Network Working Group
Internet-Draft
Updates: 5448 (if approved)
Intended status: Informational
Expires: May 3, 2018

J. Arkko
K. Norrman
V. Torvinen
Ericsson

October 30, 2017

Perfect-Forward Secrecy for the Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' PFS) draft-arkko-eap-aka-pfs-00

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 RFC 5448. The extension 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 merely passively eavesdropping the EAP-AKA' exchanges and deriving associated session keys, forcing 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, 2018.

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 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 designs can ask. Is there something that we can do to limit the consequences of attacks, should they occur?

This specification is an optional extension to the EAP-AKA' authentication method [RFC5448]. The extension 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 merely passively eavesdropping the EAP-AKA' exchanges and deriving associated session keys, forcing attackers to use active attacks instead.

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. This 00 version of the specification is an initial proposal for ensuring SIM-based authentication in EAP makes attacks difficult. Comments and suggestions are much appreciated, including design improvements.

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. There are currently no requirements mandating the use of this extension from 3GPP or otherwise, but the authors want to provide a public specification of an extension that helps defend against one aspect of pervasive surveillance. 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.

It should also be noted that the planned 5G network architecture includes the use of the EAP framework for authentication. The default authentication method within that context will be EAP-AKA', but other methods can certainly also be run.

2. Background

2.1. 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, 3rd generation 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.

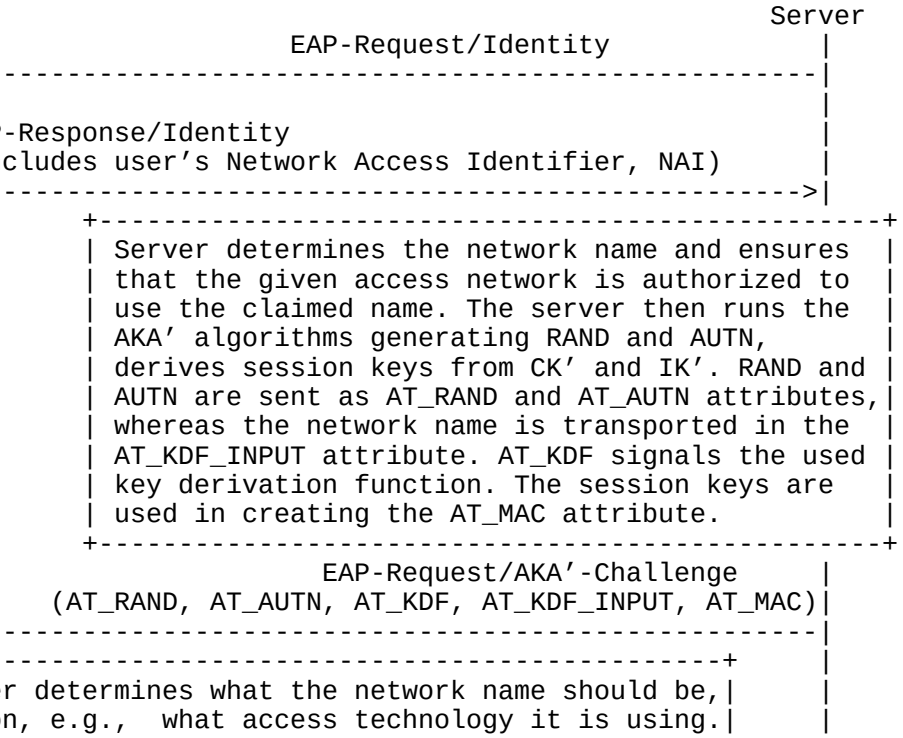
- The RAND and the AUTN are delivered to the identity module.

- 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.

2.2. EAP-AKA' Protocol

When AKA (and AKA') are embedded into EAP, the authentication on the network side is moved to the home environment; the serving network perdorms 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 [RFC5448] and [RFC4187].



