EMU
Internet-Draft
Intended status: Standards Track
Expires: 23 January 2026

A. Banerjee
T. Reddy
Nokia

22 July 2025

Enhancing Security in EAP-AKA' with Hybrid Post-Quantum Cryptography
draft-ietf-emu-hybrid-pqc-eapaka-00

Abstract

Forward Secrecy for the Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS) is specified in [RFC9678], providing updates to [RFC9048] with an optional extension that offers ephemeral key exchange using the traditional Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) key agreement algorithm for achieving perfect forward secrecy (PFS). However, it is susceptible to future threats from Cryptographically Relevant Quantum Computers, which could potentially compromise a traditional ephemeral public key. If the adversary has also obtained knowledge of the long-term key and ephemeral public key, it could compromise session keys generated as part of the authentication run in EAP-AKA'.

This draft aims to enhance the security of EAP-AKA' FS protocol by leveraging PQ/T Hybrid [I-D.ietf-pquip-pqt-hybrid-terminology] algorithms to make it quantum-safe.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-emu-hybrid-pqc-eapaka/.

Discussion of this document takes place on the emu Working Group mailing list (mailto:emu@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/emu/. Subscribe at https://www.ietf.org/mailman/listinfo/emu/.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

Forward Secrecy for the Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS) defined in [RFC9678] updates the improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA') specified in [RFC9048], with an optional extension providing ephemeral key exchange. This prevents an attacker who has gained access to the long term key from obtaining session keys established in the past, assuming these have been properly deleted. EAP-AKA' FS mitigates passive attacks (e.g., large scale pervasive monitoring) against future sessions.

Nevertheless, EAP-AKA' FS uses traditional algorithms public-key algorithms (e.g., ECDH) which will be broken by a Cryptographically Relevant Quantum Computer (CRQC) using Shor's algorithm. The presence of a CRQC would render state-of-the-art, traditional public-key algorithms deployed today obsolete and insecure, since the assumptions about the intractability of the mathematical problems for these algorithms that offer confident levels of security today no longer apply in the presence of a CRQC. A CRQC could recover the SHARED_SECRET from the ECDHE public keys (Section 6.3 of [RFC9678]). If the adversary has also obtained knowledge of the long-term key, it could then compute CK', IK', and the SHARED_SECRET, and any derived output keys. This means that the CRQC would disable the forward security capability provided by [RFC9678].

The migration to PQC is unique in the history of modern digital cryptography in that neither the traditional algorithms nor the post-quantum algorithms are fully trusted to protect data for the required data lifetimes. The post-quantum algorithms face uncertainty about the underlying mathematics, compliance issues, unknown vulnerabilities, hardware and software implementations that have not had sufficient maturing time to rule out classical cryptanalytic attacks and implementation bugs. During the transition from traditional to post-quantum algorithms, there is a desire or a requirement for protocols that use both algorithm types.

This specification defines HPKE [I-D.ietf-hpke-pq] [I-D.irtf-cfrg-hybrid-kems] for use with EAP-AKA' FS. HPKE offers a variant of public-key encryption of arbitrary-sized plaintexts for a recipient public key. HPKE works for any combination of an asymmetric key encapsulation mechanism (KEM), key derivation function (KDF), and authenticated encryption with additional data (AEAD) function. HPKE can be extended to support hybrid post-quantum Key Encapsulation Mechanisms (KEMs) as defined in [I-D.ietf-hpke-pq].

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

This document makes use of the terms defined in [I-D.ietf-pquip-pqt-hybrid-terminology]. For the purposes of this document, it is helpful to be able to divide cryptographic algorithms into two classes:

"Traditional Algorithm": An asymmetric cryptographic algorithm based on integer factorisation, finite field discrete logarithms or elliptic curve discrete logarithms. In the context of JOSE, examples of traditional key exchange algorithms include Elliptic Curve Diffie-Hellman Ephemeral Static [RFC6090] [RFC8037]. In the context of COSE, examples of traditional key exchange algorithms include Ephemeral-Static (ES) DH and Static-Static (SS) DH [RFC9052].

"Post-Quantum Algorithm": An asymmetric cryptographic algorithm that is believed to be secure against attacks using quantum computers as well as classical computers. Examples of PQC key exchange algorithms include Kyber.

