Network Working Group
Internet-Draft
Updates: RFC5448 (if approved)
Intended status: Informational
Expires: May 3, 2021

J. Arkko
K. Norrman
V. Torvinen
Ericsson

October 30, 2020

Perfect-Forward Secrecy for the Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' PFS) draft-ietf-emu-aka-pfs-05

Abstract

Many different attacks have been reported as part of revelations associated with pervasive surveillance. Some of the reported attacks involved compromising smart cards, 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.

This specification is an optional extension to the EAP-AKA' authentication method which was defined in [I-D.ietf-emu-rfc5448bis]. The extension, when negotiated, provides Perfect 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 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 http://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 May 3, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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 smart cards, 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 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 EAP-AKA'.

This specification is an optional extension to the EAP-AKA' authentication method [I-D.ietf-emu-rfc5448bis]. While optional, the use of this extension is RECOMMENDED.

The extension, when negotiated, provides Perfect 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 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.

Attacks against AKA authentication via compromising the long-term secrets in the SIM cards have been an active discussion topic in many contexts. Perfect 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 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

This 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 PFS 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 perfect 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.

- PFS 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 perfect forward secrecy being available for both EAP-TLS [RFC5216] [I-D.ietf-emu-eap-tls13] and EAP-AKA' ensures that for many practical systems perfect forward secrecy can be enabled for either all or significant fraction of users.

3. Background

3.1. AKA

AKA is based on challenge-response mechanisms and symmetric cryptography. AKA typically runs in a UMTS Subscriber Identity Module (USIM) or a CDMA2000 (Removable) User Identity Module ((R)UIM). In contrast with its earlier GSM counterparts, AKA provides long key lengths and mutual authentication.

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 are 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 behaviour, along with the definition of attributes AT_RAND, AT_AUTN, AT_MAC, and AT_RES can be found in [I-D.ietf-emu-rfc5448bis] and [RFC4187].

