbf{t}_s' = \mathbf{t}_s^*$, and hence $\mathbf{t}_e' = \mathbf{t}_e^* \bmod q$. Equivalently, $\mathbf{A}\mathbf{s}_e' + \mathbf{e}_e' = \mathbf{t}_e^* \bmod q$. Thus, the second case can occur only if a fresh seed $\mathsf{seed}' \neq \mathsf{seed}_e^*$ generates the same ephemeral public-key component as seed_e^*.

For every fresh seed $\mathsf{seed}' \neq \mathsf{seed}_e^*$, the pair $(\mathbf{s}_e', \mathbf{e}_e')$ is sampled independently according to G'. In particular, $\mathbf{e}_e' \leftarrow\!\$\ \mathcal{B}_{\eta_1}^k$ consists of kn independent coefficients sampled from CBD_{η_1}. Hence, by the definition of $p_{\eta_1} := \max_x \Pr[\mathrm{CBD}_{\eta_1} = x]$, we have $\max_{\mathbf{e}_0} \Pr_{\mathbf{e}_e' \leftarrow \mathcal{B}_{\eta_1}^k}[\mathbf{e}_e' = \mathbf{e}_0] = p_{\eta_1}^{kn}$. Therefore, for the fixed values of $\mathbf{A}$ and $\mathbf{t}_e^*$, we have

$$\begin{aligned}
&\Pr_{(\mathbf{s}_e', \mathbf{e}_e') \leftarrow\!\$\, \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k} [\mathbf{A}\mathbf{s}_e' + \mathbf{e}_e' = \mathbf{t}_e^* \bmod q] \\
&= \sum_{\mathbf{s} \in \mathcal{B}_{\eta_1}^k} \Pr_{\mathbf{s}_e' \leftarrow\!\$\, \mathcal{B}_{\eta_1}^k} [\mathbf{s}_e' = \mathbf{s}] \cdot \Pr_{\mathbf{e}_e' \leftarrow\!\$\, \mathcal{B}_{\eta_1}^k} [\mathbf{A}\mathbf{s} + \mathbf{e}_e' = \mathbf{t}_e^* \bmod q] \\
&= \sum_{\mathbf{s} \in \mathcal{B}_{\eta_1}^k} \Pr_{\mathbf{s}_e' \leftarrow\!\$\, \mathcal{B}_{\eta_1}^k} [\mathbf{s}_e' = \mathbf{s}] \cdot \Pr_{\mathbf{e}_e' \leftarrow\!\$\, \mathcal{B}_{\eta_1}^k} [\mathbf{e}_e' = \mathbf{t}_e^* - \mathbf{A}\mathbf{s} \bmod q] \\
&\le \sum_{\mathbf{s} \in \mathcal{B}_{\eta_1}^k} \Pr_{\mathbf{s}_e' \leftarrow\!\$\, \mathcal{B}_{\eta_1}^k} [\mathbf{s}_e' = \mathbf{s}] \cdot p_{\eta_1}^{kn} \\
&= p_{\eta_1}^{kn}.
\end{aligned}$$

Thus, for each fresh seed $\mathsf{seed}' \neq \mathsf{seed}_e^*$, the probability that it generates the same ephemeral public-key component as seed_e^* is at most $p_{\eta_1}^{kn}$. Since the adversary makes at most $q_{\mathsf{G}'}$ queries to G' in the ROM and outputs one final seed, a union bound over at most $q_{\mathsf{G}'} + 1$ candidate seeds shows that the second case occurs with probability at most $(q_{\mathsf{G}'} + 1)p_{\eta_1}^{kn}$. Combining the two cases, in the ROM we obtain

$$\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \le 2^{-256} + (q_{\mathsf{G}'} + 1)p_{\eta_1}^{kn}.$$

QROM Analysis. In the QROM, the same target-search event is bounded by the standard quantum search bound [BBBV97], replacing the second term by $O((q_{\mathsf{G}'} + 1)^2 p_{\eta_1}^{kn})$. Combining the two cases, in the QROM we obtain

$$\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \le 2^{-256} + O((q_{\mathsf{G}'} + 1)^2 p_{\eta_1}^{kn}).$$

□

Theorem C.3 (RSPK-Bind Security of AFS-KEM-ML in the ROM/QROM). *Let $q_{\mathsf{G}'}$ be the number of queries made by the adversary to the random oracle G', and let $p_{\eta_1} := \max_x \Pr[\mathrm{CBD}_{\eta_1} = x] = \frac{\binom{2\eta_1}{\eta_1}}{2^{2\eta_1}}$. For any PPT adversary $\mathcal{A}$ in the ROM, we have*

$$\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \leq (q_{\mathsf{G}'}+1)2^{-\lambda} + p_{\eta_1}^{kn}.$$

For any QPT adversary $\mathcal{A}$ making at most $q_{\mathsf{G}'}$ quantum queries to G', we have

$$\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \leq p_{\eta_1}^{kn} + \frac{3}{2}\sqrt{(q_{\mathsf{G}'}+1)2^{-\lambda}}.$$

Proof. Note that revealing seed_e to the adversary does not give the adversary any additional advantage. Therefore, we may equivalently analyze a game in which seed_e is sampled after cpk is fixed and is not given to the adversary.

First, write $\mathsf{pk}_s^* = (\mathsf{seed}_{\mathsf{pk}}^*, \mathbf{t}_s^*)$ and $\mathsf{cpk} = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_c)$, where cpk is output by the adversary. Let $\mathbf{A} = \mathsf{G}(\mathsf{seed}_{\mathsf{pk}})$, and write $(\mathbf{s}_e, \mathbf{e}_e) = \mathsf{G}'(\mathsf{seed}_e)$, where $\mathsf{seed}_e \leftarrow\!\$\ \{0,1\}^\lambda$. Since $\mathsf{EKeyGen}_c$ computes $\mathbf{t}_e = \mathbf{A}\mathbf{s}_e + \mathbf{e}_e \bmod q$, and $\mathsf{ExtractPK}$ outputs $\mathsf{pk}_s' = (\mathsf{seed}_{\mathsf{pk}}, \mathbf{t}_c - \mathbf{t}_e \bmod q)$, the condition $\mathsf{pk}_s' = \mathsf{pk}_s^*$ first implies $\mathsf{seed}_{\mathsf{pk}} = \mathsf{seed}_{\mathsf{pk}}^*$, and then implies $\mathbf{A}\mathbf{s}_e + \mathbf{e}_e = \mathbf{t}_c - \mathbf{t}_s^* \pmod q$.

We now bound the probability of this event in the ROM. Let S denote the set of inputs to G' queried by the adversary before it outputs cpk. Since seed_e is sampled uniformly after cpk has been fixed, we have $\Pr[\mathsf{seed}_e \in S] \leq (q_{\mathsf{G}'}+1)2^{-\lambda}$. Conditioned on $\mathsf{seed}_e \notin S$, the value $\mathsf{G}'(\mathsf{seed}_e)$ is fresh and independent of cpk. Thus, $(\mathbf{s}_e, \mathbf{e}_e) \leftarrow\!\$\ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$ independently of the fixed target $\mathbf{t}_c - \mathbf{t}_s^*$. As in the second case of the proof of Theorem C.2, for any fixed $\mathbf{A}$ and any fixed target $\mathbf{t}$, we have $\Pr_{(\mathbf{s}_e,\mathbf{e}_e)\leftarrow\!\$\mathcal{B}_{\eta_1}^k\times\mathcal{B}_{\eta_1}^k}[\mathbf{A}\mathbf{s}_e + \mathbf{e}_e = \mathbf{t} \bmod q] \leq p_{\eta_1}^{kn}$. Applying this bound to $\mathbf{t} = \mathbf{t}_c - \mathbf{t}_s^*$, we obtain $\Pr[\mathsf{pk}_s' = \mathsf{pk}_s^*] \leq q_{\mathsf{G}'}2^{-\lambda} + p_{\eta_1}^{kn}$. Therefore, in the ROM we have

$$\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \leq (q_{\mathsf{G}'}+1)2^{-\lambda} + p_{\eta_1}^{kn}.$$

QROM Analysis. We proceed in two steps. First, after the adversary has output cpk, we reprogram G' at the point seed_e so that $\mathsf{G}'(\mathsf{seed}_e)$ induces an independent fresh sample $(\widetilde{\mathbf{s}}_e, \widetilde{\mathbf{e}}_e) \leftarrow\!\$\ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$. By Lemma A.9, applied with $X = \{0,1\}^\lambda$, we have $|X| = 2^\lambda$. Hence, for any adversary making at most $q_{\mathsf{G}'}+1$ oracle queries, where the additional 1 accounts for the implicit G' query induced by $\mathsf{SKeyGen}(1^\lambda)$, the distinguishing advantage caused by this reprogramming is at most

$$\frac{3}{2}\sqrt{(q_{\mathsf{G}'}+1)2^{-\lambda}}.$$

Next, we bound the success probability in the modified experiment. In this experiment, the value $(\widetilde{\mathbf{s}}_e, \widetilde{\mathbf{e}}_e) \leftarrow\!\$\ \mathcal{B}_{\eta_1}^k \times \mathcal{B}_{\eta_1}^k$ is sampled independently of the adversary's already fixed target $\mathbf{t}_c - \mathbf{t}_s^*$. By the same argument as above, for any fixed $\mathbf{A}$ and any fixed target $\mathbf{t}$, we have

$$\Pr_{(\mathbf{s}_e,\mathbf{e}_e)\leftarrow\!\$\mathcal{B}_{\eta_1}^k\times\mathcal{B}_{\eta_1}^k}[\mathbf{A}\mathbf{s}_e + \mathbf{e}_e = \mathbf{t} \bmod q] \leq p_{\eta_1}^{kn}.$$

Applying this bound to $\mathbf{t} = \mathbf{t}_c - \mathbf{t}_s^*$, the adversary wins in the modified experiment with probability at most $p_{\eta_1}^{kn}$. Combining the analysis above, we obtain

$$\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \leq p_{\eta_1}^{kn} + \frac{3}{2}\sqrt{(q_{\mathsf{G}'}+1)2^{-\lambda}}.$$

□

Theorem C.4 (Key-Coll Security of AFS-KEM-ML in the ROM/QROM). *Suppose that H, G, and $\mathsf{H}_{\mathsf{rej}}$ are modeled as random oracles. Let q_{H}, q_{G}, and $q_{\mathsf{H}_{\mathsf{rej}}}$ be the number of queries made by the adversary to H, G, and $\mathsf{H}_{\mathsf{rej}}$, respectively. For any PPT adversary $\mathcal{A}$ in the ROM, we have*

$$\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \leq \frac{q_{\mathsf{H}} + q_{\mathsf{H}_{\mathsf{rej}}} + 2}{2^{256}}.$$

For any QPT adversary $\mathcal{A}$ making at most q_{H}, q_{G}, and $q_{\mathsf{H}_{\mathsf{rej}}}$ quantum queries to H, G, and $\mathsf{H}_{\mathsf{rej}}$, respectively, in the QROM, we have

$$\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) \leq \frac{(q_{\mathsf{H}}+1)^2 + (q_{\mathsf{H}_{\mathsf{rej}}}+1)^2}{2^{256}}.$$

Proof. The proof of Theorem C.4 is established by analyzing the adversary's advantage through a case-by-case discussion for $g \in \{1, 2, 3\}$.

Part $g = 1$: In this case, the experiment computes $(\mathsf{ct}', \mathsf{K}') \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}')$ for a public key $\mathsf{cpk}' \neq \mathsf{cpk}^*$ output by $\mathcal{A}$. The adversary wins only if it subsequently outputs a ciphertext ct^* such that $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}^*) = \mathsf{K}'$. Note that the key returned by $\mathsf{Decaps}_{\mathsf{AFS}}$ is derived either from the random oracle H or from the implicit rejection oracle $\mathsf{H}_{\mathsf{rej}}$. We bound the advantage of $\mathcal{A}$ by considering the following two cases.

- **Case 1: A second-preimage event for H.** In this case, the key returned by $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}^*)$ is derived from H. Since $\mathsf{cpk}' \neq \mathsf{cpk}^*$ and the input to H binds the combined public key, the H-input used in this decapsulation is distinct from the one that generated K' during $\mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}')$. Hence the adversary can win only if it finds a second preimage associated with cpk^* whose output equals the target key K'. In the ROM, this event occurs with probability at most $(q_{\mathsf{H}}+1)/2^{256}$, where the additional $+1$ accounts for the internal H-evaluation performed by the decapsulation algorithm in the experiment. In the QROM, by the standard quantum search bound [Zha19, Corollary 2], this probability is bounded by $(q_{\mathsf{H}}+1)^2/2^{256}$.
- **Case 2: A target-preimage event for $\mathsf{H}_{\mathsf{rej}}$.** In this case, the key returned by $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}^*)$ is derived from $\mathsf{H}_{\mathsf{rej}}$. Therefore, the adversary can win only if it finds an input to $\mathsf{H}_{\mathsf{rej}}$ whose output equals the target key K'. In the ROM, this event occurs with probability at most $(q_{\mathsf{H}_{\mathsf{rej}}}+1)/2^{256}$. In the QROM, by the standard quantum search bound, this probability is bounded by $(q_{\mathsf{H}_{\mathsf{rej}}}+1)^2/2^{256}$.

Part $g = 2$: In this case, the experiment first computes $(\mathsf{K}^*, \mathsf{ct}^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$. The adversary then outputs a public key cpk', and the experiment computes $(\mathsf{K}', \mathsf{ct}') \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}')$. The adversary wins only if $\mathsf{K}' = \mathsf{K}^*$. In this case, both keys are derived from the random oracle H. Since each encapsulation samples an independent $m \leftarrow\!\$\ \{0,1\}^{256}$, the two H-inputs coincide only if $\mathsf{cpk}' = \mathsf{cpk}^*$ and the same message is sampled, which occurs with probability at most 2^{-256}.

Otherwise, the two H-inputs are distinct. A win in this case yields a second-preimage event for H, since two distinct H-inputs produce the same target key K^*. In the ROM, this event occurs with probability at most $(q_{\mathsf{H}}+1)/2^{256}$. In the QROM, by the standard quantum search bound, this probability is bounded by $(q_{\mathsf{H}}+1)^2/2^{256}$.

Part $g = 3$: In this case, the experiment computes $(\mathsf{K}^*, \mathsf{ct}^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$. The adversary wins only if it outputs a ciphertext $\mathsf{ct}' \neq \mathsf{ct}^*$ such that $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}') = \mathsf{K}^*$. We bound the advantage of $\mathcal{A}$ by considering the following three cases.

- **Case 1: Coincidence of the H-inputs.** Suppose that the key returned by $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}')$ is derived from H, and that the decapsulation procedure recovers the same message as in $(\mathsf{K}^*, \mathsf{ct}^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$. Since the re-encryption randomness is deterministically derived from G using the same combined public key and message, the re-encryption check reconstructs the same ciphertext ct^*. Since the corresponding re-encryption check in $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}')$ is valid in this case, we have $\mathsf{ct}' = \mathsf{ct}^*$, contradicting the winning condition $\mathsf{ct}' \neq \mathsf{ct}^*$. Hence this case cannot occur.
- **Case 2: A second-preimage event for H.** In this case, the key returned by $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}')$ is derived from H, but the decapsulation procedure recovers a message different from the one used in $(\mathsf{K}^*, \mathsf{ct}^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$. Hence the corresponding H-input is distinct from the one that generated K^* in $\mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$. Therefore, the adversary can win only if it finds a second preimage for H whose output equals the target key K^*. In the ROM, this event occurs with probability at most $(q_{\mathsf{H}}+1)/2^{256}$, where the additional $+1$ accounts for the internal H-evaluation performed by the decapsulation algorithm in the experiment. In the QROM, by the standard quantum search bound, this probability is bounded by $(q_{\mathsf{H}}+1)^2/2^{256}$.
- **Case 3: A target-preimage event for $\mathsf{H}_{\mathsf{rej}}$.** In this case, the key returned by $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}')$ is derived from $\mathsf{H}_{\mathsf{rej}}$. Therefore, the adversary can win only if it finds an input to $\mathsf{H}_{\mathsf{rej}}$ whose output equals the target key K^*. In the ROM, this event occurs with probability at most $(q_{\mathsf{H}_{\mathsf{rej}}}+1)/2^{256}$. In the QROM, by the standard quantum search bound, this probability is bounded by $(q_{\mathsf{H}_{\mathsf{rej}}}+1)^2/2^{256}$.

Finally, the total advantage of $\mathcal{A}$ is bounded by

$$\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM\text{-}ML}}(\mathcal{A}) = \max\{\mathsf{Adv}^{g=1}_{\mathcal{A}}, \mathsf{Adv}^{g=2}_{\mathcal{A}}, \mathsf{Adv}^{g=3}_{\mathcal{A}}\}$$

□

D Appendix: Security Analysis for pUAKE

D.1 The pUAKE Protocol and Freshness Predicate $\mathsf{F}_{\mathsf{pUAKE}}$.

The pUAKE protocol is a three-round authenticated key exchange protocol that provides initiator-only authentication in the post-id setting. It is constructed from a single AFS-KEM and a pseudorandom generator (PRG). The protocol flow is depicted in Fig. 4 and proceeds as follows.

- *Round 1 (Initiator → Responder).* The initiator A samples a fresh random value seed_A and derives a corresponding ephemeral key pair $(\mathsf{pk}_e^A, \mathsf{sk}_e^A)$ using EKeyGen. It then combines the long-term and ephemeral keys via CombK to obtain a composite key pair $(\mathsf{cpk}_A, \mathsf{csk}_A)$, and sends cpk_A to the responder B.
- *Round 2 (Responder → Initiator).* Upon receiving cpk_A, the responder encapsulates using the AFS-KEM to obtain $(\mathsf{ct}_A, \mathsf{K}_A)$ and sends ct_A to the initiator. The initiator decapsulates ct_A using csk_A to obtain K'_A and derives the session key $\mathsf{K}_{\mathsf{SESSION}}$ together with an encryption key $\mathsf{K}^A_{\mathsf{ENC}}$ via a PRG. The initiator A then accepts the session key $\mathsf{K}_{\mathsf{SESSION}}$.
- *Round 3 (Initiator → Responder).* The initiator encrypts seed_A under $\mathsf{K}^A_{\mathsf{ENC}}$ by computing $\mathsf{c}^A_{\mathsf{seed}} = \mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{seed}_A$, and sends $(\mathsf{id}_A, \mathsf{c}^A_{\mathsf{seed}})$ to the responder. The responder recovers $\mathsf{seed}'_A = \mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{c}^A_{\mathsf{seed}}$ and reconstructs $\mathsf{pk}_e^{A'}$ via $\mathsf{EKeyGen}_\mathsf{c}$. It then extracts pk'_A using ExtractPK and verifies the binding between id_A and pk'_A using the certification authority (CA). If the verification succeeds, the responder accepts the session key $\mathsf{K}_{\mathsf{SESSION}}$.

Leakage through RevealEph. Our analysis of the pUAKE protocol accounts for an RevealEph query that may reveal the following information. For an initiator session, the query reveals the randomness seed_A used in EKeyGen, which implies knowledge of the ephemeral secret key sk_e^A. For a responder session, no information is revealed. In particular, we assume that the randomness used for encapsulation is securely erased immediately after its computation.

Definition D.1 (Freshness Predicate for pUAKE). *Using the notion in Sec. 4.3, we define the freshness predicate F_{pUAKE} as the following five cases, each corresponding to an admissible adversarial capability pattern. All pairs of cases are disjoint by construction. In the subsequent security analysis, the case FU_5 will be handled by showing that the event FU_5 occurs with negligible probability, whereas for FU_1–FU_4 we show that the adversary's advantage in winning the key secrecy game is negligible.*

$$
\begin{aligned}
\mathsf{FU}_1 &\triangleq \mathsf{AisInit} \wedge \overline{\mathsf{CorI}} \wedge \mathsf{FirstMatch} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FU}_2 &\triangleq \mathsf{AisResp} \wedge \overline{\mathsf{CorI}} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FU}_3 &\triangleq \mathsf{AisInit} \wedge \mathsf{CorI} \wedge \overline{\mathsf{EphT}} \wedge \mathsf{FirstMatch} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FU}_4 &\triangleq \mathsf{AisResp} \wedge \mathsf{CorI} \wedge \mathsf{FirstMatch} \wedge \overline{\mathsf{EphFirstMatch}} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}},\\
\mathsf{FU}_5 &\triangleq \mathsf{AisResp} \wedge \mathsf{CorIafterT} \wedge \overline{\mathsf{FirstMatch}} \wedge \overline{\mathsf{SkT}} \wedge \overline{\mathsf{SkM}}.
\end{aligned}
$$

We define

$$
\mathsf{F}_{\mathsf{pUAKE}} \triangleq \bigvee_{i=1}^{5} \mathsf{FU}_i.
$$

D.2 Key Secrecy of the pUAKE protocols

Theorem D.2 ($\mathsf{F}_{\mathsf{pUAKE}}$-Key Secrecy of pUAKE in the ROM). *Let there be at most N parties and at most S sessions activated in the $\mathsf{F}_{\mathsf{pUAKE}}$-KS game. Suppose that the underlying AFS-KEM is δ-correct, satisfies γ-spreadness, and satisfies perfect ExtractPK-correctness and ephemeral-key-generation equivalence. We model the PRG as a random oracle G, to which the adversary makes at most q_{G} queries. Then, for any PPT adversary $\mathcal{A}$ against the $\mathsf{F}_{\mathsf{pUAKE}}$-KS security of pUAKE, there exist PPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{C}, \mathcal{D}, \mathcal{F}$, and $\mathcal{Z}$, constructed below and running in time comparable*

to that of $\mathcal{A}$, such that

$$\mathsf{Adv}^{\mathsf{F_{pUAKE}\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} + NS\left(\begin{array}{l} 4\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + 2\,\mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) \\ + S\left(\begin{array}{l} 2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + 4\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) \\ + 4S\,\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + 4(N+1)\,\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) \\ + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + \dfrac{2q_{\mathsf{G}}}{|\mathcal{K}|} + 5\delta \end{array}\right) \end{array}\right).$$

where k_1 and k_2 are min-entropy bounds on honest static and combined public keys of AFS-KEM*, respectively, and $|\mathcal{K}|$ is the cardinality of the* AFS-KEM *encapsulated key space.*

Proof. First, we assume that all honest initiators have distinct long-term public keys, and that all honest initiator sessions compute distinct cpk_A. Moreover, we assume that every responder session that receives an honest cpk_A from some honest initiator session, computes a distinct ct_A. Since the min-entropy of the public key pk_A is at least k_1, the honestly combined public key cpk_A has min-entropy at least k_2, and the ciphertexts of AFS-KEM satisfy γ-spreadness, this will add $\frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}}$ to the final bound. Under these assumptions, it is impossible for a session to have more than one FirstMatch or Match session.

We assume that the tested session satisfies the $\mathsf{F_{pUAKE}}$ throughout the experiment, as otherwise the adversary loses the indistinguishability game. To bound $\mathsf{Adv}^{\mathsf{F_{pUAKE}\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A})$, we partition the event that the adversary wins into five cases according to the definition $\mathsf{F_{pUAKE}} = \bigvee_{i=1}^{5} \mathsf{FU}_i$ (Definition D.1), which yields

$$\mathrm{Adv}^{\mathsf{F_{pUAKE}\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le \sum_{i=1}^{5} \mathrm{Adv}^{\mathsf{FU}_i\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}.$$

We now give security reductions for each subcase $\mathsf{Adv}^{\mathsf{FU}_i\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$ with $i \in \{1,2,3,4,5\}$, where throughout we assume that the tested session satisfies FU_i. Otherwise the adversary trivially loses the corresponding indistinguishability game. The corresponding reductions are stated in Lemmas D.3, D.4, and D.5. For readability, we defer the proofs of these lemmas to Subsection D.3.

Lemma D.3. *With the notations and assumptions of the proof of Theorem D.2, there exist adversaries $\mathcal{B}_1$, $\mathcal{C}$, and $\mathcal{Z}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FU_1\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + (N+1)\cdot\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + \delta\right)$$

Lemma D.4. *With the notations and assumptions of the proof of Theorem D.2, there exist adversaries $\mathcal{B}_2$, $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{Z}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FU_2\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + (N+1)S\cdot\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + S\delta\right)$$

$$\mathsf{Adv}^{\mathsf{FU_5\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + S\delta\right)$$

Lemma D.5. *With the notations and assumptions of the proof of Theorem D.2, and with* PRG *modeled as a random oracle* G *to which the adversary makes at most q_{G} queries, there exist adversaries $\mathcal{B}_3$, $\mathcal{C}$, and $\mathcal{F}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FU_3\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) &\le NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|} + \delta\right) \\ &\quad + NS^2\left(2S\cdot\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + (N+1)\cdot\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})\right).\end{aligned}$$

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FU_4\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) &\le NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|} + \delta\right) \\ &\quad + NS^2\left(2S\cdot\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + (N+1)\cdot\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})\right).\end{aligned}$$

Combining all the above lemmas completes the proof. □

Theorem D.6 ($\mathsf{F}_{\mathsf{pUAKE}}$-Key Secrecy of pUAKE in the QROM). *Let there be at most N parties and at most S sessions activated in the $\mathsf{F}_{\mathsf{pUAKE}}$-KS game. Suppose that the underlying AFS-KEM is δ-correct, satisfies γ-spreadness, and satisfies perfect ExtractPK-correctness and ephemeral-key-generation equivalence. We model the PRG as a quantum-accessible random oracle G, to which the adversary makes at most q_{G} (quantum) queries. Then, for any QPT adversary $\mathcal{A}$ attacking the $\mathsf{F}_{\mathsf{pUAKE}}$-KS security of pUAKE, there exist QPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{C}, \mathcal{D}, \mathcal{F}$, and $\mathcal{Z}$, constructed below and running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{F}_{\mathsf{pUAKE}}\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le{}& \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} \\ &+ NS\left(4\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{LCK}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{RSPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{D}) + 2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{PRG}}_{\mathsf{PRG}}(\mathcal{Z})\right. \\ &\left.+ S\left(\begin{array}{l} 2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + 5\delta \\ + 4S\,\mathsf{Adv}^{\mathsf{Key}\text{-}\mathsf{Coll}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{F}) + 4(N+1)\,\mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C}) \\ + 4\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{ECPA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_3) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}\right)} \end{array}\right)\right).\end{aligned}$$

where k_1 and k_2 are the min-entropy bounds on the honest static and combined public keys of AFS-KEM, respectively. Furthermore, $|\mathcal{K}|$ denotes the cardinality of the AFS-KEM encapsulated key space.