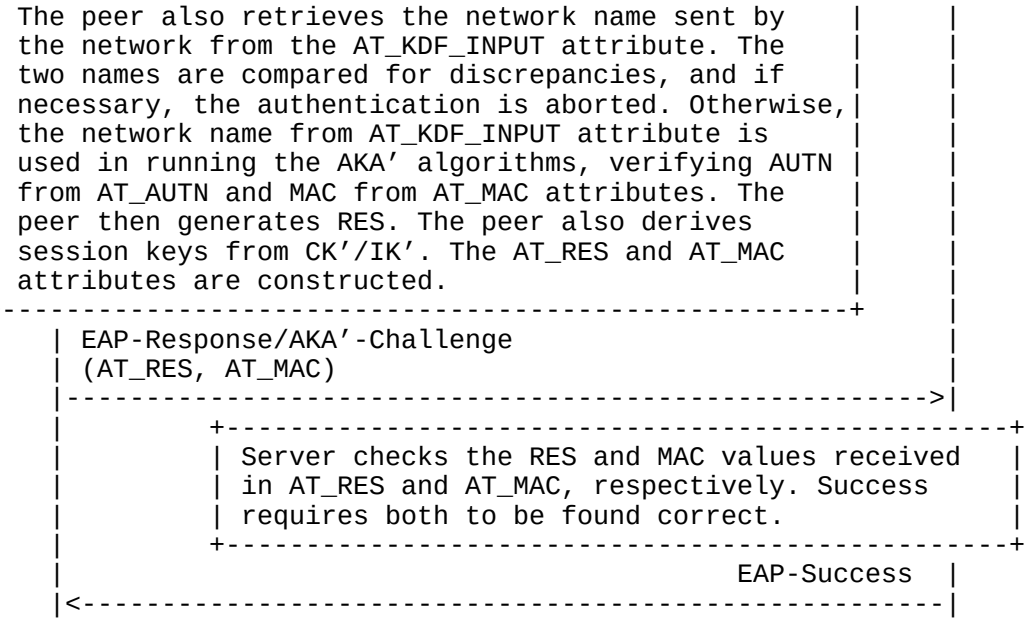


Figure 1: EAP-AKA' Authentication Process

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

Current 3GPP systems use (U)SIM pre-shared key based protocols to authenticate subscribers. Since the addition of replay protection and mutual authentication in the third generation 3GPP systems, there have been no published attacks that violate the security properties defined for the Authentication and Key Agreement (AKA) in, at least not within the assumed trust model. (However, there have been attacks using a different trust model [CB2014] [MT2012]; the protocol was not designed to counter those situations. There have also been attacks against systems where AKA is used in a different setting than initially intended, e.g. [BT2013].)

Recent reports of compromised long term pre-shared keys used in AKA [Heist2015] indicate a need to look into solutions that allow a weaker trust model, in particular for future 5G systems. It is also noted in [Heist2015] that, even if the current trust model is kept, some security can be retained in this situation by providing Perfect Forward Security (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. Introducing PFS for authentication in 3GPP systems can be achieved by adding a Diffie-Hellman (DH) exchange.

