Workgroup: Network Working Group
Internet-Draft: draft-ietf-emu-aka-pfs-12
Updates: 5448, 9048 (if approved)
Published: 19 February 2024
Intended Status: Standards Track
Expires: 22 August 2024
Authors: J. Arkko   K. Norrman   J. Preuß Mattsson
         Ericsson   Ericsson   Ericsson

Forward Secrecy for the Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS)

Abstract

This document updates RFC 9048, the improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA'), with an optional extension providing ephemeral key exchange. Similarly, this document also updates the earlier version of the EAP-AKA' specification in RFC 5448. The extension EAP-AKA' Forward Secrecy (EAP-AKA' FS), when negotiated, provides forward secrecy for the session keys generated as a part of the authentication run in EAP-AKA'. 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. In addition, EAP-AKA' FS mitigates passive attacks (e.g., large scale pervasive monitoring) against future sessions. This forces attackers to use active attacks instead.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Many different attacks have been reported as part of revelations associated with pervasive surveillance. Some of the reported attacks involved compromising the Universal Subscriber Identity Module (USIM) card supply chain. Attacks revealing the AKA long-term key may occur for instance, during the manufacturing process of USIM cards, during the transfer of the cards and associated information to the operator, and when a system is running. Since the publication of reports about such attacks [Heist2015], manufacturing and provisioning processes have gained much scrutiny and have improved.

However, the danger of resourceful attackers attempting to gain information about long-term keys is still a concern because these keys are high-value targets. Note that the attacks are largely independent of the used authentication technology; the issue is not vulnerabilities in algorithms or protocols, but rather the possibility of someone gaining unauthorized access to key material. Furthermore, an explicit goal of the IETF is to ensure that we understand the surveillance concerns related to IETF protocols and take appropriate countermeasures [RFC7258].

While strong protection of manufacturing and other processes is essential in mitigating surveillance and other risks associated with AKA long-term keys, there are also protocol mechanisms that can help.

This document updates [RFC9048], the Improved 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA') method, with an optional extension providing ephemeral key exchange minimizing the impact of long-term key compromise and strengthens the identity privacy requirements. This is important, given the large number of users of AKA in mobile networks.

The extension, when negotiated, provides Forward Secrecy (FS) [DOW1992] for the session key generated as a part of the authentication run in EAP-AKA'. This prevents an attacker who has gained access to the long-term key in a USIM card from getting access to past session keys. In addition to FS, the included Diffie-Hellman exchange, forces attackers to be active if they want access to future session keys even if they have access to the long-term key. This is beneficial, because active attacks demand much more resources to launch, and are easier to detect. As with other protocols, an active attacker with access to the long-term key material will of course be able to attack all future communications, but risks detection, particularly if done at scale.

It should also be noted that 5G network architecture [TS.33.501] includes the use of the EAP framework for authentication. While any methods can be run, the default authentication method within that context will be EAP-AKA'. As a result, improvements in EAP-AKA' security have a potential to improve security for many users.

2. Requirements Language

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. Protocol Design and Deployment Objectives

The extension specified here re-uses large portions of the current structure of 3GPP interfaces and functions, with the rationale that this will make the construction more easily adopted. In particular, the construction keeps the interface between the USIM and the mobile terminal intact. As a consequence, there is no need to roll out new credentials to existing subscribers. The work is based on an earlier paper [TrustCom2015], and uses much of the same material, but applied to EAP rather than the underlying AKA method.

It has been a goal to implement this change as an extension of the widely supported EAP-AKA' method, rather than a completely new authentication method. The extension is implemented as a set of new, optional attributes, that are provided alongside the base attributes in EAP-AKA'. Old implementations can ignore these attributes, but their presence will nevertheless be verified as part of base EAP-AKA' integrity verification process, helping protect against bidding down attacks. This extension does not increase the number of rounds necessary to complete the protocol.

The use of this extension is at the discretion of the authenticating parties. It should be noted that FS and defenses against passive attacks do not solve all problems, but they can provide a partial defense that increases the cost and risk associated with pervasive surveillance.

While adding forward secrecy to the existing mobile network infrastructure can be done in multiple different ways, this document specifies a solution that is relatively easily deployable. In particular:

* As noted above, no new credentials are needed; there is no change to USIM cards.

* FS property can be incorporated into any current or future system that supports EAP, without changing any network functions beyond the EAP endpoints.

* Key generation happens at the endpoints, enabling highest grade key material to be used both by the endpoints and the intermediate systems (such as access points that are given access to specific keys).

* While EAP-AKA' is just one EAP method, for practical purposes forward secrecy being available for both EAP-TLS [RFC5216] [RFC9190] and EAP-AKA' ensures that for many practical systems forward secrecy can be enabled for either all or significant fraction of users.

4. Background

The reader is assumed to have basic understanding of the EAP framework [RFC3748].

4.1. AKA

We use the term Authentication and Key Agreement (AKA) for the main authentication and key agreement protocol used by 3GPP mobile networks from the third generation (3G) and onward. Later generations adds new features to AKA, but the core remains the same. It is based on challenge-response mechanisms and symmetric cryptography. In contrast to its earlier GSM counterparts, AKA provides long key lengths and mutual authentication. The phone typically executes AKA in a USIM. USIM is technically just an application that can reside on a removable UICC (Universal Integrated Circuit Card), an embedded UICC, or integrated in a Trusted Execution Environment (TEE). In this document we use the term "USIM card" to refer to any Subscriber Identity Module capable of running AKA.

The goal of AKA is to mutually authenticate the USIM and the so-called home environment, which is the authentication server in the subscribers home operator's network.

AKA works in the following manner:

* The USIM and the home environment have agreed on a long-term symmetric key beforehand.

* The actual authentication process starts by having the home environment produce an authentication vector, based on the long-term key and a sequence number. The authentication vector 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.

* The authentication vector is passed to the serving network, which uses it to authenticate the device.

* The RAND and the AUTN are delivered to the USIM.

* The USIM verifies the AUTN, again based on the long-term key and the sequence number. If this process is successful (the AUTN is valid and the sequence number used to generate AUTN is within the correct range), the USIM produces an authentication result RES and sends it to the serving network.

* The serving network verifies that the result from the USIM matches the expected value in the authentication vector. If it does, the USIM is considered authenticated, and IK and CK can be used to protect further communications between the USIM and the home environment.

4.2. EAP-AKA' Protocol

When AKA is embedded into EAP, the authentication processing on the network side is moved to the home environment. The 3GPP authentication database (AD) generates authentication vectors. The 3GPP authentication server takes the role of EAP server. The USIM combined with the mobile phone takes the role of the client. The difference between EAP-AKA [RFC4187] and EAP-AKA' [RFC9048] is that EAP-AKA' binds the derived keys to the name of access network. Figure 1 describes the basic flow in the EAP-AKA' authentication process. The definition of the full protocol behavior, along with the definition of attributes AT_RAND, AT_AUTN, AT_MAC, and AT_RES can be found in [RFC9048] and [RFC4187]. Note the use of EAP-terminology from hereon. That is, the 3GPP serving network takes on the role of an EAP access network.