Proof. This proof follows the same argument as that of Theorem D.2, except for Lemma D.5. Lemma D.5 is the only lemma that relies on the random oracle model, while all other cases are analyzed in the standard model. Below, we extend Lemma D.5 to the QROM setting. For readability, the proof of Lemma D.7 is deferred to Subsection D.3.

Lemma D.7 (QROM counterpart of Lemma D.5). *With the notations and assumptions of the proof of Theorem D.6, and with PRG modeled as a quantum-accessible random oracle G to which the adversary makes at most q_{G} (quantum) queries, there exist QPT adversaries $\mathcal{B}_3$, $\mathcal{C}$, and $\mathcal{D}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FU}_3\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS^2\left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{ECPA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_3) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}\right)} + \beta\right),$$

$$\mathsf{Adv}^{\mathsf{FU}_4\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS^2\left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{ECPA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_3) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}\right)} + \beta\right),$$

where $\beta = 2S \cdot \mathsf{Adv}^{\mathsf{Key}\text{-}\mathsf{Coll}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{F}) + (N+1) \cdot \mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C}) + \delta$.

Combining all the above analyses completes the proof. □

D.3 Full Analysis for Lemmas D.3, D.4, D.5, and D.7

Lemma D.3. *With the notations and assumptions of the proof of Theorem D.2, there exist adversaries $\mathcal{B}_1$, $\mathcal{C}$, and $\mathcal{Z}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FU}_1\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_1) + (N+1)\cdot\mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + \delta\right)$$

Proof. We assume that the tested session satisfies FU_1, otherwise the adversary trivially loses the game.

Game G_0. This game corresponds to the original game for $\mathsf{Adv}^{\mathsf{FU}_1\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$, that is,

$$\mathsf{Adv}^{\mathsf{FU}_1\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1 (Guessing $\pi^{\mathsf{Test}}.\mathsf{actor}$, π^{Test}, and $\pi^{\mathsf{FirstMatch}}$). In game G_1, we first guess the actor identity of the test session, and further guess the test session π^{Test} and its corresponding (unique) FirstMatch session $\pi^{\mathsf{FirstMatch}}$. If any of the guesses is incorrect, the game is aborted. Concretely, we sample P^* uniformly from the N parties, and sample two session instances π, π' uniformly from the S sessions, respectively. If any of the guesses is incorrect (i.e., $P^* \neq \pi^{\mathsf{Test}}.\mathsf{actor}$, or $\pi \neq \pi^{\mathsf{Test}}$, or $\pi' \neq \pi^{\mathsf{FirstMatch}}$), the game is aborted. The probability of guessing correctly is $1/(NS^2)$, and therefore

$$\mathsf{Adv}^{G_0}(\mathcal{A}) \leq NS^2 \mathsf{Adv}^{G_1}(\mathcal{A}).$$

Game G_2 (Random K_A). In this game, we replace the KEM key K_A in the tested session π and its (unique) FirstMatch session π' with an independently uniform key $\tilde{k} \leftarrow\!\!\$\, \mathcal{K}$. For any adversary $\mathcal{A}$ that can distinguish G_1 from G_2, we construct an adversary $\mathcal{B}_1$ against the $\mathsf{IND\text{-}SCCA}$ security for $\mathsf{AFS\text{-}KEM}$. Adversary $\mathcal{B}_1$ interacts with its $\mathsf{AFS\text{-}KEM}$ challenger and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{seed}_e^*, \mathsf{ct}^*, \mathsf{K}_b^*)$, where K_b^* equals the real encapsulated key if $b = 0$ and a uniform random key if $b = 1$. It then simulates the view of $\mathcal{A}$ as in G_2, except for the following changes.

- $\mathcal{B}_1$ sets pk_s^* as the long-term public key of P^*, and generates long-term key pairs $(\mathsf{pk}_U, \mathsf{sk}_U)$ honestly for all other parties. It embeds $(\mathsf{cpk}^*, \mathsf{ct}^*)$ into the tested session π and its corresponding FirstMatch session π', and uses K_b^* as the KEM key in both sessions. Using seed_e^* and K_b^*, $\mathcal{B}_1$ can simulate both sessions according to the protocol specification.
- For each non-tested session $\hat{\pi} \neq \pi$ of P^* with $\hat{\pi}.\mathsf{role} = \mathsf{Init}$, $\mathcal{B}_1$ independently queries $\mathcal{O}^{\mathsf{EKeyGen}()}$ to obtain a fresh tuple $(\mathsf{cpk}, \mathsf{pk}_e, \mathsf{sk}_e, \mathsf{seed})$ for the session and sets the first protocol message to be cpk. Upon receiving the corresponding KEM ciphertext ct for $\hat{\pi}$, $\mathcal{B}_1$ queries $\mathcal{O}^{\mathsf{Dec}}(\mathsf{cpk}, \mathsf{ct})$ to obtain the KEM key K_A and continues the simulation of that session honestly. Notably, we have $\mathsf{cpk} \neq \mathsf{cpk}^*$ by the assumption that all honestly generated cpk_A are distinct.

When $b = 0$, $\mathcal{B}_1$ perfectly simulates G_1, as long as the KEM ciphertext received by π is always correctly decapsulated. In G_1, π performs the decapsulation of the KEM ciphertext, which could potentially differ from the encapsulated key. In the simulation, $\mathcal{B}_1$ always uses the real encapsulated key. This difference is quantified by the δ-correctness of the underlying $\mathsf{AFS\text{-}KEM}$. That is, $|\Pr[\mathcal{B}_1 \Rightarrow 1 : b = 0] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]| \leq \delta$. When $b = 1$, $\mathcal{B}_1$ perfectly simulates G_2, that is $\Pr[\mathcal{B}_1 \Rightarrow 1 : b = 1] = \Pr[G_2^{\mathcal{A}} \Rightarrow 1]$. Therefore, we have

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + \delta.$$

Game G_3 (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$). In this game, we replace the session keys $\mathsf{K}_{\mathsf{SESSION}}$ in π and π' with the same independently sampled uniform key $\hat{\mathsf{K}}_{\mathsf{SESSION}}$. Moreover, we replace the keys $\mathsf{K}^A_{\mathsf{ENC}}$ in π and π' with the same independently sampled uniform key $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$. For any adversary that can distinguish G_2 from G_3, we construct an adversary $\mathcal{Z}$ against the $\mathsf{IND\text{-}PRG}$ security of the PRG as follows. $\mathcal{Z}$ receives a value y_b from its challenger. If $b = 0$, then $y_b = \mathsf{PRG}(k)$, where $k \leftarrow\!\!\$\, \mathcal{K}$. If $b = 1$, then y_b is uniformly chosen from the output space of the PRG. Adversary $\mathcal{Z}$ then perfectly simulates the game for $\mathcal{A}$ as G_2, except that it uses y_b as the pair $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}})$ in the sessions π and π'. Note that when $b = 0$, the simulation is identical to G_2, and when $b = 1$, it is identical to G_3. That is $\Pr[\mathcal{Z} \Rightarrow 1 : b = 0] = \Pr[G_2^{\mathcal{A}} \Rightarrow 1]$ and $\Pr[\mathcal{Z} \Rightarrow 1 : b = 1] = \Pr[G_3^{\mathcal{A}} \Rightarrow 1]$. Therefore, we have

$$\left|\mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}).$$

Game G_4 (Peer acceptance implies a matching session). In this game, we abort if π' accepts, but π' is not the matching session of π, which we define as the BAD event. Since games G_3 and G_4 are identical unless the BAD event occurs, we have

$$\left|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})\right| \leq \Pr[\mathsf{BAD} : G_3].$$

Next, we analyze $\Pr[\mathsf{BAD} : G_3]$. If BAD occurs in G_3, then the adversary has tampered with the third message flow between π and π'. Specifically, the pair $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$ sent by π differs from $(\mathsf{id}', \mathsf{c}'_{\mathsf{seed}})$ received by π'. We consider the following two cases:

- **Case 1:** If $\mathsf{id}_{P^*} \neq \mathsf{id}'$, we construct an adversary $\mathcal{C}$ against the $\mathsf{SPK\text{-}Bind}$ security of $\mathsf{AFS\text{-}KEM}$ as follows. $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates

Game G_3 for $\mathcal{A}$, except that it first samples B^* uniformly from among the N parties and aborts if $B^* \neq \mathsf{id}'$. Notably, $\mathcal{C}$ guesses correctly with probability $1/N$. Then $\mathcal{C}$ sets $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$ and $\mathsf{pk}_{B^*} = \overline{\mathsf{pk}}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π. When π' receives a pair $(\mathsf{id}', \mathsf{c}'_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'_{\mathsf{seed}}$ and outputs the corresponding seed'_e to its challenger. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}', \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{\mathsf{id}'} = \overline{\mathsf{pk}}_s^*$, we conclude that $\mathsf{pk}'_s = \overline{\mathsf{pk}}_s^*$ by the security of CA. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 1}] \leq N \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

– **Case 2:** If $\mathsf{id}_{P^*} = \mathsf{id}'$ and $\mathsf{c}_{\mathsf{seed}} \neq \mathsf{c}'_{\mathsf{seed}}$, we also construct an adversary $\mathcal{C}$[21] against the $\mathsf{SPK\text{-}Bind}$ security of $\mathsf{AFS\text{-}KEM}$ as follows. $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_3 for $\mathcal{A}$, except that $\mathcal{C}$ sets $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π. When π' receives a valid pair $(\mathsf{id}_{P^*}, \mathsf{c}'_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'_{\mathsf{seed}}$ and outputs the corresponding seed'_e to its challenger. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{c}'_{\mathsf{seed}} \neq \mathsf{c}_{\mathsf{seed}}$, we have $\mathsf{pk}'_s = \mathsf{pk}_s^*$ and $\mathsf{seed}'_e \neq \mathsf{seed}_e^*$. Therefore, if the BAD event occurs in this case, then $\mathcal{C}$ also wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2}] \leq \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Combining the two cases, we have

$$\left|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})\right| \leq (N+1) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Analysis of G_4. In G_4, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly sampled, and there may be one accepted session (which must be the matching session) that uses the same session key $\mathsf{K}_{\mathsf{SESSION}}$. Note that all session keys in other accepted sessions are independent of $\mathsf{K}_{\mathsf{SESSION}}$ right now. Since we do not allow the adversary to reveal the session key in the matching session of π, we have

$$\mathsf{Adv}^{G_4}(\mathcal{A}) = 0.$$

Combining all the bounds derived above completes the proof. □

Lemma D.4. *With the notations and assumptions of the proof of Theorem D.2, there exist adversaries $\mathcal{B}_2$, $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{Z}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FU_2\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \leq NS\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + (N+1)S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + S\delta\right)$$

$$\mathsf{Adv}^{\mathsf{FU_5\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \leq NS\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + S\delta\right)$$

Proof. In this lemma, we analyze both $\mathsf{Adv}^{\mathsf{FU_2\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{FU_5\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A})$, since their proofs follow a similar structure. We first present a complete proof for Case $\mathsf{FU_2}$, and then show how to adapt the argument to Case $\mathsf{FU_5}$.

Game G_0. This game corresponds to the original game for $\mathsf{Adv}^{\mathsf{FU_2\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A})$, that is,

$$\mathsf{Adv}^{\mathsf{FU_2\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1 (Guessing $\pi^{\mathsf{Test}}.\mathsf{peer}$, π^{Test}). In game G_1, we first guess the peer identity of the test session, and further guess the test session π^{Test}. If either guess is incorrect, the game is aborted. Concretely, we sample P^* uniformly from the N parties, and sample one session instance π uniformly from the at most S sessions. If the guess is incorrect (i.e., $P^* \neq \pi^{\mathsf{Test}}.\mathsf{peer}$ or $\pi \neq \pi^{\mathsf{Test}}$), the game is aborted. The probability of guessing correctly is $1/(NS)$, therefore

$$\mathsf{Adv}^{G_0}(\mathcal{A}) \leq NS \cdot \mathsf{Adv}^{G_1}(\mathcal{A}).$$

Game G_2 (Random K_A). In this game, we replace the KEM key K_A in the tested session π with an independently uniform key $\tilde{k} \leftarrow\!\!\$\, \mathcal{K}$. If the unique FirstMatch session π' of π exists, we also replace the

[21] We reuse the notation $\mathcal{C}$ for simplicity.

K_A in π' with the same key $\tilde{k}$. For any adversary $\mathcal{A}$ that can distinguish G_1 from G_2, we construct an adversary $\mathcal{B}_2$ against the IND-LCK-SCCA security of AFS-KEM as follows. Adversary $\mathcal{B}_2$ first receives pk_s^* from its challenger, then sets pk_s^* as the long-term public key of P^*, and generates long-term key pairs $(\mathsf{pk}_U, \mathsf{sk}_U)$ honestly for all other parties. It then simulates the view of $\mathcal{A}$ as in G_2, except for the following changes.

- For each session $\hat{\pi}$ of P^* with $\hat{\pi}.\mathsf{role} = \mathsf{Init}$, $\mathcal{B}_2$ independently queries $\mathcal{O}^{\mathsf{EKeyGen}}$ to obtain a fresh tuple $(\mathsf{cpk}, \mathsf{pk}_e, \mathsf{sk}_e, \mathsf{seed})$ for the session and sets the first message of the session to be cpk. When π receives its cpk', $\mathcal{B}_2$ sets $\mathsf{cpk}^* = \mathsf{cpk}'$ and forwards it to its challenger. After $\mathcal{B}_2$ receives $(\mathsf{pk}_s^*, \mathsf{ct}^*, \mathsf{K}_b^*)$ from its challenger, $\mathcal{B}_2$ embeds ct^* into the π and uses K_b^* as the KEM key K_A in π. If there exists a session of P^* that is the FirstMatch session of π and we denote this session as π', $\mathcal{B}_2$ also sets K_b^* as the KEM key K_A in π'. Moreover, upon P^* receiving the corresponding KEM ciphertext ct for $\hat{\pi} \neq \pi'$ with $\hat{\pi}.\mathsf{role} = \mathsf{Init}$, $\mathcal{B}_2$ queries the decryption oracle $\mathcal{O}^{\mathsf{Dec}}(\mathsf{cpk}, \mathsf{ct})$, where cpk is the composite public key embedded into session $\hat{\pi}$ via $\mathcal{O}^{\mathsf{EKeyGen}}$, to obtain the KEM key K_A. It then continues the simulation of session $\hat{\pi}$ honestly.
- When π receives $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}}^{P^*})$, $\mathcal{B}_2$ decrypts $\mathsf{c}_{\mathsf{seed}}^{P^*}$ using K_b^* to obtain seed_{P^*} and outputs seed_{P^*} to its challenger. Since π is an accepted session, we have $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_{P^*}) = 1$, where $\mathsf{pk}'_{P^*} \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{P^*})$ and $(\mathsf{pk}_e^{P^*}, \cdot) = \mathsf{EKeyGen}_{\mathsf{c}}(\mathsf{cpk}^*, \mathsf{seed}_{P^*})$. Therefore, we have $\mathsf{pk}'_{P^*} = \mathsf{pk}_s^*$ and seed_{P^*} is a valid output for $\mathcal{B}_2$.

When $b = 0$, adversary $\mathcal{B}_2$ perfectly simulates Game G_1, except for the case where the ciphertext received by π' (if it exists) is not correctly decapsulated. In G_1, and if π' exists, π' performs the decapsulation of the ciphertext, which could potentially differ from the encapsulated key. In the simulation, $\mathcal{B}_2$ always uses the real corresponding encapsulated key in π'. This difference can be bounded by the δ-correctness of the underlying AFS-KEM. Specifically, we embed the correctness-failure experiment of AFS-KEM into the guessed session π' and the corresponding session π, where π' is guessed uniformly from among the S sessions with success probability $1/S$. Conditioned on the event that the guess is correct, the simulation is perfect, and whenever the above discrepancy occurs, the embedded correctness-failure experiment outputs 1. That is, $|\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 0] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]| \leq S\delta$. When $b = 1$, $\mathcal{B}_2$ perfectly simulates G_2, that is, $\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 1] = \Pr[G_2^{\mathcal{A}} \Rightarrow 1]$. Therefore, we have

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + S\delta.$$

Game G_3 (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$). In this game, we replace the keys $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ in π with independently sampled uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$ and $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$. If the unique FirstMatch session π' of π exists, we also replace the $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ in π' with the same keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$ and $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$. For any PPT adversary that can distinguish G_2 from G_3, we construct an adversary $\mathcal{Z}$ against the IND-PRG security of the PRG as follows. $\mathcal{Z}$ receives a value y_b from its challenger. If $b = 0$, then $y_b = \mathsf{PRG}(k)$, where k is uniformly chosen from $\mathcal{K}$. If $b = 1$, then y_b is uniformly chosen from the output space of the PRG. Adversary $\mathcal{Z}$ then perfectly simulates G_2 for $\mathcal{A}$, except that it uses y_b as the pair $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}})$ in the sessions π and π' (if it exists). Note that when $b = 0$, the simulation is identical to G_2, and when $b = 1$, it is identical to G_3. Therefore, we have

$$\left|\mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}).$$

Game G_4 (Peer acceptance implies a matching session). In this game, we abort if the unique FirstMatch session π' of π exists and π' accepts, but π' is not the matching session of π; we denote this event by BAD. Since games G_3 and G_4 are identical unless the BAD event occurs, we have

$$\left|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})\right| \leq \Pr[\mathsf{BAD} : G_3].$$

Next, we analyze $\Pr[\mathsf{BAD} : G_3]$. Note that if the BAD event occurs, then the adversary has tampered with the third message sent by π'. Specifically, the pair $(\mathsf{id}', \mathsf{c}'_{\mathsf{seed}})$ sent by π' differs from $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$ received by π. We consider the following two cases:

- **Case 1:** If $\mathsf{id}_{P^*} \neq \mathsf{id}'$, we construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_3 for $\mathcal{A}$, except that it first samples B^* uniformly from among the N parties and π' uniformly

from among the S sessions, aborting if $B^* \neq \mathsf{id}'$ or $\pi' \neq \pi^{\mathsf{FirstMatch}}$. Thus, $\mathcal{C}$ guesses correctly with probability $1/(NS)$. Then $\mathcal{C}$ sets $\mathsf{pk}_{B^*} = \mathsf{pk}_s^*$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π'. When π receives a valid pair $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}_{\mathsf{seed}}$ and outputs the corresponding seed_e to its challenger. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \underline{\mathsf{EKeyGen}}_c(\mathsf{cpk}^*, \mathsf{seed}_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}_s^*$, we have $\mathsf{pk}'_s = \overline{\mathsf{pk}}_s^*$ by the security of CA. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 1}] \leq NS \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

– **Case 2:** If $\mathsf{id}_{P^*} = \mathsf{id}'$ and $\mathsf{c}_{\mathsf{seed}} \neq \mathsf{c}'_{\mathsf{seed}}$, we also construct an adversary $\mathcal{C}$[22] against the $\mathsf{SPK\text{-}Bind}$ security of $\mathsf{AFS\text{-}KEM}$ as follows. $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_3 for $\mathcal{A}$, except that it first samples π' uniformly from among the S sessions, aborting if $\pi' \neq \pi^{\mathsf{FirstMatch}}$. Thus, $\mathcal{C}$ guesses correctly with probability $1/S$. It then sets $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π'. When π receives a valid pair $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}_{\mathsf{seed}}$ and outputs the corresponding seed_e to its challenger. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{c}_{\mathsf{seed}} \neq \mathsf{c}'_{\mathsf{seed}}$, we have $\mathsf{pk}'_s = \mathsf{pk}_s^*$ and $\mathsf{seed}_e \neq \mathsf{seed}_e^*$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ also wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2}] \leq S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Combining the two cases, we have

$$\left|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})\right| \leq (N+1)S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Analysis of G_4. In G_4, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random for $\mathcal{A}$, and there may be one accepted session (which must be the matching session) that uses the same session key $\mathsf{K}_{\mathsf{SESSION}}$ and we do not allow the adversary to reveal this session key for the matching session of π. Note that all session keys in other accepted sessions are independent of $\mathsf{K}_{\mathsf{SESSION}}$. Therefore, we have

$$\mathsf{Adv}^{G_4}(\mathcal{A}) = 0.$$

Combining all the bounds derived above completes the proof for case FU_2.

Proof for Case FU_5. In this case, we show that the event FU_5 occurs only with negligible probability. Notably, the Init-Explicit Authentication analysis for pUAKE is essentially identical to the analysis of FU_5, except for Game G'_5. Here, in G'_5 we abort if the $\mathsf{FirstMatch}$ session of the tested session does not exist. In contrast, in the Init-Explicit Authentication analysis (Theorem D.8), we abort if the corresponding Match session does not exist.

Game G'_0. This game corresponds to the original game for $\mathsf{Adv}^{\mathsf{FU_5\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A})$, that is,

$$\mathsf{Adv}^{\mathsf{FU_5\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) = \mathsf{Adv}^{G'_0}(\mathcal{A}).$$

Game G'_1 (Guessing $\pi^{\mathsf{Test}}.\mathsf{peer}$, π^{Test}). This is the same game as G_1 for FU_2. That is, we sample P^* uniformly from the N parties, and sample one session instance π uniformly from the at most S sessions. If the guess is incorrect (i.e., $P^* \neq \pi^{\mathsf{Test}}.\mathsf{peer}$ or $\pi \neq \pi^{\mathsf{Test}}$), the game is aborted. The probability of guessing correctly is $1/(NS)$; therefore,

$$\mathsf{Adv}^{G'_0}(\mathcal{A}) \leq NS \cdot \mathsf{Adv}^{G'_1}(\mathcal{A}).$$

Game G'_2 (New win condition). In this game, we change the condition for the adversary to win. An instance of G'_2 is won by the adversary $\mathcal{A}$ if and only if π^{Test} accepts and nothing makes the instance of the game invalid (as per the conditions of G'_1). We stop the game at the moment when π^{Test} accepts. Note that in G'_2, the Test query is not requested for π^{Test}, and in all valid instances, P^* remains uncorrupted. For G'_1 to be won, any instance of G'_1 needs to satisfy the win conditions of G'_2. Thus, we have

$$\mathsf{Adv}^{G'_1}(\mathcal{A}) \leq \mathsf{Adv}^{G'_2}(\mathcal{A}).$$

[22] We reuse the notation $\mathcal{C}$ for simplicity.

Game G'_3 (Random K_A). In this game, we replace the KEM key K_A in the tested session π with an independently uniform key $\tilde{k} \leftarrow\!\$\, \mathcal{K}$. If the unique FirstMatch session π' of π exists, we also replace the K_A in π' with the same key $\tilde{k} \leftarrow\!\$\, \mathcal{K}$. To study the difference in advantage between G'_2 and G'_3, we use the same argument as in Game G_2 for the case of FU_2. This argument shows that we can construct an adversary $\mathcal{B}_2$ against the IND-LCK-SCCA security of AFS-KEM such that

$$\left|\mathsf{Adv}^{G'_2}(\mathcal{A}) - \mathsf{Adv}^{G'_3}(\mathcal{A})\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + S\delta.$$

Game G'_4 (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$). In this game, we replace the keys $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ in π with independently sampled uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$ and $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$. If the unique FirstMatch session π' of π exists, we also replace the $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ in π' with the same keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$ and $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$. To study the difference in advantage between G'_3 and G'_4, we use the same argument as in Game G_3 for the case of FU_2. Thus, we can construct an adversary $\mathcal{Z}$ against the IND-PRG security of the PRG such that

$$\left|\mathsf{Adv}^{G'_3}(\mathcal{A}) - \mathsf{Adv}^{G'_4}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}).$$

Game G'_5 (FirstMatch session). In this game, we define the event that the FirstMatch session $\pi^{\mathsf{FirstMatch}}$ of the test session π does not exist as the BAD event, and abort if BAD occurs. Note that whenever the FU_5 event occurs in this game, the game aborts. Therefore,

$$\mathsf{Adv}^{G'_5}(\mathcal{A}) = 0.$$

It is evident that $|\mathsf{Adv}^{G'_4}(\mathcal{A}) - \mathsf{Adv}^{G'_5}(\mathcal{A})| \leq \Pr[\mathsf{BAD} : G'_4]$. Next, we analyze $\Pr[\mathsf{BAD} : G'_4]$. In Game G'_4, the key $\mathsf{K}^A_{\mathsf{ENC}}$ of the tested session π is uniformly random from the adversary's view. Since the test session π accepts without any FirstMatch session, the key $\mathsf{K}^A_{\mathsf{ENC}}$ is never used in any other session. Therefore, $\mathsf{seed}'_A = \mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{c}^A_{\mathsf{seed}}$ is uniformly random from the adversary's view. In this case, we can construct an adversary $\mathcal{D}$ against the RSPK-Bind security of AFS-KEM as follows. Specifically, $\mathcal{D}$ receives $(\mathsf{pk}^*_s, \mathsf{sk}^*_s)$ and perfectly simulates the game for $\mathcal{A}$ as in G'_4, except that it sets $\mathsf{pk}_{P^*} = \mathsf{pk}^*_s$ and outputs the composite key cpk_A in session π to its challenger. After $\mathcal{D}$ receives seed_e, it defines $\mathsf{K}^A_{\mathsf{ENC}} = \mathsf{seed}_e \oplus \mathsf{c}^A_{\mathsf{seed}}$ in session π, where $\mathsf{c}^A_{\mathsf{seed}}$ is received in π. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) = \mathsf{EKeyGen}_c(\mathsf{cpk}_A, \mathsf{seed}_e)$ and $\mathsf{pk}'_s = \mathsf{ExtractPK}(\mathsf{cpk}_A, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{P^*} = \mathsf{pk}^*_s$, we have $\mathsf{pk}'_s = \mathsf{pk}^*_s$ by the security of CA. Therefore, if the BAD event occurs in this case, then $\mathcal{D}$ wins the RSPK-Bind game. Consequently, we have $\Pr[\mathsf{BAD} : G'_4] \leq \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})$ and

$$\left|\mathsf{Adv}^{G'_4}(\mathcal{A}) - \mathsf{Adv}^{G'_5}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}).$$

Combining all the bounds derived above completes the proof for case FU_5. □

Lemma D.5. *With the notations and assumptions of the proof of Theorem D.2, and with* PRG *modeled as a random oracle* G *to which the adversary makes at most q_{G} queries, there exist adversaries $\mathcal{B}_3$, $\mathcal{C}$, and $\mathcal{F}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{FU_3\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) &\leq NS^2\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|} + \delta\Big) \\
&\quad + NS^2\Big(2S \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + (N+1) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})\Big). \\
\mathsf{Adv}^{\mathsf{FU_4\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) &\leq NS^2\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|} + \delta\Big) \\
&\quad + NS^2\Big(2S \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + (N+1) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})\Big).
\end{aligned}$$

Proof. We present a unified analysis for Case FU_3 and Case FU_4 by considering $\mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$. It follows that $\mathsf{Adv}^{\mathsf{FU_3\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \leq \mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{FU_4\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \leq \mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$.