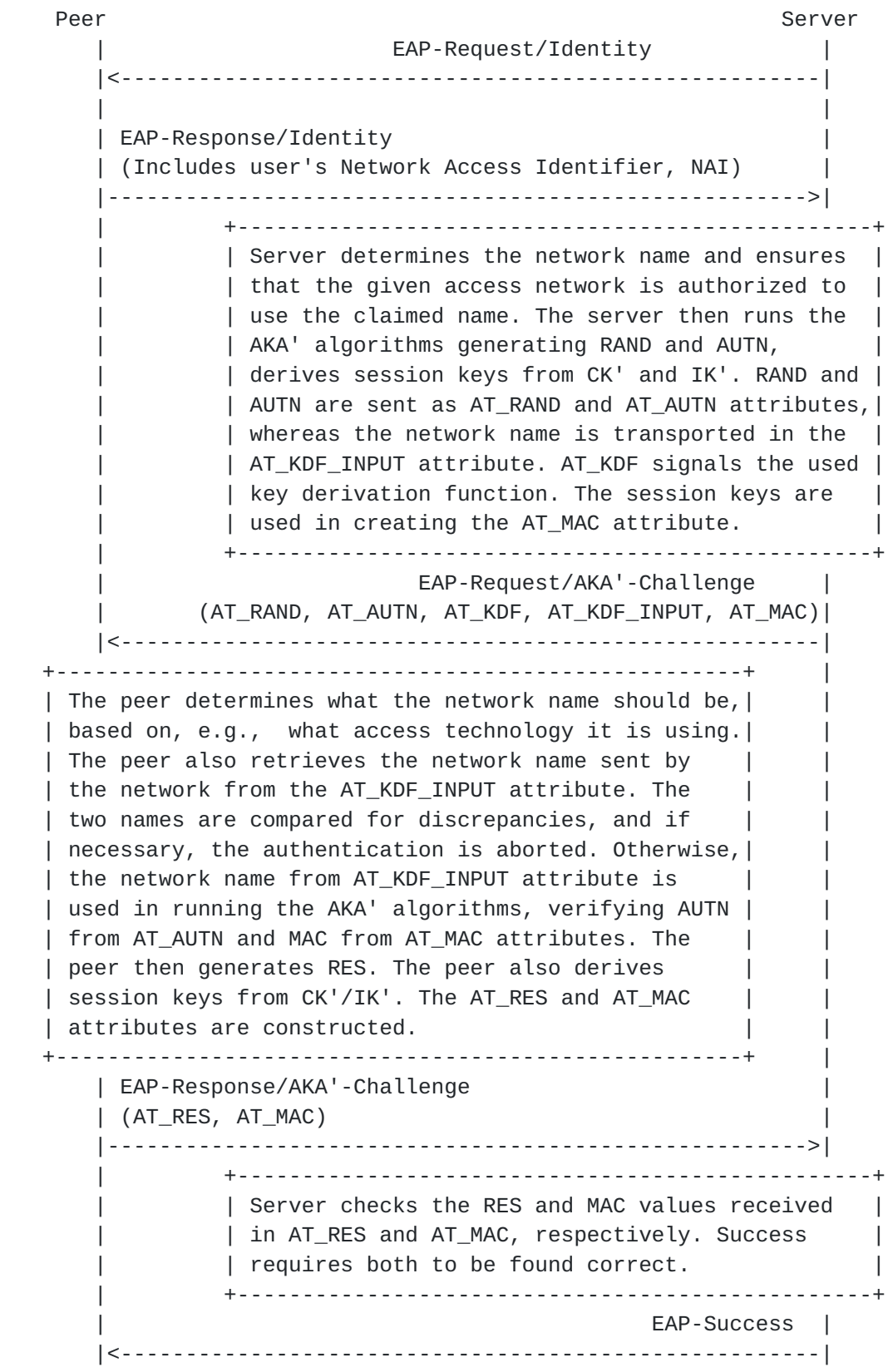


Figure 1: EAP-AKA' Authentication Process

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 [I-D.ietf-emu-rfc5448bis].

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 Perfect Forward Secrecy (PFS) [DOW1992] for the session key. If AKA would have provided PFS, 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.