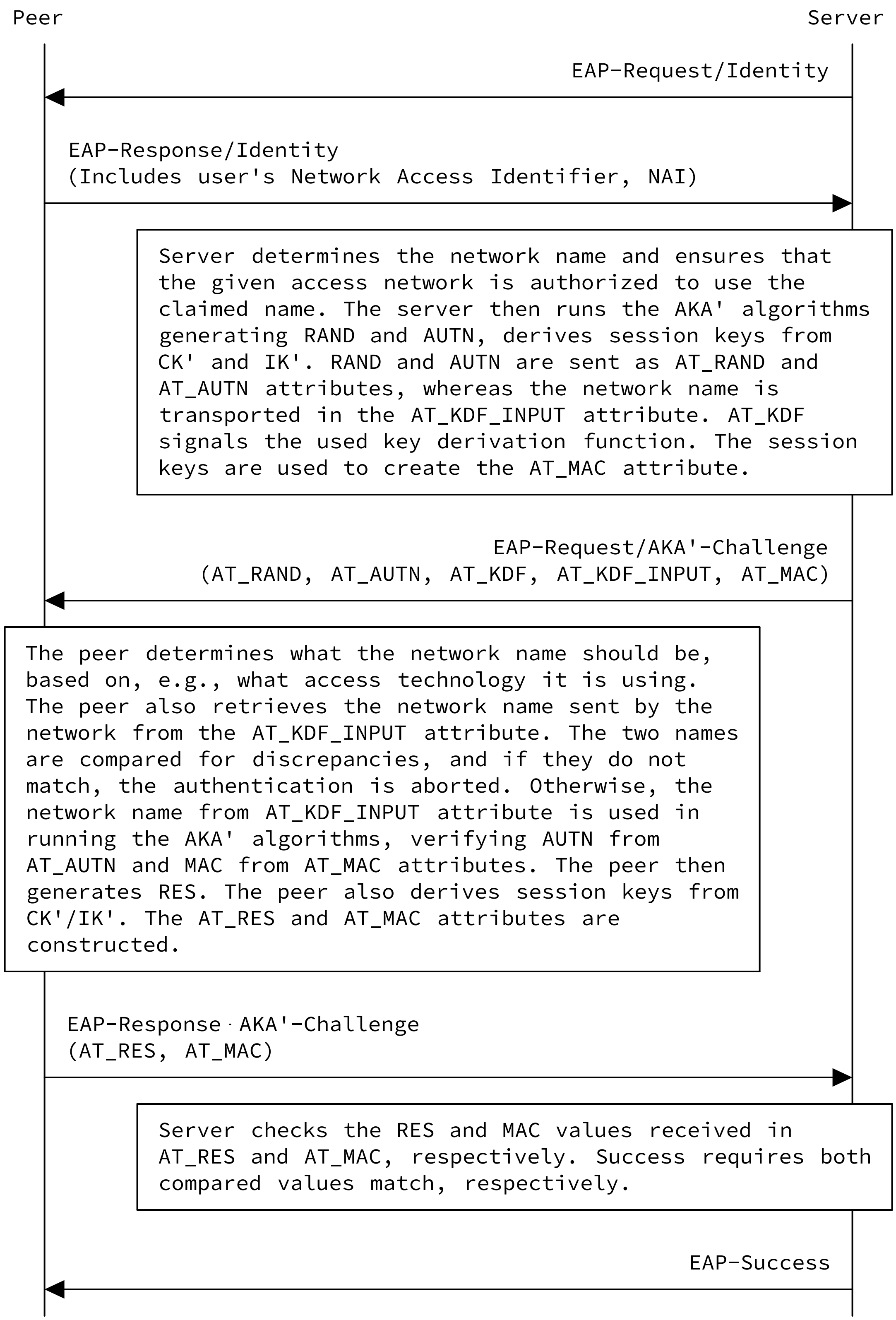

Figure 1: EAP-AKA' Authentication Process

4.3. Attacks Against Long-Term Keys in Smart Cards

The general security properties and potential vulnerabilities of AKA and EAP-AKA' are discussed in [RFC9048].

An important question in that discussion relates to the potential compromise of long-term keys, as discussed earlier. Attacks on long-term keys are not specific to AKA or EAP-AKA', and all security systems fail at least to some extent if key material is stolen. However, it would be preferable to retain some security even in the face of such attacks. This document specifies a mechanism that reduces risks to compromise of key material belonging to previous sessions, before the long-term keys were compromised. It also forces attackers to be active even after the compromise.

5. Protocol Overview

Forward secrecy for EAP-AKA' is achieved by using an Elliptic Curve Diffie-Hellman (ECDH) exchange [RFC7748]. To provide FS, the exchange must be run in an ephemeral manner, i.e., both sides generate temporary keys according to the negotiated ciphersuite, e.g., for X25519 this is done as specified in [RFC7748]. This method is referred to as ECDHE, where the last 'E' stands for Ephemeral. The two initially registered elliptic curves and their wire formats are chosen to align with the elliptic curves and formats specified for Subscription Concealed Identifier (SUCI) encryption in Appendix C.3.4 of 3GPP TS 33.501 [TS.33.501].

The enhancements in the EAP-AKA' FS protocol are compatible with the signaling flow and other basic structures of both AKA and EAP-AKA'. The intent is to implement the enhancement as optional attributes that legacy implementations ignore.

The purpose of the protocol is to achieve mutual authentication between the EAP server and peer, and to establish keying material for secure communication between the two. This document specifies the calculation of key material, providing new properties that are not present in key material provided by EAP-AKA' in its original form.

Figure 2 below describes the overall process. Since the goal has been to not require new infrastructure or credentials, the flow diagrams also show the conceptual interaction with the USIM card and the home environment. Recall that the home environment represent the 3GPP Authentication Database (AD) and server. The details of those interactions are outside the scope of this document, however, and the reader is referred to the 3GPP specifications. For 5G this is specified in 3GPP TS 33.501 [TS.33.501]