"Hybrid" key exchange, in this context, means the use of two key exchange algorithms based on different cryptographic assumptions, e.g., one traditional algorithm and one Post-Quantum algorithm, with the purpose of the final shared secret key being secure as long as at least one of the component key exchange algorithms remains unbroken. It is referred to as PQ/T Hybrid Scheme in [I-D.ietf-pquip-pqt-hybrid-terminology].

PQ/T Hybrid Key Encapsulation Mechanism: A Key Encapsulation mechanism (KEM) made up of two or more component KEM algorithms where at least one is a post-quantum algorithm and at least one is a traditional algorithm.

4. Background on EAP-AKA’ with perfect forward secrecy

In EAP-AKA’, The authentication vector (AV) contains a random part RAND, an authenticator part AUTN used for authenticating the network to the USIM, an expected result part XRES, a 128-bit session key for integrity check IK, and a 128-bit session key for encryption CK.

As described in the draft [RFC9678], the server has the EAP identity of the peer. The server asks the AD to run AKA algorithm to generate RAND, AUTN, XRES, CK and IK. Further it also derives CK' and IK' keys which are tied to a particular network name. The server now generates the ephemeral key pair and sends the public key of that key pair and the first EAP method message to the peer. In this EAP message, AT_PUB_ECDHE (carries public key) and the AT_KDF_FS(carries other FS related parameters). Both of these might be ignored of USIM doesn' t support the Forward Secrecy extension. The peer checks if it wants to have a Forward extension in EAP AKA'. If yes, then it will eventually respond with AT_PUB_ECDHE and MAC. If not, it will ignore AT_PUB_ECDHE. If the peer wants to participate in FS extension, it will then generate its ECDH key pair, calculate a shared key based on its private key and server public key. The server will receive the RES from peer and AT_PUB_ECDHE. The shared key will be generated both in the peer and the server with key pairs exchanged, and later master key is also generated.

MK_ECDHE = PRF'(IK'| CK'|SHARED_SECRET,"EAP-AKA' FS"|Identity)

5. Hybrid Enhancements by Design

We suggest the following changes and enhancements:

* A new attribute, AT_PUB_HYBRID, is defined to carry the public key, which is the concatenation of traditional and PQC KEM public keys from the EAP server. The AT_PUB_HYBRID attribute will carry the encapsulated key, which is formed by concatenating the encapsulated key (enc) from the traditional KEM algorithm and the ciphertext (ct) from the PQC KEM Encapsulation function from the EAP peer.

* The AT_KDF_FS attribute is updated to indicate the PQ/T Hybrid KEM in HPKE and HKDF for generating the Hybrid Master Key MK_HYBRID.

