Workgroup: Network Working Group
Internet-Draft: draft-ietf-emu-aka-pfs-10
Updates: 5448, 9048 (if approved)
Published: 26 January 2023
Intended Status: Informational
Expires: 30 July 2023
Authors: J. Arkko Ericsson
K. Norrman Ericsson
V. Torvinen Ericsson
J. Preuß Mattsson Ericsson

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

Abstract

Many different attacks have been reported as part of revelations associated with pervasive surveillance. Some of the reported attacks involved compromising the smart card supply chain, such as attacking SIM card manufacturers and operators in an effort to compromise shared secrets stored on these cards. Since the publication of those reports, manufacturing and provisioning processes have gained much scrutiny and have improved. However, the danger of resourceful attackers for these systems is still a concern. Always assuming breach such as key compromise and minimizing the impact of breach are essential zero-trust principles.

This specification updates RFC 9048, the improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA'), with an optional extension. Similarly, this specification also updates the earlier version of the EAP-AKA' specification in RFC 5448. The extension, when negotiated, provides Forward Secrecy 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 pre-shared secret in a Subscriber Identity Module (SIM) card from being able to decrypt any past communications. In addition, if the attacker stays merely a passive eavesdropper, the extension prevents attacks against future sessions. This forces attackers to use active attacks instead.

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/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 July 2023.

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 smart card supply chain, such as attacking SIM card manufacturers and operators in an effort to compromise shared secrets stored on these cards. Such attacks are conceivable, for instance, during the manufacturing process of cards, or during the transfer of cards and associated information to the operator. Since the publication of reports about such attacks, manufacturing and provisioning processes have gained much scrutiny and have improved.

However, the danger of resourceful attackers attempting to gain information about Subscriber Identity Module (SIM) cards is still a concern. They are a high-value target and concern a large number of people. 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 unlawful access to key material. While the better protection of manufacturing and other processes is essential in protecting against this, there is one question that we as protocol designers can ask. Is there something that we can do to limit the consequences of attacks, should they occur?

The authors want to provide a public specification of an extension that helps defend against one aspect of pervasive surveillance. This is important, given the large number of users such practices may affect. It is also a stated goal of the IETF to ensure that we understand the surveillance concerns related to IETF protocols and take appropriate countermeasures [RFC7258]. This document does that

for the improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA').

This specification updates [RFC9048], the EAP-AKA' authentication method, with an optional extension and strengthens the identity privacy requirements. While optional, the use of this extension is strongly RECOMMENDED.

The extension, when negotiated, provides Forward Secrecy (FS) 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 pre-shared secret in a SIM card from being able to decrypt any past communications. In addition, if the attacker stays merely a passive eavesdropper, the extension prevents attacks against future sessions. This forces attackers to use active attacks instead. This is beneficial, because active attacks demand much more resources to launch, and can generally be detected much easier. 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. The attacker is forced to attempt to exfiltrate key material, if it can, on a continuous basis, as opposed to learning it once [RFC7624].

Attacks against Authentication and Key Agreement (AKA) authentication via compromising the long-term secrets in the SIM cards have been an active discussion topic in many contexts. Forward secrecy is on the list of features for the next release of 3GPP (5G Phase 2), and this document provides a basis for providing this feature in a particular fashion.

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 large number of users.

2. 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 maintains the interface between the Universal Subscriber Identification Module (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 are by no means a panacea, 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, the authors believe that the approach chosen here is relatively easily deployable. In particular:

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

3. Background

3.1. AKA

Authentication and Key Agreement (AKA) is based on challenge-response mechanisms and symmetric cryptography. In contrast with its earlier GSM counterparts, AKA provides long key lengths and mutual authentication. AKA typically runs in a Universal Subscriber Identity Module (USIM). USIM is technically just an application that can reside on a removable UICC, an embedded UICC, or integrated in a Trusted Execution Environment (TEE). In this document we use the

term "SIM card" to refer to any Subscriber Identity Module capable of running AKA.

AKA works in the following manner:

* The identity module and the home environment have agreed on a secret key beforehand.

* The actual authentication process starts by having the home environment produce an authentication vector, based on the secret key and a sequence number. The authentication vector contains a random part RAND, an authenticator part AUTN used for authenticating the network to the identity module, 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 identity module.