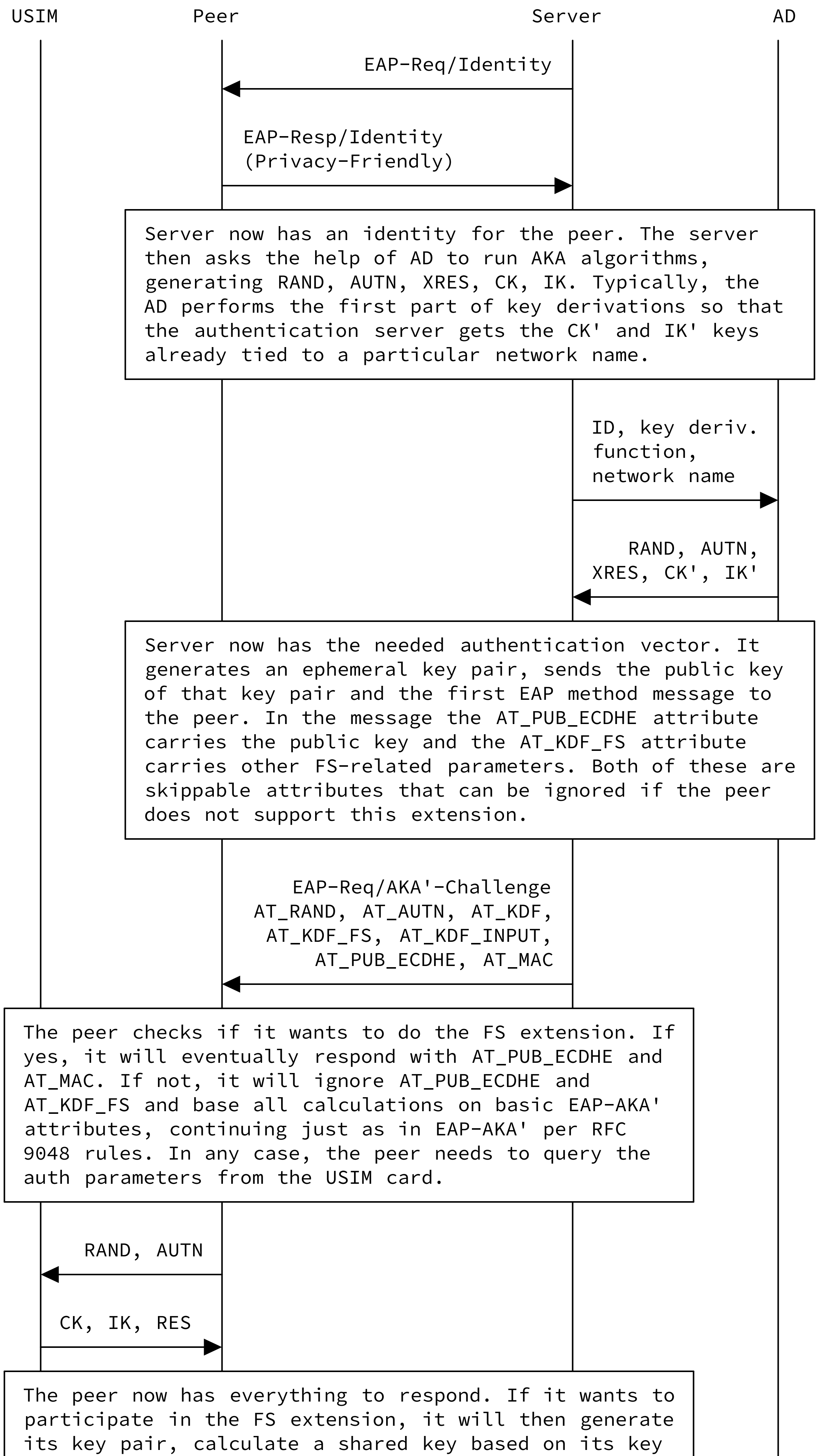
USIM
Peer
Server
AD
EAP-Req/Identity
EAP-Resp/Identity (Privacy-Friendly)
Server now has an identity for the peer. The server then asks the help of AD to run AKA algorithms, generating RAND, AUTN, XRES, CK, IK. Typically, the AD performs the first part of key derivations so that the authentication server gets the CK' and IK' keys already tied to a particular network name.
ID, key deriv. function, network name
RAND, AUTN, XRES, CK', IK'
Server now has the needed authentication vector. It generates an ephemeral key pair, sends the public key of that key pair and the first EAP method message to the peer. In the message the AT_PUB_ECDHE attribute carries the public key and the AT_KDF_FS attribute carries other FS-related parameters. Both of these are skippable attributes that can be ignored if the peer does not support this extension.
EAP-Req/AKA'-Challenge AT_RAND, AT_AUTN, AT_KDF, AT_KDF_FS, AT_KDF_INPUT, AT_PUB_ECDHE, AT_MAC
The peer checks if it wants to do the FS extension. If yes, it will eventually respond with AT_PUB_ECDHE and AT_MAC. If not, it will ignore AT_PUB_ECDHE 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 key pair, calculate a shared key based on its key

Figure 2: EAP-AKA' FS Authentication Process

6. Extensions to EAP-AKA'

6.1. AT_PUB_ECDHE

The AT_PUB_ECDHE carries an ECDHE value.

The format of the AT_PUB_ECDHE attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

AT_PUB_ECDHE	Length	Value

The fields are as follows:

AT_PUB_ECDHE
This is set to TBA1 BY IANA.

Length
The length of the attribute, set as other attributes in EAP-AKA [RFC4187]. The length is expressed in multiples of 4 bytes. The length includes the attribute type field, the Length field itself, and the Value field (along with any padding).

Value
This value is the sender's ECDHE public key. The value depends on AT_KDF_FS and is calculated as follows:

* For X25519, the length of this value is 32 bytes, encoded as specified in [RFC7748] Section 5.

* For P-256, the length of this value is 33 bytes, encoded using the compressed form specified in Section 2.3.3 of [SEC1].

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.

6.2. AT_KDF_FS

The AT_KDF_FS indicates the used or desired forward secrecy key generation function, if the Forward Secrecy (FS) extension is used. It will also indicate the used or desired ECDHE group. A new attribute is needed to carry this information, as AT_KDF carries the basic KDF value which is still used together with the forward secrecy KDF value. The basic KDF value is also used by those EAP peers that cannot or do not want to use this extension.

This document only specifies the behavior relating to the following combinations of basic KDF values and forward secrecy KDF values: The basic KDF value in AT_KDF is 1, as specified in [RFC5448] and [RFC9048], and the forward secrecy KDF values in AT_KDF_FS are 1 or 2, as specified below and in Section 6.3.

Any future specifications that add either new basic KDF or new forward secrecy KDF values need to specify how they are treated and what combinations are allowed. This requirement is an update to how [RFC5448] and [RFC9048] may be extended in the future.