Game G_0. This game corresponds to the original game for $\mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$. We use the π, π' to denote the tested session and its unique FirstMatch session, respectively. Moreover, we denote by π^{Init} (resp. π^{Resp}) the initiator (resp. responder) session among π and π'.Thus,

$$\mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1 (KEM-Correctness). In this game, we abort if the decapsulation of ct_A in π^{Init} does not equal the corresponding encapsulated key K_A in π^{Resp}. This abort event can be bounded by the δ-correctness of AFS-KEM. Specifically, the reduction first guesses π^{Init} and π^{Resp} uniformly from among the S sessions, and guesses $\pi^{\mathsf{Init}}.\mathsf{actor}$ uniformly from among the N parties. Notably, the guess is correct with probability $1/(NS^2)$. Conditioned on the event that the guess is correct, the reduction can perfectly embed the correctness-failure experiment of AFS-KEM into the guessed sessions π^{Init} and π^{Resp}. Whenever the abort event occurs, the embedded correctness-failure experiment outputs 1. By the δ-correctness of AFS-KEM, we have

$$\left|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})\right| \leq NS^2\delta.$$

Game G_2. In this game, for notational convenience, let K^{Test} denote the encapsulated key K_A used in session π. We define the COLL event as the existence of a session $\hat{\pi} \notin \{\pi, \pi'\}$ such that $\hat{\pi}.\mathsf{K}_A = K^{\mathsf{Test}}$. If the COLL event occurs, we abort the game. Since games G_1 and G_2 are identical unless the COLL event occurs, we have

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \leq \Pr[\mathsf{COLL} : G_1].$$

Next we analysis the $\Pr[\mathsf{COLL} : G_1]$ by considering the following two cases:

- **Case 1:** $\hat{\pi}.\mathsf{role} = \mathsf{Init}$. We construct an adversary $\mathcal{F}$ against the Key-Coll game for AFS-KEM as follows. Adversary $\mathcal{F}$ first chooses $g = 1$ and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \mathsf{seed}_e^*)$ from its challenger. Adversary $\mathcal{F}$ guesses the sessions $\hat{\pi}$, π, π' and the party $\hat{\pi}.\mathsf{actor}$ by sampling $\hat{\pi}$, π, π' uniformly from the S sessions and sampling $\hat{\pi}.\mathsf{actor}$ uniformly from the N parties. Notably, the probability that $\mathcal{F}$ guesses correctly is at least $1/(NS^3)$.
  Adversary $\mathcal{F}$ then perfectly simulates Game G_1 for $\mathcal{A}$, except that it sets $(\mathsf{pk}_s^*, \mathsf{sk}_s^*)$ as the long-term key pair for $\hat{\pi}.\mathsf{actor}$ and embeds $(\mathsf{cpk}^*, \mathsf{seed}_e^*)$ into the session $\hat{\pi}$. Then, $\mathcal{F}$ forwards the cpk in the session π and π' to its challenger. After receiving the corresponding (ct', K') from its challenger, $\mathcal{F}$ continues the simulation by embedding $(\mathsf{ct}_A = \mathsf{ct}', \mathsf{K}_A = K')$ into π and π'. Finally, when session $\hat{\pi}$ receives its ciphertext ct'_A, $\mathcal{F}$ outputs ct'_A to its challenger. Notably, $\mathcal{F}$ perfectly simulates Game G_1. If the event $[\mathsf{COLL} : \text{Case 1}]$ occurs, then we have $\mathsf{cpk} \neq \mathsf{cpk}^*$ by the assumption that all honestly generated cpk values are distinct (as stated at the beginning of Theorem D.2) and $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}'_A) = K'$. Therefore, adversary $\mathcal{F}$ wins the Key-Coll game, and hence $\Pr[\mathsf{COLL} : \text{Case 1}] \leq NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$.
- **Case 2:** $\hat{\pi}.\mathsf{role} = \mathsf{Resp}$. We also construct an adversary $\mathcal{F}$ against the Key-Coll game for AFS-KEM as follows. Adversary $\mathcal{F}$ first chooses $g = 2$ and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \mathsf{seed}_e^*, K^*, \mathsf{ct}^*)$ from its challenger. Adversary $\mathcal{F}$ first guesses the sessions $\hat{\pi}$, π^{Init}, π^{Resp}, and the party $\pi^{\mathsf{Init}}.\mathsf{actor}$ by sampling $\hat{\pi}$, π^{Init}, and π^{Resp} uniformly from the S sessions and sampling $\pi^{\mathsf{Init}}.\mathsf{actor}$ uniformly from the N parties. Notably, the probability that $\mathcal{F}$ guesses correctly is at least $1/(NS^3)$.
  Adversary $\mathcal{F}$ then perfectly simulates Game G_1 for $\mathcal{A}$, except that it sets $(\mathsf{pk}_s^*, \mathsf{sk}_s^*)$ as the long-term key pair for $\pi^{\mathsf{Init}}.\mathsf{actor}$ and sets $(\mathsf{cpk}_A, \mathsf{seed}_A) = (\mathsf{cpk}^*, \mathsf{seed}_e^*)$ in the session π^{Init}, and embeds $(\mathsf{ct}_A, \mathsf{K}_A) = (\mathsf{ct}^*, K^*)$ into the session π^{Init} and π^{Resp}. When session $\hat{\pi}$ receives its composite public key cpk', $\mathcal{F}$ forwards cpk' to its challenger and receives the corresponding (ct', K'). It then embeds (ct', K') into the session $\hat{\pi}$. Notably, $\mathcal{F}$ perfectly simulates Game G_1 for this case. If the event $[\mathsf{COLL} : \text{Case 2}]$ occurs, then we have $K' = K^*$. Thus, adversary $\mathcal{F}$ wins the Key-Coll game, and therefore $\Pr[\mathsf{COLL} : \text{Case 2}] \leq NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$.

Combining the two cases, we obtain

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \leq 2NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}^{23}).$$

Game G_3 (Guessing Actor, π^{Test}, and $\pi^{\mathsf{FirstMatch}}$). In Game G_3, we guess the test session π^{Test} and its corresponding FirstMatch session $\pi^{\mathsf{FirstMatch}}$, and further guess the identity of $\pi^{\mathsf{Test}}.\mathsf{actor}$ (resp. $\pi^{\mathsf{FirstMatch}}.\mathsf{actor}$) if $\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Init}$ (resp. $\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Resp}$). If any of these guesses is incorrect, the game is aborted. Concretely, we sample a party P^* uniformly from the N parties and sample two session instances π and π' uniformly from the S sessions. The game aborts if any of the following conditions holds: (i) $\pi \neq \pi^{\mathsf{Test}}$, (ii) $\pi' \neq \pi^{\mathsf{FirstMatch}}$, (iii) $\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Init}$ and $P^* \neq \pi^{\mathsf{Test}}.\mathsf{actor}$, or (iv)

[23] We reuse the notation $\mathcal{F}$ for simplicity.

$\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Resp}$ and $P^* \neq \pi^{\mathsf{FirstMatch}}.\mathsf{actor}$. The probability of guessing correctly is $1/(NS^2)$, and therefore

$$\mathsf{Adv}^{G_2}(\mathcal{A}) \leq NS^2 \cdot \mathsf{Adv}^{G_3}(\mathcal{A}).$$

Game G_4 (Peer acceptance implies a matching session). In this game, we define the BAD event as the event that the FirstMatch session π' of π accepts but is not the matching session of π. If the BAD event occurs, we abort the game. If the BAD event does not occur, games G_3 and G_4 are identical. Therefore, we have $|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})| \leq \Pr[\mathsf{BAD} : G_3]$. Importantly, after this game hop, the session π^{Resp} accepts the session key if and only if it is the matching session of π^{Init}, otherwise π^{Resp} aborts.

Next, we analyze $\Pr[\mathsf{BAD} : G_3]$. Note that if the BAD event occurs, then the pair $(\mathsf{id}, \mathsf{c}_{\mathsf{seed}})$ sent by π^{Init} differs from $(\mathsf{id}', \mathsf{c}'_{\mathsf{seed}})$ received by π^{Resp}, and both π^{Init} and π^{Resp} accept the session key. We consider the following two cases:

- **Case 1:** If $\mathsf{id} \neq \mathsf{id}'$, we construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives a challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_3 for $\mathcal{A}$, except that it first samples B^* uniformly from among the N parties and aborts if $B^* \neq \pi^{\mathsf{Resp}}.\mathsf{peer}$. Notably, $\mathcal{C}$ guesses correctly with probability $1/N$. Then $\mathcal{C}$ sets $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$ and $\mathsf{pk}_{B^*} = \overline{\mathsf{pk}}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π^{Init}. When π^{Resp} receives a valid pair $(\mathsf{id}' = B^*, \mathsf{c}'_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'_{\mathsf{seed}}$ and outputs the corresponding seed'_e to its challenger. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(B^*, \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{B^*} = \overline{\mathsf{pk}}_s^*$, we conclude that $\mathsf{pk}'_s = \overline{\mathsf{pk}}_s^*$. Therefore, if the BAD event occurs and $\mathcal{C}$ guesses correctly, then adversary $\mathcal{C}$ wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 1}] \leq N \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
- **Case 2:** If $\mathsf{id} = \mathsf{id}'$ and $\mathsf{c}_{\mathsf{seed}} \neq \mathsf{c}'_{\mathsf{seed}}$, we also construct an adversary $\mathcal{C}$[24] against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_3 for $\mathcal{A}$, except that it sets $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π^{Init}. When π^{Resp} receives a valid pair $(\mathsf{id}_{P^*}, \mathsf{c}'_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'_{\mathsf{seed}}$ and outputs the corresponding seed'_e to its challenger. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{c}'_{\mathsf{seed}} \neq \mathsf{c}_{\mathsf{seed}}$, we have $\mathsf{pk}'_s = \mathsf{pk}_s^*$ and $\mathsf{seed}'_e \neq \mathsf{seed}_e^*$. Therefore, if the BAD event occurs in this case, then $\mathcal{C}$ also wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2}] \leq \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Combining the two cases, we have

$$\left|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})\right| \leq (N+1) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Game G_5 (random $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}$). In this game, we replace the $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}$ in π, π' with independent uniformly random values. Define QUERY as the event that $\mathcal{A}$ queries G on input K^{Test}, where K^{Test} denotes the encapsulated key K_A in sessions π and π'. Then, G_4 and G_5 are identical unless the event QUERY happens, and we have

$$\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq \Pr[\mathsf{QUERY} : G_5].$$

At this point, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random from the view of $\mathcal{A}$, and there may be one accepted session (which must be the matching session) that uses the same session key $\mathsf{K}_{\mathsf{SESSION}}$. Note that all session keys in other accepted sessions are independent of $\mathsf{K}_{\mathsf{SESSION}}$. Since we do not allow the adversary to reveal the session key of the matching session of π, we have

$$\mathsf{Adv}^{G_5}(\mathcal{A}) = 0.$$

Game G_6 (Randomized $c^{P^*}_{\mathsf{seed}}$). In this game, we modify the computation of $c^{P^*}_{\mathsf{seed}} = \mathsf{K}^{P^*}_{\mathsf{ENC}} \oplus \mathsf{seed}_{P^*}$ in π^{Init} by replacing it with $\hat{c}^{P^*}_{\mathsf{seed}} = \mathsf{K}^{P^*}_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_{P^*}$, where $\hat{\mathsf{seed}}_{P^*} \leftarrow\!\!\$\ \{0,1\}^\lambda$ is sampled uniformly and independently of seed_{P^*}. Importantly, after Game G_4, the session π^{Resp} accepts the session key if and only if it is the matching session of π^{Init}, otherwise π^{Resp} aborts. Since $\mathsf{K}^{P^*}_{\mathsf{ENC}}$ is uniformly random and

[24] We reuse the notation $\mathcal{C}$ for simplicity.

independent of the adversary's view in Game G_5, the values $\mathsf{K}_{\mathsf{ENC}}^{P^*} \oplus \mathsf{seed}_{P^*}$ and $\mathsf{K}_{\mathsf{ENC}}^{P^*} \oplus \hat{\mathsf{seed}}_{P^*}$ are identically distributed from the adversary's view. Hence, games G_5 and G_6 are identical from the adversary's view, and therefore

$$\Pr[\mathsf{QUERY} : G_5] = \Pr[\mathsf{QUERY} : G_6].$$

Game G_7 (Random K^{Test}). In this game, we replace the KEM key K^{Test} in π and π' with a uniformly random key. To bound $\big|\Pr[\mathsf{QUERY} : G_6] - \Pr[\mathsf{QUERY} : G_7]\big|$, we can construct an IND-ECPA adversary $\mathcal{B}$ against AFS-KEM. Specifically, $\mathcal{B}$ receives $(\mathsf{pk}_s^*, \mathsf{pk}_e^*, \mathsf{cpk}^*, \mathsf{sk}_s^*, \mathsf{ct}^*, \mathsf{K}_b^*)$ from its challenger. Adversary $\mathcal{B}$ honestly simulates the view of $\mathcal{A}$ as in G_6, except that it sets pk_s^* as the public key of P^* and embeds cpk^* and ct^* into the sessions π and π', respectively. Moreover, $\mathcal{B}$ uses K_b^* as the KEM key K^{Test} in both π and π'. When $b = 0$, $\mathcal{B}$ perfectly simulates game G_6, and when $b = 1$, $\mathcal{B}$ perfectly simulates game G_7. That is, $\Pr[\mathcal{B} \Rightarrow 1 : b = 0] = \Pr[\mathsf{QUERY} : G_6]$ and $\Pr[\mathcal{B} \Rightarrow 1 : b = 1] = \Pr[\mathsf{QUERY} : G_7]$. Therefore, we have

$$\big|\Pr[\mathsf{QUERY} : G_6] - \Pr[\mathsf{QUERY} : G_7]\big| \le 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3).$$

Analysis of $\Pr[\mathsf{QUERY} : G_7]$. Note that in this game the key K^{Test} is independent of ct and cpk in session π and π', and is uniformly random from the adversary's view. It follows that

$$\Pr[\mathsf{QUERY} : G_7] \le \frac{q_{\mathsf{G}}}{|\mathcal{K}|}.$$

Combining all the bounds derived above completes the proof. □

Lemma D.7 (QROM counterpart of Lemma D.5). *With the notations and assumptions of the proof of Theorem D.6, and with* PRG *modeled as a quantum-accessible random oracle* G *to which the adversary makes at most q_{G} (quantum) queries, there exist QPT adversaries $\mathcal{B}_3$, $\mathcal{C}$, and $\mathcal{D}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FU_3\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS^2\Big(2\sqrt{(q_{\mathsf{G}}+1)\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}\Big)} + \beta\Big),$$

$$\mathsf{Adv}^{\mathsf{FU_4\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le NS^2\Big(2\sqrt{(q_{\mathsf{G}}+1)\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}\Big)} + \beta\Big),$$

where $\beta = 2S \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + (N+1) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + \delta$.

Proof. We also present a unified analysis for Case $\mathsf{FU_3}$ and Case $\mathsf{FU_4}$ by considering $\mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$. It follows that $\mathsf{Adv}^{\mathsf{FU_3\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le \mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{FU_4\text{-}KS}}_{\mathsf{pUAKE}}(\mathcal{A}) \le \mathsf{Adv}^{(\mathsf{FU_3}\vee\mathsf{FU_4})\text{-}\mathsf{KS}}_{\mathsf{pUAKE}}(\mathcal{A})$.

Game $G_0 - G_4$. The games $G_0 - G_4$ are the same as the $G_0 - G_4$ in proof of Lemma D.5. Notably, we use π and π' to denote the tested session and its (unique) FirstMatch session, respectively. We further denote by π^{Init} (resp. π^{Resp}) the initiator (resp. responder) session among π and π'. K^* denotes the encapsulated key K_A shared by π and π'.

Game G_5 (Random $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}_{\mathsf{ENC}}^A$). In this game, the random oracle G accessed by $\mathcal{A}$ is replaced by an oracle G' where $\mathsf{G}'(x) = \mathsf{G}(x)$ if $x \neq K^*$, and $\mathsf{G}'(K^*) = y^*$, where y^* is randomly sampled from the output space of the random oracle G. Notably, the challenger still uses $\mathsf{G}(K^*)$ in π and π' to compute $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}_{\mathsf{ENC}}^A)$.

Let $z = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, K^*)$ where $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$, $\mathsf{seed}^* \leftarrow\!\$\ \{0,1\}^\lambda$, $(\mathsf{pk}_e^*, \mathsf{sk}_e^*) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}^*, \mathsf{seed}^*)$, $\mathsf{cpk}^* \leftarrow \mathsf{CombPK}(\mathsf{pk}^*, \mathsf{pk}_e^*)$, and $(\mathsf{ct}^*, K^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$. Let $A_{G_i}^{|O\rangle}(z)$ be an oracle algorithm that perfectly simulates the game G_i for $\mathcal{A}$, except that it first uses the random oracle O to simulate the random oracle accessed by $\mathcal{A}$. Moreover, A uses pk^* as the long-term public key of $\pi^{\mathsf{Init}}.\mathsf{actor}$, and embeds $\mathsf{seed}^*, \mathsf{ct}^*, K^*$ into the sessions π^{Init} and π^{Resp}, and finally returns 1 if $\mathcal{A}$ wins the game. Thus, we have $\mathsf{Adv}^{G_4}(\mathcal{A}) = \Pr[A_{G_4}^{\mathsf{G}}(z) \Rightarrow 1]$ and $\mathsf{Adv}^{G_5}(\mathcal{A}) = \Pr[A_{G_4}^{\mathsf{G}'}(z) \Rightarrow 1]$.

Lemma A.10 states that $|\Pr[A^{\mathsf{G}}_{G_4}(z) \Rightarrow 1] - \Pr[A^{\mathsf{G}'}_{G_4}(z) \Rightarrow 1]| \leq 2\sqrt{(q_G+1)\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_4}(z)]}$. Here $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_4}(z)]$ denotes the probability that the oracle $\mathcal{O}^{SC}_{K^*}$ ever outputs 1 during the execution of $A^{\mathsf{G}'\setminus K^*}_{G_4}(z)$. Note that by definition we have $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_4}(z)] = \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_5}(z)]$, therefore

$$\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq 2\sqrt{(q_G+1)\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_5}(z)]}.$$

At this point, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random from the view of $\mathcal{A}$, and there may be one accepted session (which must be the matching session) that uses the same session key $\mathsf{K}_{\mathsf{SESSION}}$. Note that all session keys in other accepted sessions are independent of $\mathsf{K}_{\mathsf{SESSION}}$. Since we do not allow the adversary to reveal the session key of the matching session of π, we have

$$\mathsf{Adv}^{G_5}(\mathcal{A}) = 0.$$

It remains to analyze $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_5}(z)]$.

Game G_6. This game is identical to G_5, except that the random oracle accessed by $\mathcal{A}$ is replaced with $\mathsf{G}' \setminus K^*$. By the definition of A, we have

$$\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_5}(z)] = \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_6}(z)].$$

Game G_7 (Randomized $c^{P^*}_{\mathsf{seed}}$). In this game, we modify the computation of $c^{P^*}_{\mathsf{seed}} = \mathsf{K}^{P^*}_{\mathsf{ENC}} \oplus \mathsf{seed}_{P^*}$ in π^{Init} by replacing it with $\hat{c}^{P^*}_{\mathsf{seed}} = \mathsf{K}^{P^*}_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_{P^*}$, where $\hat{\mathsf{seed}}_{P^*} \leftarrow\!\!\$\ \{0,1\}^\lambda$ is sampled uniformly and independently of seed_{P^*}. Importantly, after Game G_4, the session π^{Resp} accepts the session key if and only if it is the matching session of π^{Init}, otherwise π^{Resp} aborts. In particular, π^{Resp} need not recover the seed from the received value $c^{P^*}_{\mathsf{seed}}$ in order to decide whether to accept or abort.

Since $\mathsf{K}^{P^*}_{\mathsf{ENC}}$ is derived from the hidden value $\mathsf{G}(K^*)$, while $\mathcal{A}$ only has access to $\mathsf{G}' \setminus K^*$, the key $\mathsf{K}^{P^*}_{\mathsf{ENC}}$ is uniformly random and independent of the adversary's view in the execution of $A^{\mathsf{G}'\setminus K^*}_{G_6}(z)$. Therefore, the values $\mathsf{K}^{P^*}_{\mathsf{ENC}} \oplus \mathsf{seed}_{P^*}$ and $\mathsf{K}^{P^*}_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_{P^*}$ are identically distributed from the adversary's view. Hence, the executions $A^{\mathsf{G}'\setminus K^*}_{G_6}(z)$ and $A^{\mathsf{G}'\setminus K^*}_{G_7}(z)$ induce identical distributions on all variables relevant to the event FIND. Therefore,

$$\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_6}(z)] = \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_7}(z)].$$

Game G_8 (Random K^*). Game G_8 is identical to Game G_7, except that the KEM key K^* used in sessions π and π' is replaced by a uniformly random key $\hat{K}^*$, and the random oracle $\mathsf{G}' \setminus K^*$ that the adversary accessed is replaced with $\mathsf{G}' \setminus \hat{K}^*$.

Let $\hat{z} = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, \hat{K}^*)$. To bound $\left|\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_7}(z)] - \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{K}^*}_{G_8}(\hat{z})]\right|$, we construct an $\mathsf{IND\text{-}ECPA}$ adversary $\mathcal{B}$ against $\mathsf{AFS\text{-}KEM}$ as follows. Specifically, $\mathcal{B}$ receives $(\mathsf{pk}^*_s, \mathsf{pk}^*_e, \mathsf{cpk}^*, \mathsf{sk}^*_s, \mathsf{ct}^*, \mathsf{K}^*_b)$ from its challenger. Adversary $\mathcal{B}$ honestly simulates the view of $\mathcal{A}$ as $A^{\mathsf{G}'\setminus K^*}_{G_7}(z)$, except that it sets pk^*_s as the public key of P^* and embeds cpk^* and ct^* into sessions π and π', respectively. Moreover, $\mathcal{B}$ uses K^*_b as the KEM key K in both π and π', it samples a quantum random oracle G' and uses the oracle $\mathsf{G}' \setminus \mathsf{K}^*_b$ to answer adversary (quantum) RO queries. If the FIND event occurs, $\mathcal{B}$ returns 1. When $b = 0$, adversary $\mathcal{D}$ perfectly simulates Game for $\mathcal{A}$ as $A^{\mathsf{G}'\setminus K^*}_{G_7}(z)$, whereas when $b = 1$, it perfectly simulates Game as $A^{\mathsf{G}'\setminus K^*}_{G_7}(z)$. That is $\Pr[\mathcal{D} \Rightarrow 1 : b = 0] = \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_7}(z)]$ and $\Pr[\mathcal{D} \Rightarrow 1 : b = 1] = \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_8}(\hat{z})]$. Therefore,

$$\left|\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus K^*}_{G_7}(z)] - \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{K}^*}_{G_8}(\hat{z})]\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}).$$

Analysis of $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{K}^*}_{G_8}(\hat{z})]$. Recall that $\hat{z} = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, \hat{K}^*)$, and that $A^{\mathsf{G}'\setminus \hat{K}^*}_{G_8}(\hat{z})$ still uses the value $\mathsf{G}(\hat{K}^*)$ to derive $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ for the sessions π and π', while the adversary $\mathcal{A}$ does not access G. Therefore, we define an algorithm $\hat{A}^{\mathsf{G}'\setminus \hat{K}^*}_{G_8}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*)$ that is identical to $A^{\mathsf{G}'\setminus \hat{K}^*}_{G_8}(\hat{z})$, except that it samples $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ uniformly at random for π and π' (instead

of computing them via $\mathsf{G}(\hat{K}^*)$). It is immediate that $\hat{A}_{G_8}^{\mathsf{G}'\backslash\hat{K}^*}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*)$ and $A_{G_8}^{\mathsf{G}'\backslash\hat{K}^*}(\hat{z})$ induce identical views for $\mathcal{A}$. Hence, $\Pr[\mathsf{FIND} : A_{G_8}^{\mathsf{G}'\backslash\hat{K}^*}(\hat{z})] = \Pr[\mathsf{FIND} : \hat{A}_{G_8}^{\mathsf{G}'\backslash\hat{K}^*}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*)]$.

