Internet Engineering Task Force (IETF)
Request for Comments: 5998
Updates: 5996
Category: Standards Track
ISSN: 2070-1721

P. Eronen
Independent
H. Tschofenig
Nokia Siemens Networks
Y. Sheffer
Independent

September 2010

# An Extension for EAP-Only Authentication in IKEv2

## Abstract

IKEv2 specifies that Extensible Authentication Protocol (EAP) authentication must be used together with responder authentication based on public key signatures. This is necessary with old EAP methods that provide only unilateral authentication using, e.g., one-time passwords or token cards.

This document specifies how EAP methods that provide mutual authentication and key agreement can be used to provide extensible responder authentication for IKEv2 based on methods other than public key signatures.

## Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5998.

## 1. Introduction

The Extensible Authentication Protocol (EAP), defined in [RFC3748], is an authentication framework that supports multiple authentication mechanisms. Today, EAP has been implemented at end hosts and routers that connect via switched circuits or dial-up lines using PPP [RFC1661], IEEE 802 wired switches [IEEE8021X], and IEEE 802.11 wireless access points [IEEE80211i].

One of the advantages of the EAP architecture is its flexibility. EAP is used to select a specific authentication mechanism, typically after the authenticator requests more information in order to determine the specific authentication method to be used. Rather than requiring the authenticator (e.g., wireless LAN access point) to be updated to support each new authentication method, EAP permits the use of a backend authentication server that may implement some or all authentication methods.

IKEv2 ([RFC4306] and [RFC5996]) is a component of IPsec used for performing mutual authentication and establishing and maintaining Security Associations (SAs) for IPsec ESP and Authentication Header (AH). In addition to supporting authentication using public key signatures and shared secrets, IKEv2 also supports EAP authentication.

IKEv2 provides EAP authentication since it was recognized that public key signatures and shared secrets are not flexible enough to meet the requirements of many deployment scenarios. By using EAP, IKEv2 can leverage existing authentication infrastructure and credential databases, since EAP allows users to choose a method suitable for existing credentials, and also makes separation of the IKEv2 responder (VPN gateway) from the EAP authentication endpoint (backend Authentication, Authorization, and Accounting (AAA) server) easier.

Some older EAP methods are designed for unilateral authentication only (that is, EAP peer to EAP server). These methods are used in conjunction with IKEv2 public-key-based authentication of the responder to the initiator. It is expected that this approach is especially useful for "road warrior" VPN gateways that use, for instance, one-time passwords or token cards to authenticate the clients.

However, most newer EAP methods, such as those typically used with IEEE 802.11i wireless LANs, provide mutual authentication and key agreement. Currently, IKEv2 specifies that these EAP methods must also be used together with responder authentication based on public key signatures.

In order for the public key signature authentication of the gateway to be effective, a deployment of Public Key Infrastructure (PKI) is required, which has to include management of trust anchors on all supplicants. In many environments, this is not realistic, and the security of the gateway public key is the same as the security of a self-signed certificate. Mutually authenticating EAP methods alone can provide a sufficient level of security in many circumstances, and in fact, in some deployments, IEEE 802.11i uses EAP without any PKI for authenticating the Wireless Local Area Network (WLAN) access points.

This document specifies how EAP methods that offer mutual authentication and key agreement can be used to provide responder authentication in IKEv2 completely based on EAP.

### 1.1. Terminology

All notation in this protocol extension is taken from [RFC4306].

Numbered messages refer to the IKEv2 message sequence when using EAP.

Thus:

- Message 1 is the request message of IKE_SA_INIT.
- Message 2 is the response message of IKE_SA_INIT.
- Message 3 is the first request of IKE_AUTH.
- Message 4 is the first response of IKE_AUTH.

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].

## 2. Scenarios

In this section, we describe two scenarios for extensible authentication within IKEv2. These scenarios are intended to be illustrative examples rather than specifying how things should be done.

Figure 1 shows a configuration where the EAP and the IKEv2 endpoints are co-located. Authenticating the IKEv2 responder using both EAP and public key signatures is redundant. Offering EAP-based authentication has the advantage that multiple different authentication and key exchange protocols are available with EAP with different security properties (such as strong password-based protocols, protocols offering user identity confidentiality, and many more).