The format of the AT_KDF_FS attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------------------------------+
| AT_KDF_FS     | Length        | FS Key Derivation Function    |
+---------------+---------------+-------------------------------+
```

The fields are as follows:

AT_KDF_FS
This is set to TBA2 BY IANA.

Length
The length of the attribute, MUST be set to 1.

FS Key Derivation Function
An enumerated value representing the forward secrecy key derivation function that the server (or peer) wishes to use. See Section 6.3 for the functions specified in this document. Note: This field has a different name space than the similar field in the AT_KDF attribute Key Derivation Function defined in [RFC9048].

Servers MUST send one or more AT_KDF_FS attributes in the EAP-Request/AKA'-Challenge message. These attributes represent the desired functions ordered by preference, the most preferred function being the first attribute. The most preferred function is the only one that the server includes a public key value for, however. So for a set of AT_KDF_FS attributes, there is always only one AT_PUB_ECDHE attribute.

Upon receiving a set of these attributes:

* If the peer supports and is willing to use the FS Key Derivation Function indicated by the first AT_KDF_FS attribute, and is willing and able to use the extension defined in this document, the function is taken into use without any further negotiation.

* If the peer does not support this function or is unwilling to use it, it responds to the server with an indication that a different function is needed. Similarly with the negotiation process defined in [RFC9048] for AT_KDF, the peer sends EAP-Response/AKA'-Challenge message that contains only one attribute, AT_KDF_FS with the value set to the desired alternative function from among the ones suggested by the server earlier. If there is no suitable alternative, the peer has a choice of either falling back to EAP-AKA' or behaving as if AUTN had been incorrect and failing authentication (see Figure 3 of [RFC4187]). The peer MUST fail the authentication if there are any duplicate values within the list of AT_KDF_FS attributes (except where the duplication is due to a request to change the key derivation function; see below for further information).

* If the peer does not recognize the extension defined in this document or is unwilling to use it, it ignores the AT_KDF_FS attribute.

Upon receiving an EAP-Response/AKA'-Challenge with AT_KDF_FS from the peer, the server checks that the suggested AT_KDF_FS value was one of the alternatives in its offer. The first AT_KDF_FS value in the message from the server is not a valid alternative. If the peer has replied with the first AT_KDF_FS value, the server behaves as if AT_MAC of the response had been incorrect and fails the authentication. For an overview of the failed authentication process in the server side, see Section 3 and Figure 2 in [RFC4187]. Otherwise, the server re-sends the EAP-Response/AKA'-Challenge message, but adds the selected alternative to the beginning of the list of AT_KDF_FS attributes, and retains the entire list following it. Note that this means that the selected alternative appears twice in the set of AT_KDF values. Responding to the peer's request to change the FS Key Derivation Function is the only valid situation where such duplication may occur.

When the peer receives the new EAP-Request/AKA'-Challenge message, it MUST check that the requested change, and only the requested change occurred in the list of AT_KDF_FS attributes. If yes, it continues. If not, it behaves as if AT_MAC had been incorrect and fails the authentication. If the peer receives multiple EAP-Request/AKA'-Challenge messages with differing AT_KDF_FS attributes without having requested negotiation, the peer MUST behave as if AT_MAC had been incorrect and fail the authentication.

6.3. Forward Secrecy Key Derivation Functions

Two new FS Key Derivation Function types are defined for "EAP-AKA' with ECDHE and X25519", represented by value 1, and "EAP-AKA' with ECDHE and P-256", represented by value 2. These represent a particular choice of key derivation function and at the same time selects an ECDHE group to be used.

The FS Key Derivation Function type value is only used in the AT_KDF_FS attribute. When the forward secrecy extension is used, the AT_KDF_FS attribute determines how to derive the keys MK_ECDHE, K_re, MSK, and EMSK. The AT_KDF_FS attribute should not be confused with the different range of key derivation functions that can be represented in the AT_KDF attribute as defined in [RFC9048]. When the forward secrecy extension is used, the AT_KDF attribute only specifies how to derive the keys MK, K_encr, and K_aut.

Key derivation in this extension produces exactly the same keys for internal use within one authentication run as EAP-AKA' [RFC9048] does. For instance, K_aut that is used in AT_MAC is still exactly as it was in EAP-AKA'. The only change to key derivation is in re-authentication keys and keys exported out of the EAP method, MSK and EMSK. As a result, EAP-AKA' attributes such as AT_MAC continue to be usable even when this extension is in use.

When the FS Key Derivation Function field in the AT_KDF_FS attribute is set to 1 or 2 and the Key Derivation Function field in the AT_KDF attribute is set to 1, the Master Key (MK) and accompanying keys are derived as follows.

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

Requirements for how to securely generate, validate, and process the ephemeral public keys depend on the elliptic curve.

For P-256 the SHARED_SECRET is the shared secret computed as specified in Section 5.7.1.2 of [SP-800-56A]. Public key validation requirements are defined in Section 5 of [SP-800-56A]. At least partial public-key validation MUST be done for the ephemeral public keys. The uncompressed y-coordinate can be computed as described in Section 2.3.4 of [SEC1].

For X25519 the SHARED_SECRET is the shared secret computed as specified in Section 6.1 of [RFC7748]. Both the peer and the server MAY check for zero-value shared secret as specified in Section 6.1 of [RFC7748].

> Note: The way that shared secret is tested for zero can, if performed inappropriately, provide an ability for attackers to listen to CPU power usage side channels. Refer to [RFC7748] for a description of how to perform this check in a way that it does not become a problem.

If validation of the other party's ephemeral public key or the shared secret fails, a party MUST behave as if the current EAP-AKA' authentication process starts again from the beginning.

The rest of computation proceeds as defined in Section 3.3 of [RFC9048].

For readability, an explanation of the notation used above is copied here: [n..m] denotes the substring from bit n to m. PRF' is a new pseudo-random function specified in [RFC9048]. K_encr is the encryption key, 128 bits, K_aut is the authentication key, 256 bits, K_re is the re-authentication key, 256 bits, MSK is the Master Session Key, 512 bits, and EMSK is the Extended Master Session Key, 512 bits. MSK and EMSK are outputs from a successful EAP method run [RFC3748].

CK and IK are produced by the AKA algorithm. IK' and CK' are derived as specified in [RFC9048] from IK and CK.

The value "EAP-AKA'" is an eight-characters-long ASCII string. It is used as is, without any trailing NUL characters. Similarly, "EAP-AKA' FS" is an eleven-characters-long ASCII string, also used as is.

Identity is the peer identity as specified in Section 7 of [RFC4187]. A privacy-friendly identifier [RFC9048] SHALL be used.

6.4. ECDHE Groups

The selection of suitable groups for the elliptic curve computation is necessary. The choice of a group is made at the same time as deciding to use of particular key derivation function in AT_KDF_FS.

For "EAP-AKA' with ECDHE and X25519" the group is the Curve25519 group specified in [RFC7748]. The support for this group is REQUIRED.

For "EAP-AKA' with ECDHE and P-256" the group is the NIST P-256 group (SEC group secp256r1), specified in Section 3.2.1.3 of

[SP-800-186] or alternatively Section 2.4.2 of [SEC2]. The support for this group is REQUIRED.

The term "support" here means that the group MUST be implemented.

6.5. Message Processing