Note that the test key $\hat{K}^*$ is independent of $\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*$, and is uniformly random from the adversary's view. By Lemma A.11, it follows that

$$\Pr[\mathsf{FIND} : \hat{A}_{G_8}^{\mathsf{G}'\backslash\hat{K}^*}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*)] \leq \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}.$$

Combining all the bounds derived above completes the proof. □

D.4 Init-Explicit Authentication Analysis for pUAKE

Theorem D.8. *Let there be at most N parties and at most S sessions activated in the* Init-EA *game. Suppose that the underlying* AFS-KEM *is δ-correct, γ-spreadness, and satisfies perfect* ExtractPK*-correctness and ephemeral-key-generation equivalence. Then, for any polynomial-time adversary $\mathcal{A}$ attacking the* Init-EA *security of* pUAKE*, there exist polynomial-time adversaries $\mathcal{B}, \mathcal{C}, \mathcal{D}$, and $\mathcal{Z}$, constructed below and running times are about that of $\mathcal{A}$, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Init\text{-}EA}}_{\mathsf{pUAKE}}(\mathcal{A}) \leq{}& \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} \\ &+ NS\begin{pmatrix} 2\ \mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}) + S\delta \\ + (N+1)S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) \end{pmatrix},\end{aligned}$$

where k_1 and k_2 are the min-entropy bounds on the honest static and combined public keys of AFS-KEM*, respectively.*

Proof. First, we assume that all honest initiators have distinct long-term public keys, and that all honest initiator sessions compute distinct cpk_A. Moreover, we assume that every responder session that receives an honest cpk_A from some honest initiator session computes a distinct ct_A. Since the min-entropy of the public key pk_A is at least k_1, the honestly combined public key cpk_A has min-entropy at least k_2, and the ciphertexts of AFS-KEM satisfy γ-spreadness, this will add $\frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}}$ to the final bound. Under these assumptions, it is impossible for a session to have more than one FirstMatch or Match session. We assume the adversary outputs a π that satisfies $\mathsf{MalAcc}_{\mathsf{Resp}}(\pi) = \mathsf{true}$. otherwise, the adversary trivially loses the game.

Game G_0. This game corresponds to the original game with the above assumption for $\mathsf{Adv}^{\mathsf{Init\text{-}EA}}_{\mathsf{pUAKE}}(\mathcal{A})$, that is,

$$\mathsf{Adv}^{\mathsf{Init\text{-}EA}}_{\mathsf{pUAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1 (Guessing $\pi.\mathsf{peer}$, π). In this game, we guess the session π output by the adversary and its $\pi.\mathsf{peer}$, and abort if the guess is incorrect. Formally, we sample P^* uniformly from the N parties, and sample one session instance π uniformly from at most S sessions. If the guess is incorrect (i.e., $P^* \neq \pi.\mathsf{peer}$ or $\pi \neq \pi$), where π is the session output by the adversary, the game is aborted. The probability of guessing correctly is $1/(NS)$; therefore,

$$\mathsf{Adv}^{G_0}(\mathcal{A}) \leq NS \cdot \mathsf{Adv}^{G_1}(\mathcal{A}).$$

Game G_2 (New win condition). In this game, we change the condition for the adversary to win. An instance of G_2 is won by the adversary $\mathcal{A}$ if and only if π accepts where $\mathsf{MalAcc}_{\mathsf{Resp}}(\pi) = \mathsf{true}$. We stop the game at the moment when π accepts. Note that in G_2, P^* remains uncorrupted. For G_1 to be won, any instance of G_1 needs to satisfy the win conditions of G_2. Thus, we have

$$\mathsf{Adv}^{G_1}(\mathcal{A}) \leq \mathsf{Adv}^{G_2}(\mathcal{A}).$$

Game G_3 (Random K_A). In this game, we replace the KEM key K_A in the session π with an independently uniform key $\tilde{k} \leftarrow\!\!\$\ \mathcal{K}$. If the unique FirstMatch session π' of π exists, we also replace the

K_A in π' with the same key $\tilde{k} \leftarrow\!\!\$\ \mathcal{K}$. To study the difference in advantage between G_2 and G_3, we use the same argument as in Game G_2 in Lemma D.4 for the case of FU_2. This argument shows that we can construct an adversary $\mathcal{B}$ against the IND-LCK-SCCA security of AFS-KEM such that

$$\left|\mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}) + S\delta.$$

Game G_4 (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$). In this game, we replace the keys $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ in π with independently sampled uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$ and $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$. If the unique FirstMatch session π' of π exists, we also replace the $\mathsf{K}_{\mathsf{SESSION}}$ and $\mathsf{K}^A_{\mathsf{ENC}}$ in π' with the same keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$ and $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$. To study the difference in advantage between G_3 and G_4, we use the same argument as in Game G_3 in Lemma D.4 for the case of FU_2. Thus, we can construct an adversary $\mathcal{Z}$ against the IND-PRG security of the PRG such that

$$\left|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{IND\text{-}PRG}}_{\mathsf{PRG}}(\mathcal{Z}).$$

Game G_5 (Matching session). In this game, we define the event that the matching session π^{Match} of the test session π does not exist as the BAD event, and abort if BAD occurs. Note that the adversary can win only if BAD occurs and Game G_5 aborts whenever BAD occurs, we have

$$\mathsf{Adv}^{G_5}(\mathcal{A}) = 0.$$

It is evident that $\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq \Pr[\mathsf{BAD} : G_4]$. Next, we analyze $\Pr[\mathsf{BAD} : G_4]$. In Game G_4, the key $\mathsf{K}^A_{\mathsf{ENC}}$ of the test session π is uniformly random from the adversary's view. Moreover, if there exists a FirstMatch session of π, denoted by π', then π' uses the same encryption key $\mathsf{K}^A_{\mathsf{ENC}}$. We analyze the event BAD by considering the following cases.

Case 1: π' does not exist. Since the session π accepts without any FirstMatch session, the key $\mathsf{K}^A_{\mathsf{ENC}}$ is never used in any other session. Therefore, $\mathsf{seed}'_A = \mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{c}^A_{\mathsf{seed}}$ in π is uniformly random from the adversary's view. In this case, we can construct an adversary $\mathcal{D}$ against the RSPK-Bind security of AFS-KEM as follows. Specifically, $\mathcal{D}$ receives $(\mathsf{pk}^*_s, \mathsf{sk}^*_s)$ and perfectly simulates the game for $\mathcal{A}$ as in G_4, except that it sets $\mathsf{pk}_{P^*} = \mathsf{pk}^*_s$ and outputs the composite key cpk_A in session π to its challenger. After $\mathcal{D}$ receives seed_e from its challenger, it defines $\mathsf{K}^A_{\mathsf{ENC}} = \mathsf{seed}_e \oplus \mathsf{c}^A_{\mathsf{seed}}$ in session π, where $\mathsf{c}^A_{\mathsf{seed}}$ is received in π. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) = \mathsf{EKeyGen}_c(\mathsf{cpk}_A, \mathsf{seed}_e)$ and $\mathsf{pk}'_s = \mathsf{ExtractPK}(\mathsf{cpk}_A, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{P^*} = \mathsf{pk}^*_s$, we have $\mathsf{pk}'_s = \mathsf{pk}^*_s$ by the security of CA. Therefore, if the BAD event occurs in this case, then $\mathcal{D}$ wins the RSPK-Bind game. Hence, $\Pr[\mathsf{BAD} : \text{Case 1}] \leq \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})$.

Case 2: π' exists. In this case, the session π' must accept. The BAD event implies that the adversary has tampered with the third message sent by π'. Let $(\mathsf{id}', \mathsf{c}'_{\mathsf{seed}})$ be the message sent by π', and let $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$ be the message received by π, where $(\mathsf{id}', \mathsf{c}'_{\mathsf{seed}}) \neq (\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$. We further distinguish the following subcases.

- **Case 2.a:** If $\mathsf{id}_{P^*} \neq \mathsf{id}'$, we construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}^*_e, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \overline{\mathsf{pk}}^*_s, \overline{\mathsf{sk}}^*_s)$ and perfectly simulates Game G_4 for $\mathcal{A}$, except that it first samples B^* uniformly from among the N parties and π' uniformly from among the S sessions, aborting if $B^* \neq \mathsf{id}'$ or $\pi' \neq \pi^{\mathsf{FirstMatch}}$. Thus, $\mathcal{C}$ guesses correctly with probability $1/(NS)$. Then $\mathcal{C}$ sets $\mathsf{pk}_{B^*} = \mathsf{pk}^*_s$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}^*_s$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}^*_e$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π'. When π receives a valid pair $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}_{\mathsf{seed}}$ and outputs the corresponding seed_e to its challenger. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}^*_s$, we have $\mathsf{pk}'_s = \overline{\mathsf{pk}}^*_s$ by the security of CA. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2.a}] \leq NS \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
- **Case 2.b:** If $\mathsf{id}_{P^*} = \mathsf{id}'$ and $\mathsf{c}_{\mathsf{seed}} \neq \mathsf{c}'_{\mathsf{seed}}$, we also construct an adversary $\mathcal{C}$[25] against the SPK-Bind security of AFS-KEM as follows. $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}^*_e, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \overline{\mathsf{pk}}^*_s, \overline{\mathsf{sk}}^*_s)$ and perfectly simulates Game G_4 for $\mathcal{A}$, except that it first samples π' uniformly from among the S sessions, aborting if $\pi' \neq \pi^{\mathsf{FirstMatch}}$. Thus, $\mathcal{C}$ guesses correctly with probability $1/S$. It then sets $\mathsf{pk}_{P^*} = \mathsf{pk}^*_s$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}^*_e$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π'. When π receives a valid pair $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}_{\mathsf{seed}}$ and outputs the corresponding seed_e to its challenger. Let

[25] We reuse the notation $\mathcal{C}$ for simplicity.

$(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{c}_{\mathsf{seed}} \neq \mathsf{c}'_{\mathsf{seed}}$, we have $\mathsf{pk}'_s = \mathsf{pk}^*_s$ and $\mathsf{seed}_e \neq \mathsf{seed}^*_e$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ also wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2.b}] \leq S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Combining all the cases above, we conclude that

$$\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq (N+1)S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}).$$

Combining all the bounds derived above completes the proof. □

E Appendix: Security Analysis for pMAKE

E.1 Key Secrecy of the pMAKE protocols.

Theorem 5.3 ($\mathsf{F}_{\mathsf{pMAKE}}$-Key Secrecy of pMAKE in the ROM and QROM). *Let there be at most N parties and at most S sessions activated in the $\mathsf{F}_{\mathsf{pMAKE}}$-KS game. Suppose that the underlying AFS-KEM is δ-correct, satisfies γ-spreadness, and satisfies perfect ExtractPK-correctness and ephemeral-key-generation equivalence. We model the PRF as a random oracle G, to which the adversary makes at most q_{G} queries. Then, for any PPT adversary $\mathcal{A}$ attacking the $\mathsf{F}_{\mathsf{pMAKE}}$-KS security of pMAKE, there exist PPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{C}, \mathcal{D}, \mathcal{F}$, and $\mathcal{Z}$, constructed below and running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{F}_{\mathsf{pMAKE}}\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq\ & \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} \\
& + NS \left(\begin{array}{l} 4\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + 2(S+1)\,\mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) \\ \quad + 4(2NS+1)\,\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) \\ \quad + S \left(\begin{array}{l} 4\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + 8\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) \\ \quad + 8(S+1)\,\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + \dfrac{4q_{\mathsf{G}}}{|\mathcal{K}|} + 16\delta \\ \quad + (8N^2 + 29N + 20)\,\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) \end{array} \right) \end{array} \right).
\end{aligned}$$

If we model PRF as a quantum-accessible random oracle G, to which the adversary makes at most q_{G} quantum queries, then, for any QPT adversary $\mathcal{A}$ attacking the $\mathsf{F}_{\mathsf{pMAKE}}$-KS security of pMAKE, there exist QPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{C}, \mathcal{D}, \mathcal{F}$, and $\mathcal{Z}$, running in time comparable to that of $\mathcal{A}$, such that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{F}_{\mathsf{pMAKE}}\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq\ & \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} \\
& + NS \left(\begin{array}{l} 4\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + 2(S+1)\,\mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 4\,\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) \\ \quad + S \left(\begin{array}{l} 4\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + 8(S+1)\,\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ \quad + (5N+4)\,\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + 16\delta \end{array} \right) \end{array} \right) \\
& + 4NS^2 \left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|} + \beta\right)} + \beta \right),
\end{aligned}$$

where $\beta = N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$. Here k_1 and k_2 are the min-entropy bounds on the honest static and combined public keys of AFS-KEM, respectively. Furthermore, $|\mathcal{K}|$ denotes the cardinality of the AFS-KEM encapsulated key space.

Proof. First, we assume that all honest users have distinct long-term public keys, and that all honest sessions compute distinct combined public keys cpk, including cpk_A for initiator sessions and cpk_B for responder sessions. Moreover, we assume that every honest responder session that receives an honestly generated cpk_A from some honest initiator session computes a distinct ciphertext ct_A. Similarly, we

assume that every honest initiator session that receives an honestly generated cpk_B from some honest responder session computes a distinct ciphertext ct_B.

Since the long-term public keys have min-entropy at least k_1, the combined public keys generated honestly from a fixed long-term public key have min-entropy at least k_2[26], and the ciphertexts of AFS-KEM satisfy γ-spreadness, these assumptions contribute an additive term $\frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}}$ to the final security bound. Notably, under these assumptions, no session can have more than one PartialMatch session or more than one Match session.

We assume that the tested session satisfies $\mathsf{F}_{\mathsf{pMAKE}}$ throughout the experiment; otherwise, the adversary loses the indistinguishability game. To bound $\mathsf{Adv}^{\mathsf{F}_{\mathsf{pMAKE}}\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$, we partition the winning event into six cases according to $\mathsf{F}_{\mathsf{pMAKE}} = \bigvee_{i=1}^{6} \mathsf{FM}_i$ (Definition 5.2), which yields

$$\mathsf{Adv}^{\mathsf{F}_{\mathsf{pMAKE}}\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq \sum_{i=1}^{6} \mathsf{Adv}^{\mathsf{FM}_i\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})\,.$$

We now give security reductions for each subcase $\mathsf{Adv}^{\mathsf{FM}_i\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$ with $i \in \{1,2,3,4,5,6\}$, where throughout we assume that the tested session satisfies FM_i (otherwise the adversary trivially loses the corresponding game). The corresponding reductions are stated in Lemmas E.1, E.2, and E.3. For readability, we defer the proofs of these lemmas to Subsection E.2.

Lemma E.1. *With the notation and assumptions of Theorem 5.3, there exist adversaries $\mathcal{B}_1$, $\mathcal{C}$, and $\mathcal{Z}$ running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FM}_1\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq NS^2\Big(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{dual}\text{-}\mathsf{PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 2(N+1)\,\mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C}) + 2\delta\Big).$$

$$\mathsf{Adv}^{\mathsf{FM}_2\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq NS^2\Big(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{dual}\text{-}\mathsf{PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + (3N+2)\,\mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C}) + 2\delta\Big).$$

Lemma E.2. *With the notations and assumptions of the proof of Theorem 5.3, and with* PRF *modeled as a random oracle* G *to which the adversary makes at most q_{G} queries, there exist adversaries $\mathcal{B}_2$, $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{F}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FM}_3\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq{}& 2NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{ECPA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_2) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|}\right) + 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key}\text{-}\mathsf{Coll}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{F})\\ &+ 4N^2S^2 \cdot \mathsf{Adv}^{\mathsf{RSPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{D}) + 4N(N+1)(N+2)S^2 \cdot \mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C}).\end{aligned}$$

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FM}_4\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq{}& 2NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{ECPA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_2) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|}\right) + 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key}\text{-}\mathsf{Coll}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{F})\\ &+ 4N^2S^2 \cdot \mathsf{Adv}^{\mathsf{RSPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{D}) + 4N(N+1)(N+2)S^2 \cdot \mathsf{Adv}^{\mathsf{SPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{C}).\end{aligned}$$

Lemma E.3. *With the notation and assumptions of Theorem 5.3, there exist adversaries $\mathcal{B}_3$, $\mathcal{D}$, $\mathcal{F}$, and $\mathcal{Z}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FM}_5\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq{}& NS\Big(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{LCK}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_3) + \mathsf{Adv}^{\mathsf{dual}\text{-}\mathsf{PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 2\,\mathsf{Adv}^{\mathsf{RSPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{D})\Big)\\ &+ 2NS^2\Big(2\,\mathsf{Adv}^{\mathsf{Key}\text{-}\mathsf{Coll}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{F}) + \delta\Big).\end{aligned}$$

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FM}_6\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq{}& NS\Big(2\,\mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{LCK}\text{-}\mathsf{SCCA}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{B}_3) + \mathsf{Adv}^{\mathsf{dual}\text{-}\mathsf{PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 2\,\mathsf{Adv}^{\mathsf{RSPK}\text{-}\mathsf{Bind}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{D})\Big)\\ &+ 2NS^2\Big(2\,\mathsf{Adv}^{\mathsf{Key}\text{-}\mathsf{Coll}}_{\mathsf{AFS}\text{-}\mathsf{KEM}}(\mathcal{F}) + \delta\Big).\end{aligned}$$

Combining all the above analyses completes the proof in the ROM.

Proof in the QROM. This QROM proof follows the same structure as the ROM proof, except for Lemma E.2, where the function $G = \mathsf{PRF}$ is modeled as a random oracle and therefore must be lifted to the QROM. For readability, we defer the proof of Lemma E.4 to Subsection E.2.

[26] Note that in the definition of the min-entropy of the combined public key, the long-term public key pk_s is fixed, and the min-entropy is taken over honestly generated combined public keys derived from pk_s.

Lemma E.4 (QROM counterpart of Lemma E.2). *With the notation and assumptions of Theorem 5.3, and modeling* PRF *as a quantum-accessible random oracle* G *to which the adversary makes at most q_{G} quantum queries, there exist QPT adversaries $\mathcal{B}_2$, $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{F}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FM_3\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) &\le 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ &\quad + 2NS^2\left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|} + \beta\right)} + \beta\right).\\ \mathsf{Adv}^{\mathsf{FM_4\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) &\le 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ &\quad + 2NS^2\left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|} + \beta\right)} + \beta\right).\end{aligned}$$

where $\beta = N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Combining all the above analyses completes the proof in the QROM. □

E.2 Full Analysis for Lemmas E.1, E.2, E.3, and E.4

Lemma E.1. *With the notation and assumptions of Theorem 5.3, there exist adversaries $\mathcal{B}_1$, $\mathcal{C}$, and $\mathcal{Z}$ running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FM_1\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le NS^2\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 2(N+1)\,\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + 2\delta\Big).$$

$$\mathsf{Adv}^{\mathsf{FM_2\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le NS^2\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + (3N+2)\,\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + 2\delta\Big).$$

Proof. We begin by providing a formal analysis for case FM_1 and assume that the tested session satisfies FM_1 (otherwise, the adversary trivially loses the corresponding game). We then derive the proof for Case FM_2, which follows the same structure as Case FM_1 except for G_4'.

Game G_0. This game corresponds to the original game for $\mathsf{Adv}^{\mathsf{FM_1\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A})$, thus,

$$\mathsf{Adv}^{\mathsf{FM_1\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1 (KEM Correctness). In this game, we abort if the decapsulation of ct_A or ct_B in π^{Test} does not equal the corresponding encapsulated key. Let ERR_A and ERR_B denote the abort events caused by ct_A and ct_B, respectively. We first bound $\Pr[\mathsf{ERR}_A]$ using the δ-correctness of AFS-KEM. The reduction guesses π^{Test} and $\pi^{\mathsf{PartialMatch}}$ uniformly from S sessions, and guesses $\pi^{\mathsf{Init}}.\mathsf{actor}$ uniformly from N parties. Notably, the guess is correct with probability at least $1/(NS^2)$. Conditioned on this guess being correct, the reduction can perfectly embed the δ-correctness experiment of AFS-KEM into the guessed sessions. Therefore, by the δ-correctness of AFS-KEM, we have $\Pr[\mathsf{ERR}_A] \le NS^2\delta$. By a similar argument, we have $\Pr[\mathsf{ERR}_B] \le NS^2\delta$. By a union bound, $\Pr[\mathsf{ERR}_A \vee \mathsf{ERR}_B] \le 2NS^2\delta$. Since games G_0 and G_1 differ only if $\mathsf{ERR}_A \vee \mathsf{ERR}_B$ occurs, it follows that

$$\left|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})\right| \le 2NS^2\delta.$$

Game G_2 (Peer acceptance implies a matching session). In this game, we define the BAD event as the event that the PartialMatch session π' accepts but is not the matching session of π. If the BAD event occurs, we abort the game immediately. Note that when the BAD event does not occur, Games G_1 and G_2 are identical from the adversary's view. Therefore,

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \le \Pr[\mathsf{BAD} : G_1].$$

Next, we analyze $\Pr[\mathsf{BAD} : G_1]$. Let π^{Init} (resp. π^{Resp}) be the initiator (resp. responder) session among π, π'. If BAD occurs in G_1, then both π^{Init} and π^{Resp} accept, but the tuple $(\mathsf{id}_A, \mathsf{c}^A_{\mathsf{seed}}, \mathsf{id}_B, \mathsf{c}^B_{\mathsf{seed}})$ maintained in π^{Init} is different from the tuple $(\mathsf{id}'_A, \mathsf{c}'^A_{\mathsf{seed}}, \mathsf{id}'_B, \mathsf{c}'^B_{\mathsf{seed}})$ maintained in π^{Resp}. We consider the following four cases.

– **Case 1:** If $\mathsf{id}_A \neq \mathsf{id}'_A$, we construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_1 for $\mathcal{A}$, except that it first guesses the sessions π^{Init} and π^{Resp} uniformly over the S sessions, and guesses the identities id_A and id'_A uniformly over the N parties. Notably, $\mathcal{C}$ guesses correctly with probability at least $1/(N^2S^2)$. Then $\mathcal{C}$ sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$ and $\mathsf{pk}_{\mathsf{id}'_A} = \overline{\mathsf{pk}}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into the session π^{Init}. When π^{Resp} receives $(\mathsf{id}'_A, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Let $(\mathsf{pk}_e^{A'}, \mathsf{sk}_e^{A'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_A)$ and $\mathsf{pk}'_A \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{A'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}'_A, \mathsf{pk}'_A) = 1$ and $\mathsf{pk}_{\mathsf{id}'_A} = \overline{\mathsf{pk}}_s^*$, we have $\mathsf{pk}'_A = \overline{\mathsf{pk}}_s^*$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 1}] \leq N^2S^2 \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
– **Case 2:** If $\mathsf{id}_B \neq \mathsf{id}'_B$, the analysis is similar to Case 1. We also construct an adversary $\mathcal{C}$[27] against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_1 for $\mathcal{A}$, except that it first randomly guesses π^{Init}, π^{Resp}, id_B, and id'_B, with success probability at least $1/(N^2S^2)$. Then $\mathcal{C}$ sets $\mathsf{pk}_{\mathsf{id}'_B} = \mathsf{pk}_s^*$ and $\mathsf{pk}_{\mathsf{id}_B} = \overline{\mathsf{pk}}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_B = \mathsf{seed}_e^*$ and $\mathsf{cpk}_B = \mathsf{cpk}^*$ into the session π^{Resp}. When π^{Init} receives $(\mathsf{id}'_B, \mathsf{c}'^B_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^B_{\mathsf{seed}}$ and outputs the corresponding seed'_B to its challenger. Let $(\mathsf{pk}_e^{B'}, \mathsf{sk}_e^{B'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_B)$ and $\mathsf{pk}'_B \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{B'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_B, \mathsf{pk}'_B) = 1$ and $\mathsf{pk}_{\mathsf{id}_B} = \overline{\mathsf{pk}}_s^*$, we have $\mathsf{pk}'_B = \overline{\mathsf{pk}}_s^*$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ also wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2}] \leq N^2S^2 \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
– **Case 3:** If $\mathsf{id}_A = \mathsf{id}'_A$ and $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$, we also construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_1 for $\mathcal{A}$, except that it first randomly guesses π^{Init}, π^{Resp}, and id_A, with success probability at least $1/(NS^2)$. $\mathcal{C}$ then sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$ and embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π^{Init}. When π^{Resp} receives a pair $(\mathsf{id}_A, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Let $(\mathsf{pk}_e^{A'}, \mathsf{sk}_e^{A'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_A)$ and $\mathsf{pk}'_A \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{A'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_A, \mathsf{pk}'_A) = 1$ and $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$, we have $\mathsf{pk}'_s = \mathsf{pk}_s^*$. Moreover, since $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$ and the OTP key used to recover the seed is the same, we have $\mathsf{seed}'_A \neq \mathsf{seed}_e^*$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 3}] \leq NS^2\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
– **Case 4:** If $\mathsf{id}_B = \mathsf{id}'_B$ and $\mathsf{c}^B_{\mathsf{seed}} \neq \mathsf{c}'^B_{\mathsf{seed}}$, the analysis is analogous to Case 3. We construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_1 for $\mathcal{A}$, except that it first randomly guesses π^{Init}, π^{Resp}, and id_B, with success probability at least $1/(NS^2)$. $\mathcal{C}$ sets $\mathsf{pk}_{\mathsf{id}_B} = \mathsf{pk}_s^*$ and embeds $\mathsf{seed}_B = \mathsf{seed}_e^*$ and $\mathsf{cpk}_B = \mathsf{cpk}^*$ into π^{Resp}. When π^{Init} receives $(\mathsf{id}_B, \mathsf{c}'^B_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^B_{\mathsf{seed}}$ and outputs the corresponding seed'_B to its challenger. Let $(\mathsf{pk}_e^{B'}, \mathsf{sk}_e^{B'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_B)$ and $\mathsf{pk}'_B \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{B'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_B, \mathsf{pk}'_B) = 1$ and $\mathsf{pk}_{\mathsf{id}_B} = \mathsf{pk}_s^*$, we have $\mathsf{pk}'_B = \mathsf{pk}_s^*$. Moreover, we have $\mathsf{seed}'_B \neq \mathsf{seed}_e^*$ since $\mathsf{c}^B_{\mathsf{seed}} \neq \mathsf{c}'^B_{\mathsf{seed}}$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 4}] \leq NS^2\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Combining the above cases, we have

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \leq 2(N+1)NS^2 \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Game G_3 (Guessing $\pi^{\mathsf{Test}}.\mathsf{actor}$, π^{Test}, and $\pi^{\mathsf{PartialMatch}}$). In game G_3, we first guess the actor identity of the test session, and further guess the test session π^{Test} and its corresponding (unique) PartialMatch session $\pi^{\mathsf{PartialMatch}}$. If any of the guesses is incorrect, the game is aborted. Concretely, we sample P^* uniformly from the N parties, and sample two session instances π, π' uniformly from the at most S sessions, respectively. If any of the guesses is incorrect (i.e., $P^* \neq \pi^{\mathsf{Test}}.\mathsf{actor}$, or $\pi \neq \pi^{\mathsf{Test}}$, or

[27] We reuse the notation $\mathcal{C}$ for simplicity in these four cases.