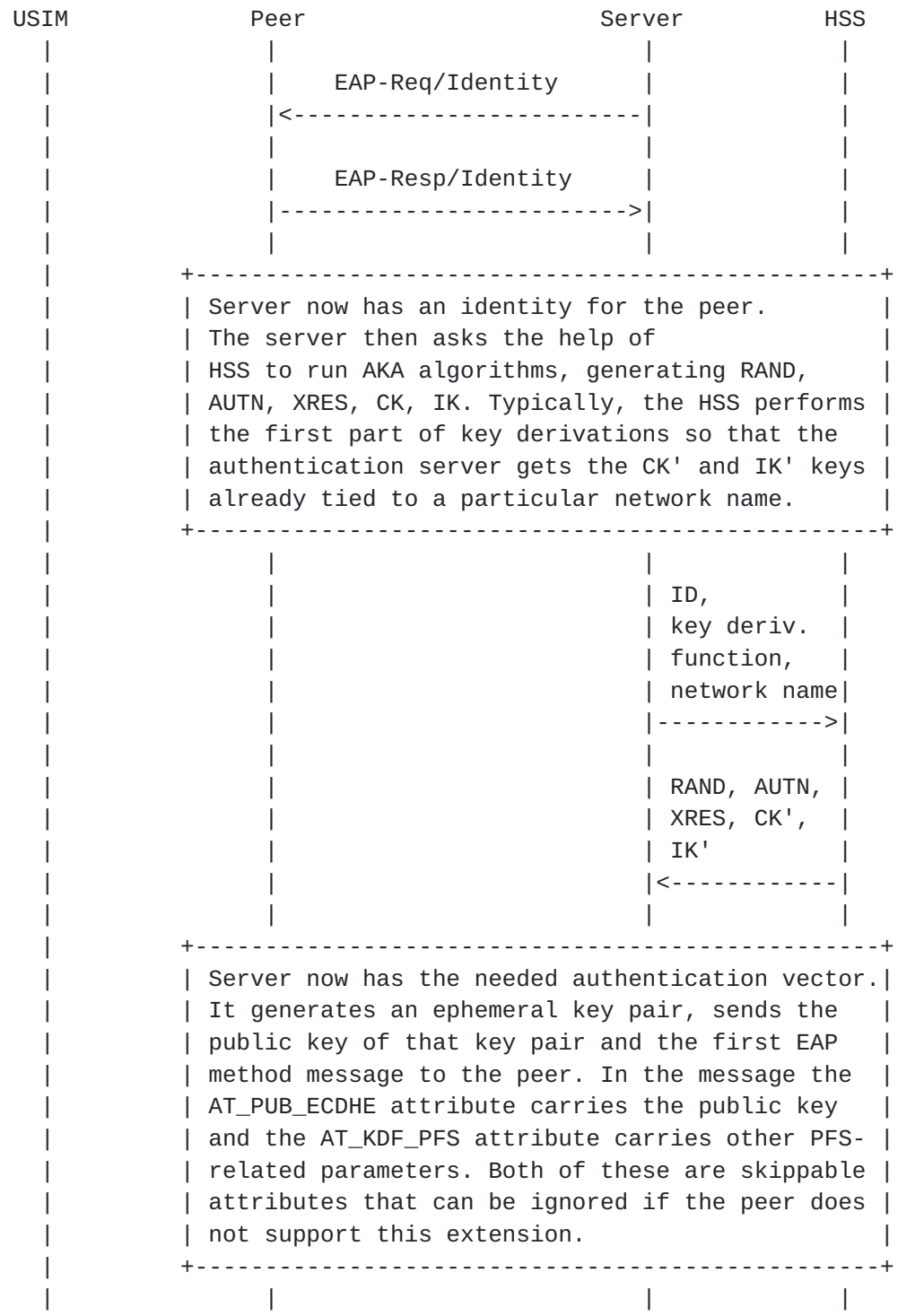
5. Protocol Overview

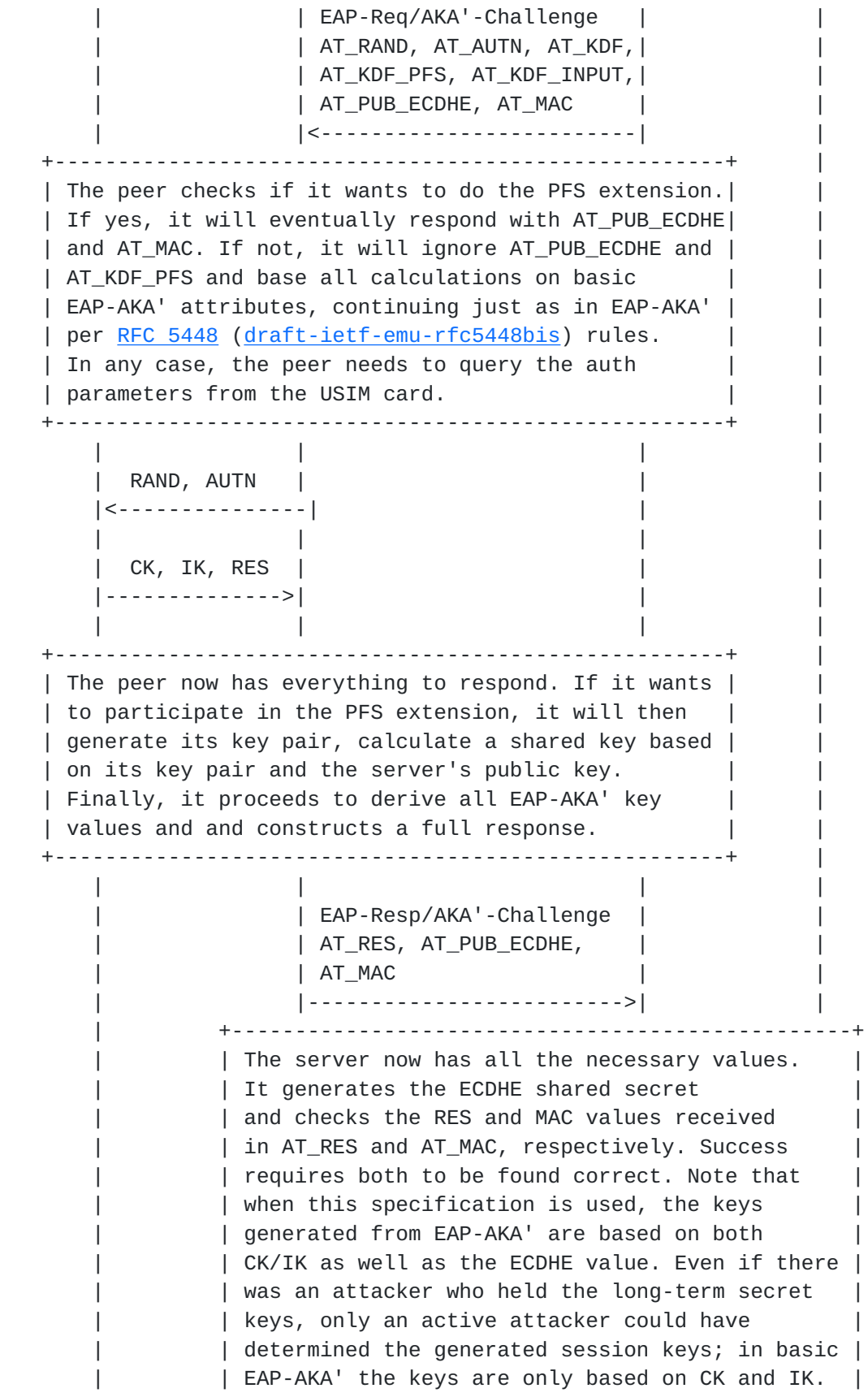
Introducing PFS for EAP-AKA' can be achieved by using an Elliptic Curve Diffie-Hellman (ECDH) exchange [RFC7748]. In EAP-AKA' PFS this exchange is run in an ephemeral manner, i.e., both sides generate temporary keys as specified in [RFC7748]. This method is referred to as ECDHE, where the last 'E' stands for Ephemeral.