This section specifies the changes related to message processing when this extension is used in EAP-AKA'. It specifies when a message may be transmitted or accepted, which attributes are allowed in a message, which attributes are required in a message, and other message-specific details, where those details are different for this extension than the base EAP-AKA' or EAP-AKA protocol. Unless otherwise specified here, the rules from [RFC9048] or [RFC4187] apply.

6.5.1. EAP-Request/AKA'-Identity

No changes, except that the AT_KDF_FS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these attributes in a received message MUST be ignored.

6.5.2. EAP-Response/AKA'-Identity

No changes, except that the AT_KDF_FS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these attributes in a received message MUST be ignored. The peer identifier SHALL comply with the privacy-friendly requirements of [RFC9190]. An example of a compliant way of constructing a privacy-friendly peer identifier is using a non-NULL SUCI [TS.33.501].

6.5.3. EAP-Request/AKA'-Challenge

The server sends the EAP-Request/AKA'-Challenge on full authentication as specified by [RFC4187] and [RFC9048]. The attributes AT_RAND, AT_AUTN, and AT_MAC MUST be included and checked on reception as specified in [RFC4187]. They are also necessary for backwards compatibility.

In EAP-Request/AKA'-Challenge, there is no message-specific data covered by the MAC for the AT_MAC attribute. The AT_KDF_FS and AT_PUB_ECDHE attributes MUST be included. The AT_PUB_ECDHE attribute carries the server's public Diffie-Hellman key. If either AT_KDF_FS or AT_PUB_ECDHE is missing on reception, the peer MUST treat it as if neither one was sent, and the assume that the extension defined in this document is not in use.

The AT_RESULT_IND, AT_CHECKCODE, AT_IV, AT_ENCR_DATA, AT_PADDING, AT_NEXT_PSEUDONYM, AT_NEXT_REAUTH_ID and other attributes may be included as specified in Section 9.3 of [RFC4187].

When processing this message, the peer MUST process AT_RAND, AT_AUTN, AT_KDF_FS, AT_PUB_ECDHE before processing other attributes. Only if these attributes are verified to be valid, the peer derives keys and verifies AT_MAC. If the peer is unable or unwilling to perform the extension specified in this document, it proceeds as defined in [RFC9048]. Finally, if there is an error error, see Section 6.3.1. of [RFC4187].

6.5.4. EAP-Response/AKA'-Challenge

The peer sends EAP-Response/AKA'-Challenge in response to a valid EAP-Request/AKA'-Challenge message, as specified by [RFC4187] and [RFC9048]. If the peer supports and is willing to perform the extension specified in this protocol, and the server had made a valid request involving the attributes specified in Section 6.5.3, the peer responds per the rules specified below. Otherwise, the peer responds as specified in [RFC4187] and [RFC9048] and ignores the attributes related to this extension. If the peer has not received attributes related to this extension from the Server, and has a policy that requires it to always use this extension, it behaves as if AUTN had been incorrect and fails the authentication.

The AT_MAC attribute MUST be included and checked as specified in [RFC9048]. In EAP-Response/AKA'-Challenge, there is no message-specific data covered by the MAC. The AT_PUB_ECDHE attribute MUST be included, and carries the peer's public Diffie-Hellman key.

The AT_RES attribute MUST be included and checked as specified in [RFC4187]. When processing this message, the Server MUST process AT_RES before processing other attributes. The Server derives keys and verifies AT_MAC only when this attribute is verified to be valid.

If the Server has proposed the use of the extension specified in this protocol, but the peer ignores and continues the basic EAP-AKA' authentication, the Server makes policy decision of whether this is allowed. If this is allowed, it continues the EAP-AKA' authentication to completion. If it is not allowed, the Server MUST behave as if authentication failed.

The AT_CHECKCODE, AT_RESULT_IND, AT_IV, AT_ENCR_DATA and other attributes may be included as specified in Section 9.4 of [RFC4187].

6.5.5. EAP-Request/AKA'-Reauthentication

No changes, but note that the re-authentication process uses the keys generated in the original EAP-AKA' authentication, which, if the extension specified in this document is in use, employs key material from the Diffie-Hellman procedure.

6.5.6. EAP-Response/AKA'-Reauthentication

No changes, but as discussed in Section 6.5.5, re-authentication is based on the key material generated by EAP-AKA' and the extension defined in this document.

6.5.7. EAP-Response/AKA'-Synchronization-Failure

No changes, except that the AT_KDF_FS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these attributes in a received message MUST be ignored.

6.5.8. EAP-Response/AKA'-Authentication-Reject

No changes, except that the AT_KDF_FS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these attributes in a received message MUST be ignored.

6.5.9. EAP-Response/AKA'-Client-Error

No changes, except that the AT_KDF_FS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these attributes in a received message MUST be ignored.

6.5.10. EAP-Request/AKA'-Notification

No changes.

6.5.11. EAP-Response/AKA'-Notification

No changes.

7. Security Considerations

This section deals only with the changes to security considerations as they differ from EAP-AKA', or as new information has been gathered since the publication of [RFC9048].

As discussed in Section 1, forward secrecy is an important countermeasure against adversaries who gain access to the long-term keys. The long-term keys can be best protected with good processes, e.g., restricting access to the key material within a factory or among personnel, etc. Even so, not all attacks can be entirely ruled out. For instance, well-resourced adversaries may be able to coerce insiders to collaborate, despite any technical protection measures. The zero trust principles suggest that we assume that breaches are inevitable or have potentially already occurred, and that we need to minimize the impact of these breaches [NSA-ZT] [NIST-ZT]. One type of breach is key compromise or key exfiltration.

If a mechanism without ephemeral key exchange such as (5G-AKA, EAP-AKA') is used the effects of key compromise are devastating. There are serious consequences of not properly providing forward secrecy for the key establishment. For both control and user plane, and both directions:

1. An attacker can decrypt 5G communication that they previously recorded.

2. A passive attacker can eavesdrop (decrypt) all future 5G communication.

3. An active attacker can impersonate the UE or the Network and inject messages in an ongoing 5G connection between the real UE and the real network.

Best practice security today is to mandate forward secrecy (as is done in WPA3, EAP-TLS 1.3, EAP-TTLS 1.3, IKEv2, SSH, QUIC, WireGuard, Signal, etc.). It is recommended that in deployments, EAP-AKA methods without forward secrecy be phased out in the long term.

This extension provide assistance against passive attacks from attackers that have compromised the key material on USIM cards. Passive attacks are attractive for attackers performing large scale pervasive monitoring as they require much less resources and are much harder to detect. The extension also provides protection against active attacks as the attacker is forced to be on path during the AKA run and subsequent communication between the parties. Without forward secrecy an active attacker that has compromised the long-term key can inject messages in an connection between the real Peer and the real server without being on path. This extension is most useful when used in a context where the MSK/EMSK are used in protocols not providing forward secrecy. For instance, if used with IKEv2 [RFC7296], the session keys produced by IKEv2 have this property, so better characteristics of the MSK and EMSK is not that useful. However, typical link layer usage of EAP does not involve running another, forward secure, key exchange. Therefore, using EAP to authenticate access to a network is one situation where the extension defined in this document can be helpful.