3. Requirements Language

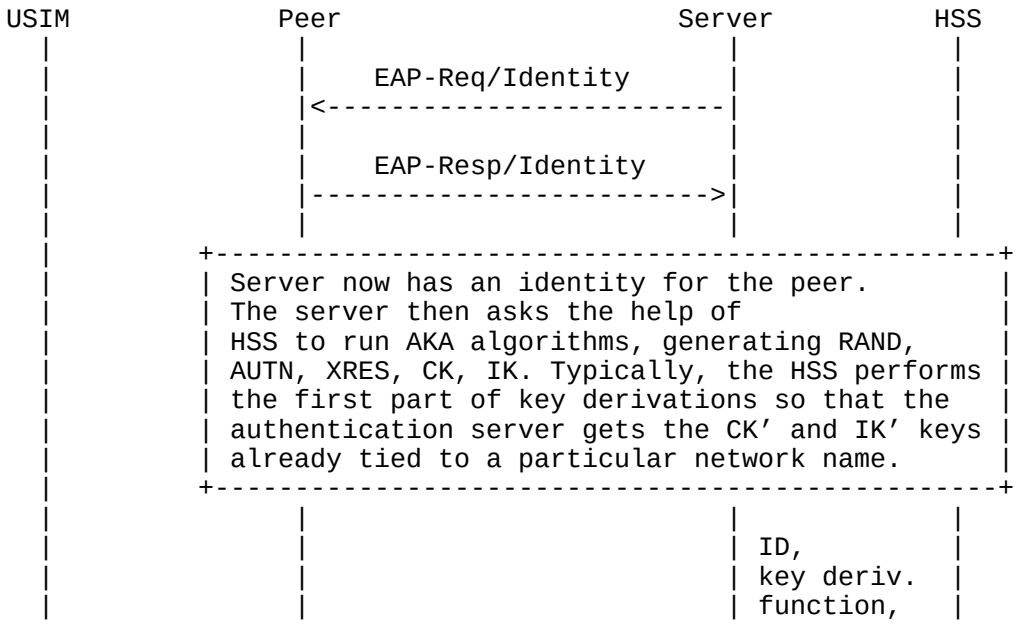
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

