PANA/IKEv2: an Internet Authentication Protocol for Heterogeneous Access

Paulo S. Pagliusi and Chris J. Mitchell

Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
{P.S.Pagliusi, C.Mitchell}@rhul.ac.uk
http://www.isg.rhul.ac.uk

Abstract. Currently there are no Internet access authentication protocols available that support both symmetric and asymmetric cryptographic techniques, can be carried over arbitrary access networks, and are flexible enough to be re-used in all the likely future ubiquitous mobility access contexts. This article proposes the PANA/IKEv2 authentication protocol for heterogeneous network access as a step towards filling this gap. A security analysis of the PANA/IKEv2 protocol is also provided. This article aims primarily at contributing to the design of authentication protocols suitable for use in future heterogeneous Internet access environments supporting ubiquitous mobility.

1 Introduction

According to the PAMPAS Project [1, p135], "the increasing heterogeneity of the networking environment is one of the long-term trends which requires new security approaches. The main challenges include the investigation and development of unified, secure and convenient authentication mechanisms that can be used in different access networks". Authentication and key agreement are the central components of secure access procedures for heterogeneous network access supporting ubiquitous mobility.

Currently there are no authentication protocols available that can be carried over arbitrary access networks, and are flexible enough for use with all the various access technologies likely to be deployed to support future ubiquitous mobility. Furthermore, existing access procedures need to be made resistant to Denial-of-Service (DoS) attacks; they also do not provide non-repudiation. In addition to being limited to specific access media (e.g. 802.1aa [2] for IEEE 802 links), some of these protocols are limited to specific network topologies (e.g. PPP [3] for point-to-point links) and are not scalable.

Additionally, the cryptography used to support the access procedures can be based either on secret key (symmetric) or public key (asymmetric) techniques. Whereas the former requires the involvement of the home network during the

initial authentication process between a user and visited network, the latter allows for architectures that avoid on-line involvement of the home network, since authentication may then be based on certificates. Nevertheless, asymmetric techniques typically require a Public Key Infrastructure to support key distribution, and use of this PKI may require on-line certificate status checking. While symmetric techniques are used almost exclusively today, it seems likely that asymmetric techniques will gain greater importance in future ubiquitous mobility access networks because of their greater flexibility.

The recent IETF PANA (Protocol for carrying Authentication for Network Access[1]) work aims to provide a protocol [4] that will be a flexible and scalable network-layer authentication carrier for access networks that support IP. PANA will be capable of transporting any EAP (Extensible Authentication Protocol) method [5] to enable access authentication. In addition, the EAP-IKEv2 protocol [6] specifies a way of encapsulating the first phase of the Internet Key Exchange (IKEv2) Protocol [7], which supports both symmetric and asymmetric authentication, within EAP. Once inside EAP, the IKEv2 parameters can thus be carried by PANA. In this paper we present a proposal for combining IKEv2 authentication with EAP-IKEv2 and PANA, which we call PANA/IKEv2.

The goal of the PANA/IKEv2 protocol is to provide an IP compatible, flexible and scalable authentication method that allows a client to be authenticated by either symmetric or asymmetric techniques in a heterogeneous network access environment. The proposal adapts the security techniques used in IKEv2 to the PANA structure. The protocol runs between a client device and an agent device in the access network, where the agent may be a client of an AAA (Authentication, Authorization and Accounting) infrastructure.

Section 2 summarises the authentication and key exchange phase of the IKEv2 protocol, Section 3 describes the EAP-IKEv2 method, and Section 4 explains the PANA protocol. Section 5 then describes the proposed new PANA/ IKEv2 authentication scheme. Section 6 analyses the threats to the PANA/IKEv2 protocol, Section 7 considers its advantages and disadvantages and, finally, Sections 8, 9, and 10 present possible further work, conclusions and acknowledgements.

2 Authentication and Key Exchange via IKEv2 Protocol

IKEv2 [7] is a component of IPsec (IP Security Protocol[2]) used for performing mutual authentication and establishing and maintaining security associations (SAs). IKEv2 is a protocol which consists of two phases:

1. An authentication and key exchange protocol, which establishes an IKE-SA,
2. Messages and payloads which allow negotiation of parameters (e.g. algorithms, traffic selectors) in order to establish IPsec SAs (i.e. Child-SAs).

[1] http://www.ietf.org/html.charters/pana-charter.html
[2] http://www.ietf.org/html.charters/ipsec-charter.html

In addition, IKEv2 also includes certain payloads and messages which allow configuration parameters to be exchanged for remote access scenarios. The PANA/IKEv2 protocol defined here uses the IKEv2 payloads and messages from phase 1.

IKEv2 is designed to address certain issues with IKEv1 [8], as described in Appendix A of [7]. Important here are the reduced number of initial exchanges, support of legacy authentication, decreased latency of the initial exchange, optional DoS protection capability, and the resolution of certain known security defects. IKEv2 is a protocol that has received a considerable amount of expert review, and that benefits from the experience gained from IKEv1. The goal of PANA/IKEv2 is to inherit these properties through the EAP-IKEv2 method.

IKEv2 also provides authentication and key exchange capabilities which allow an entity to use symmetric as well as asymmetric cryptographic techniques, in addition to legacy authentication[3] support, within a single protocol. Such flexibility seems likely to be important for heterogeneous network access supporting ubiquitous mobility, and is provided by PANA/IKEv2.

For further information on IKEv2 and its design rationale, see Perlman [10].