This extension generates keying material using the ECDHE exchange in order to gain the FS property. This means that once an EAP-AKA' authentication run ends, the session that it was used to protect is closed, and the corresponding keys are destroyed, even someone who has recorded all of the data from the authentication run and session and gets access to all of the AKA long-term keys cannot reconstruct the keys used to protect the session or any previous session, without doing a brute force search of the session key space.

Even if a compromise of the long-term keys has occurred, FS is still provided for all future sessions, as long as the attacker does not become an active attacker.

The extension does not provide protection against active attackers with access to the long-term key that mount an on-path attack on future EAP-AKA' runs will be able to eavesdrop on the traffic protected by the resulting session key(s). Still, past sessions where FS was in use remain protected.

Using EAP-AKA' FS once provides forward secrecy. Forward secrecy limits the effect of key leakage in one direction (compromise of a key at time T2 does not compromise some key at time T1 where T1 < T2). Protection in the other direction (compromise at time T1 does not compromise keys at time T2) can be achieved by rerunning ECDHE frequently. If a long-term authentication key has been compromised, rerunning EAP-AKA' FS gives protection against passive attackers. Using the terms in [RFC7624], forward secrecy without rerunning ECDHE does not stop an attacker from doing static key exfiltration. Frequently rerunning EC(DHE) forces an attacker to do dynamic key exfiltration (or content exfiltration).

7.1. Deployment Considerations

Achieving FS requires that when a connection is closed, each endpoint MUST destroy not only the ephemeral keys used by the connection but also any information that could be used to recompute those keys.

Similarly, other parts of the system matter. For instance, when the keys generated by EAP are transported to a pass-through authenticator, such transport must also provide forward secure encryption with respect to the long-term keys used to establish its security. Otherwise, an adversary may attack the transport connection used to carry keys from EAP, and use this method to gain access to current and past keys from EAP, which in turn would lead to the compromise of anything protected by those EAP keys.

Of course, these considerations apply to any EAP method, not only this one.

7.2. Security Properties

The following security properties of EAP-AKA' are impacted through this extension:

Protected ciphersuite negotiation

EAP-AKA' has a negotiation mechanism for selecting the key derivation functions, and this mechanism has been extended by the extension specified in this document. The resulting mechanism continues to be secure against bidding down attacks.

There are two specific needs in the negotiation mechanism:

Negotiating key derivation function within the extension

The negotiation mechanism allows changing the offered key derivation function, but the change is visible in the final EAP- Request/AKA'-Challenge message that the server sends to the peer. This message is authenticated via the AT_MAC attribute, and carries both the chosen alternative and the initially offered list. The peer refuses to accept a change it did not initiate. As a result, both parties are aware that a change is being made and what the original offer was.

Negotiating the use of this extension

This extension is offered by the server through presenting the AT_KDF_FS and AT_PUB_ECDHE attributes in the EAP-Request/AKA'-Challenge message. These attributes are protected by AT_MAC, so attempts to change or omit them by an adversary will be detected.

Except of course, if the adversary holds the long-term key and is willing to engage in an active attack. Such an attack can, for instance, forge the negotiation process so that no FS will be provided. However, as noted above, an attacker with these capabilities will in any case be able to impersonate any party in the protocol and perform on-path attacks. That is not a situation that can be improved by a technical solution. However, as discussed in the introduction, even an attacker with access to the long-term keys is required to be on path on each AKA run and subsequent communication, which makes mass surveillance more laborious.

The security properties of the extension also depend on a policy choice. As discussed in Section 6.5.4, both the peer and the server make a policy decision of what to do when it was willing to perform the extension specified in this protocol, but the other side does not wish to use the extension. Allowing this has the benefit of allowing backwards compatibility to equipment that did not yet support the extension. When the extension is not supported or negotiated by the parties, no FS can obviously be provided.

If turning off the extension specified in this protocol is not allowed by policy, the use of legacy equipment that does not support this protocol is no longer possible. This may be appropriate when, for instance, support for the extension is sufficiently widespread, or required in a particular version of a mobile network.

Key derivation

This extension provides forward secrecy. As described in several places in this specification, this can be roughly summarized as that an attacker with access to long-term keys is unable to obtain session keys of ended past sessions, assuming these sessions deleted all relevant session key material. This extension does not change the properties related to re-authentication. No new Diffie-Hellman run is performed during the re-authentication allowed by EAP-AKA'. However, if this extension was in use when the original EAP-AKA' authentication was performed, the keys used for re-authentication (K_re) are based on the Diffie-Hellman keys, and hence continue to be equally safe against expose of the long-term key as the original authentication.

7.3. Denial-of-Service

In addition, it is worthwhile to discuss Denial-of-Service attacks and their impact on this protocol. The calculations involved in public key cryptography require computing power, which could be used in an attack to overpower either the peer or the server. While some forms of Denial-of-Service attacks are always possible, the following factors help mitigate the concerns relating to public key cryptography and EAP-AKA' FS.

* In 5G context, other parts of the connection setup involve public key cryptography, so while performing additional operations in EAP-AKA' is an additional concern, it does not change the overall situation. As a result, the relevant system components need to be dimensioned appropriately, and detection and management mechanisms to reduce the effect of attacks need to be in place.

* This specification is constructed so that a separation between the USIM and Peer on client side and the Server and AD on network side is possible. This ensures that the most sensitive (or legacy) system components cannot be the target of the attack. For instance, EAP-AKA' and public key cryptography takes place in the phone and not the low-power USIM card.

* EAP-AKA' has been designed so that the first actual message in the authentication process comes from the Server, and that this message will not be sent unless the user has been identified as an active subscriber of the operator in question. While the initial identity can be spoofed before authentication has succeeded, this reduces the efficiency of an attack.

* Finally, this memo specifies an order in which computations and checks must occur. When processing the EAP-Request/AKA'-Challenge message, for instance, the AKA authentication must be checked and succeed before the peer proceeds to calculating or processing the FS related parameters (see Section 6.5.4). The same is true of EAP-Response/AKA'-Challenge (see Section 6.5.4). This ensures that the parties need to show possession of the long-term key in some way, and only then will the FS calculations become active. This limits the Denial-of-Service to specific, identified subscribers. While botnets and other forms of malicious parties could take advantage of actual subscribers and their key material, at least such attacks are (a) limited in terms of subscribers they control, and (b) identifiable for the purposes of blocking the affected subscribers.

7.4. Identity Privacy