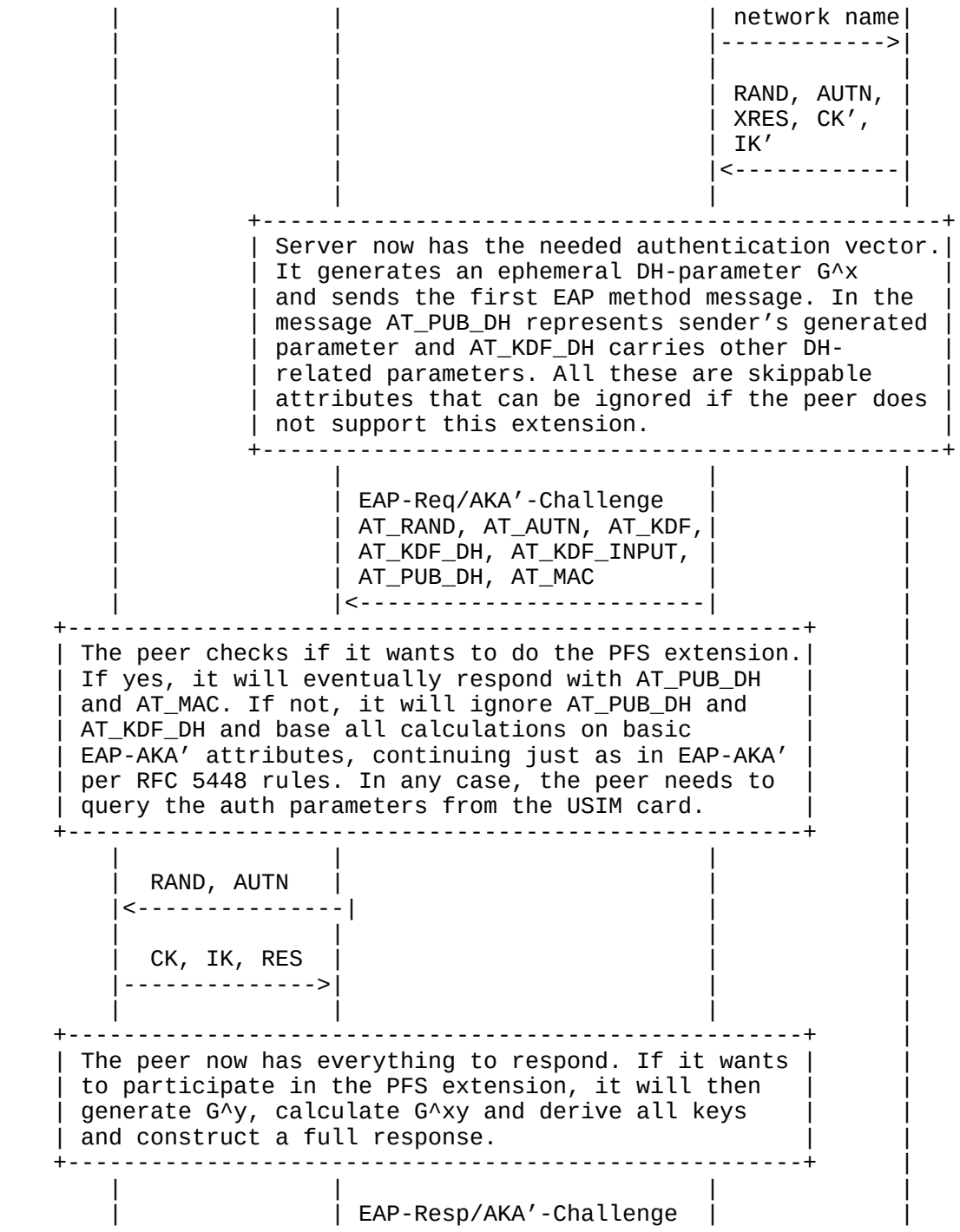
4. Protocol Overview

The enhancements in the protocol specified here 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. The enhancements brought in this document change 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 the 3GPP specifications .




network name
RAND, AUTN,
XRES, CK',
IK'
Server now has the needed authentication vector.
It generates an ephemeral DH-parameter G^x
and sends the first EAP method message. In the
message AT_PUB_DH represents sender's generated
parameter and AT_KDF_DH carries other DH-
related parameters. All 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_DH, AT_KDF_INPUT,
AT_PUB_DH, AT_MAC
The peer checks if it wants to do the PFS extension.
If yes, it will eventually respond with AT_PUB_DH
and AT_MAC. If not, it will ignore AT_PUB_DH and
AT_KDF_DH and base all calculations on basic
EAP-AKA' attributes, continuing just as in EAP-AKA'
per RFC 5448 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 G^y, calculate G^xy and derive all keys
and construct a full response.
EAP-Resp/AKA'-Challenge