* The identity module verifies the AUTN, again based on the secret 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 identity module produces an authentication result RES and sends it to the serving network.

* The serving network verifies the correct result from the identity module. If the result is correct, IK and CK can be used to protect further communications between the identity module and the home environment.

3.2. EAP-AKA' Protocol

When AKA is embedded into EAP, the authentication on the network side is moved to the home environment; the serving network performs the role of a pass-through authenticator. 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].

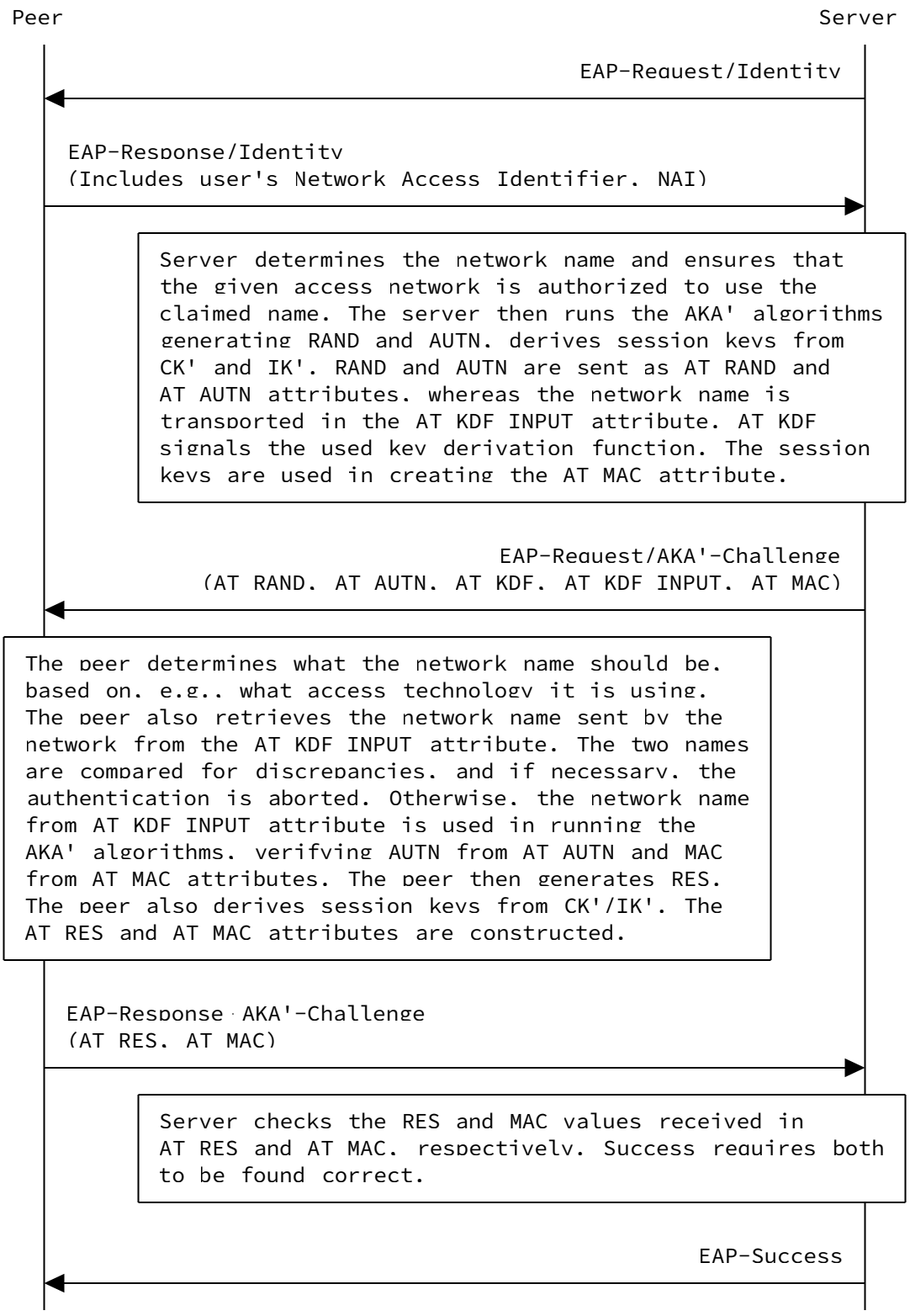


Figure 1: EAP-AKA' Authentication Process

3.3. Attacks Against Long-Term Shared Secrets in Smart Cards

Current 3GPP systems use SIM pre-shared key-based protocols and Authentication and Key Agreement (AKA) to authenticate subscribers. The general security properties and potential vulnerabilities of AKA and EAP-AKA' are discussed in [RFC9048].