$\pi' \neq \pi^{\mathsf{PartialMatch}}$), the game is aborted. The probability of guessing correctly is $1/(NS^2)$, and therefore

$$\mathsf{Adv}^{G_0}(\mathcal{A}) \leq NS^2 \mathsf{Adv}^{G_3}(\mathcal{A}).$$

Game G_4 (Random K_A or K_B). In this game, if $\pi.\mathsf{role} = \mathsf{Init}$ (resp. $\pi.\mathsf{role} = \mathsf{Resp}$), we replace the KEM key K_A (resp. K_B) in the tested session π and its (unique) $\mathsf{PartialMatch}$ session π' with an independently uniform key $\tilde{k} \leftarrow\!\!\$\, \mathcal{K}$.

For any adversary $\mathcal{A}$ that can distinguish G_3 from G_4, we construct an adversary $\mathcal{B}_1$ against the $\mathsf{IND\text{-}SCCA}$ security of $\mathsf{AFS\text{-}KEM}$. Adversary $\mathcal{B}_1$ interacts with its $\mathsf{AFS\text{-}KEM}$ challenger and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}_s^*, \mathsf{seed}_e^*, \mathsf{ct}^*, \mathsf{K}_b^*)$, where K_b^* equals the real encapsulated key if $b = 0$ and a uniform random key if $b = 1$. It then simulates the view of $\mathcal{A}$ as in G_4, except that

- $\mathcal{B}_1$ sets pk_s^* as the long-term public key of P^*. It embeds $(\mathsf{cpk}_A, \mathsf{ct}_A) = (\mathsf{cpk}^*, \mathsf{ct}^*)$ (resp. $(\mathsf{cpk}_B, \mathsf{ct}_B) = (\mathsf{cpk}^*, \mathsf{ct}^*)$) into the tested session π and its $\mathsf{PartialMatch}$ session π' when $\pi.\mathsf{role} = \mathsf{Init}$ (resp. $\pi.\mathsf{role} = \mathsf{Resp}$), and uses K_b^* as the KEM key K_A (resp. K_B) in both sessions π and π'. Using seed_e^* and K_b^*, $\mathcal{B}_1$ can honestly simulate both sessions.
- For each non-tested session $\hat{\pi} \neq \pi$ of P^*, $\mathcal{B}_1$ independently queries $\mathcal{O}^{\mathsf{EKeyGen}}()$ to obtain a fresh tuple $(\mathsf{cpk}, \mathsf{pk}_e, \mathsf{sk}_e, \mathsf{seed})$ for the session, and sets the cpk_A message of $\hat{\pi}$ to be cpk (resp. sets the cpk_B message to be cpk) when $\hat{\pi}.\mathsf{role} = \mathsf{Init}$ (resp. $\hat{\pi}.\mathsf{role} = \mathsf{Resp}$). Upon receiving the corresponding KEM ciphertext ct for $\hat{\pi}$, $\mathcal{B}_1$ queries $\mathcal{O}^{\mathsf{Dec}_s}(\mathsf{cpk}, \mathsf{ct})$ to obtain the KEM key K_A (resp. K_B) when $\hat{\pi}.\mathsf{role} = \mathsf{Init}$ (resp. $\hat{\pi}.\mathsf{role} = \mathsf{Resp}$), and continues the simulation of that session honestly.

When $b = 0$, $\mathcal{B}_1$ perfectly simulates G_3, and when $b = 1$, $\mathcal{B}_1$ perfectly simulates G_4. That is, $\Pr[\mathcal{B}_1 \Rightarrow 1 : b = 0] = \Pr[G_3^{\mathcal{A}} \Rightarrow 1]$ and $\Pr[\mathcal{B}_1 \Rightarrow 1 : b = 1] = \Pr[G_4^{\mathcal{A}} \Rightarrow 1]$. Therefore, we have

$$\left|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1).$$

Game G_5 (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, $\mathsf{K}^B_{\mathsf{ENC}}$). In this game, we replace the keys $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$ in the sessions π and π' with independently sampled uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}, \hat{\mathsf{K}}^A_{\mathsf{ENC}}, \hat{\mathsf{K}}^B_{\mathsf{ENC}}$, respectively.

For any adversary $\mathcal{A}$ that can distinguish G_4 from G_5, we construct an adversary $\mathcal{Z}$ against the $\mathsf{dual\text{-}PRF}$ security of the underlying PRF. Adversary $\mathcal{Z}$ is given oracle access to $O_b(\cdot)$, where O_0 implements the real PRF and O_1 implements a uniformly random function. Adversary $\mathcal{Z}$ simulates the view of $\mathcal{A}$ exactly as in Game G_4, except that it computes $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}})$ for the sessions π and π' by querying its oracle $O_b(\cdot)$. Notably, if $\pi.\mathsf{role} = \mathsf{Init}$, $\mathcal{Z}$ queries the corresponding $O_b(\mathsf{K}_B)$; otherwise, it queries the corresponding $O_b(\mathsf{K}_A)$ in π and π'. When $b = 0$, the oracle O_0 corresponds to the real PRF, and the simulation is identical to Game G_4. When $b = 1$, the oracle O_1 is a truly random function, and the simulation is identical to Game G_5. Thus, $\Pr[\mathcal{Z} \Rightarrow 1 \mid b = 0] = \Pr[G_4^{\mathcal{A}} \Rightarrow 1]$ and $\Pr[\mathcal{Z} \Rightarrow 1 \mid b = 1] = \Pr[G_5^{\mathcal{A}} \Rightarrow 1]$. Therefore, we obtain

$$\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}).$$

Analysis of G_5. In G_5, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random for $\mathcal{A}$, and there may be one accepted session (which must be the matching session) that uses the same session key $\mathsf{K}_{\mathsf{SESSION}}$, and we do not allow the adversary to reveal this session key for the matching session of π. Note that all session keys in other accepted sessions are independent of $\mathsf{K}_{\mathsf{SESSION}}$. Therefore, we have

$$\mathsf{Adv}^{G_5}(\mathcal{A}) = 0.$$

Combining all the bounds derived above completes the proof for case FM_1. □

Proof of Case FM_2. This proof follows the same structure as that for case FM_1, except for G_4'. We only highlight the differences below. Notably, we assume that the tested session satisfies FM_2.

Game G_0'. This game corresponds to the original game defining $\mathsf{Adv}^{\mathsf{FM_2\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A})$. Thus,

$$\mathsf{Adv}^{\mathsf{FM_2\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0'}(\mathcal{A}).$$

Game G_1' (KEM-Correctness). In this game, we abort if the decapsulation of ct_A or ct_B in π^{Test} does not equal the corresponding encapsulated key. Similar to the analysis of Game G_1 in case FM_1, we have

$$\left|\mathsf{Adv}^{G_0'}(\mathcal{A}) - \mathsf{Adv}^{G_1'}(\mathcal{A})\right| \leq 2NS^2\delta.$$

Game G_2' (Peer acceptance implies a matching session). In this game, we define the BAD event as the event that the PartialMatch session π' accepts but is not the matching session of π. If the BAD event occurs, we abort the game immediately. The analysis is identical to that of Game G_2 in case FM_1, yielding

$$\left|\mathsf{Adv}^{G_1'}(\mathcal{A}) - \mathsf{Adv}^{G_2'}(\mathcal{A})\right| \leq 2(N+1)NS^2 \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Game G_3' (Guessing $\pi^{\mathsf{Test}}.\mathsf{peer}$, π^{Test}, and $\pi^{\mathsf{PartialMatch}}$). In game G_3', we first guess the peer identity of the test session, and further guess the test session π^{Test} and its corresponding (unique) PartialMatch session $\pi^{\mathsf{PartialMatch}}$. If any of the guesses is incorrect, the game is aborted. Concretely, we sample P^* uniformly from the N parties, and sample two session instances π, π' uniformly from the at most S sessions, respectively. If any of the guesses is incorrect, i.e., $P^* \neq \pi^{\mathsf{Test}}.\mathsf{peer}$, $\pi \neq \pi^{\mathsf{Test}}$, or $\pi' \neq \pi^{\mathsf{PartialMatch}}$, the game is aborted. The probability of guessing correctly is $1/(NS^2)$. Therefore,

$$\mathsf{Adv}^{G_2'}(\mathcal{A}) \leq NS^2\mathsf{Adv}^{G_3'}(\mathcal{A}).$$

Game G_4'. In this game, we define the flag event as $\pi^{\mathsf{PartialMatch}}.\mathsf{actor} \neq P^*$. If the flag event occurs, the game aborts immediately. In particular, after this game hop, we must have $\pi^{\mathsf{PartialMatch}}.\mathsf{actor} = P^*$, where P^* is not corrupted by the adversary. Since G_3' and G_4' are identical unless the flag event occurs, we have

$$\left|\mathsf{Adv}^{G_3'}(\mathcal{A}) - \mathsf{Adv}^{G_4'}(\mathcal{A})\right| \leq \Pr[\mathsf{flag}].$$

If the flag event occurs, we construct an adversary $\mathcal{C}$ against the static public key binding security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives a challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_3' for $\mathcal{A}$, except that it samples B^* uniformly from N parties and aborts if $B^* \neq \pi^{\mathsf{PartialMatch}}.\mathsf{actor}$. Notably, $\mathcal{C}$ guesses correctly with probability $1/N$.

Then $\mathcal{C}$ sets $\mathsf{pk}_{B^*} = \mathsf{pk}_s^*$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ (resp. $\mathsf{seed}_B = \mathsf{seed}_e^*$ and $\mathsf{cpk}_B = \mathsf{cpk}^*$) into $\pi^{\mathsf{PartialMatch}}$ if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$). When π receives an OTP ciphertext c, $\mathcal{C}$ honestly decrypts c and outputs the corresponding seed_e' to its challenger. Let $(\mathsf{pk}_e', \mathsf{sk}_e') \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}_e')$ and $\mathsf{pk}_s' \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e')$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}_s') = 1$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}_s^*$, we conclude that $\mathsf{pk}_s' = \overline{\mathsf{pk}}_s^*$. Therefore, if the flag event occurs and $\mathcal{C}$ guesses correctly, adversary $\mathcal{C}$ wins its static public key binding game. Hence,

$$\left|\mathsf{Adv}^{G_3'}(\mathcal{A}) - \mathsf{Adv}^{G_4'}(\mathcal{A})\right| \leq N \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Game G_5' (Random K_A or K_B). In this game, if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$), we replace the KEM key K_A (resp. K_B) in the tested session π and its (unique) PartialMatch session π' with an independently uniform key $\tilde{k} \leftarrow\!\!\$\, \mathcal{K}$. Similarly to the analysis of Game G_4 for case FM_1, it suffices to interchange the conditions $\pi.\mathsf{role} = \mathsf{Init}$ and $\pi.\mathsf{role} = \mathsf{Resp}$ in G_4 to obtain the analysis here. Hence, we can construct an adversary $\mathcal{B}_1$ against the IND-SCCA security of AFS-KEM such that

$$\left|\mathsf{Adv}^{G_4'}(\mathcal{A}) - \mathsf{Adv}^{G_5'}(\mathcal{A})\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_1).$$

Game G_6' (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, $\mathsf{K}^B_{\mathsf{ENC}}$). In this game, we replace the keys $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$ in the sessions π and π' with independently sampled uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}, \hat{\mathsf{K}}^A_{\mathsf{ENC}}, \hat{\mathsf{K}}^B_{\mathsf{ENC}}$, respectively. Similar to the analysis of Game G_5 in case FM_1, we obtain

$$\left|\mathsf{Adv}^{G_5'}(\mathcal{A}) - \mathsf{Adv}^{G_6'}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}).$$

Analysis of G_6'. In Game G_6', the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random from the adversary's view. Therefore,

$$\mathsf{Adv}^{G_6'}(\mathcal{A}) = 0.$$

Combining all the bounds above completes the proof for case FM_2. □

Lemma E.3. *With the notation and assumptions of Theorem 5.3, there exist adversaries $\mathcal{B}_3$, $\mathcal{D}$, $\mathcal{F}$, and $\mathcal{Z}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FM}_5\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le NS\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 2\,\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})\Big) + 2NS^2\Big(2\,\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + \delta\Big).$$

$$\mathsf{Adv}^{\mathsf{FM}_6\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le NS\Big(2\,\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3) + \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 2\,\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})\Big) + 2NS^2\Big(2\,\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + \delta\Big).$$

Proof. In this lemma, we provide a unified analysis by considering $\mathsf{Adv}^{(\mathsf{FM}_5\vee\mathsf{FM}_6)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$ and showing that the event $\mathsf{FM}_5\vee\mathsf{FM}_6$ occurs only with negligible probability. In particular, it holds that $\mathsf{Adv}^{\mathsf{FM}_5\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le \mathsf{Adv}^{(\mathsf{FM}_5\vee\mathsf{FM}_6)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{FM}_6\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le \mathsf{Adv}^{(\mathsf{FM}_5\vee\mathsf{FM}_6)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$. The proof of explicit authentication for pMAKE (Theorem E.5) largely follows the analysis of the cases $\mathsf{FM}_5 \vee \mathsf{FM}_6$, except for Game G_6. In Game G_6 here, we show that a PartialMatch session of the tested session must exist, whereas in the proof of Theorem E.5, we show that a matching session must exist.

Game G_0. This game corresponds to the original game for $\mathsf{Adv}^{\mathsf{FM}_5\vee\mathsf{FM}_6\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$ and we assume that the freshness condition $\mathsf{FM}_5 \vee \mathsf{FM}_6$ holds for the tested session. Let π denote the tested session in the game. Let π' denote the (unique, if it exists) session such that $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) = (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$ (resp. $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) = (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$) if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. if $\pi.\mathsf{role} = \mathsf{Init}$). Notably, we have

$$\mathsf{Adv}^{\mathsf{FM}_5\vee\mathsf{FM}_6\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Game G_1 (Abort if Collision in K_A or K_B). In this game, we define the COLL event as follows: π' exists and $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) \neq (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$ (resp. $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) \neq (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$) but $\pi.\mathsf{K}_B = \pi'.\mathsf{K}_B$ (resp. $\pi.\mathsf{K}_A = \pi'.\mathsf{K}_A$) when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$), then the COLL event occurs. If the COLL event occurs, we abort the game. Since Games G_0 and G_1 are identical unless the COLL event occurs, we have

$$\left|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})\right| \le \Pr[\mathsf{COLL} : G_0].$$

For simplicity, we consider the case $\pi.\mathsf{role} = \mathsf{Resp}$. If the COLL event occurs in this case, then $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) \neq (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$ while $\pi.\mathsf{K}_B = \pi'.\mathsf{K}_B$. We consider the following two subcases.

- **Case 1:** $\pi'.\mathsf{cpk}_B \neq \pi.\mathsf{cpk}_B$. We construct an adversary $\mathcal{F}$ against the Key-Coll game for AFS-KEM as follows. Adversary $\mathcal{F}$ first chooses $g = 1$ and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \mathsf{seed}^*_e)$ from its challenger. It then perfectly simulates Game G_0 for $\mathcal{A}$, except that $\mathcal{F}$ first guesses the identity of $\pi.\mathsf{actor}$ uniformly from the N users and guesses the session π', π uniformly from the S sessions. Notably, the probability that $\mathcal{F}$ guesses both correctly is $1/(NS^2)$. Adversary $\mathcal{F}$ then sets $(\mathsf{pk}^*_s, \mathsf{sk}^*_s)$ as the long-term key pair of $\pi.\mathsf{actor}$, and embeds $(\mathsf{cpk}_B = \mathsf{cpk}^*, \mathsf{seed}_B = \mathsf{seed}^*_e)$ in session π.
  Next, $\mathcal{F}$ outputs the composite public key cpk'_B in session π' and forwards it to its challenger. Upon receiving the corresponding (ct', K') from the challenger, $\mathcal{F}$ continues the simulation by embedding $(\mathsf{ct}_B = \mathsf{ct}', \mathsf{K}_B = K')$ into session π'. Finally, when session π receives a ciphertext ct'_B, $\mathcal{F}$ outputs ct'_B to its challenger. If the event [COLL : Case 1] occurs, then $\mathsf{cpk}'_B \neq \mathsf{cpk}^*$ and $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}'_B) = \mathsf{K}_B$. Hence, adversary $\mathcal{F}$ wins the Key-Coll game, and we have $\Pr[\mathsf{COLL} : \text{Case 1}] \le NS^2 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$.
- **Case 2:** $\pi'.\mathsf{cpk}_B = \pi.\mathsf{cpk}_B$ but $\pi'.\mathsf{ct}_B \neq \pi.\mathsf{ct}_B$. We again construct an adversary $\mathcal{F}$ against the Key-Coll game for AFS-KEM as follows. Adversary $\mathcal{F}$[28] chooses $g = 3$ and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \mathsf{seed}^*_e, K^*, \mathsf{ct}^*)$ from its challenger.
  As before, $\mathcal{F}$ guesses the identity of $\pi.\mathsf{actor}$ uniformly from the N users and guesses the session π', π uniformly from the S sessions, with success probability $1/(NS^2)$. It sets $(\mathsf{pk}^*_s, \mathsf{sk}^*_s)$ as the long-term key pair of $\pi.\mathsf{actor}$, sets $\mathsf{cpk}_B = \mathsf{cpk}^*, \mathsf{seed}_B = \mathsf{seed}^*_e$ in session π, and embeds $\mathsf{K}_B = K^*, \mathsf{ct}_B = \mathsf{ct}^*$ into session π'. When session π receives a ciphertext $\mathsf{ct}'_B \neq \mathsf{ct}^*$, adversary $\mathcal{F}$ outputs ct'_B to its challenger. If the event [COLL : Case 2] occurs, then $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}'_B) = K^*$, and thus $\mathcal{F}$ wins the Key-Coll game. Therefore, $\Pr[\mathsf{COLL} : \text{Case 2}] \le NS^2 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$.

[28] For simplicity, we reuse the notation $\mathcal{F}$ for the adversary.

Combining the two subcases, we obtain $\Pr[\mathsf{COLL} : \pi.\mathsf{role} = \mathsf{Resp}] \leq 2NS^2 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$. By a symmetric analysis, the same bound holds for the case $\pi.\mathsf{role} = \mathsf{Init}$, that is, $\Pr[\mathsf{COLL} : \pi.\mathsf{role} = \mathsf{Init}] \leq 2NS^2 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$. Therefore, we have

$$\left|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})\right| \leq 4NS^2 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}).$$

Game G_2 (KEM Correctness). In this game, we define the event ERR_1 to occur if the decapsulation of ct_A (resp. ct_B) in π' (if it exists) when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. when $\pi.\mathsf{role} = \mathsf{Init}$) does not equal the corresponding encapsulated key. Moreover, if π' is further to be a $\mathsf{PartialMatch}$ session of π, we define the event ERR_2 to occur if the decapsulation of ct_B (resp. ct_A) in π' (if it exists) when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. when $\pi.\mathsf{role} = \mathsf{Init}$) does not equal the corresponding encapsulated key.

We first bound $\Pr[\mathsf{ERR}_1]$ using the δ-correctness of $\mathsf{AFS\text{-}KEM}$. The reduction guesses the identity of $\pi'.\mathsf{actor}$ and the sessions π and π' uniformly at random, with success probability at least $1/(NS^2)$, and embeds the δ-correctness experiment into the corresponding KEM instance $(\mathsf{cpk}_A, \mathsf{ct}_A)$ (resp. $(\mathsf{cpk}_B, \mathsf{ct}_B)$) of π and π' when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. when $\pi.\mathsf{role} = \mathsf{Init}$). Since $\mathsf{AFS\text{-}KEM}$ is δ-correct, it follows that $\Pr[\mathsf{ERR}_1] \leq NS^2\delta$. A similar analysis applies to ERR_2, yielding $\Pr[\mathsf{ERR}_2] \leq NS^2\delta$. By a union bound, games G_{A2} and G_{A3} differ with probability at most $2NS^2\delta$. Therefore,

$$\left|\mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})\right| \leq 2NS^2\delta.$$

Game G_3 (Guessing $\pi^{\mathsf{Test}}.\mathsf{peer}$, π^{Test}). In game G_2, we first guess the peer identity of the test session, and further guess the test session π^{Test}. If either guess is incorrect, the game is aborted. Concretely, we sample P^* uniformly from the N parties, and sample one session instance π uniformly from the at most S sessions. If the guess is incorrect (i.e., $P^* \neq \pi^{\mathsf{Test}}.\mathsf{peer}$ or $\pi \neq \pi^{\mathsf{Test}}$), the game is aborted. The probability of guessing correctly is $1/(NS)$, therefore

$$\mathsf{Adv}^{G_2}(\mathcal{A}) \leq NS \cdot \mathsf{Adv}^{G_3}(\mathcal{A}).$$

Game G_4 (New win condition). In this game, we change the condition for the adversary to win. An instance of G_4 is won by the adversary $\mathcal{A}$ if and only if the tested session π accepts and nothing makes the instance of the game invalid (as per the conditions of G_3). We stop the game at the moment when π accepts. Note that in G_4, the Test query is not requested for π, and in all valid instances and P^* remains uncorrupted. It is evident that any adversary that wins the game G_3 must also win the game G_4. Thus, we have

$$\mathsf{Adv}^{G_3}(\mathcal{A}) \leq \mathsf{Adv}^{G_4}(\mathcal{A}).$$

Game G_5 (Random K_A or K_B). In this game, we replace the KEM key K_A (resp. K_B) if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. if $\pi.\mathsf{role} = \mathsf{Init}$) in the tested session π with an independently uniform key $\tilde{k} \leftarrow\!\!\$\ \mathcal{K}$. Recall that π' denote the (unique, if it exists) session such that $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) = (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$(resp $.(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) = (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$) if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. if $\pi.\mathsf{role} = \mathsf{Init}$). If such π' exists, we also replace the corresponding KEM key K_A (resp. K_B) in π' with the same key $\tilde{k}$.

For any adversary $\mathcal{A}$ that can distinguish G_4 from G_5, we construct an adversary $\mathcal{B}_3$ against the $\mathsf{IND\text{-}LCK\text{-}SCCA}$ security of $\mathsf{AFS\text{-}KEM}$. Adversary $\mathcal{B}_3$ first receives pk^*_s from its challenger, sets pk^*_s as the long-term public key of P^*, and generates long-term key pairs $(\mathsf{pk}_U, \mathsf{sk}_U)$ honestly for all other parties. It then simulates the view of $\mathcal{A}$ as in G_5, except for the following changes.

- For each session $\hat{\pi}$ of P^* with $\hat{\pi}.\mathsf{role} = \mathsf{Init}$ (resp. Resp), $\mathcal{B}_3$ independently queries $\mathcal{O}^{\mathsf{EKeyGen}}$ to obtain a fresh tuple $(\mathsf{cpk}, \mathsf{pk}_e, \mathsf{sk}_e, \mathsf{seed})$ for the session, and sets $\mathsf{cpk}_A = \mathsf{cpk}$ (resp. $\mathsf{cpk}_B = \mathsf{cpk}$). When tested session π receives the corresponding composite public key cpk', $\mathcal{B}_3$ sets $\mathsf{cpk}^* = \mathsf{cpk}'$ and forwards it to its challenger. After $\mathcal{B}_3$ receives $(\mathsf{pk}^*_s, \mathsf{ct}^*, \mathsf{K}^*_b)$ from its challenger, it embeds $\mathsf{ct}_A = \mathsf{ct}^*$ (resp. $\mathsf{ct}_B = \mathsf{ct}^*$) into the tested session π if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$), and uses K^*_b as the corresponding KEM key K_A (resp. K_B) in π. If the π' session exists, then $\mathcal{B}_3$ also sets K^*_b as the corresponding KEM key K_A (resp. K_B) in π' if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$).
  Moreover, when P^* receives a KEM ciphertext ct for a session $\hat{\pi} \neq \pi'$, the adversary $\mathcal{B}_3$ queries the corresponding decryption oracle $\mathcal{O}^{\mathsf{Dec}}(\mathsf{cpk}, \mathsf{ct})$, where cpk is the composite public key embedded into session $\hat{\pi}$ via $\mathcal{O}^{\mathsf{EKeyGen}}$, to obtain the KEM key K_A (resp. K_B) if $\hat{\pi}.\mathsf{role} = \mathsf{Init}$ (resp. $\hat{\pi}.\mathsf{role} = \mathsf{Resp}$). It then continues the simulation of session $\hat{\pi}$ honestly.

– When π receives $(\mathsf{id}_{P^*}, \mathsf{c}_{\mathsf{seed}}^{P^*})$, $\mathcal{B}_3$ decrypts $\mathsf{c}_{\mathsf{seed}}^{P^*}$ using K_b^* to obtain seed_{P^*} and outputs seed_{P^*} to its challenger. Since π is an accepted session, we have $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_{P^*}) = 1$, where $\mathsf{pk}'_{P^*} \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}', \mathsf{pk}_e^{P^*})$ and $(\mathsf{pk}_e^{P^*}, \cdot) = \mathsf{EKeyGen}_{\mathsf{c}}(\mathsf{cpk}', \mathsf{seed}_{P^*})$. Therefore, $\mathsf{pk}'_{P^*} = \mathsf{pk}_s^*$ and seed_{P^*} is a valid output for $\mathcal{B}_3$.