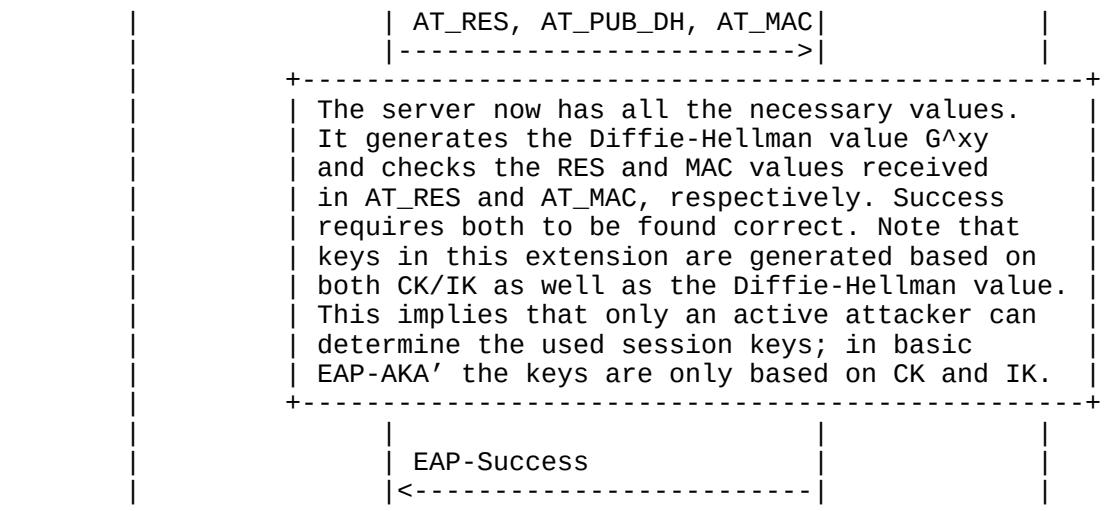


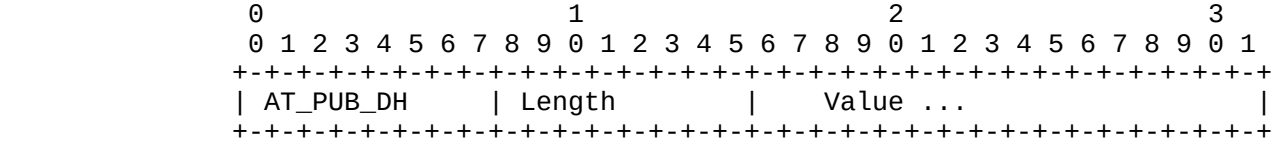
Figure 2: EAP-AKA' PFS Authentication Process

5. Extensions to EAP-AKA'

5.1. AT_PUB_DH

The AT_PUB_DH carries a Diffie-Hellman value.

The format of the AT_PUB_DH attribute is shown below.



The fields are as follows:

AT_PUB_DH

This is set to TBA1 BY IANA.

Length

The length of the attribute, set as other attributes in EAP-AKA [RFC4187].

Value

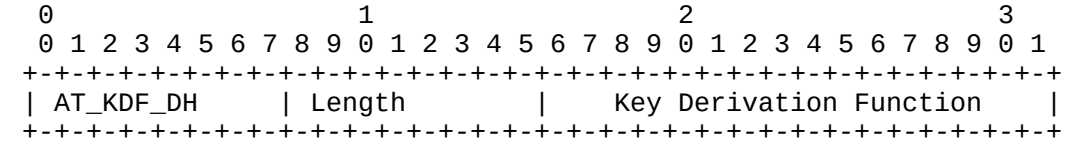
This value is the sender's Diffie-Hellman public value. For Curve25519, the length of this value is 32 bytes, represented as specified in [RFC8031] and [RFC7748].

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

5.2. AT_KDF_DH

The AT_KDF_DH 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 Diffie-Hellman 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_DH attribute is shown below.



The fields are as follows:

AT_KDF_DH

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 5.3 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 [RFC5448].

Servers MUST send one or more AT_KDF_DH 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.

Upon receiving a set of these attributes, if the peer supports and is willing to use the key derivation function indicated by the first attribute, and is willing and able to use the extension defined in this specification, the function is taken into use without any further negotiation. However, 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 [RFC5448] for AT_KDF, the peer sends EAP-Response/AKA'-Challenge message that contains only one attribute, AT_KDF_DH 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_DH 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_DH attribute.

Upon receiving an EAP-Response/AKA'-Challenge with AT_KDF_DH from the peer, the server checks that the suggested AT_KDF_DH value was one of the alternatives in its offer. The first AT_KDF_DH value in the message from the server is not a valid alternative. If the peer has replied with the first AT_KDF_DH 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_DH 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_DH 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_DH attributes without having requested negotiation, the peer MUST behave as if AT_MAC had been incorrect and fail the authentication.

5.3. New Key Derivation Function

A new Key Derivation Function type is defined for "EAP-AKA' with DH and Curve25519", represented by value 1. It represents a particular choice of key derivation function and at the same time selects a Diffie-Hellman group to be used.

The Key Derivation Function type value is only used in the AT_KDF_DH 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 [RFC5448].

Key derivation in this extension produces exactly the same keys for internal use within one authentication run as RFC 5448 EAP-AKA' did. 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_DH 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 and as follows below.

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

The rest of computation proceeds as defined in Section 3.3 of [RFC5448].

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 [RFC5448]. 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 [RFC5448] 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].

5.4. Diffie-Hellman Groups

The selection of suitable groups for the Diffie-Hellman 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_DH. For "EAP-AKA' with DH and Curve25519" the Diffie-Hellman group is the Curve25519 group specified in [RFC8031].