An important vulnerability in that discussion relates to the recent reports of compromised long term pre-shared keys used in AKA [Heist2015]. These attacks are not specific to AKA or EAP-AKA', as all security systems fail at least to some extent if key material is stolen. However, the reports indicate a need to look into solutions that can operate at least to an extent under these types of attacks. It is noted in [Heist2015] that some security can be retained even in the face of the attacks by providing Forward Secrecy (FS) [DOW1992] for the session key. If AKA would have provided FS, compromising the pre-shared key would not be sufficient to perform passive attacks; the attacker is, in addition, forced to be a Man-In-The-Middle (MITM) during the AKA run and subsequent communication between the parties.

4. 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.

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 format is 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 can 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 our goal has been to not require new infrastructure or credentials, the flow diagrams also show the conceptual interaction with the USIM card and the 3GPP authentication server (HSS). The details of those interactions are outside the scope of this document, however, and the reader is referred to the 3GPP specifications.

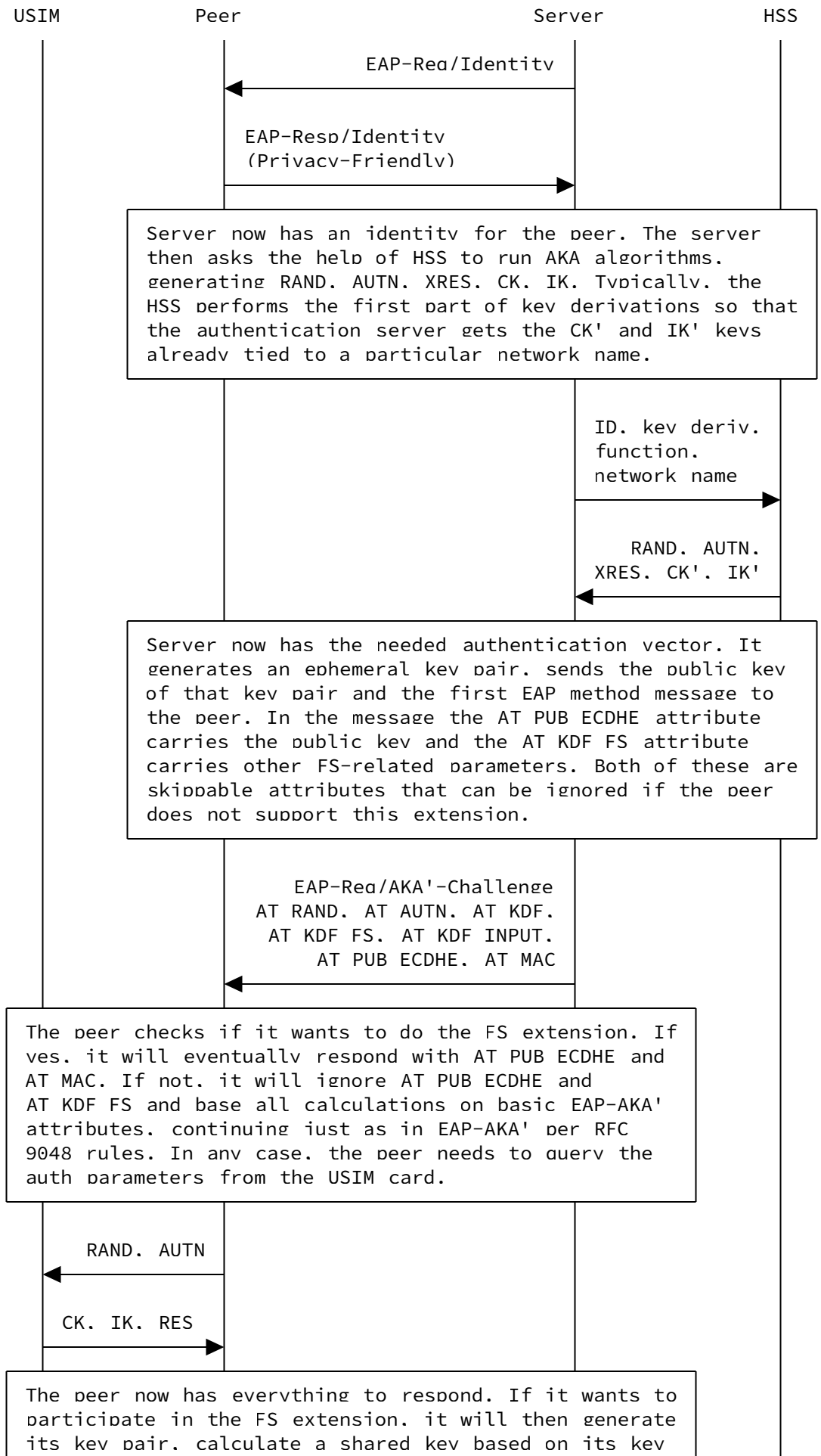

USIM
Peer
Server
HSS
EAP-Req/Identity
EAP-Resp/Identity
(Privacy-Friendly)
Server now has an identity for the peer. The server then asks the help of HSS to run AKA algorithms, generating RAND, AUTN, XRES, CK, IK. Typically, the HSS performs the first part of key derivations so that the authentication 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

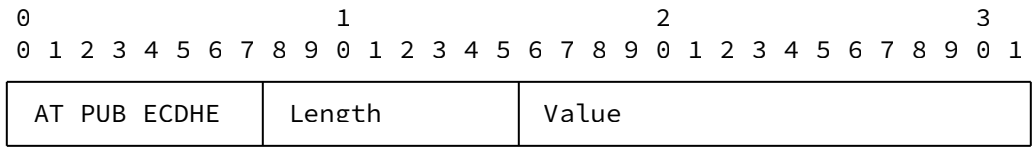
Figure 2: EAP-AKA' FS Authentication Process

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.



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].

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

- For X25519/Curve25519, 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].

To retain the security of the keys, the sender SHALL generate a fresh value for each run of the protocol.

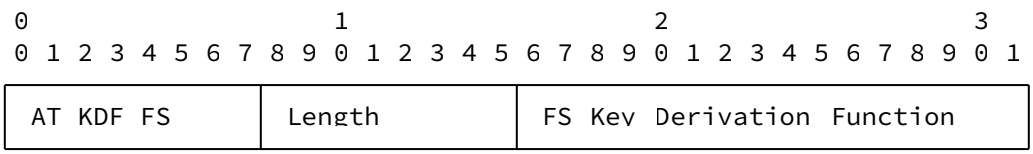
6.2. AT_KDF_FS

The AT_KDF_FS indicates the used or desired forward secrecy key generation function, if the Forward Secrecy extension is taken into use. It will also at the same time 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 specification 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.