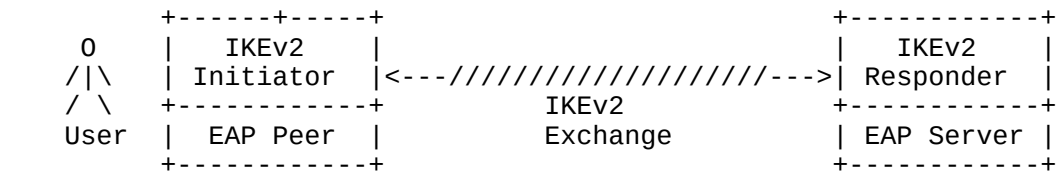


Figure 1: EAP and IKEv2 Endpoints Are Co-Located

Figure 2 shows a typical corporate network access scenario. The initiator (client) interacts with the responder (VPN gateway) in the corporate network. The EAP exchange within IKE runs between the

client and the home AAA server. As a result of a successful EAP authentication protocol run, session keys are established and sent from the AAA server to the VPN gateway, and then used to authenticate the IKEv2 SA with AUTH payloads.

The protocol used between the VPN gateway and AAA server could be, for instance, Diameter [RFC4072] or RADIUS [RFC3579]. See Section 6 for related security considerations.

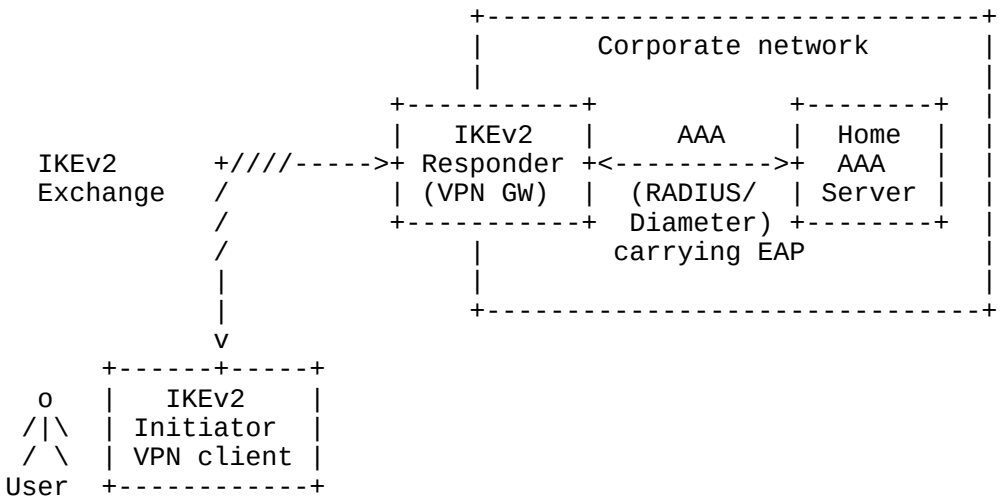


Figure 2: Corporate Network Access

## 3. Solution

IKEv2 specifies that when the EAP method establishes a shared secret key, that key is used by both the initiator and responder to generate an AUTH payload (thus authenticating the IKEv2 SA set up by messages 1 and 2).

When used together with public key responder authentication, the responder is, in effect, authenticated using two different methods: the public key signature AUTH payload in message 4, and the EAP-based AUTH payload later.

If the initiator does not wish to use public-key-based responder authentication, it includes an EAP_ONLY_AUTHENTICATION notification payload (16417) in message 3. The Protocol ID and Security Parameter Index (SPI) size fields are set to zero, and there is no additional data associated with this notification.

If the responder supports this notification and chooses to use it, it omits the public-key-based AUTH payload and CERT payloads from message 4.

If the responder does not support the EAP_ONLY_AUTHENTICATION notification or does not wish to use it, it ignores the notification payload, and includes the AUTH payload in message 4. In this case, the initiator MUST verify that payload and any associated certificates, as per [RFC4306].

When receiving message 4, the initiator MUST verify that the proposed EAP method is allowed by this specification, and MUST abort the protocol immediately otherwise.

Both the initiator and responder MUST verify that the EAP method actually used provided mutual authentication and established a shared secret key. The AUTH payloads sent after EAP Success MUST use the EAP-generated key, and MUST NOT use SK_pi or SK_pr (see Section 2.15 of [RFC5996]).