When $b = 0$, adversary $\mathcal{B}_3$ perfectly simulates Game G_4, i.e., $\Pr[\mathcal{B}_3 \Rightarrow 1 : b = 0] = \Pr[G_4^{\mathcal{A}} \Rightarrow 1]$. When $b = 1$, $\mathcal{B}_3$ perfectly simulates G_5, i.e., $\Pr[\mathcal{B}_3 \Rightarrow 1 : b = 1] = \Pr[G_5^{\mathcal{A}} \Rightarrow 1]$. Therefore,

$$\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq 2\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_3).$$

Game G_6 (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, $\mathsf{K}^B_{\mathsf{ENC}}$). In this game, we replace the keys $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$ in the session π with independently sampled uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}, \hat{\mathsf{K}}^A_{\mathsf{ENC}}, \hat{\mathsf{K}}^B_{\mathsf{ENC}}$, respectively. If π' exists and is the PartialMatch session of π, we use the same uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}, \hat{\mathsf{K}}^A_{\mathsf{ENC}}, \hat{\mathsf{K}}^B_{\mathsf{ENC}}$ as in π. Otherwise, if π' exists but is not a PartialMatch session of π, we use independently sampled uniform keys $\overline{\mathsf{K}}_{\mathsf{SESSION}}, \overline{\mathsf{K}}^A_{\mathsf{ENC}}, \overline{\mathsf{K}}^B_{\mathsf{ENC}}$ in π'. Notably, after Game G_1, if π' exists but is not a PartialMatch session of π, then we must have $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) \neq (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$ (resp. $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) \neq (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$), but $\pi.\mathsf{K}_B \neq \pi'.\mathsf{K}_B$ (resp. $\pi.\mathsf{K}_A \neq \pi'.\mathsf{K}_A$) when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$).

For any adversary $\mathcal{A}$ that can distinguish G_5 from G_6, we construct an adversary $\mathcal{Z}$ against the dual-PRF security of the underlying PRF. Adversary $\mathcal{Z}$ is given oracle access to $O_b(\cdot)$, where O_0 implements the real PRF and O_1 implements a uniformly random function. Adversary $\mathcal{Z}$ simulates the view of $\mathcal{A}$ exactly as in Game G_5, except that it computes $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}})$ for the sessions π and π' (if it exists) by querying its oracle $O_b(\mathsf{K}_B)$ (resp. $O_b(\mathsf{K}_A)$) if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$). When $b = 0$, the oracle O_0 corresponds to the real PRF, and the simulation is identical to Game G_5. When $b = 1$, the oracle O_1 is a truly random function, and the simulation is identical to Game G_6. Thus, $\Pr[\mathcal{Z} \Rightarrow 1 \mid b = 0] = \Pr[G_5^{\mathcal{A}} \Rightarrow 1]$ and $\Pr[\mathcal{Z} \Rightarrow 1 \mid b = 1] = \Pr[G_6^{\mathcal{A}} \Rightarrow 1]$ Therefore, we obtain

$$\left|\mathsf{Adv}^{G_5}(\mathcal{A}) - \mathsf{Adv}^{G_6}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}).$$

Game G_7 (PartialMatch session). In this game, we define BAD as the event that the PartialMatch session π^{Match} of the test session π does not exist. If BAD occurs, we abort the game immediately. At this point, whenever the $\mathsf{FM}_5 \vee \mathsf{FM}_6$ event occurs, the game aborts. Therefore,

$$\mathsf{Adv}^{G_7}(\mathcal{A}) = 0.$$

It is evident that $\left|\mathsf{Adv}^{G_6}(\mathcal{A}) - \mathsf{Adv}^{G_7}(\mathcal{A})\right| \leq \Pr[\mathsf{BAD} : G_6]$. We next analyze $\Pr[\mathsf{BAD} : G_6]$. In G_6, the keys $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$ and $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$ of the tested session π are uniform from the adversary's view. Since π accepts without any FirstMatch session, these keys are never used in any other session. Therefore, $\mathsf{seed}'_A = \hat{\mathsf{K}}^A_{\mathsf{ENC}} \oplus \mathsf{c}^A_{\mathsf{seed}}$ (resp. $\mathsf{seed}'_B = \hat{\mathsf{K}}^B_{\mathsf{ENC}} \oplus \mathsf{c}^B_{\mathsf{seed}}$) computed in π is uniform from the adversary's view when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$).

We first consider the case $\pi.\mathsf{role} = \mathsf{Resp}$. We construct an adversary $\mathcal{D}$ against the RSPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{D}$ receives $(\mathsf{pk}_s^*, \mathsf{sk}_s^*)$ and perfectly simulates Game G_6 for $\mathcal{A}$, except that it sets $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$. It outputs the composite public key cpk_A in session π to its challenger. After receiving seed_e from the challenger, $\mathcal{D}$ defines $\mathsf{K}^A_{\mathsf{ENC}} = \mathsf{seed}_e \oplus \mathsf{c}^A_{\mathsf{seed}}$, where $\mathsf{c}^A_{\mathsf{seed}}$ is received in π. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}_A, \mathsf{seed}_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}_A, \mathsf{pk}'_e)$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$, we have $\mathsf{pk}'_s = \mathsf{pk}_s^*$ by the security of CA. Therefore, if the BAD event occurs in this case, then $\mathcal{D}$ wins the RSPK-Bind game.

The analysis case $\pi.\mathsf{role} = \mathsf{Init}$ is similar. In this case, $\mathcal{D}$ outputs the composite public key cpk_B in session π to its challenger. After receiving seed_e from the challenger, $\mathcal{D}$ defines $\mathsf{K}^B_{\mathsf{ENC}} = \mathsf{seed}_e \oplus \mathsf{c}^B_{\mathsf{seed}}$, where $\mathsf{c}^B_{\mathsf{seed}}$ is received in π. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}_B, \mathsf{seed}_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}_B, \mathsf{pk}'_e)$. Again, since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{P^*}, \mathsf{pk}'_s) = 1$ and $\mathsf{pk}_{P^*} = \mathsf{pk}_s^*$, we have $\mathsf{pk}'_s = \mathsf{pk}_s^*$ by the security of CA. Therefore, if the BAD event occurs in this case, then $\mathcal{D}$ wins the RSPK-Bind game. Consequently, $\Pr[\mathsf{BAD} : G_6] \leq 2\ \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})$. Thus,

$$\left|\mathsf{Adv}^{G_6}(\mathcal{A}) - \mathsf{Adv}^{G_7}(\mathcal{A})\right| \leq 2\ \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}).$$

Combining all the bounds derived above completes the proof. □

Lemma E.2. *With the notations and assumptions of the proof of Theorem 5.3, and with* PRF *modeled as a random oracle* G *to which the adversary makes at most* q_{G} *queries, there exist adversaries* $\mathcal{B}_2$, $\mathcal{C}$, $\mathcal{D}$, *and* $\mathcal{F}$, *running in time comparable to that of* $\mathcal{A}$, *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FM_3\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le 2NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|}\right) + 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ + 4N^2S^2 \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + 4N(N+1)(N+2)S^2 \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).\end{aligned}$$

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FM_4\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le 2NS^2\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{q_{\mathsf{G}}}{|\mathcal{K}|}\right) + 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ + 4N^2S^2 \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + 4N(N+1)(N+2)S^2 \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).\end{aligned}$$

Proof. We provide a unified analysis for Case FM_3 and Case FM_4, as their proofs are essentially identical. We assume that the tested session satisfies $\mathsf{FM}_3 \vee \mathsf{FM}_4$; otherwise, the adversary trivially loses the game. In particular, it holds that $\mathsf{Adv}^{\mathsf{FM_3\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le \mathsf{Adv}^{(\mathsf{FM_3}\vee\mathsf{FM_4})\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{FM_4\text{-}KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \le \mathsf{Adv}^{(\mathsf{FM_3}\vee\mathsf{FM_4})\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$.

Game G_0. This game corresponds to the original game defining $\mathsf{Adv}^{(\mathsf{FM_3}\vee\mathsf{FM_4})\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$. Thus,

$$\mathsf{Adv}^{(\mathsf{FM_3}\vee\mathsf{FM_4})\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Case Distinction. Let π denote the tested session and let π' denote its (unique) PartialMatch session. We further write π^{Init} and π^{Resp} to denote the initiator and responder sessions among $\{\pi, \pi'\}$, respectively. Next, we consider the following two cases in Game G_0. Case A (denoted by G_A): the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Init}})$. Case B (denoted by G_B): the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Resp}})$. Accordingly, we can bound the adversary's success probability by

$$\mathsf{Adv}^{G_0}(\mathcal{A}) \le \mathsf{Adv}^{G_A}(\mathcal{A}) + \mathsf{Adv}^{G_B}(\mathcal{A}).$$

Case A: the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Init}})$.

Game G_{A1} (KEM Correctness). In this game, we define ERR_1 as the event that the decapsulation of ct_A in π^{Init} does not equal the corresponding encapsulated key, and define ERR_2 analogously for ct_B in π^{Resp}. If either event occurs, we abort. We bound $\Pr[\mathsf{ERR}_1]$ by the δ-correctness of AFS-KEM. The reduction randomly guesses π^{Init}, π^{Resp}, and $\pi^{\mathsf{Init}}.\mathsf{actor}$, with success probability at least $1/(NS^2)$, and embeds the correctness-failure experiment into $(\mathsf{cpk}_A, \mathsf{ct}_A)$. Hence $\Pr[\mathsf{ERR}_1] \le NS^2\delta$. Similarly, $\Pr[\mathsf{ERR}_2] \le NS^2\delta$. By a union bound, we have

$$\left|\mathsf{Adv}^{G_A}(\mathcal{A}) - \mathsf{Adv}^{G_{A1}}(\mathcal{A})\right| \le 2NS^2\delta.$$

Game G_{A2}. In this game, let $\mathsf{K}^{\mathsf{Test}}_A = \pi^{\mathsf{Resp}}.\mathsf{K}_A$. We define the COLL event as the existence of a session $\hat{\pi} \notin \{\pi, \pi'\}$ such that $\hat{\pi}.\mathsf{K}_A = \mathsf{K}^{\mathsf{Test}}_A$. Game G_{A2} is identical to Game G_{A1}, except that the game aborts whenever the COLL event occurs. Since Games G_{A1} and G_{A2} are identical unless the COLL event occurs, we have

$$\left|\mathsf{Adv}^{G_{A1}}(\mathcal{A}) - \mathsf{Adv}^{G_{A2}}(\mathcal{A})\right| \le \Pr[\mathsf{COLL} : G_{A1}].$$

If the COLL event occurs, we consider the following two cases.

- **Case 1:** $\hat{\pi}.\mathsf{role} = \mathsf{Init}$. We construct an adversary $\mathcal{F}$ against the Key-Coll game for AFS-KEM as follows. Adversary $\mathcal{F}$ first chooses $g = 1$ and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \mathsf{seed}^*_e)$ from its challenger. Adversary $\mathcal{F}$ guesses the sessions $\hat{\pi}$, π^{Init}, and π^{Resp} uniformly over the S sessions, and guesses $\hat{\pi}.\mathsf{actor}$ uniformly over the N parties. The probability that $\mathcal{F}$ guesses correctly is at least $1/(NS^3)$.
  Adversary $\mathcal{F}$ then perfectly simulates Game G_{A1} for $\mathcal{A}$, except that it sets $(\mathsf{pk}^*_s, \mathsf{sk}^*_s)$ as the long-term key pair for $\hat{\pi}.\mathsf{actor}$ and embeds $(\mathsf{cpk}_A, \mathsf{seed}_A) = (\mathsf{cpk}^*, \mathsf{seed}^*_e)$ into session $\hat{\pi}$. It then outputs the cpk_A received by π^{Resp} to its challenger. After receiving (ct', K') from the challenger,

$\mathcal{F}$ embeds $(\mathsf{ct}_A = \mathsf{ct}', \mathsf{K}_A = K')$ into π^{Init} and π^{Resp}. Finally, when session $\hat{\pi}$ receives its ciphertext ct'_A, $\mathcal{F}$ outputs ct'_A to its challenger.
Note that $\mathcal{F}$ perfectly simulates Game G_{A1}. If the event [COLL : Case 1] occurs, then $\mathsf{cpk} \neq \mathsf{cpk}^*$ since all honestly generated cpk values are distinct, and $\mathsf{Decaps}_{\mathsf{AFS}}(\mathsf{csk}^*, \mathsf{ct}'_A) = K'$. Therefore, $\mathcal{F}$ wins the Key-Coll game, and hence $\Pr[\mathsf{COLL} : \text{Case 1}] \leq NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$.

– **Case 2:** $\hat{\pi}.\mathsf{role} = \mathsf{Resp}$. We again construct an adversary $\mathcal{F}$ against the Key-Coll game for AFS-KEM as follows. Adversary $\mathcal{F}$ first chooses $g = 2$ and receives a tuple $(\mathsf{cpk}^*, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \mathsf{seed}^*_e, K^*, \mathsf{ct}^*)$ from its challenger. It guesses the sessions $\hat{\pi}$, π^{Init}, and π^{Resp} uniformly over the S sessions, and guesses $\pi^{\mathsf{Init}}.\mathsf{actor}$ uniformly over the N parties. The probability of guessing correctly is at least $1/(NS^3)$. Adversary $\mathcal{F}$ then perfectly simulates Game G_{A1} for $\mathcal{A}$, except that it sets $(\mathsf{pk}^*_s, \mathsf{sk}^*_s)$ as the long-term key pair for $\pi^{\mathsf{Init}}.\mathsf{actor}$, embeds $(\mathsf{cpk}_A, \mathsf{seed}_A) = (\mathsf{cpk}^*, \mathsf{seed}^*_e)$ into session π^{Init}, and embeds $(\mathsf{ct}_A, \mathsf{K}_A) = (\mathsf{ct}^*, K^*)$ into session π^{Resp}. When session $\hat{\pi}$ receives its composite public key cpk'_A, $\mathcal{F}$ forwards cpk'_A to its challenger and receives (ct', K'). It then embeds $(\mathsf{ct}_A = \mathsf{ct}', \mathsf{K}_A = K')$ into session $\hat{\pi}$.
Adversary $\mathcal{F}$ perfectly simulates Game G_{A1} for this case. If the event [COLL : Case 2] occurs, then $K' = K^*$. Thus, $\mathcal{F}$ wins the Key-Coll game, and therefore $\Pr[\mathsf{COLL} : \text{Case 2}] \leq NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$.

Combining the two cases, we obtain

$$\left|\mathsf{Adv}^{G_{A1}}(\mathcal{A}) - \mathsf{Adv}^{G_{A2}}(\mathcal{A})\right| \leq 2NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}).$$

Game G_{A3} (Guessing π^{Test}, $\pi^{\mathsf{PartialMatch}}$, and Actor). In this game, we guess the test session π, its corresponding PartialMatch session π', and the identity of $\pi^{\mathsf{Init}}.\mathsf{actor}$. Concretely, we sample two session instances π and π' uniformly from the S sessions, and sample a party P^* uniformly from the N parties. The game aborts if $\pi \neq \pi^{\mathsf{Test}}$ or $\pi' \neq \pi^{\mathsf{PartialMatch}}$, or $\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Init}$ and $P^* \neq \pi^{\mathsf{Test}}.\mathsf{actor}$, or if $\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Resp}$ and $P^* \neq \pi^{\mathsf{PartialMatch}}.\mathsf{actor}$. The probability of guessing correctly is $1/(NS^2)$. Therefore,

$$\mathsf{Adv}^{G_{A2}}(\mathcal{A}) \leq NS^2 \cdot \mathsf{Adv}^{G_{A3}}(\mathcal{A}).$$

Game G_{A4} (Random $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$). In this game, we replace $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, and $\mathsf{K}^B_{\mathsf{ENC}}$ in the guessed sessions π and π' with uniformly random values. Define QUERY as the event that $\mathcal{A}$ queries G on input $(\mathsf{K}^{\mathsf{Test}}_A, \mathsf{K}^{\mathsf{Test}}_B)$, where $\mathsf{K}^{\mathsf{Test}}_A$ and $\mathsf{K}^{\mathsf{Test}}_B$ denote the encapsulated keys K_A and K_B in sessions π and π', respectively. Notably, after Game G_{A2}, every other session $\hat{\pi} \notin \{\pi, \pi'\}$ satisfies $\hat{\pi}.\mathsf{K}_A \neq \mathsf{K}^{\mathsf{Test}}_A$. Then Games G_{A3} and G_{A4} are identical unless the event QUERY occurs, and we have

$$\left|\mathsf{Adv}^{G_{A3}}(\mathcal{A}) - \mathsf{Adv}^{G_{A4}}(\mathcal{A})\right| \leq \Pr[\mathsf{QUERY} : G_{A4}].$$

Game G_{A5} (Abort on BAD). In this game, we define the event BAD as follows. The event BAD occurs if either (i) the responder session π^{Resp} accepts the session key but the tuple $(\mathsf{id}_A, \mathsf{c}^A_{\mathsf{seed}})$ sent by π^{Init} is different from $(\mathsf{id}'_A, \mathsf{c}'^A_{\mathsf{seed}})$ received by π^{Resp}, or (ii) the initiator session π^{Init} accepts the session key but π^{Resp} is not the matching session of π^{Init}. If the event BAD occurs, we abort the game. Notably, after this game hop, π^{Resp} accepts the session key only if it receives the honest third message from π^{Init}, and π^{Init} accepts the session key only if π^{Resp} is its matching session.

At this point, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random from the adversary's view. Moreover, there may be one accepted session using the same session key, but such a session must be the matching session of π. All session keys held by other accepted sessions are independent of $\mathsf{K}_{\mathsf{SESSION}}$. Therefore,

$$\mathsf{Adv}^{G_{A5}}(\mathcal{A}) = 0.$$

Since Games G_{A4} and G_{A5} are identical unless the BAD event occurs, we have $\left|\mathsf{Adv}^{G_{A4}}(\mathcal{A}) - \mathsf{Adv}^{G_{A5}}(\mathcal{A})\right| \leq \Pr[\mathsf{BAD} : G_{A4}]$. Similarly, $\left|\Pr[\mathsf{QUERY} : G_{A4}] - \Pr[\mathsf{QUERY} : G_{A5}]\right| \leq \Pr[\mathsf{BAD} : G_{A4}]$. Next, we analyze $\Pr[\mathsf{BAD} : G_{A4}]$. We consider the following cases.

Case 1: The responder session π^{Resp} accepts, but the tuple $(\mathsf{id}_A = P^*, \mathsf{c}^A_{\mathsf{seed}})$ sent by π^{Init} is different from the tuple $(\mathsf{id}'_A, \mathsf{c}'^A_{\mathsf{seed}})$ received by π^{Resp}. We consider the following two subcases.

– **Case 1.a:** If $\mathsf{id}_A \neq \mathsf{id}'_A$, we construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_{A4} for $\mathcal{A}$, except that it first guesses the id'_A uniformly over the N parties. Notably, $\mathcal{C}$ guesses correctly with probability at least $1/N$. Then $\mathcal{C}$ sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$ and $\mathsf{pk}_{\mathsf{id}'_A} = \overline{\mathsf{pk}}_s^*$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into the session π^{Init}. When π^{Resp} receives $(\mathsf{id}'_A, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Let $(\mathsf{pk}_e^{A'}, \mathsf{sk}_e^{A'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_A)$ and $\mathsf{pk}'_A \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{A'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}'_A, \mathsf{pk}'_A) = 1$ and $\mathsf{pk}_{\mathsf{id}'_A} = \overline{\mathsf{pk}}_s^*$, we have $\mathsf{pk}'_A = \overline{\mathsf{pk}}_s^*$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 1.a}] \leq N \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
– **Case 1.b:** If $\mathsf{id}_A = \mathsf{id}'_A$ and $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$, we also construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}_e^*, \mathsf{pk}_s^*, \mathsf{sk}_s^*, \overline{\mathsf{pk}}_s^*, \overline{\mathsf{sk}}_s^*)$ and perfectly simulates Game G_{A4} for $\mathcal{A}$. $\mathcal{C}$ then sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$ and embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π^{Init}. When π^{Resp} receives $(\mathsf{id}_A, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Let $(\mathsf{pk}_e^{A'}, \mathsf{sk}_e^{A'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_A)$ and $\mathsf{pk}'_A \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{A'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_A, \mathsf{pk}'_A) = 1$ and $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$, we have $\mathsf{pk}'_A = \mathsf{pk}_s^*$. Moreover, since $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$ and the OTP key used to recover the seed is the same, we have $\mathsf{seed}'_A \neq \mathsf{seed}_e^*$. Therefore, if the BAD event occurs in this case, then $\mathcal{C}$ wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 1.b}] \leq \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Case 2: The initiator session π^{Init} accepts but π^{Resp} is not the matching session of π^{Init}. First, suppose that π^{Resp} aborts, i.e., it does not accept its session key, and denote this event by $[\mathsf{BAD} : \text{Case 2, abort}]$. Then the corresponding encryption key $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$ is uniform from the adversary's view and is never used in any other session. Thus, the value $\mathsf{seed}'_B = \hat{\mathsf{K}}^B_{\mathsf{ENC}} \oplus \mathsf{c}^B_{\mathsf{seed}}$ computed in π^{Init} is uniform from the adversary's view. In this case, we construct an adversary $\mathcal{D}$ against the RSPK-Bind security of AFS-KEM as follows.

Adversary $\mathcal{D}$ receives $(\mathsf{pk}_s^*, \mathsf{sk}_s^*)$ and perfectly simulates Game G_{A4} for $\mathcal{A}$, except that it first guesses the peer identity B^* of π^{Init}, with success probability at least $1/N$. Then $\mathcal{D}$ sets $\mathsf{pk}_{B^*} = \mathsf{pk}_s^*$ and outputs the composite public key cpk_B in session π^{Init} to its challenger. After receiving seed_e from the challenger, $\mathcal{D}$ defines $\mathsf{K}^B_{\mathsf{ENC}} = \mathsf{seed}_e \oplus \mathsf{c}^B_{\mathsf{seed}}$, where $\mathsf{c}^B_{\mathsf{seed}}$ is received in π^{Init}. Let $(\mathsf{pk}'_e, \mathsf{sk}'_e) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}_B, \mathsf{seed}_e)$ and $\mathsf{pk}'_s \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}_B, \mathsf{pk}'_e)$. Since π^{Init} accepts, the extracted public key is accepted for the peer identity B^*, i.e., $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_{B^*}, \mathsf{pk}'_s) = 1$. Since $\mathsf{pk}_{B^*} = \mathsf{pk}_s^*$, we have $\mathsf{pk}'_s = \mathsf{pk}_s^*$ by the security of CA. Therefore, if the BAD event occurs in this case and $\mathcal{D}$ guesses correctly, then $\mathcal{D}$ wins the RSPK-Bind game. Consequently, $\Pr[\mathsf{BAD} : \text{Case 2, abort}] \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})$.

It remains to consider the case where π^{Resp} accepts. In this case, BAD implies that the adversary has tampered with either the third or the fourth protocol message between π^{Init} and π^{Resp}, but both sessions accept. Specifically, the tuple $(\mathsf{id}_A, \mathsf{c}^A_{\mathsf{seed}}, \mathsf{id}_B, \mathsf{c}^B_{\mathsf{seed}})$ maintained by π^{Init} differs from $(\mathsf{id}'_A, \mathsf{c}'^A_{\mathsf{seed}}, \mathsf{id}'_B, \mathsf{c}'^B_{\mathsf{seed}})$ maintained by π^{Resp}. We consider the following four subcases.

– **Case 2.a:** If $\mathsf{id}_A \neq \mathsf{id}'_A$, then similarly to Case 1.a, we can construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM such that $\Pr[\mathsf{BAD} : \text{Case 2.a}] \leq N \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
– **Case 2.b:** If $\mathsf{id}_B \neq \mathsf{id}'_B$, then a similar SPK-Bind reduction to that in Case 1.a applies. Notably, in this case, the reduction needs to guess both id_B and id'_B. Hence, $\Pr[\mathsf{BAD} : \text{Case 2.b}] \leq N^2 \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
– **Case 2.c:** If $\mathsf{id}_A = \mathsf{id}'_A$ and $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$, then the same SPK-Bind reduction as in Case 1.b applies. Hence, $\Pr[\mathsf{BAD} : \text{Case 2.c}] \leq \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
– **Case 2.d:** If $\mathsf{id}_B = \mathsf{id}'_B$ and $\mathsf{c}^B_{\mathsf{seed}} \neq \mathsf{c}'^B_{\mathsf{seed}}$, then the same SPK-Bind reduction as in Case 1.b applies. Notably, in this case, the reduction needs to guess id_B. Hence, $\Pr[\mathsf{BAD} : \text{Case 2.d}] \leq N \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

For simplicity, we reuse the notation $\mathcal{C}$ for the SPK-Bind reductions above. Combining the analysis for Case 1 and Case 2, we obtain $\Pr[\mathsf{BAD} : G_{A4}] \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$. Therefore,

$$\left|\mathsf{Adv}^{G_{A4}}(\mathcal{A}) - \mathsf{Adv}^{G_{A5}}(\mathcal{A})\right| \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

$$\left|\Pr[\mathsf{QUERY} : G_{A4}] - \Pr[\mathsf{QUERY} : G_{A5}]\right| \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Recall that $\mathsf{Adv}^{G_{A5}}(\mathcal{A}) = 0$. It remains to analyze $\Pr[\mathsf{QUERY} : G_{A5}]$.