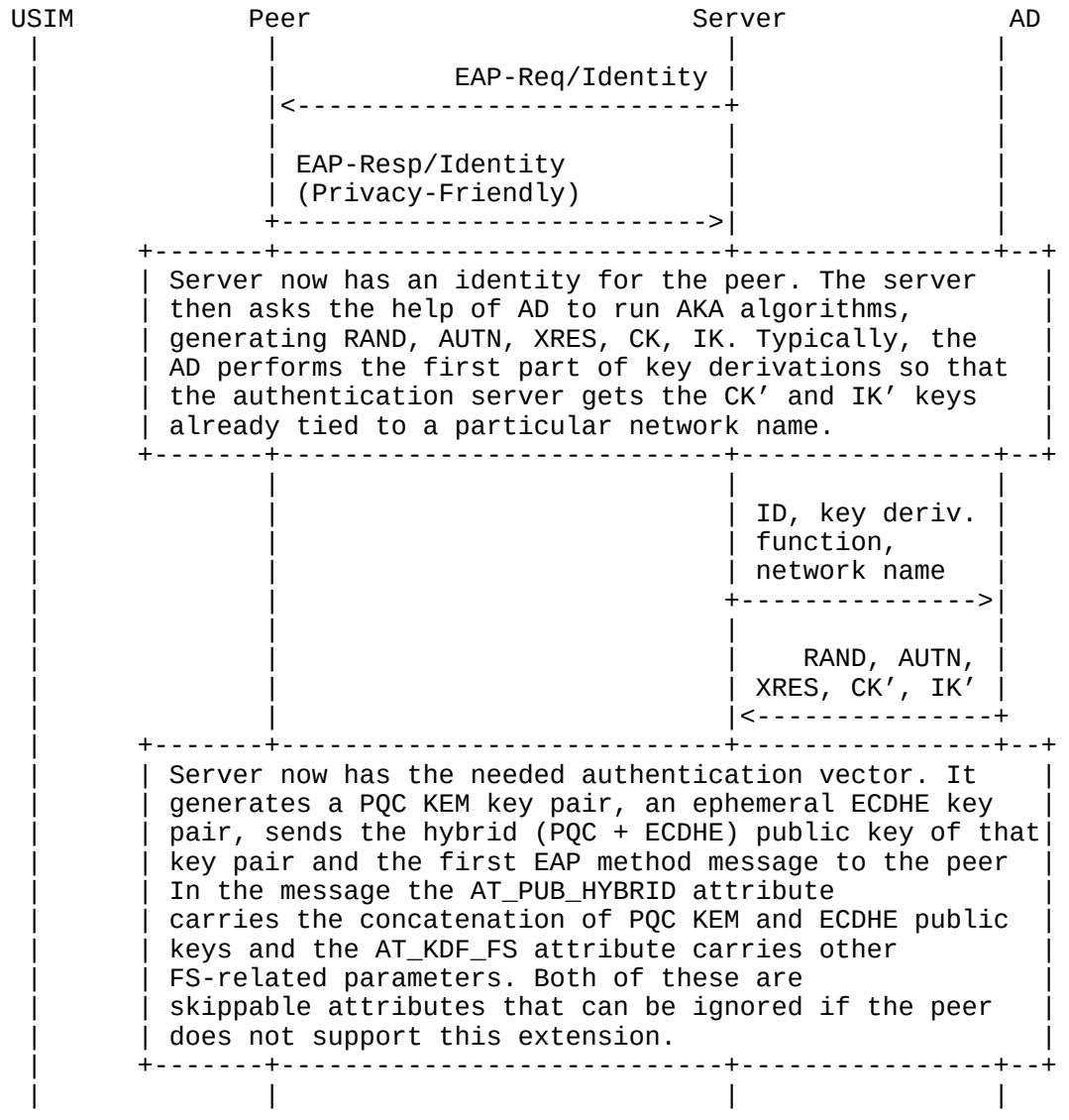
* Multiple AT_KDF_FS attributes is included in the EAP-Request to handle the EAP peer not supporting PQ/T Hybrid KEM in HPKE.

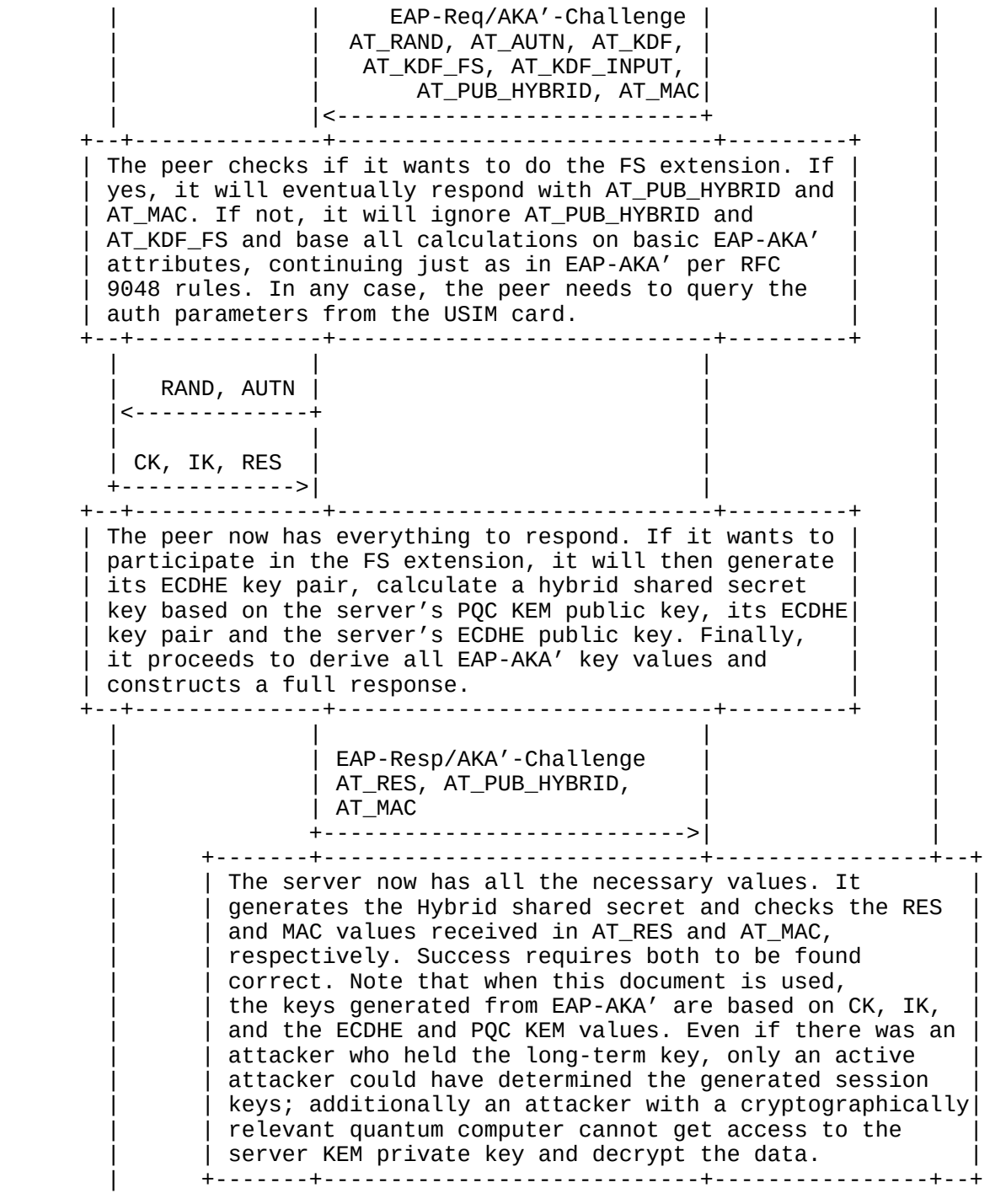
* The Hybrid key derivation function will be included first in the EAP-Request to indicate a higher priority than the traditional key derivation function.