An IKEv2 message exchange with this modification is shown below:

```
   Initiator                   Responder
  -----------                 -----------
   HDR, SAi1, KEi, Ni,
        [N(NAT_DETECTION_SOURCE_IP),
         N(NAT_DETECTION_DESTINATION_IP)]  -->

                          <--   HDR, SAr1, KEr, Nr, [CERTREQ],
                                     [N(NAT_DETECTION_SOURCE_IP),
                                      N(NAT_DETECTION_DESTINATION_IP)]

   HDR, SK { IDi, [IDr], SAi2, TSi, TSr,
             N(EAP_ONLY_AUTHENTICATION),
             [CP(CFG_REQUEST)] }  -->

                          <--   HDR, SK { IDr, EAP(Request) }

   HDR, SK { EAP(Response) }  -->

                          <--   HDR, SK { EAP(Request) }

   HDR, SK { EAP(Response) }  -->

                          <--   HDR, SK { EAP(Success) }

   HDR, SK { AUTH }  -->

                          <--   HDR, SK { AUTH, SAr2, TSi, TSr,
                                           [CP(CFG_REPLY] }
```

Note: all notation in the above protocol sequence and elsewhere in this specification is as defined in [RFC4306], and see in particular Sec. 1.2 of [RFC4306] for payload types.

The NAT detection and Configuration payloads are shown for informative purposes only; they do not change how EAP authentication works.

An IKE SA that was set up with this extension can be resumed using the mechanism described in [RFC5723]. However, session resumption does not change the authentication method. Therefore, during the IKE_AUTH exchange of the resumed session, this extension MUST NOT be sent by the initiator.

## 4. Safe EAP Methods

EAP methods to be used with this extension MUST have the following properties:

1. The method provides mutual authentication of the peers.

2. The method is key-generating.

3. The method is resistant to dictionary attacks.

The authors believe that the following EAP methods are secure when used with the current extension. The list is not inclusive, and there are likely other safe methods that have not been listed here.

| Method Name | Allows Channel Binding? | Reference |
|---|---|---|
| EAP-SIM | No | [RFC4186] |
| EAP-AKA | Yes | [RFC4187] |
| EAP-AKA' | Yes | [RFC5448] |
| EAP-GPSK | Yes | [RFC5433] |
| EAP-pwd | No | [RFC5931] |
| EAP-EKE | Yes | [EMU-EAP-EKE] |
| EAP-PAX | Yes | [RFC4746] |
| EAP-SAKE | No | [RFC4763] |
| EAP-SRP | No | [EAP-SRP] |
| EAP-POTP (mutual authentication variant) | Yes | [RFC4793] |
| EAP-TLS | No | [RFC5216] |
| EAP-FAST | No | [RFC4851] |
| EAP-TTLS | No | [RFC5281] |

The "Allows channel binding?" column denotes protocols where protected identity information may be sent between the EAP endpoints. This third, optional property of the method provides protection against certain types of attacks (see Section 6.2 for an explanation), and therefore in some scenarios, methods that allow for channel binding are to be preferred. It is noted that at the time of writing, even when such capabilities are provided, they are not fully specified in an interoperable manner. In particular, no RFC specifies what identities should be sent under the protection of the channel binding mechanism, or what policy is to be used to correlate identities at the different layers.

## 5. IANA Considerations

This document defines a new IKEv2 Notification Payload type, EAP_ONLY_AUTHENTICATION, described in Section 3. This payload has been assigned the type number 16417 from the "Status Types" range.

## 6. Security Considerations

Security considerations applicable to all EAP methods are discussed in [RFC3748]. The EAP Key Management Framework [RFC5247] deals with issues that arise when EAP is used as a part of a larger system.

### 6.1. Authentication of IKEv2 SA

It is important to note that the IKEv2 SA is not authenticated by just running an EAP conversation: the crucial step is the AUTH payload based on the EAP-generated key. Thus, EAP methods that do not provide mutual authentication or establish a shared secret key MUST NOT be used with the modifications presented in this document.

### 6.2. Authentication with Separated IKEv2 Responder / EAP Server

As described in Section 2, the EAP conversation can terminate either at the IKEv2 responder or at a backend AAA server.

If the EAP method is terminated at the IKEv2 responder, then no key transport via the AAA infrastructure is required. Pre-shared secret and public-key-based authentication offered by IKEv2 is then replaced by a wider range of authentication and key exchange methods.