Game G_{A6} (Randomized $\mathsf{c}^A_{\mathsf{seed}}$). In this game, we modify the computation of $\mathsf{c}^A_{\mathsf{seed}} = \mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{seed}_A$ in π^{Init} by replacing it with $\hat{\mathsf{c}}^A_{\mathsf{seed}} = \mathsf{K}^A_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_A$, where $\hat{\mathsf{seed}}_A \leftarrow\!\$\ \{0,1\}^\lambda$ is sampled uniformly and independently of seed_A. Importantly, after Game G_{A5}, the responder session π^{Resp} accepts the session key only if it receives the honest third message from π^{Init}; otherwise, it aborts. Since $\mathsf{K}^A_{\mathsf{ENC}}$ is uniformly random and independent of the adversary's view in Game G_{A5}, the values $\mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{seed}_A$ and $\mathsf{K}^A_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_A$ are identically distributed from the adversary's view. Hence, Games G_{A5} and G_{A6} are identical from the adversary's view. Therefore,

$$\Pr[\mathsf{QUERY} : G_{A5}] = \Pr[\mathsf{QUERY} : G_{A6}].$$

Game G_{A7} (Random $\mathsf{K}^{\mathsf{Test}}_A$). In this game, we replace the KEM key $\mathsf{K}^{\mathsf{Test}}_A$ in π and π' with a uniformly random key. To bound $\left|\Pr[\mathsf{QUERY} : G_{A6}] - \Pr[\mathsf{QUERY} : G_{A7}]\right|$, we construct an IND-ECPA adversary $\mathcal{B}_2$ against AFS-KEM. Specifically, $\mathcal{B}_2$ receives $(\mathsf{pk}^*_s, \mathsf{pk}^*_e, \mathsf{cpk}^*, \mathsf{sk}^*_s, \mathsf{ct}^*, \mathsf{K}^*_b)$ from its challenger. Adversary $\mathcal{B}_2$ honestly simulates the view of $\mathcal{A}$ as in Game G_{A6}, using the party P^* guessed in Game G_{A3}. It sets pk^*_s as the public key of P^* and embeds $\mathsf{cpk}_A = \mathsf{cpk}^*$ in π^{Init} and $\mathsf{ct}_A = \mathsf{ct}^*$ in π^{Resp}. Moreover, $\mathcal{B}_2$ uses K^*_b as the KEM key K_A in both π and π'. When $b = 0$, $\mathcal{B}_2$ perfectly simulates Game G_{A6}, and when $b = 1$, $\mathcal{B}_2$ perfectly simulates Game G_{A7}. That is, $\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 0] = \Pr[\mathsf{QUERY} : G_{A6}]$ and $\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 1] = \Pr[\mathsf{QUERY} : G_{A7}]$. Therefore,

$$\left|\Pr[\mathsf{QUERY} : G_{A6}] - \Pr[\mathsf{QUERY} : G_{A7}]\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2).$$

Analysis of $\Pr[\mathsf{QUERY} : G_{A7}]$. In this game, the key $\mathsf{K}^{\mathsf{Test}}_A$ is independent of ct and cpk in sessions π and π', and is uniformly random from the adversary's view. It follows that

$$\Pr[\mathsf{QUERY} : G_{A7}] \leq \frac{q_{\mathsf{G}}}{|\mathcal{K}|}.$$

Combining all the bounds derived above completes the proof of Case A.

Case B: the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Resp}})$.

The analysis of Case B is analogous to that of Case A. We therefore only provide a proof sketch highlighting the differences.

Game G_{B1} (KEM Correctness). In this game, we abort if the decapsulation of ct_A or ct_B in π^{Init} does not equal the corresponding encapsulated key. By an argument analogous to that for Game G_{A1}, we obtain $\left|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_{B1}}(\mathcal{A})\right| \leq 2NS^2\delta$.

Game G_{B2} (Key Collision). In this game, let $\mathsf{K}^{\mathsf{Test}}_B = \pi^{\mathsf{Resp}}.\mathsf{K}_B$. We define the event COLL as the existence of a session $\hat{\pi} \notin \{\pi, \pi'\}$ such that $\hat{\pi}.\mathsf{K}_B = \mathsf{K}^{\mathsf{Test}}_B$. Game G_{B2} is identical to Game G_{B1}, except that it aborts whenever COLL occurs. By an argument analogous to that for Game G_{A2}, we have $\left|\mathsf{Adv}^{G_{B1}}(\mathcal{A}) - \mathsf{Adv}^{G_{B2}}(\mathcal{A})\right| \leq 2NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F})$.

Game G_{B3} (Guessing π^{Test}, $\pi^{\mathsf{PartialMatch}}$, and Actor). In this game, we guess the test session π, its corresponding PartialMatch session π', and the identity of $\pi^{\mathsf{Resp}}.\mathsf{actor}$. Concretely, we sample two session instances π and π' uniformly from the S sessions, and sample a party P^* uniformly from the N parties. The game aborts if $\pi \neq \pi^{\mathsf{Test}}$ or $\pi' \neq \pi^{\mathsf{PartialMatch}}$, or $\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Resp}$ and $P^* \neq \pi^{\mathsf{Test}}.\mathsf{actor}$, or if $\pi^{\mathsf{Test}}.\mathsf{role} = \mathsf{Init}$ and $P^* \neq \pi^{\mathsf{PartialMatch}}.\mathsf{actor}$. Equivalently, P^* is a guess for $\pi^{\mathsf{Resp}}.\mathsf{actor}$. The probability of guessing correctly is $1/(NS^2)$. Therefore,

$$\mathsf{Adv}^{G_{B2}}(\mathcal{A}) \leq NS^2 \cdot \mathsf{Adv}^{G_{B3}}(\mathcal{A}).$$

Game G_{B4} (Random $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$). This game applies the same modification as in Game G_{A4}. Specifically, we replace $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, and $\mathsf{K}^B_{\mathsf{ENC}}$ in the guessed sessions π and π' with independently sampled uniform values. Define QUERY as the event that $\mathcal{A}$ queries G on input $(\mathsf{K}^{\mathsf{Test}}_A, \mathsf{K}^{\mathsf{Test}}_B)$, where $\mathsf{K}^{\mathsf{Test}}_A$ and $\mathsf{K}^{\mathsf{Test}}_B$ denote the encapsulated keys K_A and K_B in sessions π

and π', respectively. Then Games G_{B3} and G_{B4} are identical unless QUERY occurs. Therefore, $\left|\mathsf{Adv}^{G_{B3}}(\mathcal{A}) - \mathsf{Adv}^{G_{B4}}(\mathcal{A})\right| \leq \Pr[\mathsf{QUERY} : G_{B4}]$.

Game G_{B5} (Abort on BAD). This game applies the same modification as in Game G_{A5}. We define the event BAD as follows. The event BAD occurs if either (i) the responder session π^{Resp} accepts the session key but the tuple $(\mathsf{id}_A, \mathsf{c}^A_{\mathsf{seed}})$ sent by π^{Init} is different from $(\mathsf{id}'_A, \mathsf{c}'^A_{\mathsf{seed}})$ received by π^{Resp}, or (ii) the initiator session π^{Init} accepts the session key but π^{Resp} is not the matching session of π^{Init}. If BAD occurs, the game aborts. By an argument analogous to that for Game G_{A5}, we obtain $\left|\mathsf{Adv}^{G_{B4}}(\mathcal{A}) - \mathsf{Adv}^{G_{B5}}(\mathcal{A})\right| \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$. Similarly, $\left|\Pr[\mathsf{QUERY} : G_{B4}] - \Pr[\mathsf{QUERY} : G_{B5}]\right| \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

At this point, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random from the adversary's view. Moreover, there may be one accepted session using the same session key, but such a session must be the matching session of π. All session keys held by other accepted sessions are independent of $\mathsf{K}_{\mathsf{SESSION}}$. Therefore, $\mathsf{Adv}^{G_{B5}}(\mathcal{A}) = 0$. It remains to bound $\Pr[\mathsf{QUERY} : G_{B5}]$.

Game G_{B6} (Randomized $\mathsf{c}^B_{\mathsf{seed}}$). In this game, we modify the computation of $\mathsf{c}^B_{\mathsf{seed}} = \mathsf{K}^B_{\mathsf{ENC}} \oplus \mathsf{seed}_B$ in π^{Resp} by replacing it with $\hat{\mathsf{c}}^B_{\mathsf{seed}} = \mathsf{K}^B_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_B$, where $\hat{\mathsf{seed}}_B \leftarrow\!\!\$\ \{0,1\}^\lambda$ is sampled uniformly and independently of seed_B. Importantly, after Game G_{B5}, the initiator session π^{Init} accepts the session key only if π^{Resp} is its matching session; otherwise, π^{Init} aborts. Since $\mathsf{K}^B_{\mathsf{ENC}}$ is uniformly random and independent of the adversary's view in Game G_{B5}, the values $\mathsf{K}^B_{\mathsf{ENC}} \oplus \mathsf{seed}_B$ and $\mathsf{K}^B_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_B$ are identically distributed from the adversary's view. Hence, Games G_{B5} and G_{B6} are identical from the adversary's view. Therefore, $\Pr[\mathsf{QUERY} : G_{B5}] = \Pr[\mathsf{QUERY} : G_{B6}]$.

Game G_{B7} (Random $\mathsf{K}^{\mathsf{Test}}_B$). In this game, we replace the KEM key $\mathsf{K}^{\mathsf{Test}}_B$ in π and π' with a uniformly random key. By an argument analogous to that for Game G_{A7}, using the party P^* guessed in Game G_{B3}, we have

$$\left|\Pr[\mathsf{QUERY} : G_{B6}] - \Pr[\mathsf{QUERY} : G_{B7}]\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2).$$

Analysis of $\Pr[\mathsf{QUERY} : G_{B7}]$. In this game, the key $\mathsf{K}^{\mathsf{Test}}_B$ is uniformly random and independent of ct and cpk in sessions π and π'. Therefore, we have

$$\Pr[\mathsf{QUERY} : G_{B7}] \leq \frac{q_{\mathsf{G}}}{|\mathcal{K}|}.$$

Combining all the bounds derived above completes the proof of Case B. □

Lemma E.4 (QROM counterpart of Lemma E.2). *With the notation and assumptions of Theorem 5.3, and modeling PRF as a quantum-accessible random oracle G to which the adversary makes at most q_{G} quantum queries, there exist QPT adversaries $\mathcal{B}_2$, $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{F}$, running in time comparable to that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{FM}_3\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + 2NS^2\left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|} + \beta\right)} + \beta\right).$$

$$\mathsf{Adv}^{\mathsf{FM}_4\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq 4NS^2\delta + 4NS^3 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) + 2NS^2\left(2\sqrt{(q_{\mathsf{G}}+1)\left(2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2) + \frac{4q_{\mathsf{G}}}{|\mathcal{K}|} + \beta\right)} + \beta\right).$$

where $\beta = N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Proof. Similar to the ROM analysis, we consider $\mathsf{Adv}^{(\mathsf{FM}_3 \vee \mathsf{FM}_4)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$ and assume that the tested session satisfies $\mathsf{FM}_3 \vee \mathsf{FM}_4$; otherwise, the adversary trivially loses the game. In particular, it holds that $\mathsf{Adv}^{\mathsf{FM}_3\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq \mathsf{Adv}^{(\mathsf{FM}_3 \vee \mathsf{FM}_4)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{FM}_4\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq \mathsf{Adv}^{(\mathsf{FM}_3 \vee \mathsf{FM}_4)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$. In the following, we mainly highlight the games where the analysis deviates from the ROM proof (Lemma E.2).

Game G_0. This game corresponds to the original game defining $\mathsf{Adv}^{(\mathsf{FM}_3\vee\mathsf{FM}_4)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A})$. Thus,

$$\mathsf{Adv}^{(\mathsf{FM}_3\vee\mathsf{FM}_4)\text{-}\mathsf{KS}}_{\mathsf{pMAKE}}(\mathcal{A}) = \mathsf{Adv}^{G_0}(\mathcal{A}).$$

Case Distinction. Let π denote the tested session and let π' denote its (unique) PartialMatch session. We further write π^{Init} and π^{Resp} to denote the initiator and responder sessions among $\{\pi, \pi'\}$, respectively. Next, we consider the following two cases in Game G_0. Case A (denoted by G_A): the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Init}})$. Case B (denoted by G_B): the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Resp}})$. Accordingly, we can bound the adversary's success probability by

$$\mathsf{Adv}^{G_0}(\mathcal{A}) \le \mathsf{Adv}^{G_A}(\mathcal{A}) + \mathsf{Adv}^{G_B}(\mathcal{A}).$$

Case A: the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Init}})$.

Game $G_{A1}-G_{A3}$. The games $G_{A1}-G_{A3}$ are the same as Games $G_{A1}-G_{A3}$ in the proof of Lemma E.2.

Game G_{A4} (Random $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$). In this game, the random oracle G accessed by $\mathcal{A}$ is replaced by an oracle G' defined as follows. Let $\overline{\mathsf{G}}$ be an independently sampled random oracle, and let $\mathsf{K}^{\mathsf{Test}}_A$ denote the K_A value in the sessions π and π'. For each $\mathsf{K}_B \in \mathcal{K}$, we define $\mathsf{G}'(\mathsf{K}^{\mathsf{Test}}_A, \mathsf{K}_B) = \overline{\mathsf{G}}(\mathsf{K}^{\mathsf{Test}}_A, \mathsf{K}_B)$ and set $\mathsf{G}'(x) = \mathsf{G}(x)$ for all other inputs x. Notably, the challenger still uses $\mathsf{G}(\mathsf{K}^{\mathsf{Test}}_A, \cdot)$ in π and π' to compute $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}})$.

Let $z = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, K^*)$ where $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow\!\$\ \mathsf{SKeyGen}(1^\lambda)$, $\mathsf{seed}^* \leftarrow\!\$\ \{0,1\}^\lambda$, $(\mathsf{pk}^*_e, \mathsf{sk}^*_e) \leftarrow \mathsf{EKeyGen}(\mathsf{pk}^*, \mathsf{seed}^*)$, $\mathsf{cpk}^* \leftarrow \mathsf{CombPK}(\mathsf{pk}^*, \mathsf{pk}^*_e)$, and $(\mathsf{ct}^*, K^*) \leftarrow\!\$\ \mathsf{Encaps}_{\mathsf{AFS}}(\mathsf{cpk}^*)$. Sample an independent random oracle $\overline{\mathsf{G}}$, and define G' as above. Define $S^* = \{(K^*, \mathsf{K}_B) \mid \mathsf{K}_B \in \mathcal{K}\}$. For every $x \in S^*$, define $\mathsf{G}(x) = \overline{\mathsf{G}}(x)$ and set $\mathsf{G}(x) = \mathsf{G}'(x)$ for all other inputs $x \notin S^*$. Let $A^{|O\rangle}_{G_{A4}}(z)$ be an oracle algorithm that perfectly simulates the game G_{A4} for $\mathcal{A}$, except that it first uses the random oracle O to simulate the random oracle accessed by $\mathcal{A}$. Moreover, A uses pk^* as the long-term public key of $\pi_{\mathsf{Init}}.\mathsf{actor}$, and embeds $\mathsf{seed}_A = \mathsf{seed}^*$, $\mathsf{ct}_A = \mathsf{ct}^*$, and $\mathsf{K}_A = K^*$ into the sessions π^{Init} and π^{Resp}, and finally returns 1 if $\mathcal{A}$ wins the game. Thus, we have $\mathsf{Adv}^{G_{A3}}(\mathcal{A}) = \Pr[A^{\mathsf{G}}_{G_{A4}}(z) \Rightarrow 1]$ and $\mathsf{Adv}^{G_{A4}}(\mathcal{A}) = \Pr[A^{\mathsf{G}'}_{G_{A4}}(z) \Rightarrow 1]$.

Lemma A.10 states that $\left|\Pr[A^{\mathsf{G}}_{G_{A4}}(z) \Rightarrow 1] - \Pr[A^{\mathsf{G}'}_{G_{A4}}(z) \Rightarrow 1]\right| \le 2\sqrt{(q_G+1)\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{A4}}(z)]}$. Here $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{A4}}(z)]$ denotes the probability that the oracle $\mathcal{O}^{SC}_{S^*}$ ever outputs 1 during the execution of $A^{\mathsf{G}'\setminus S^*}_{G_{A4}}(z)$. Therefore,

$$\left|\mathsf{Adv}^{G_{A3}}(\mathcal{A}) - \mathsf{Adv}^{G_{A4}}(\mathcal{A})\right| \le 2\sqrt{(q_G+1)\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{A4}}(z)]}.$$

Game G_{A5} (Abort on BAD). This game applies the same modification as in Game G_{A5} of Lemma E.2. We define the event BAD as follows: BAD occurs if either (i) the responder session π^{Resp} accepts but $(\pi^{\mathsf{Init}}.\mathsf{id}_A, \pi^{\mathsf{Init}}.\mathsf{c}^A_{\mathsf{seed}}) \neq (\pi^{\mathsf{Resp}}.\mathsf{id}_A, \pi^{\mathsf{Resp}}.\mathsf{c}^A_{\mathsf{seed}})$, or (ii) the initiator session π^{Init} accepts but π^{Resp} is not the matching session of π^{Init}. If the event BAD occurs, the game aborts. At this point, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random from the view of $\mathcal{A}$, and we have

$$\mathsf{Adv}^{G_{A5}}(\mathcal{A}) = 0.$$

If the BAD event does not occur, games G_{A4} and G_{A5} are identical. Therefore, we have $\left|\mathsf{Adv}^{G_{A4}}(\mathcal{A}) - \mathsf{Adv}^{G_{A5}}(\mathcal{A})\right| \le \Pr[\mathsf{BAD} : G_{A4}]$. Similarly, we have $\left|\Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{A4}}(z)] - \Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{A5}}(z)]\right| \le \Pr[\mathsf{BAD} : G_{A4}]$. By an argument analogous to that for Game G_{A5} in Lemma E.2, we obtain $\Pr[\mathsf{BAD} : G_{A4}] \le N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$. Therefore,

$$\left|\mathsf{Adv}^{G_{A4}}(\mathcal{A}) - \mathsf{Adv}^{G_{A5}}(\mathcal{A})\right| \le N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}),$$

$$\left|\Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{A4}}(z)] \le \Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{A5}}(z)]\right| \le N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Game G_{A6} (Randomized c^A_{seed}). In this game, we modify the computation of $c^A_{\mathsf{seed}} = \mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{seed}_A$ in π^{Init} by replacing it with $\hat{c}^A_{\mathsf{seed}} = \mathsf{K}^A_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_A$, where $\hat{\mathsf{seed}}_A \leftarrow\!\$\ \{0,1\}^\lambda$ is sampled uniformly and

independently of seed_A. Importantly, after Game G_{A5}, the responder session π^{Resp} accepts the session key only if it receives the honest third message from π^{Init}, otherwise π^{Resp} aborts. Since $\mathsf{K}^A_{\mathsf{ENC}}$ is derived from the hidden value $\mathsf{G}(\mathsf{K}^{\mathsf{Test}}_A, \mathsf{K}^{\mathsf{Test}}_B)$, while $\mathcal{A}$ only has access to $\mathsf{G}' \setminus S^*$, the key $\mathsf{K}^A_{\mathsf{ENC}}$ is uniformly random and independent of the adversary's view in the execution of $A^{\mathsf{G}'\setminus S^*}_{G_{A5}}(z)$. Therefore, the values $\mathsf{K}^A_{\mathsf{ENC}} \oplus \mathsf{seed}_A$ and $\mathsf{K}^A_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_A$ are identically distributed from the adversary's view. Hence, the executions $A^{\mathsf{G}'\setminus S^*}_{G_{A5}}(z)$ and $A^{\mathsf{G}'\setminus S^*}_{G_{A6}}(z)$ induce identical distributions on all variables relevant to the event FIND. Therefore,

$$\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{A5}}(z)] = \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{A6}}(z)].$$

Game G'_{A6} (Random K^*). It remains to bound $\Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{A6}}(z)]$. In this game, we replace the K^* in z and S^* for $A^{G'\setminus S^*}_{G_{A6}}(z)$ with a uniformly random $\hat{\mathsf{K}}^*$. That is, let $\hat{z} = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, \hat{\mathsf{K}}^*)$ and $\hat{S}^* = \{(\hat{\mathsf{K}}^*, \mathsf{K}_B) \mid \mathsf{K}_B \in \mathcal{K}\}$. To bound $\left|\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{A6}}(z)] - \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})]\right|$, we construct an IND-ECPA adversary $\mathcal{B}_2$ against AFS-KEM as follows. Specifically, $\mathcal{B}_2$ receives $(\mathsf{pk}^*_s, \mathsf{pk}^*_e, \mathsf{cpk}^*, \mathsf{sk}^*_s, \mathsf{ct}^*, \mathsf{K}^*_b)$ from its challenger. Adversary $\mathcal{B}_2$ honestly simulates the view of $\mathcal{A}$ as $A^{\mathsf{G}'\setminus S^*}_{G_{A6}}(z)$, except that it sets pk^*_s as the public key of P^* and embeds $\mathsf{cpk}_A = \mathsf{cpk}^*$ and $\mathsf{ct}_A = \mathsf{ct}^*$ into sessions π and π', respectively. Moreover, $\mathcal{B}_2$ uses K^*_b as the KEM key K_A in both π and π', samples a quantum random oracle G', and uses the oracle $\mathsf{G}' \setminus S^*_b$ to answer adversary (quantum) RO queries, where $S^*_b = \{(\mathsf{K}^*_b, \mathsf{K}_B) \mid \mathsf{K}_B \in \mathcal{K}\}$. If the FIND event occurs, $\mathcal{B}_2$ returns 1. When $b = 0$, we have $\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 0] = \Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{A6}}(z)]$. When $b = 1$, we have $\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 1] = \Pr[\mathsf{FIND} : A^{G'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})]$. Therefore,

$$\left|\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{A6}}(z)] - \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})]\right| \le 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2).$$

Analysis of $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})]$. Recall that $\hat{z} = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, \hat{\mathsf{K}}^*)$, and that $A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})$ still uses the value $\mathsf{G}(\hat{\mathsf{K}}^*, \cdot)$ to derive $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$ for the sessions π and π', while the adversary $\mathcal{A}$ does not access G. Therefore, we define an algorithm $\hat{A}^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*)$ that is identical to $A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})$, except that it samples $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$ uniformly at random for π and π' (instead of computing them via $\mathsf{G}(\hat{\mathsf{K}}^*, \cdot)$). It is immediate that we have $\hat{A}^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*)$ and $A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})$ induce identical views for $\mathcal{A}$. Hence, $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\hat{z})] = \Pr[\mathsf{FIND} : \hat{A}^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*)]$. Note that the test key $\hat{S}^*$ is independent of $\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*$, and is uniformly random from the adversary's view. By Lemma A.11, it follows that

$$\Pr[\mathsf{FIND} : \hat{A}^{\mathsf{G}'\setminus \hat{S}^*}_{G_{A6}}(\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*)] \le \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}.$$

Combining all the bounds derived above completes the proof of Case A.

Case B: the adversary never issues $\mathsf{RevealEph}(\pi^{\mathsf{Resp}})$. In Case B, we can obtain the same bound as in Case A. The analysis of Case B closely follows that of Case A. For completeness, we provide a proof sketch that highlights the only points where the two cases differ.

Game $G_{B1} - G_{B3}$. The games $G_{B1} - G_{B3}$ are identical to the corresponding games $G_{B1} - G_{B3}$ in the ROM analysis (Lemma E.2) of Case B.

Game G_{B4} (Random $\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}}$). This game applies a similar modification as in Game G_{A4}, except that we replace $\mathsf{K}^{\mathsf{Test}}_A$ with $\mathsf{K}^{\mathsf{Test}}_B$. More specifically, let $\mathsf{K}^{\mathsf{Test}}_B$ denote the encapsulated key K_B in sessions π and π'. We define a modified random oracle G' as follows. Let $\overline{\mathsf{G}}$ be an independently sampled random oracle. For every $\mathsf{K}_A \in \mathcal{K}$, define $\mathsf{G}'(\mathsf{K}_A, \mathsf{K}^{\mathsf{Test}}_B) = \overline{\mathsf{G}}(\mathsf{K}_A, \mathsf{K}^{\mathsf{Test}}_B)$, and set $\mathsf{G}'(x) = \mathsf{G}(x)$ for all other inputs x. Notably, the challenger still uses $\mathsf{G}(\cdot, \mathsf{K}^{\mathsf{Test}}_B)$ in sessions π and π' to compute $(\mathsf{K}_{\mathsf{SESSION}}, \mathsf{K}^A_{\mathsf{ENC}}, \mathsf{K}^B_{\mathsf{ENC}})$.