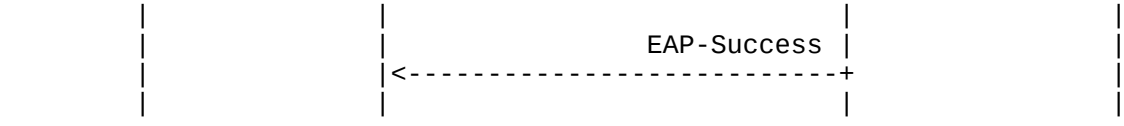
6. Protocol Construction

This section defines the construction for hybrid key exchange in EAP-AKA' FS. Hybrid key exchange refers to using multiple key exchange algorithms simultaneously and combining the result with the goal of providing security even if all but one of the component algorithms is broken. It is motivated by transition to post-quantum cryptography.

6.1. Protocol Call Flow




EAP-Req/AKA'-Challenge
AT_RAND, AT_AUTN, AT_KDF,
AT_KDF_FS, AT_KDF_INPUT,
AT_PUB_HYBRID, AT_MAC
The peer checks if it wants to do the FS extension. If yes, it will eventually respond with AT_PUB_HYBRID and AT_MAC. If not, it will ignore AT_PUB_HYBRID and AT_KDF_FS and base all calculations on basic EAP-AKA' attributes, continuing just as in EAP-AKA' per RFC 9048 rules. In any case, the peer needs to query the auth parameters from the USIM card.
RAND, AUTN
CK, IK, RES
The peer now has everything to respond. If it wants to participate in the FS extension, it will then generate its ECDHE key pair, calculate a hybrid shared secret key based on the server's PQC KEM public key, its ECDHE key pair and the server's ECDHE public key. Finally, it proceeds to derive all EAP-AKA' key values and constructs a full response.
EAP-Resp/AKA'-Challenge
AT_RES, AT_PUB_HYBRID,
AT_MAC
The server now has all the necessary values. It generates the Hybrid shared secret and checks the RES and MAC values received in AT_RES and AT_MAC, respectively. Success requires both to be found correct. Note that when this document is used, the keys generated from EAP-AKA' are based on CK, IK, and the ECDHE and PQC KEM values. Even if there was an attacker who held the long-term key, only an active attacker could have determined the generated session keys; additionally an attacker with a cryptographically relevant quantum computer cannot get access to the server KEM private key and decrypt the data.



6.2. Key Steps in protocol construction

We outline the following key steps in the protocol:

* Server generates the PQC KEM Public key(pq_PK), private key (pq_SK) pair and the ECDH public key (trad_PK), private key (trad_SK) pair. The server will generate and send the EAP AKA' Authentication Vector (AV).