However, typically EAP will be used with a backend AAA server. See [RFC5247] for a more complete discussion of the related security issues; here we provide only a short summary.

When a backend server is used, there are actually two authentication exchanges: the EAP method between the client and the AAA server, and another authentication between the AAA server and IKEv2 gateway. The AAA server authenticates the client using the selected EAP method, and they establish a session key. The AAA server then sends this key to the IKEv2 gateway over a connection authenticated using, e.g., IPsec or Transport Layer Security (TLS).

Some EAP methods do not have any concept of pass-through authenticator (e.g., Network Access Server (NAS) or IKEv2 gateway) identity, and these two authentications remain quite independent of each other. That is, after the client has verified the AUTH payload sent by the IKEv2 gateway, it knows that it is talking to SOME gateway trusted by the home AAA server, but not which one. The

situation is somewhat similar if a single cryptographic hardware accelerator, containing a single private key, would be shared between multiple IKEv2 gateways (perhaps in some kind of cluster configuration). In particular, if one of the gateways is compromised, it can impersonate any of the other gateways towards the user (until the compromise is discovered and access rights revoked).

In some environments it is not desirable to trust the IKEv2 gateways this much (also known as the "Lying NAS Problem"). EAP methods that provide what is called "connection binding" or "channel binding" transport some identity or identities of the gateway (or WLAN access point / NAS) inside the EAP method. Then the AAA server can check that it is indeed sending the key to the gateway expected by the client. A potential solution is described in [EAP-SERVICE], see also [EMU-AAAPAY].

In some deployment configurations, AAA proxies may be present between the IKEv2 gateway and the backend AAA server. These AAA proxies MUST be trusted for secure operation, and therefore SHOULD be avoided when possible; see Section 2.3.4 of [RFC4072] and Section 4.3.7 of [RFC3579] for more discussion.

### 6.3. Protection of EAP Payloads

Although the EAP payloads are encrypted and integrity protected with SK_e/SK_a, this does not provide any protection against active attackers. Until the AUTH payload has been received and verified, a man-in-the-middle can change the KEi/KEr payloads and eavesdrop or modify the EAP payloads.

In IEEE 802.11i wireless LANs, the EAP payloads are neither encrypted nor integrity protected (by the link layer), so EAP methods are typically designed to take that into account.

In particular, EAP methods that are vulnerable to dictionary attacks when used in WLANs are still vulnerable (to active attackers) when run inside IKEv2.

The rules in Section 4 are designed to avoid this potential vulnerability.

### 6.4. Identities and Authenticated Identities

When using this protocol, each of the peers sends two identity values:

1. An identity contained in the IKE ID payload.

2. An identity transferred within the specific EAP method's messages.

(IKEv2 omits the EAP Identity request/response pair, see Section 3.16 of [RFC5996].) The first identity value can be used by the recipient to route AAA messages and/or to select authentication and EAP types. But it is only the second identity that is directly authenticated by the EAP method. The reader is referred to Section 2.16 of [RFC5996] regarding the need to base IPsec policy decisions on the authenticated identity. In the context of the extension described here, this guidance on IPsec policy applies both to the authentication of the client by the gateway and vice versa.

### 6.5. User Identity Confidentiality

IKEv2 provides confidentiality for the initiator identity against passive eavesdroppers, but not against active attackers. The initiator announces its identity first (in message 3), before the responder has been authenticated. The usage of EAP in IKEv2 does not change this situation, since the ID payload in message 3 is used instead of the EAP Identity Request/Response exchange. This is somewhat unfortunate since when EAP is used with public key authentication of the responder, it would be possible to provide active user identity confidentiality for the initiator.

IKEv2 protects the responder's identity even against active attacks. This property cannot be provided when using EAP. If public key responder authentication is used in addition to EAP, the responder reveals its identity before authenticating the initiator. If only EAP is used (as proposed in this document), the situation depends on the EAP method used (in some EAP methods, the server reveals its identity first).

Hence, if active user identity confidentiality for the responder is required then EAP methods that offer this functionality have to be used (see [RFC3748], Section 7.3).

## 7. Acknowledgments

This document borrows some text from [RFC3748], [RFC4306], and [RFC4072]. We would also like to thank Hugo Krawczyk for interesting discussions about this topic, Dan Harkins, and David Harrington for their comments.