As specified in Section 6.5, the peer identity sent in the Identity Response message needs to follow the privacy-friendly requirements in [RFC9190].

7.5. Unprotected Data and Privacy

Unprotected data and metadata can reveal sensitive information and need to be selected with care. In particular, this applies to AT_KDF, AT_KDF_FS, AT_PUB_ECDHE, and AT_KDF_INPUT. AT_KDF, AT_KDF_FS, and AT_PUB_ECDHE reveal the used cryptographic algorithms, if these depend on the peer identity they leak information about the peer. AT_KDF_INPUT reveals the network name, although that is done on purpose to bind the authentication to a particular context.

An attacker observing network traffic may use the above types of information for traffic flow analysis or to track an endpoint.

7.6. Forward Secrecy within AT_ENCR

They keys K_encr and K_aut are calculated and used before the shared secret from the ephemeral key exchange is available.

K_encr and K_aut are used to encrypt and MAC data in the EAP-Req/AKA'-Challenge message, especially the DH g^x ephemeral pub key. At that point the server does not yet have the corresponding g^y from the peer and cannot compute the shared secret. K_aut is then used as the authentication key for the shared secret.

For K_encr though, none of the encrypted data sent in the EAP-Req/AKA'-Challenge message in the AT_ENCR attribute will be forward secret. That data may include re-authentication pseudonyms, so an adversary compromising the long-term key would be able to link re-authentication protocol-runs when pseudonyms are used, within a sequence of runs followed after a full EAP-AKA' authentication. No such linking would be possible across different full authentaction runs. If the pseudonum linkage risk is not acceptable, one way to avoid the linkage is to always require full EAP-AKA' authentication.

7.7. Post-Quantum Considerations

As of the publication of this document, it is unclear when or even if a quantum computer of sufficient size and power to exploit elliptic curve cryptography will exist. Deployments that need to consider risks decades into the future should transition to Post-Quantum Cryptography (PQC) in the not-too-distant future. Other systems may employ PQC when the quantum threat is more imminent. Current PQC algorithms have limitations compared to Elliptic Curve Cryptography (ECC) and the data sizes could be problematic for some constrained systems. If a Cryptographically Relevant Quantum Computer (CRQC) is built it could recover the SHARED_SECRET from the ECDHE public keys.

This would not affect the ability of EAP-AKA' - with or without this extension - to authenticate properly, however. As symmetric key cryptography is safe even if CRQCs are built, an adversary still will not be able to disrupt authentication as it requires computing a correct AT_MAC value. This computation requires the K_aut key which is based on MK and, ultimately, CK' and IK', but not SHARED_SECRET.

Other output keys do include SHARED_SECRET via MK_ECDHE, but still include also CK' and IK' which are entirely based on symmetric cryptography. As a result, an adversary with a quantum computer still cannot compute the other output keys either.

However, if the adversary has also obtained knowledge of the long-term key, they could then compute CK', IK', and SHARED_SECRET, and any derived output keys. This means that the introduction of a powerful enough quantum computer would disable this protocol extension's ability to provide the forward security capability. This would make it necessary to update the current ECC algorithms in this document to PQC algorithms. This document does not add such algorithms, but a future update can do that.

Symmetric algorithms used in EAP-AKA' FS such as HMAC-SHA-256 and the algorithms use to generate AT_AUTN and AT_RES are practically secure against even large robust quantum computers. EAP-AKA' FS is currently only specified for use with ECDHE key exchange algorithms, but use of any Key Encapsulation Method (KEM), including Post-Quantum Cryptography (PQC) KEMs, can be specified in the future. While the key exchange is specified with terms of the Diffie-Hellman protocol, the key exchange adheres to a KEM interface. AT_PUB_ECDHE would then contain either the ephemeral public key of the server or the SHARED_SECRET encapsulated with the server's public key. Note that the use of a KEM might require other changes such as including the ephemeral public key of the server in the key derivation to retain the property that both parties contribute randomness to the session key.

8. IANA Considerations

This extension of EAP-AKA' shares its attribute space and subtypes with Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM) [RFC4186], EAP-AKA [RFC4187], and EAP-AKA' [RFC9048].

Two new values (TBA1, TBA2) in the skippable range need to be assigned for AT_PUB_ECDHE (Section 6.1) and AT_KDF_FS (Section 6.2) in the "Attribute Types" registry under the "EAP-AKA and EAP-SIM Parameters" group.

Also, IANA is requested to create a new registry "EAP-AKA' AT_KDF_FS Key Derivation Function Values" to represent FS Key Derivation Function types. The "EAP-AKA' with ECDHE and X25519" and "EAP-AKA' with ECDHE and P-256" types (1 and 2, see Section 6.3) need to be assigned, along with one reserved value. The initial contents of this registry is illustrated in Table 1; new values can be created through the Specification Required policy [RFC8126]. Expert reviewers should ensure that the referenced specification is clearly identified and stable, and that the proposed addition is reasonable for the given category of allocation.

Value	Description	Reference
0	Reserved	[TBD BY IANA: THIS RFC]
1	EAP-AKA' with ECDHE and X25519	[TBD BY IANA: THIS RFC]
2	EAP-AKA' with ECDHE and P-256	[TBD BY IANA: THIS RFC]
3-65535	Unassigned	[TBD BY IANA: THIS RFC]

Table 1: Initial Content of the EAP-AKA' AT_KDF_FS Key Derivation Function Values Registry

9. References

9.1. 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/info/rfc2119>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/info/rfc3748>.

[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/info/rfc4187>.

[RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, DOI 10.17487/RFC5448, May 2009, <https://www.rfc-editor.org/info/rfc5448>.

[RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <https://www.rfc-editor.org/info/rfc7624>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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/info/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/info/rfc9048>.

[SP-800-186] NIST, "Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters", NIST

Special Publication 800-186, February 2023, <https://doi.org/10.6028/NIST.SP.800-186>.

[SEC1] Certicom Research, "SEC 1: Elliptic Curve Cryptography", Standards for Efficient Cryptography 1 (SEC 1) Version 2.0, May 2009, <https://www.secg.org/sec1-v2.pdf>.

[SEC2] Certicom Research, "SEC 2: Recommended Elliptic Curve Domain Parameters", Standards for Efficient Cryptography 2 (SEC 2) Version 2.0, January 2010, <https://www.secg.org/sec2-v2.pdf>.

[SP-800-56A] Barker, E., Chen, L., Roginsky, A., Vassilev, A., and R. Davis, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography", NIST Special Publication 800-56A Revision 3, April 2018, <https://doi.org/10.6028/NIST.SP.800-56Ar3>.

9.2. Informative References

[RFC4186] Haverinen, H., Ed. and J. Salowey, Ed., "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)", RFC 4186, DOI 10.17487/RFC4186, January 2006, <https://www.rfc-editor.org/info/rfc4186>.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <https://www.rfc-editor.org/info/rfc5216>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC9190] Preuß Mattsson, J. and M. Sethi, "EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3", RFC 9190, DOI 10.17487/RFC9190, February 2022, <https://www.rfc-editor.org/info/rfc9190>.

[TrustCom2015] Arkko, J., Norrman, K., Näslund, M., and B. Sahlin, "A USIM compatible 5G AKA protocol with perfect forward secrecy", Proceedings of IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) 2015, August 2015, <https://doi.org/10.1109/Trustcom.2015.506>.

[Heist2015]
Scahill, J. and J. Begley, "The Great SIM Heist", February 2015, <https://theintercept.com/2015/02/19/great-sim-heist/>.