* The server will store the expected response XRES, the ECDH private key trad_SK and the PQC KEM private key pq_SK. The server will forward the EAP AKA' AV to peer along with pq_PK and trad_PK.

* The USIM will validate the AUTH received, also verifies the MAC. After the verification is successful and if the peer also supports the Forward secrecy, peer will invoke Encapsulate using concat(pq_PK, trad_PK) as defined in section 5.4 of [I-D.irtf-cfrg-hybrid-kems].

"ct" is the concatenation of the ciphertext from PQC KEM and encapsulated key from ECDH whereas "ss" is hybrid shared secret key. Hybrid shared key ss is generated by the peer using the Encapsulate() ([I-D.irtf-cfrg-hybrid-kems]).

* The peer will send the Authentication response RES and ct to the server.

* The server will verify the RES with XRES. The server will use the ct, PQC KEM private key pq_SK and ECDH private key trad_SK to generate shared secret.

The generated ss from Decapsulate is the hybrid shared secret key derived from PQC KEM and traditional ECDH. The peer and the server first generate the MK_HYBRID and subsequently generate MSK, EMSK as shown below:

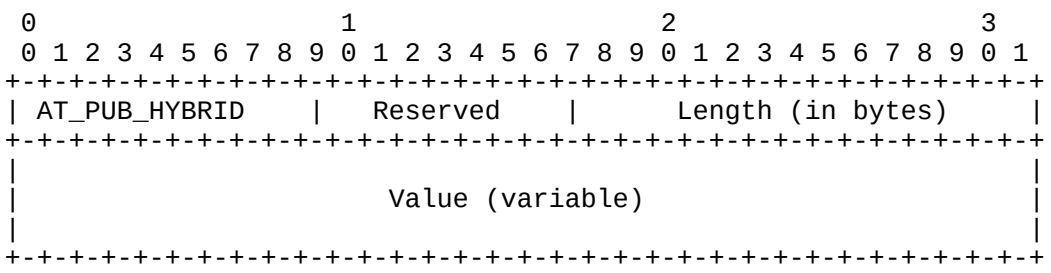
```
MK = PRF'(IK'|CK',"EAP-AKA'"|Identity)
HYBRID_SHARED_SECRET, ct = Encapsulate(pKR)
MK_HYBRID = PRF'(IK'|CK'| HYBRID_SHARED_SECRET,"EAP-AKA' FS"| Identity)
K_encr = MK[0..127]
K_aut = MK[128..383]
K_re = MK_HYBRID [0..255]
MSK = MK_HYBRID [256..767]
EMSK = MK_HYBRID [768..1279]
```

where, pkR is concatenation of PQC KEM and traditional public keys of the receiver, ct is concatenation of the ciphertext from the PQC KEM and encapsulated key from ECDH; the Encapsulate() function is perfomed by the peer only.

7. Extensions to EAP-AKA' FS

7.1. AT_PUB_HYBRID

The format of the AT_PUB_HYBRID attribute is shown below.



The fields are as follows:

AT_PUB_HYBID:

This is set to TBA1 BY IANA.

Reserved:

A 1-byte field reserved for future use. Including this field ensures that the fixed header (Type, Reserved, Length) is 4 bytes long, maintaining 4-byte alignment for the following Value field. The Reserved field MUST be set to 0 on transmission and ignored on receipt.

Length:

A 2-byte unsigned integer indicating the total length of the attribute in bytes, including the Type, Reserved, Length, and Value fields, as well as any padding. The length is expressed in multiples of 4 bytes.

This differs from the attribute format used in EAP-AKA [RFC4187], where the Length field is 1 byte. The modification is necessary because PQ/T Hybrid KEM public keys, such as X25519 and ML-KEM-768 (e.g., X-Wing), would exceed the 1024-byte limit imposed by the original EAP-AKA format.

Value:

* EAP-Request: It contains the public key, which is the concatenation of traditional and PQC KEM public keys from the EAP server.
* EAP-Response: It contains the encapsulated key, which is formed by concatenating the ciphertext (ct) from the PQC KEM Encapsulation function and the encapsulated key (enc) from the traditional KEM algorithm and from the EAP peer.

Because the length of the attribute must be a multiple of 4 bytes, the sender pads the Value field with zero bytes when necessary. To retain the security of the keys, the sender SHALL generate a fresh value for each run of the protocol.

8. Security Considerations

ML-KEM is IND-CCA2 secure based on multiple analyses. The ML-KEM variant and its underlying components should be selected consistently with the desired security level. For further clarity on the sizes and security levels of ML-KEM variants, please refer to the tables in Sections 12 and 13 of [I-D.ietf-pquip-pqc-engineers].