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC5723] Sheffer, Y. and H. Tschofenig, "Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption", RFC 5723, January 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

### 8.2. Informative References

[EAP-SERVICE] Arkko, J. and P. Eronen, "Authenticated Service Information for the Extensible Authentication Protocol (EAP)", Work in Progress, October 2005.

[EAP-SRP] Carlson, J., Aboba, B., and H. Haverinen, "EAP SRP-SHA1 Authentication Protocol", Work in Progress, July 2001.

[EMU-AAAPAY] Clancy, C., Lior, A., Zorn, G., and K. Hoeper, "EAP Method Support for Transporting AAA Payloads", Work in Progress, May 2010.

[EMU-EAP-EKE] Sheffer, Y., Zorn, G., Tschofenig, H., and S. Fluhrer, "An EAP Authentication Method Based on the EKE Protocol", Work in Progress, August 2010.

[IEEE80211i] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements", IEEE Standard 802.11i-2004, July 2004.

[IEEE8021X] Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Port-Based Network Access Control", IEEE Standard 802.1X-2001, 2001.

[RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

[RFC4072] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.

[RFC4186] Haverinen, H. and J. Salowey, "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)", RFC 4186, January 2006.

[RFC4187] Arkko, J. and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)", RFC 4187, January 2006.

[RFC4746] Clancy, T. and W. Arbaugh, "Extensible Authentication Protocol (EAP) Password Authenticated Exchange", RFC 4746, November 2006.

[RFC4763] Vanderveen, M. and H. Soliman, "Extensible Authentication Protocol Method for Shared-secret Authentication and Key Establishment (EAP-SAKE)", RFC 4763, November 2006.

[RFC4793] Nystroem, M., "The EAP Protected One-Time Password Protocol (EAP-POTP)", RFC 4793, February 2007.

[RFC4851] Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)", RFC 4851, May 2007.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, March 2008.

[RFC5247] Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, August 2008.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008.

[RFC5433] Clancy, T. and H. Tschofenig, "Extensible Authentication Protocol - Generalized Pre-Shared Key (EAP-GPSK) Method", RFC 5433, February 2009.

[RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, May 2009.

[RFC5931] Harkins, D. and G. Zorn, "Extensible Authentication Protocol (EAP) Authentication Using Only A Password", RFC 5931, August 2010.

## Appendix A. Alternative Approaches

In this section, we list alternatives that have been considered during the work on this document. We concluded that the solution presented in Section 3 seems to fit better into IKEv2.

### A.1. Ignore AUTH Payload at the Initiator

With this approach, the initiator simply ignores the AUTH payload in message 4 (but obviously must check the second AUTH payload later!). The main advantage of this approach is that no protocol modifications are required and no signature verification is required. A significant disadvantage is that the EAP method to be used cannot be selected to take this behavior into account.

The initiator could signal to the responder (using a notification payload) that it did not verify the first AUTH payload.

### A.2. Unauthenticated Public Keys in AUTH Payload (Message 4)

Another solution approach suggests the use of unauthenticated public keys in the public key signature AUTH payload (for message 4).

That is, the initiator verifies the signature in the AUTH payload, but does not verify that the public key indeed belongs to the intended party (using certificates) -- since it doesn't have a PKI that would allow this. This could be used with X.509 certificates (the initiator ignores all other fields of the certificate except the public key), or "Raw RSA Key" CERT payloads.

This approach has the advantage that initiators that wish to perform certificate-based responder authentication (in addition to EAP) may do so, without requiring the responder to handle these cases separately. A disadvantage here, again, is that the EAP method selection cannot take into account the incomplete validation of the responder's certificate.

If using RSA, the overhead of signature verification is quite small, compared to the g^xy calculation required by the Diffie-Hellman exchange.

### A.3. Using EAP Derived Session Keys for IKEv2

It has been proposed that when using an EAP method that provides mutual authentication and key agreement, the IKEv2 Diffie-Hellman exchange could also be omitted. This would mean that the session keys for IPsec SAs established later would rely only on EAP-provided keys.

It seems the only benefit of this approach is saving some computation time (g^xy calculation). This approach requires designing a completely new protocol (which would not resemble IKEv2 anymore); we do not believe that it should be considered. Nevertheless, we include it for completeness.

## Authors' Addresses

Pasi Eronen
Independent

EMail: pe@iki.fi

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
EMail: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at

Yaron Sheffer
Independent

EMail: yaronf.ietf@gmail.com