Let $z = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, K^*)$ be defined analogously as in Game G_{A4}. Define $S^* = \{(\mathsf{K}_A, K^*) \mid \mathsf{K}_A \in \mathcal{K}\}$. For every $x \in S^*$, define $\mathsf{G}(x) = \overline{\mathsf{G}}(x)$ and set $\mathsf{G}(x) = \mathsf{G}'(x)$ for all other inputs $x \notin S^*$. Let $A^{|O\rangle}_{G_{B4}}(z)$ be an oracle algorithm that perfectly simulates the game G_{B4} for $\mathcal{A}$, except that it first uses

the random oracle O to simulate the random oracle accessed by $\mathcal{A}$. Moreover, A uses pk^* as the long-term public key of $\pi_{\mathsf{Resp}}.\mathsf{actor}$, and embeds $\mathsf{seed}_B = \mathsf{seed}^*$, $\mathsf{ct}_B = \mathsf{ct}^*$, and $\mathsf{K}_B = K^*$ into the sessions π^{Init} and π^{Resp}, and finally returns 1 if $\mathcal{A}$ wins the game. Thus, we have $\mathsf{Adv}^{G_{B3}}(\mathcal{A}) = \Pr[A^{\mathsf{G}}_{G_{B4}}(z) \Rightarrow 1]$ and $\mathsf{Adv}^{G_{B4}}(\mathcal{A}) = \Pr[A^{\mathsf{G}'}_{G_{B4}}(z) \Rightarrow 1]$. Using Lemma A.10 with the same argument as in Case A, we have

$$\left|\mathsf{Adv}^{G_{B3}}(\mathcal{A}) - \mathsf{Adv}^{G_{B4}}(\mathcal{A})\right| \leq 2\sqrt{(q_{\mathsf{G}}+1)\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{B4}}(z)]}.$$

Game G_{B5} (Abort on BAD). This game applies the same modification as in Game G_{A5}. We define the event BAD as follows: BAD occurs if either (i) the responder session π^{Resp} accepts but $(\pi^{\mathsf{Init}}.\mathsf{id}_A, \pi^{\mathsf{Init}}.\mathsf{c}^A_{\mathsf{seed}}) \neq (\pi^{\mathsf{Resp}}.\mathsf{id}_A, \pi^{\mathsf{Resp}}.\mathsf{c}^A_{\mathsf{seed}})$, or (ii) the initiator session π^{Init} accepts but π^{Resp} is not the matching session of π^{Init}. If the event BAD occurs, the game aborts. At this point, the session key $\mathsf{K}_{\mathsf{SESSION}}$ of the tested session π is uniformly random from the view of $\mathcal{A}$, and we have

$$\mathsf{Adv}^{G_{B5}}(\mathcal{A}) = 0.$$

Moreover, by an argument analogous to that for Game G_{A5}, we obtain

$$\left|\mathsf{Adv}^{G_{B4}}(\mathcal{A}) - \mathsf{Adv}^{G_{B5}}(\mathcal{A})\right| \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + 3N(N+1) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}),$$

$$\left|\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{B4}}(z)] - \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{B5}}(z)]\right| \leq N \cdot \mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + 3N(N+1) \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Game G_{B6} (Randomized c^B_{seed}). In this game, we modify the computation of $c^B_{\mathsf{seed}} = \mathsf{K}^B_{\mathsf{ENC}} \oplus \mathsf{seed}_B$ in π^{Resp} by replacing it with $\hat{c}^B_{\mathsf{seed}} = \mathsf{K}^B_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_B$, where $\hat{\mathsf{seed}}_B \leftarrow\!\$\ \{0,1\}^\lambda$ is sampled uniformly and independently of seed_B. Importantly, after Game G_{B5}, the initiator session π^{Init} accepts the session key if and only if it is the matching session of π^{Resp}; otherwise, π^{Init} aborts. In particular, π^{Init} does not need to recover the seed from the received value c^B_{seed} in any execution relevant to the event FIND. Since $\mathsf{K}^B_{\mathsf{ENC}}$ is derived from the hidden value $\mathsf{G}(\mathsf{K}^{\mathsf{Test}}_A, \mathsf{K}^{\mathsf{Test}}_B)$, while $\mathcal{A}$ only has access to $\mathsf{G}' \setminus S^*$, the key $\mathsf{K}^B_{\mathsf{ENC}}$ is uniformly random and independent of the adversary's view in the execution of $A^{\mathsf{G}'\setminus S^*}_{G_{B5}}(z)$. Therefore, the values $\mathsf{K}^B_{\mathsf{ENC}} \oplus \mathsf{seed}_B$ and $\mathsf{K}^B_{\mathsf{ENC}} \oplus \hat{\mathsf{seed}}_B$ are identically distributed from the adversary's view. Hence, the executions $A^{\mathsf{G}'\setminus S^*}_{G_{B5}}(z)$ and $A^{\mathsf{G}'\setminus S^*}_{G_{B6}}(z)$ induce identical distributions on all variables relevant to the event FIND. Therefore,

$$\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{B5}}(z)] = \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{B6}}(z)].$$

Game G'_{B6} (Random K^*). In this game, we replace the K^* in z and S^* for $A^{G'\setminus S^*}_{G_{B6}}(z)$ with a uniformly random $\hat{\mathsf{K}}^*$. That is, let $\hat{z} = (\mathsf{pk}^*, \mathsf{sk}^*, \mathsf{seed}^*, \mathsf{ct}^*, \hat{\mathsf{K}}^*)$ and $\hat{S}^* = \{(\mathsf{K}_A, \hat{\mathsf{K}}^*) \mid \mathsf{K}_A \in \mathcal{K}\}$. To bound $\left|\Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{B6}}(z)] - \Pr[\mathsf{FIND} : A^{G'\setminus \hat{S}^*}_{G_{B6}}(\hat{z})]\right|$, we construct an $\mathsf{IND\text{-}ECPA}$ adversary $\mathcal{B}_2$ against $\mathsf{AFS\text{-}KEM}$ as follows. Specifically, $\mathcal{B}_2$ receives $(\mathsf{pk}^*_s, \mathsf{pk}^*_e, \mathsf{cpk}^*, \mathsf{sk}^*_s, \mathsf{ct}^*, \mathsf{K}^*_b)$ from its challenger. Adversary $\mathcal{B}_2$ honestly simulates the view of $\mathcal{A}$ as $A^{\mathsf{G}'\setminus S^*}_{G_{B6}}(z)$, except that it sets pk^*_s as the public key of P^* and embeds $\mathsf{cpk}_B = \mathsf{cpk}^*$ and $\mathsf{ct}_B = \mathsf{ct}^*$ into sessions π and π', respectively. Moreover, $\mathcal{B}_2$ uses K^*_b as the KEM key K_B in both π and π', samples a quantum random oracle G', and uses the oracle $\mathsf{G}' \setminus S^*_b$ to answer adversary (quantum) RO queries, where $S^*_b = \{(\mathsf{K}_A, \mathsf{K}^*_b) \mid \mathsf{K}_A \in \mathcal{K}\}$. If the FIND event occurs, $\mathcal{B}_2$ returns 1. When $b = 0$, we have $\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 0] = \Pr[\mathsf{FIND} : A^{G'\setminus S^*}_{G_{B6}}(z)]$. When $b = 1$, we have $\Pr[\mathcal{B}_2 \Rightarrow 1 : b = 1] = \Pr[\mathsf{FIND} : A^{G'\setminus \hat{S}^*}_{G_{B6}}(\hat{z})]$. Therefore,

$$\left|\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus S^*}_{G_{B6}}(z)] - \Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{B6}}(\hat{z})]\right| \leq 2\,\mathsf{Adv}^{\mathsf{IND\text{-}ECPA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}_2).$$

Analysis of $\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{B6}}(\hat{z})]$. Since $\hat{\mathsf{K}}^*$ is uniformly random from the adversary's view, Lemma A.11 implies that

$$\Pr[\mathsf{FIND} : A^{\mathsf{G}'\setminus \hat{S}^*}_{G_{B6}}(\hat{z})] \leq \frac{4q_{\mathsf{G}}}{|\mathcal{K}|}.$$

Combining all the bounds above completes the proof of Case B. □

E.3 Explicit Authentication Analysis for pMAKE

Theorem E.5. *Let there be at most N parties and at most S sessions activated in the* Init-EA *and* Resp-EA *games. Suppose that the underlying* AFS-KEM *is δ-correct, γ-spreadness, and satisfies perfect* ExtractPK*-correctness and ephemeral-key-generation equivalence. Then, for any polynomial-time adversary $\mathcal{A}$ attacking the* Init-EA *or* Resp-EA *security of* pMAKE*, there exist polynomial-time adversaries $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{F}$, and $\mathcal{Z}$, constructed below and running in time comparable to that of $\mathcal{A}$, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Init\text{-}EA}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq{}& \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} + 2NS^2\delta \\ &+ NS\begin{pmatrix} 2\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 4S\cdot\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ + (N+1)(N+2)S\cdot\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + 3\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) \end{pmatrix}.\end{aligned}$$

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Resp\text{-}EA}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq{}& \frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}} + 2NS^2\delta \\ &+ NS\begin{pmatrix} 2\mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}) + 4S\cdot\mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}) \\ + (N+1)(N+2)S\cdot\mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}) + 3\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) \end{pmatrix}.\end{aligned}$$

Proof. We give a unified analysis for Init-EA and Resp-EA. We begin by defining game G_0, which is identical to the Init-EA game except for the win condition. Specifically, the adversary is said to win the experiment if the session π output by the adversary satisfies $\mathsf{MalAcc}_{\mathsf{Init}}(\pi) = \mathsf{true}$ or $\mathsf{MalAcc}_{\mathsf{Resp}}(\pi) = \mathsf{true}$. It is immediate that $\mathsf{Adv}^{\mathsf{Init\text{-}EA}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq \mathsf{Adv}^{G_0}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{Resp\text{-}EA}}_{\mathsf{pMAKE}}(\mathcal{A}) \leq \mathsf{Adv}^{G_0}(\mathcal{A})$. Therefore, it suffices to bound $\mathsf{Adv}^{G_0}(\mathcal{A})$.

We first assume that all honest users possess distinct long-term public keys, and that all honest sessions compute distinct combined public keys cpk, including cpk_A for initiator sessions and cpk_B for responder sessions. Moreover, we assume that every honest responder session that receives an honestly generated cpk_A from an honest initiator session computes a distinct ciphertext ct_A. Similarly, every honest initiator session that receives an honestly generated cpk_B from an honest responder session computes a distinct ciphertext ct_B.

Since long-term public keys have min-entropy at least k_1, combined public keys generated honestly from a fixed long-term public key have min-entropy at least k_2, and the ciphertexts of AFS-KEM satisfy γ-spreadness, these assumptions contribute $\frac{N^2}{2^{k_1}} + \frac{S^2}{2^{k_2}} + \frac{S^2}{2^{\gamma}}$ to the final bound. Under these assumptions, it is impossible for any session to have more than one PartialMatch session or more than one Match session.

This proof largely mirrors the analysis for the cases $\mathsf{FM}_5 \vee \mathsf{FM}_6$ in Lemma E.3, except for the treatment of Game G_7. In the following analysis, we assume that the session π output by the adversary satisfies $\mathsf{MalAccInit}(\pi) = \mathsf{true}$ or $\mathsf{MalAccResp}(\pi) = \mathsf{true}$, otherwise the adversary trivially loses the game.

Game G_1 (Abort if Collision in K_A or K_B). Let π denote the session output by the adversary in the game. Let π' denote the unique session, if it exists, such that $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) = (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$ when $\pi.\mathsf{role} = \mathsf{Resp}$, and such that $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) = (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$ when $\pi.\mathsf{role} = \mathsf{Init}$.

In this game, we define the COLL event as follows. The event COLL occurs if π' exists and $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) \neq (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$ (resp. $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) \neq (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$), but $\pi.\mathsf{K}_B = \pi'.\mathsf{K}_B$ (resp. $\pi.\mathsf{K}_A = \pi'.\mathsf{K}_A$), when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$). If the COLL event occurs, we abort the game. Since Games G_0 and G_1 are identical unless COLL occurs, we have $\left|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})\right| \leq \Pr[\mathsf{COLL} : G_0]$. Using the same argument as in Game G_1 of Lemma E.3, we obtain

$$\left|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})\right| \leq 4NS^2 \cdot \mathsf{Adv}^{\mathsf{Key\text{-}Coll}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{F}).$$

Game G_2 (KEM Correctness). In this game, we define the event ERR_1 to occur if the decapsulation of ct_A (resp. ct_B) in π' (if it exists) does not equal the corresponding encapsulated key when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$). Moreover, if π' is further a PartialMatch session of π, we define the event ERR_2 to occur if the decapsulation of ct_B (resp. ct_A) in π' does not equal the corresponding

encapsulated key when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$). If ERR_1 or ERR_2 occurs, we abort the game. Using the same argument for δ-correctness property for AFS-KEM as in Game G_2 of Lemma E.3, we obtain

$$\left|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})\right| \leq 2NS^2\delta.$$

Game G_3 (Guessing $\pi.\mathsf{peer}$ and π). In this game, we guess the session π output by the adversary and its peer identity $\pi.\mathsf{peer}$, and abort if either guess is incorrect. Formally, we sample P^* uniformly from the N parties and sample one session instance π^* uniformly over the at most S sessions. If $P^* \neq \pi.\mathsf{peer}$ or $\pi^* \neq \pi$, where π is the session output by the adversary, the game is aborted. Since the probability of guessing correctly is $1/(NS)$, we have

$$\mathsf{Adv}^{G_2}(\mathcal{A}) \leq NS \cdot \mathsf{Adv}^{G_3}(\mathcal{A}).$$

Game G_4 (New win condition). In this game, we modify the win condition as follows. The adversary $\mathcal{A}$ wins if and only if the guessed session π accepts and satisfies $\mathsf{MalAcc}_{\mathsf{Init}}(\pi) = \mathsf{true}$ or $\mathsf{MalAcc}_{\mathsf{Resp}}(\pi) = \mathsf{true}$. The game is stopped at the moment when π accepts. Notably, the party P^* remains uncorrupted in G_4. It is evident that any adversary that wins Game G_3 also wins Game G_4. Thus,

$$\mathsf{Adv}^{G_3}(\mathcal{A}) \leq \mathsf{Adv}^{G_4}(\mathcal{A}).$$

Game G_5 (Random K_A or K_B). In this game, we replace the KEM key K_A (resp. K_B) in the tested session π with an independently uniform key $\tilde{k} \leftarrow\!\!\$\ \mathcal{K}$ if $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$). Let π' denote the unique session, if it exists, such that $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) = (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$ when $\pi.\mathsf{role} = \mathsf{Resp}$, and such that $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) = (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$ when $\pi.\mathsf{role} = \mathsf{Init}$. If such a session π' exists, we replace the corresponding KEM key in π' with the same key $\tilde{k}$. Using the same argument as in Game G_5 of Lemma E.3, we obtain

$$\left|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_5}(\mathcal{A})\right| \leq 2\ \mathsf{Adv}^{\mathsf{IND\text{-}LCK\text{-}SCCA}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{B}).$$

Game G_6 (Random $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, $\mathsf{K}^B_{\mathsf{ENC}}$). In this game, we replace the keys $\mathsf{K}_{\mathsf{SESSION}}$, $\mathsf{K}^A_{\mathsf{ENC}}$, and $\mathsf{K}^B_{\mathsf{ENC}}$ in the session π with independently sampled uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$, $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$, and $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$, respectively. If π' exists and is the PartialMatch session of π, we use the same uniform keys $\hat{\mathsf{K}}_{\mathsf{SESSION}}$, $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$, and $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$ in π'. Otherwise, if π' exists but is not a PartialMatch session of π, we use independently sampled uniform keys $\overline{\mathsf{K}}_{\mathsf{SESSION}}$, $\overline{\mathsf{K}}^A_{\mathsf{ENC}}$, and $\overline{\mathsf{K}}^B_{\mathsf{ENC}}$ in π'. Notably, after Game G_1, if π' exists but is not a PartialMatch session of π, then we must have $(\pi'.\mathsf{cpk}_B, \pi'.\mathsf{ct}_B) \neq (\pi.\mathsf{cpk}_B, \pi.\mathsf{ct}_B)$ (resp. $(\pi'.\mathsf{cpk}_A, \pi'.\mathsf{ct}_A) \neq (\pi.\mathsf{cpk}_A, \pi.\mathsf{ct}_A)$), and $\pi.\mathsf{K}_B \neq \pi'.\mathsf{K}_B$ (resp. $\pi.\mathsf{K}_A \neq \pi'.\mathsf{K}_A$) when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$).

Using the same argument as in Game G_6 of Lemma E.3, we can construct an adversary $\mathcal{Z}$ against the dual-PRF security of the PRF such that

$$\left|\mathsf{Adv}^{G_5}(\mathcal{A}) - \mathsf{Adv}^{G_6}(\mathcal{A})\right| \leq \mathsf{Adv}^{\mathsf{dual\text{-}PRF}}_{\mathsf{PRF}}(\mathcal{Z}).$$

Game G_7 (Abort on BAD). In this game, we define the BAD event as follows: BAD occurs if the guessed session π satisfies $\mathsf{MalAcc}_{\mathsf{Init}}(\pi) = \mathsf{true}$ or $\mathsf{MalAcc}_{\mathsf{Resp}}(\pi) = \mathsf{true}$. If BAD occurs, we abort the game. Since every winning execution is aborted in Game G_7, we have

$$\mathsf{Adv}^{G_7}(\mathcal{A}) = 0.$$

It follows that $\left|\mathsf{Adv}^{G_6}(\mathcal{A}) - \mathsf{Adv}^{G_7}(\mathcal{A})\right| \leq \Pr[\mathsf{BAD} : G_6]$. We now analyze $\Pr[\mathsf{BAD} : G_6]$. In Game G_6, the keys $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$ and $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$ of the session π are uniform from the adversary's view. Moreover, if the PartialMatch session $\pi^{\mathsf{PartialMatch}}$ of π exists, then it uses the same keys $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$ and $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$. We consider the following cases.

Case 1: The PartialMatch session $\pi^{\mathsf{PartialMatch}}$ of π does not exist. Here, the keys $\hat{\mathsf{K}}^A_{\mathsf{ENC}}$ and $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$ of the tested session π are uniform from the adversary's view and are never used in any other session. Therefore, $\mathsf{seed}'_A = \hat{\mathsf{K}}^A_{\mathsf{ENC}} \oplus \mathsf{c}^A_{\mathsf{seed}}$ (resp. $\mathsf{seed}'_B = \hat{\mathsf{K}}^B_{\mathsf{ENC}} \oplus \mathsf{c}^B_{\mathsf{seed}}$) computed in π is uniform from the

adversary's view when $\pi.\mathsf{role} = \mathsf{Resp}$ (resp. $\pi.\mathsf{role} = \mathsf{Init}$). Using the same argument as in Game G_7 of Lemma E.3, we can construct an adversary $\mathcal{D}$ against the RSPK-Bind security of AFS-KEM such that $\Pr[\mathsf{BAD} : \text{Case 1}] \leq 2\,\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})$.

Case 2: The unique PartialMatch session $\pi^{\mathsf{PartialMatch}}$ of π exists and $\pi.\mathsf{role} = \mathsf{Resp}$. In this case, BAD implies that the adversary has tampered with the third protocol message. That is, $(\mathsf{id}_A, \mathsf{c}^A_{\mathsf{seed}})$ sent by the PartialMatch session differs from $(P^*, \mathsf{c}'^A_{\mathsf{seed}})$ received by π.

- **Case 2.a:** If $\mathsf{id}_A \neq P^*$, we construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}^*_e, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \overline{\mathsf{pk}}^*_s, \overline{\mathsf{sk}}^*_s)$ and perfectly simulates Game G_6 for $\mathcal{A}$, except that it first guesses the PartialMatch session $\pi^{\mathsf{PartialMatch}}$ of π uniformly over the S sessions, and guesses the identity id_A uniformly over the N parties. Thus, $\mathcal{C}$ guesses correctly with probability at least $1/(NS)$. Then $\mathcal{C}$ sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}^*_s$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}^*_s$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}^*_e$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into the session $\pi^{\mathsf{PartialMatch}}$. When π receives $(P^*, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Let $(\mathsf{pk}_e^{A'}, \mathsf{sk}_e^{A'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_A)$ and $\mathsf{pk}'_A \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{A'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(P^*, \mathsf{pk}'_A) = 1$ and $\mathsf{pk}_{P^*} = \overline{\mathsf{pk}}^*_s$, we have $\mathsf{pk}'_A = \overline{\mathsf{pk}}^*_s$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2.a}] \leq NS \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
- **Case 2.b:** If $\mathsf{id}_A = P^*$ but $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$, we again construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}^*_e, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \overline{\mathsf{pk}}^*_s, \overline{\mathsf{sk}}^*_s)$ and perfectly simulates Game G_6 for $\mathcal{A}$, except that it first guesses $\pi^{\mathsf{PartialMatch}}$ uniformly over the S sessions, with success probability at least $1/S$. $\mathcal{C}$ then sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}^*_s$ and embeds $\mathsf{seed}_A = \mathsf{seed}^*_e$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into $\pi^{\mathsf{PartialMatch}}$. When π receives $(\mathsf{id}_A, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Let $(\mathsf{pk}_e^{A'}, \mathsf{sk}_e^{A'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_A)$ and $\mathsf{pk}'_A \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{A'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_A, \mathsf{pk}'_A) = 1$ and $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}^*_s$, we have $\mathsf{pk}'_A = \mathsf{pk}^*_s$. Moreover, since $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$ and the OTP key used to recover the seed is the same, we have $\mathsf{seed}'_A \neq \mathsf{seed}^*_e$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 2.b}] \leq S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

Case 3: The unique PartialMatch session $\pi^{\mathsf{PartialMatch}}$ of π exists and $\pi.\mathsf{role} = \mathsf{Init}$. If the PartialMatch session aborts, then $\hat{\mathsf{K}}^B_{\mathsf{ENC}}$ is never used in any other session. Therefore, the value $\mathsf{seed}'_B = \hat{\mathsf{K}}^B_{\mathsf{ENC}} \oplus \mathsf{c}^B_{\mathsf{seed}}$ computed in π is uniform from the adversary's view. Similarly to the analysis in Case 1, this case can be bounded by $\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D})$.

It remains to consider the case where $\pi^{\mathsf{PartialMatch}}$ accepts. In this case, BAD implies that the adversary has tampered with either the third or the fourth protocol message between π and $\pi^{\mathsf{PartialMatch}}$. Specifically, the tuple $(\mathsf{id}_A, \mathsf{c}^A_{\mathsf{seed}}, \mathsf{id}_B = P^*, \mathsf{c}^B_{\mathsf{seed}})$ maintained by π differs from $(\mathsf{id}'_A, \mathsf{c}'^A_{\mathsf{seed}}, \mathsf{id}'_B, \mathsf{c}'^B_{\mathsf{seed}})$ maintained by $\pi^{\mathsf{PartialMatch}}$. We consider the following subcases.

- **Case 3.a:** If $\mathsf{id}_A \neq \mathsf{id}'_A$, this analysis is similar to Case 2.a. We construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}^*_e, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \overline{\mathsf{pk}}^*_s, \overline{\mathsf{sk}}^*_s)$ and perfectly simulates Game G_6 for $\mathcal{A}$, except that it first guesses the session $\pi^{\mathsf{PartialMatch}}$ uniformly over the S sessions, and guesses the identities id_A and id'_A uniformly over the N parties. Notably, $\mathcal{C}$ guesses correctly with probability at least $1/(N^2S)$. Then $\mathcal{C}$ sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}^*_s$ and $\mathsf{pk}_{\mathsf{id}'_A} = \overline{\mathsf{pk}}^*_s$. Moreover, $\mathcal{C}$ embeds $\mathsf{seed}_A = \mathsf{seed}^*_e$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into the session π. When $\pi^{\mathsf{PartialMatch}}$ receives $(\mathsf{id}'_A, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Similarly to Case 2.a, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins its static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 3.a}] \leq N^2S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
- **Case 3.b:** If $\mathsf{id}_B \neq \mathsf{id}'_B$, this analysis is similar to Case 2.a. We also construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM. Since $\mathsf{id}_B = P^*$ has already been guessed in G_3, $\mathcal{C}$ only needs to guess $\pi^{\mathsf{PartialMatch}}$ and id'_B, with success probability at least $1/(NS)$. Thus, $\Pr[\mathsf{BAD} : \text{Case 3.b}] \leq NS \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.
- **Case 3.c:** If $\mathsf{id}_A = \mathsf{id}'_A$ and $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$, this analysis is similar to Case 2.b. We also construct an adversary $\mathcal{C}$ against the SPK-Bind security of AFS-KEM as follows. Adversary $\mathcal{C}$ receives the challenge tuple $(\mathsf{cpk}^*, \mathsf{seed}^*_e, \mathsf{pk}^*_s, \mathsf{sk}^*_s, \overline{\mathsf{pk}}^*_s, \overline{\mathsf{sk}}^*_s)$ and perfectly simulates Game G_6 for $\mathcal{A}$, except that

it first guesses $\pi^{\mathsf{PartialMatch}}$ uniformly over the S sessions and guesses id_A uniformly over the N parties, with success probability at least $1/(NS)$. $\mathcal{C}$ then sets $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$ and embeds $\mathsf{seed}_A = \mathsf{seed}_e^*$ and $\mathsf{cpk}_A = \mathsf{cpk}^*$ into π. When $\pi^{\mathsf{PartialMatch}}$ receives $(\mathsf{id}_A, \mathsf{ct}_B, \mathsf{c}'^A_{\mathsf{seed}})$, $\mathcal{C}$ honestly decrypts $\mathsf{c}'^A_{\mathsf{seed}}$ and outputs the corresponding seed'_A to its challenger. Let $(\mathsf{pk}_e^{A'}, \mathsf{sk}_e^{A'}) \leftarrow \mathsf{EKeyGen}_c(\mathsf{cpk}^*, \mathsf{seed}'_A)$ and $\mathsf{pk}'_A \leftarrow \mathsf{ExtractPK}(\mathsf{cpk}^*, \mathsf{pk}_e^{A'})$. Since $\mathsf{CA}_{\mathsf{Verf}}(\mathsf{id}_A, \mathsf{pk}'_A) = 1$ and $\mathsf{pk}_{\mathsf{id}_A} = \mathsf{pk}_s^*$, we have $\mathsf{pk}'_A = \mathsf{pk}_s^*$. Moreover, since $\mathsf{c}^A_{\mathsf{seed}} \neq \mathsf{c}'^A_{\mathsf{seed}}$ and the OTP key used to recover the seed is the same, we have $\mathsf{seed}'_A \neq \mathsf{seed}_e^*$. Therefore, if the BAD event occurs in this case and $\mathcal{C}$ guesses correctly, then $\mathcal{C}$ wins the static public-key binding game. Hence, $\Pr[\mathsf{BAD} : \text{Case 3.c}] \leq NS \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

– **Case 3.d:** If $\mathsf{id}_B = \mathsf{id}'_B$ and $\mathsf{c}^B_{\mathsf{seed}} \neq \mathsf{c}'^B_{\mathsf{seed}}$, the analysis is analogous to Case 3.c. Since $\mathsf{id}_B = P^*$ has already been guessed, $\mathcal{C}$ only needs to guess $\pi^{\mathsf{PartialMatch}}$, with success probability at least $1/S$. Thus, $\Pr[\mathsf{BAD} : \text{Case 3.d}] \leq S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C})$.

For simplicity, we reuse the notation $\mathcal{C}$ for the $\mathsf{SPK\text{-}Bind}$ reductions in the above cases. Combining the above cases, we obtain

$$\left|\mathsf{Adv}^{G_6}(\mathcal{A}) - \mathsf{Adv}^{G_7}(\mathcal{A})\right| \leq 3\,\mathsf{Adv}^{\mathsf{RSPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{D}) + (N+1)(N+2)S \cdot \mathsf{Adv}^{\mathsf{SPK\text{-}Bind}}_{\mathsf{AFS\text{-}KEM}}(\mathcal{C}).$$

Combining all the bounds derived above completes the proof. □