[DOW1992] Diffie, W., Van Oorschot, P., and M. Wiener, "Authentication and Authenticated Key Exchanges", Designs, Codes and Cryptography 2 pp. 107-125, June 1992, <https://doi.org/10.1007/BF00124891>.

[TS.33.501] 3GPP, "Security architecture and procedures for 5G System", 3GPP TS 33.501 18.1.0, March 2023.

[NIST-ZT] National Institute of Standards and Technology, "Implementing a Zero Trust Architecture", December 2022, <https://www.nccoe.nist.gov/sites/default/files/2022-12/zta-nist-sp-1800-35b-preliminary-draft-2.pdf>.

[NSA-ZT] National Security Agency, "Embracing a Zero Trust Security Model", February 2021, <https://media.defense.gov/2021/Feb/25/2002588479/-1/-1/0/CSI_EMBRACING_ZT_SECURITY_MODEL_UOO115131-21.PDF>.

Appendix A. Change Log

RFC Editor: Please remove this appendix.

The -12 version of the WG draft has the following changes, most due to IESG review comments in January 2023:

* Update the draft track to Standards Track.

* Clarified the calculation of the Length field in the AT_ECDHE attribute, along with padding requirements.

* Avoided the use of keywords in operational recommendations, e.g., about deployment.

* Changed the definition of what "supported" means to focus on feature being implemented, but not require that it is usable during a protocol run, because configuration, new security information, etc. might imply that a particular feature is implemented but disabled for policy reasons.

* Changed the MITM terminology to be on-path attacks.

* Corrected a reference typo in the IANA considerations section.

* Shortened the abstract and introduction to the key aspects and removed duplication.

* Several editorial changes.

The -11 version of the WG draft has the following changes:

* Addressed IETF Last Call comments from directorates, Security AD, Meiling Cheng, and a detailed review from the author Karl. In particular:

* Replaced the reference to the deprecated FIPS 186-4 with SP 800-186.

* Changed HSS (Home Subscriber Server) to Authentication Database (AD) as HSS is a 4G term.

* Explained difference between EAP-AKA and EAP-AKA'

* Explained that the emphemeral key exhange provide more that forward secrecy and how this is important to mitigate pervasive monitoring.

* Included links for the zero trust principles.

* Explained why K_encr and K_auth not being protected by the ECDHE addition.

* Added that a future introduction of KEM might require additional changes.

* Explained how ephemeral key exchange is linked to pervasive monitoring.

* Changed SIM to USIM everywhere. A USIM is required for AKA.

* Changed to long-term key instead of long-term secret or long-term shared secret.

* Reference updates.

* Various editorial improvements.

The -10 version of the WG draft has the following changes:

* Various nits found by Peter Yee.

The -09 version of the WG draft has the following changes:

* Scalable Vector Graphics (SVG) versions for all figures has been added and the figures has been slightly modified to render nicely with aasvg.

* A reference has been added to the Section in SEC1 describing how to do decompression.

* The strengthened identity protection requirements are now mentioned in the introduction.

* Corrections and clarifications were made in the IANA considerations. The table in the IANA section has been made into a proper xml table.

* Reference updates.

* Various editorial improvements.

The -08 version of the WG draft has the following changes:

* Further clarification of key calculation in Section 6.3.

* Support for the NIST P-256 group has been made mandatory in Section 6.4, in order to align the requirements with 3GPP SUCI encryption requirements.

* The interaction between AT_KDF and AT_KDF_FS has been specified more clearly, including specifying how future specifications need to specify the treatment of new combinations.

* Addition of a discussion about the impacts of potential future quantum computing attacks with specific impacts to this extension.

* Addition of a discussion about metadata/unprotected data in Section 7.5.

* Reference updates.

* Various editorial improvements.

The -07 version of the WG draft has the following changes:

* The impact of forward secrecy explanation has been improved in the abstract and security considerations.

* The draft now more forcefully explains why the authors believe it is important to migrate existing systems to use forward secrecy, and makes a recommendation for this migration.

* The draft does no longer refer to issues within the smart cards but rather the smart card supply chain.

* The rationale for chosen algorithms is explained.

* Also, the authors have checked the language relating to the public value encoding, and believe it is exactly according to the references ([RFC7748] Section 6.1 and [SEC2] Section 2.7.1)

The -06 version of the WG draft is a refresh and a reference update. However, the following should be noted:

* The draft now uses "forward secrecy" terminology and references RFC 7624 per recommendations on mailing list discussion.

* There's been mailing list discussion about the encoding of the public values; the current text requires confirmation from the working group that it is sufficient.

The -05 version of the WG draft takes into account feedback from the working group list, about the number of bytes needed to encode P-256 values.

The -04 version of the WG draft takes into account feedback from the May 2020 WG interim meeting, correcting the reference to the NIST P-256 specification.

The -03 version of the WG draft is first of all a refresh; there are no issues that we think need addressing, beyond the one for which there is a suggestion in -03: The document now suggests an alternate group/curve as an optional one besides X25519. The specific choice of particular groups and algorithms is still up to the working group.

The -02 version of the WG draft took into account additional reviews, and changed the document to update RFC 5448 (or rather, its successor, [RFC9048]), changed the wording of the recommendation with regards to the use of this extension, clarified the references to the definition of X25519 and Curve25519, clarified the distinction to ECDH methods that use partially static keys, and simplified the use of AKA and USIM card terminology. Some editorial changes were also made.

The -00 and -01 versions of the WG draft made no major changes, only updates to some references.

The -05 version is merely a refresh while the draft was waiting for WG adoption.

The -04 version of this draft made only editorial changes.

The -03 version of this draft changed the naming of various protocol components, values, and notation to match with the use of ECDH in ephemeral mode. The AT_KDF_FS negotiation process was clarified in that exactly one key is ever sent in AT_KDF_ECDHE. The option of checking for zero key values IN ECDHE was added. The format of the actual key in AT_PUB_ECDHE was specified. Denial-of-service considerations for the FS process have been updated. Bidding down attacks against this extension itself are discussed extensively. This version also addressed comments from reviewers, including the August review from Mohit Sethi, and comments made during IETF-102 discussion.

Acknowledgments

The authors would like to note that the technical solution in this document came out of the TrustCom paper [TrustCom2015], whose authors were J. Arkko, K. Norrman, M. Näslund, and B. Sahlin. This document uses also a lot of material from [RFC4187] by J. Arkko and H. Haverinen as well as [RFC5448] by J. Arkko, V. Lehtovirta, and P. Eronen.

The authors would also like to thank Ben Campbell, Meiling Chen, Roman Danyliw, Linda Dunbar, Tim Evans, Zhang Fu, Russ Housley, Tero Kivinen, Murray Kucherawy, Warren Kumari, Eliot Lear, Vesa Lehtovirta, Kathleen Moriarty, Prajwol Kumar Nakarmi, Francesca Palombini, Anand R. Prasad, Michael Richardson, Göran Rune, Bengt Sahlin, Joseph Salowey, Mohit Sethi, Orie Steele, Rene Struik, Vesa Torvinen, Sean Turner, Helena Vahidi Mazinani, Robert Wilton, Paul Wouters, Bo Wu, Peter Yee, and many other people at the IETF, GSMA and 3GPP groups for interesting discussions in this problem space.

Authors' Addresses

Jari Arkko
Ericsson
FI-02420 Jorvas
Finland

Email: jari.arkko@piuha.net

Karl Norrman
Ericsson
SE-16483 Stockholm
Sweden

Email: karl.norrman@ericsson.com

John Preuß Mattsson
Ericsson
SE-164 40 Kista
Sweden

Email: john.mattsson@ericsson.com