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

specification, 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 specification 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 [RFC9048] EAP-AKA' 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) is derived as follows below.

```
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. 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 public key or the shared secret fails, both parties 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 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 Appendix D.1.2.3 of [FIPS186-4] or alternatively Section 2.4.2 of [SEC2]. The support for this group is REQUIRED.

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 and that a privacy-friendly identifier MUST be used. The appearance of these attributes in a received message MUST be ignored.

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 them as if neither one was sent, and the assume that the extension defined in this specification 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, the operation in case an error occurs is specified in 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. Only if these attribute is verified to be valid, the Server derives keys and verifies AT_MAC.

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].

The possibility of attacks against key storage offered in SIM or other smart cards has been a known threat. But as the discussion in Section 3.3 shows, the likelihood of practically feasible attacks such as breaches in the smart card supply chain has increased. Many of these attacks can be best dealt with improved processes, e.g., limiting the access to the key material within the factory or personnel, etc. But not all attacks can be entirely ruled out for well-resourced adversaries, irrespective of what the technical algorithms and protection measures are. Always assuming breach such as key compromise and minimizing the impact of breach are essential zero-trust principles.

If a mechanism without forward secrecy such as (5G-AKA, EAP-AKA') is used the effects of key compromise are devastating. The serious consequences of breach somewhere in the supply chain or after

delivery that are possible when 5G-AKA or EAP-AKA' is used but not when something with forward secrecy like EAP-AKA-FS is used are:

1. A passive attacker can eavesdrop (decrypt) all future 5G communication (control and user plane both directions),

2. A passive attacker can decrypt 5G communication that they previously recorded in the past (control and user plane both directions), and

3. An active attacker can impersonate UE and Network and inject messages in an ongoing 5G connection between the real UE and the real network (control and user plane both directions).

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 to long term completely phase out AKA without forward secrecy.

This extension can provide assistance in situations where there is a danger of attacks against the key material on SIM cards by adversaries that cannot or who are unwilling to mount active attacks against a large number of sessions. The extension also provides protection against active attacks as they are forced to be a Man-In-The-Middle (MITM) 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 a man-in-the-middle. This extension is most useful when used in a context where EAP keys are used without further mixing that can provide Forward Secrecy. For instance, when used with IKEv2 [RFC7296], the session keys produced by IKEv2 have this property, so better characteristics of EAP keys is not that useful. However, typical link layer usage of EAP does not involve running Diffie-Hellman, so 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 forgotten, 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. Of course, as with other protocols, if the attacker has learned the keys and does become an active attacker, there is no protection that that can be provided for future sessions. Among other things, such an active attacker can impersonate any legitimate endpoint in EAP-AKA', become a MITM in EAP-AKA' or the extension defined in this document, retrieve all keys, or turn off FS. Still, past sessions where FS was in use remain protected.

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

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. 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 shared secret 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 MITM 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 a MITM 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 key material that is based on the Diffie-Hellman keys, yet bound to the authentication through the SIM card. This means that subsequent payload communications between the parties are protected with keys that are not solely based on information in the clear (such as the RAND) and information derivable from the long-term shared secrets on the SIM card. As a result, if anyone successfully recovers shared secret information, they are unable to decrypt communications protected by the keys generated through this extension. Note that the

recovery of shared secret information could occur either before or after the time that the protected communications are used. When this extension is used, communications at time t0 can be protected if at some later time t1 an adversary learns of long-term shared secret and has access to a recording of the encrypted communications.

Obviously, this extension is still vulnerable to attackers that are willing to perform an active attack and who at the time of the attack have access to the long-term shared secret.

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 secrets as the original authentication.

7.2. 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 HSS 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 SIM 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 secret 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.

7.3. Identity Privacy

Best practice privacy today is to mandate client identity protection as is done in EAP-TLS 1.3, EAP-TTLS 1.3, etc. A client supporting EAP-AKA' FS MUST NOT send its username (or any other permanent identifiers) in cleartext in the Identity Response (or any message used instead of the Identity Response).

7.4. 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.

7.5. Post-Quantum Considerations

As of the publication of this specification, 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 secrets associated with the SIM card, 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 specification to PQC algorithms. This specification 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.

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 [RFC4186], 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, a new registry "EAP-AKA' AT_KDF_FS Key Derivation Function Values" should be created 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].

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>.

[FIPS186-4] NIST, "Digital Signature Standard (DSS)", FIPS 186-4, July 2013, <https://doi.org/10.6028/NIST.FIPS.186-4>.

[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 18.0.0, December 2022.

Appendix A. Change Log

RFC Editor: Please remove this appendix.

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 7.4.

* 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 specification 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 SIM 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, Tim Evans, Zhang Fu, Russ Housley, Tero Kivinen, Eliot Lear, Vesa Lehtovirta, Kathleen Moriarty, Prajwol Kumar Nakarmi, Anand R. Prasad, Michael Richardson, Göran Rune, Bengt Sahlin, Joseph Salowey, Mohit Sethi, Rene Struik, Sean Turner, Helena Vahidi Mazinani, 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

Vesa Torvinen
Ericsson
FI-02420 Jorvas
Finland

Email: vesa.torvinen@ericsson.com

John Preuß Mattsson
Ericsson

SE-164 40 Kista
Sweden

Email: john.mattsson@ericsson.com