The enhancements in the EAP-AKA' PFS 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 .




EAP-Req/AKA'-Challenge
AT_RAND, AT_AUTN, AT_KDF,
AT_KDF_PFS, AT_KDF_INPUT,
AT_PUB_ECDHE, AT_MAC
The peer checks if it wants to do the PFS 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_PFS and base all calculations on basic
EAP-AKA' attributes, continuing just as in EAP-AKA'
per RFC 5448 (draft-ietf-emu-rfc5448bis) 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 PFS extension, it will then
generate its key pair, calculate a shared key based
on its key pair and the server's public key.
Finally, it proceeds to derive all EAP-AKA' key
values and and constructs a full response.
EAP-Resp/AKA'-Challenge
AT_RES, AT_PUB_ECDHE,
AT_MAC
The server now has all the necessary values.
It generates the ECDHE shared secret
and checks the RES and MAC values received
in AT_RES and AT_MAC, respectively. Success
requires both to be found correct. Note that
when this specification is used, the keys
generated from EAP-AKA' are based on both
CK/IK as well as the ECDHE value. Even if there
was an attacker who held the long-term secret
keys, only an active attacker could have
determined the generated session keys; in basic
EAP-AKA' the keys are only based on CK and IK.

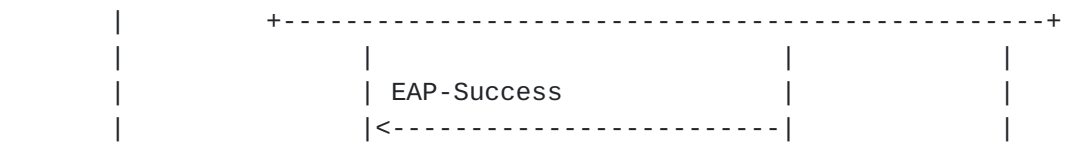


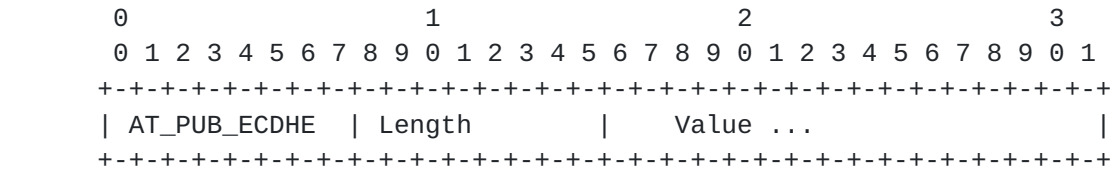
Figure 2: EAP-AKA' PFS Authentication Process