The security of the ML-KEM algorithm depends on a high-quality pseudo-random number generator. For further discussion on random number generation, see [RFC4086].

In general, good cryptographic practice dictates that a given ML-KEM key pair should be used in only one EAP session. This practice mitigates the risk that compromising one EAP session will not compromise the security of another EAP session and is essential for maintaining forward security.

For security properties of traditional ECDHE for EAP-AKA FS, see section 7 of [RFC9678]. The overall Hybrid scheme needs to be IND-CCA2 robust; i.e., atleast one of the schemes should be IND-CCA2 secure.

9. IANA Considerations

Two new values (TBA2) and (TBA3) in the skippable range need to be assigned by IANA for AT_PUB_HYBRID (Section 7.1) in the "Attribute Types" registry under the "EAP-AKA and EAP-SIM Parameters" group.

IANA is requested to update the registry "EAP-AKA' AT_KDF_FS Key Derivation Function Values" with the Hybrid key derivation function entry:

Value	Description	Reference
TBA2	QSF-KEM(ML-KEM-768,P-256)-XOF(SHAKE256)-KDF(SHA3-256)	[TBD BY IANA: THIS RFC]
TBA3	KitchenSink-KEM(ML-KEM-768,X25519)-XOF(SHAKE256)-KDF(HKDF-SHA-256)	[TBD BY IANA: THIS RFC]

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4187] Arkko, J. and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)", RFC 4187, DOI 10.17487/RFC4187, January 2006, <https://www.rfc-editor.org/rfc/rfc4187>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9048] Arkko, J., Lehtovirta, V., Torvinen, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA')", RFC 9048, DOI 10.17487/RFC9048, October 2021, <https://www.rfc-editor.org/rfc/rfc9048>.

[RFC9180] Barnes, R., Bhargavan, K., Lipp, B., and C. Wood, "Hybrid Public Key Encryption", RFC 9180, DOI 10.17487/RFC9180, February 2022, <https://www.rfc-editor.org/rfc/rfc9180>.

[RFC9678] Arkko, J., Norrman, K., and J. Preu Mattsson, "Forward Secrecy Extension to the Improved Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS)", RFC 9678, DOI 10.17487/RFC9678, March 2025, <https://www.rfc-editor.org/rfc/rfc9678>.

10.2. Informative References

[FIPS203] "FIPS-203: Module-Lattice-based Key-Encapsulation Mechanism Standard", <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf>.

[I-D.ietf-hpke-pq] Barnes, R., "Post-Quantum and Post-Quantum/Traditional Hybrid Algorithms for HPKE", Work in Progress, Internet-Draft, draft-ietf-hpke-pq-01, 30 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-hpke-pq-01>.

[I-D.ietf-pquip-pqc-engineers] Banerjee, A., Reddy.K, T., Schoinianakis, D., Hollebeek, T., and M. Ounsworth, "Post-Quantum Cryptography for Engineers", Work in Progress, Internet-Draft, draft-ietf-pquip-pqc-engineers-13, 1 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqc-engineers-13>.

[I-D.ietf-pquip-pqt-hybrid-terminology] D, F., P, M., and B. Hale, "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft, draft-ietf-pquip-pqt-hybrid-terminology-06, 10 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqt-hybrid-terminology-06>.

[I-D.irtf-cfrg-hybrid-kems] Connolly, D., Barnes, R., and P. Grubbs, "Hybrid PQ/T Key Encapsulation Mechanisms", Work in Progress, Internet-Draft, draft-irtf-cfrg-hybrid-kems-05, 20 July 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-hybrid-kems-05>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/rfc/rfc4086>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI 10.17487/RFC6090, February 2011, <https://www.rfc-editor.org/rfc/rfc6090>.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curve Diffie-Hellman (ECDH) and Signatures in JSON Object Signing and Encryption (JOSE)", RFC 8037, DOI 10.17487/RFC8037, January 2017, <https://www.rfc-editor.org/rfc/rfc8037>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/rfc/rfc9052>.

Appendix A. Acknowledgements

This draft leverages text from [RFC9678].

Authors' Addresses

Aritra Banerjee
Nokia
London
United Kingdom
Email: aritra.banerjee@nokia.com

Tirumaleswar Reddy
Nokia
Bangalore
Karnataka
India
Email: kondtir@gmail.com