5.5. 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 [RFC5448] or [RFC4187] apply.

5.5.1. EAP-Request/AKA'-Identity

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

5.5.2. EAP-Response/AKA'-Identity

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

5.5.3. EAP-Request/AKA'-Challenge

The server sends the EAP-Request/AKA'-Challenge on full authentication as specified by [RFC4187] and [RFC5448]. The attributes AT_RAND, AT_AUTN, and AT_MAC MUST be included and checked on reception as specified in 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_DH and AT_PUB_DH attributes MUST be included. The AT_PUB_DH attribute

carries the server's public Diffie-Hellman key. If either AT_KDF_DH or AT_PUB_DH 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_DH, AT_PUB_DH 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 [RFC5448]. Finally, the operation in case an error occurs is specified in Section 6.3.1. of [RFC4187].

5.5.4. 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 [RFC5448]. 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 5.5.3, the peer responds per the rules specified below. Otherwise, the peer responds as specified in [RFC4187] and [RFC5448] and ignores the attributes related to this extension.

The AT_MAC attribute MUST be included and checked as specified in [RFC5448]. In EAP-Response/AKA'-Challenge, there is no message-specific data covered by the MAC. The AT_PUB_DH 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].

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].

5.5.5. 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.

5.5.6. EAP-Response/AKA'-Reauthentication

No changes, but as discussed in Section 5.5.5, re-authentication is based on the key material generated by EAP-AKA' and the extension defined in this document.

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

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

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

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

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

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

5.5.10. EAP-Request/AKA'-Notification

No changes.

5.5.11. EAP-Response/AKA'-Notification

No changes.

6. 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 [RFC5448].

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 2.3 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, 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.

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_DH and AT_PUB_DH 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, but that is a case that cannot be solved by this protocol, or any protocol for that matter.) However, as discussed in the introduction, even an attacker with access to the long-term keys is required to be MITM on each AKA run, which makes mass survailance slightly more laborous.

Key derivation

This extension provides key material that is based on the Diffie-Hellman keys, yet bound to the authentication through the (U)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 (U)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 of 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. IANA Considerations

This extension of EAP-AKA' shares its attribute space and subtypes with EAP-SIM [RFC4186], EAP-AKA [RFC4186], and EAP-AKA' [RFC5448].

Two new Attribute Type value (TBA1, TBA2) in the skippable range need to be assigned for AT_PUB_DH (Section 5.1) and AT_KDF_DH (Section 5.2 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 DH and Curve25519" type (1, see Section 5.3) needs 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 DH and Curve25519	[TBD BY IANA: THIS RFC]
2-65535	Unassigned	

8. References

8.1. Normative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[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>.

[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>.

[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>.

[RFC8031] Nir, Y. and S. Josefsson, "Curve25519 and Curve448 for the Internet Key Exchange Protocol Version 2 (IKEv2) Key Agreement", RFC 8031, DOI 10.17487/RFC8031, December 2016, <https://www.rfc-editor.org/info/rfc8031>.

[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>.

8.2. 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>.

[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.

[CB2014] Choudhary, A. and R. Bhandari, "3GPP AKA Protocol: Simplified Authentication Process", December 2014, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 12.

[MT2012] Mjolsnes, S. and J-K. Tsay, "A vulnerability in the UMTS and LTE authentication and key agreement protocols", October 2012, in Proceedings of the 6th international conference on Mathematical Methods, Models and Architectures for Computer Network Security: computer network security.

[BT2013] Beekman, J. and C. Thompson, "Breaking Cell Phone Authentication: Vulnerabilities in AKA, IMS and Android", August 2013, in 7th USENIX Workshop on Offensive Technologies, WOOT '13.

[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. 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 John Mattson, Mohit Sethi, Vesa Lehtovirta, Bengt Sahlin, Prajwol Kumar Nakarmi and many people at the 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