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 value. It is calculated as follows:

* For X25519/Curve25519, the length of this value is 32 bytes, encoded in binary as specified [RFC7748] Section 6.1.

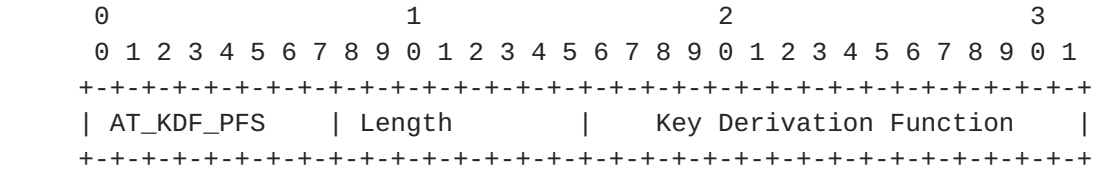
* For P-256, the length of this value is 33 bytes, encoded in binary as specified in [FIPS186-4], using the compressed form from Section 2.7.1 of [SEC2].

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

6.2. AT_KDF_PFS

The AT_KDF_PFS indicates the used or desired key generation function, if the Perfect 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 legacy KDF value for those EAP peers that cannot or do not want to use this extension.

The format of the AT_KDF_PFS attribute is shown below.



The fields are as follows:

AT_KDF_PFS

This is set to TBA2 BY IANA.

Length

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

Key Derivation Function

An enumerated value representing the 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 [I-D.ietf-emu-rfc5448bis].

Servers MUST send one or more AT_KDF_PFS 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_PFS 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 key derivation function indicated by the first AT_KDF_PFS 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 [I-D.ietf-emu-rfc5448bis] for AT_KDF, the peer sends EAP-Response/AKA'-Challenge message that contains only one attribute, AT_KDF_PFS 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_PFS 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_PFS attribute.

Upon receiving an EAP-Response/AKA'-Challenge with AT_KDF_PFS from the peer, the server checks that the suggested AT_KDF_PFS value was one of the alternatives in its offer. The first AT_KDF_PFS value in the message from the server is not a valid alternative. If the peer has replied with the first AT_KDF_PFS 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_PFS 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 key derivation function is the only legal 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_PFS 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_PFS attributes without

having requested negotiation, the peer MUST behave as if AT_MAC had been incorrect and fail the authentication.

6.3. New Key Derivation Functions

Two new 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 Key Derivation Function type value is only used in the AT_KDF_PFS attribute, and should not be confused with the different range of key derivation functions that can be represented in the AT_KDF attribute as defined in [I-D.ietf-emu-rfc5448bis].

Key derivation in this extension produces exactly the same keys for internal use within one authentication run as [I-D.ietf-emu-rfc5448bis] 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 Key Derivation Function field in the AT_KDF_PFS attribute is set to 1 and the Key Derivation Function field in the AT_KDF attribute is also 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' PFS"|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]
```

Where SHARED_SECRET is the shared secret computed via ECDHE, 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]. If such checking is performed and the SHARED_SECRET has a zero value, both parties MUST behave as if the current EAP-AKA' authentication process starts again from the beginning.

> 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.

The rest of computation proceeds as defined in Section 3.3 of [I-D.ietf-emu-rfc5448bis].

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 [I-D.ietf-emu-rfc5448bis]. 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 [I-D.ietf-emu-rfc5448bis] 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' PFS" is a twelve-characters-long ASCII string, also used as is.

Identity is the peer identity as specified in Section 7 of [RFC4187].

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_PFS.

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 [FIPS186-4]. The support for this group is OPTIONAL.

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 [I-D.ietf-emu-rfc5448bis] or [RFC4187] apply.

6.5.1. EAP-Request/AKA'-Identity

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

6.5.2. EAP-Response/AKA'-Identity

No changes, except that the AT_KDF_PFS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these messages 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 [I-D.ietf-emu-rfc5448bis]. 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_PFS 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_PFS 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_PFS, 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 [I-D.ietf-emu-rfc5448bis]. 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 [I-D.ietf-emu-rfc5448bis]. 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 [I-D.ietf-emu-rfc5448bis] 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 [I-D.ietf-emu-rfc5448bis]. 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 documents 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_PFS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these messages in a received message MUST be ignored.

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

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

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

No changes, except that the AT_KDF_PFS or AT_PUB_ECDHE attributes MUST NOT be added to this message. The appearance of these messages 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 [I-D.ietf-emu-rfc5448bis].

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 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.

This extension can provide assistance in situations where there is a danger of attacks against the key material on SIM cards by adversaries that can not or who are unwilling to mount active attacks against large number of sessions. This extension is most useful when used in a context where EAP keys are used without further mixing that can provide Perfect 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 PFS 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, PFS 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 PFS. Still, past sessions where PFS was in use remain protected.

Achieving PFS 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.

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_PFS 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 PFS 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 laborous.

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 peform 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 PFS can obviously 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.

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' PFS.

- 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 can not 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 PFS 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 PFS 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.

8. IANA Considerations

This extension of EAP-AKA' shares its attribute space and subtypes with EAP-SIM [RFC4186], EAP-AKA [RFC4186], and EAP-AKA' [I-D.ietf-emu-rfc5448bis].

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

Also, a new registry should be created to represent Diffie-Hellman 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 namespace are therefore as below; 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]

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>.

[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>.

[I-D.ietf-emu-rfc5448bis] Arkko, J., Lehtovirta, V., Torvinen, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA')", draft-ietf-emu-rfc5448bis-07 (work in progress), March 2020.

[FIPS186-4] NIST, , "Digital Signature Standard (DSS)", July 2013.

[SEC2] Certicom Research, , "SEC 2: Recommended Elliptic Curve Domain Parameters", September 2000.

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>.

[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>.

[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>.

[I-D.ietf-emu-eap-tls13] Mattsson, J. and M. Sethi, "Using EAP-TLS with TLS 1.3", draft-ietf-emu-eap-tls13-11 (work in progress), October 2020.

[TrustCom2015] Arkko, J., Norrman, K., Naslund, M., and B. Sahlin, "A USIM compatible 5G AKA protocol with perfect forward secrecy", August 2015 in Proceedings of the TrustCom 2015, IEEE.

[Heist2015] Scahill, J. and J. Begley, "The great SIM heist", February 2015, in https://firstlook.org/theintercept/2015/02/19/great-sim-heist/ .

[DOW1992] Diffie, W., vanOorschot, P., and M. Wiener, "Authentication and Authenticated Key Exchanges", June 1992, in Designs, Codes and Cryptography 2 (2): pp. 107-125.

Appendix A. Change Log

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, [I-D.ietf-emu-rfc5448bis]), 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_PFS 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 PFS 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.

Appendix B. 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. Naslund, 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 Tero Kivinen, John Mattsson, Mohit Sethi, Vesa Lehtovirta, Russ Housley, Sean Turner, Eliot Lear, Joseph Salowey, Kathleen Moriarty, Zhang Fu, Bengt Sahlin, Ben Campbell, Prajwol Kumar Nakarmi, Goran Rune, Tim Evans, Helena Vahidi Mazinani, Anand R. Prasad, Rene Struik, and many other people at the IETF, GSMA and 3GPP groups for interesting discussions in this problem space.

Authors' Addresses

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Karl Norrman
Ericsson
Stockholm 16483
Sweden

Email: karl.norrman@ericsson.com

Vesa Torvinen
Ericsson
Jorvas 02420
Finland

Email: vesa.torvinen@ericsson.com