3 An EAP Mechanism for Carrying IKEv2

The EAP-IKEv2 protocol [6] is an EAP mechanism for authentication and session key distribution that uses IKEv2 [7]. It offers the security benefits of IKEv2 without aiming to establish IPsec SAs. The authentication method used within EAP-IKEv2 differs from IKEv2 only in the computation of the $AUTH$[4] payload.

Figure 1 shows an EAP-IKEv2 message flow, which occurs between the Initiator (I) and the Responder (R). I is also referred to here as the *Client* (acting on behalf of a user), whereas R is referred to as the *Authenticating Party*. R may be co-located with the *EAP server*, which is the network element that terminates the EAP protocol [5]. However, the EAP server is typically implemented on a separate AAA server in the user's home Internet AAA network, with whom R communicates using an AAA protocol. The core EAP-IKEv2 exchange (1) consists of three round trips, which may be reduced to two if the optional IKEv2 DoS protection (2) is not used.

In the EAP/IKEv2 authentication procedure, an identity request/response message pair (a, b) is first exchanged. Next, R sends (c) an EAP-Request/EAP-IKEv2 (Start) message. I sends back (d) a message that contains an IKEv2 header (HDR[5]), a payload with the cryptographic suites supported by I for

[3] *Legacy authentication* involves methods that are not strong enough to be used in networks where attackers can easily eavesdrop and spoof on the link (e.g. EAP-MD5 [9] over wireless links). They also may not be able to produce enough keying material. Use of legacy methods can be enabled by carrying them over a secure channel (see also [4, 7]).

[4] $AUTH$ contains data used for authentication purposes; see subsection 3.8 of [7].

[5] HDR contains Security Parameter Indexes (SPIs), version numbers, and flags of various sorts. SPIs are values chosen by I and R to identify a unique IKE_SA.

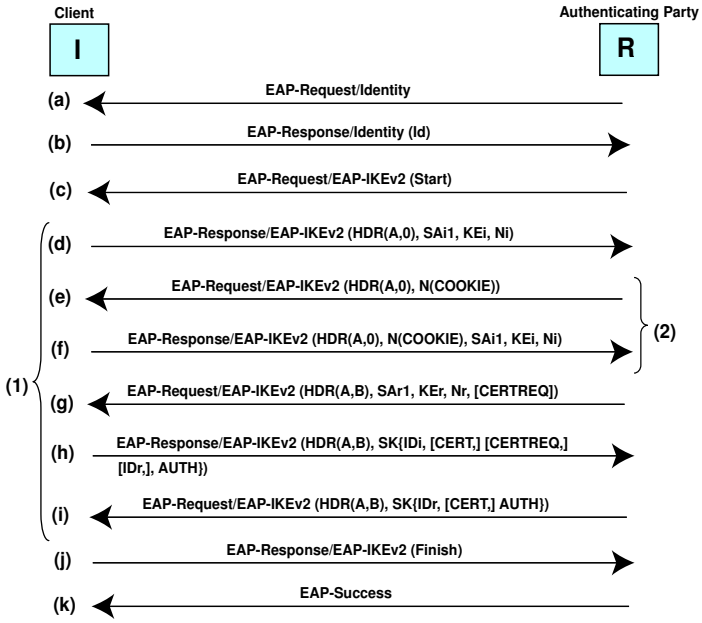

Fig. 1. EAP-IKEv2 message flow. The name of each message is shown, followed by the contents of the message in round brackets. Square brackets are used to denote optional fields.

the IKE_SA ($SAi1$), a Diffie-Hellman [11] value (KEi), and a nonce (Ni). Next, we may optionally have an IKEv2 DoS protection round trip (e, f) based on 'cookies' [7]. After that, R sends a message (g) that contains its choice of a cryptographic suite from among I's offers ($SAr1$), its value to complete the Diffie-Hellman exchange (KEr), and its nonce[6] (Nr). At this point, each party can generate the $SKEYSEED$ value (computed as a pseudo random function (prf) of Ni, Nr and the Diffie-Hellman shared secret), from which the keying material for the IKE_SA is derived. All but the IKEv2 headers of the messages that follow are encrypted and integrity protected[7], and this is indicated using the notation $SK\{\dots\}$.

I sends back (h) a message to assert its identity (IDi), to prove knowledge of the secret corresponding to IDi, and to integrity protect the contents of the first two messages with $AUTH$ (see subsection 2.15 of [7]). It may also send its certificate ($CERT$) and a list of its 'trust anchors', i.e. the names of the

HDR(A,0) means that I assigned the SPI *'A'* and R did not assign its SPI yet, while *HDR(A,B)* means that I assigned the SPI *'A'* and R assigned the SPI *'B'*.

[6] *Nonces* are inputs to cryptographic functions; they contain pseudo random data used to guarantee liveliness during an exchange, and protect against replay attacks.

[7] The recipients must verify that all signatures and *MACs* are computed correctly, and that the *ID* names correspond to the keys used to generate the $AUTH$ payload.

CAs whose public keys it trusts [12] (*CERTREQ*); the optional *IDr* payload enables I to specify which of R's identities it wants to talk to (e.g. when R is hosting multiple *IDs* at the same IP address). R then asserts its identity (*IDr*), optionally sends one or more certificates (*CERT*), and authenticates its identity with $AUTH$ (i). Start (c) and Finish (j) messages are required due to the asymmetric nature of IKEv2, and due to the Request/Response message handling of EAP. The message flow finishes with an EAP-Success message (k).

Man-in-the-middle attacks discovered in the context of tunnelled authentication protocols (see [13] and [14]) are applicable to IKEv2 if legacy authentication is used with the inner EAP [9]. To counter this threat, IKEv2 provides a compound authentication by including the inner EAP session key inside the $AUTH$ payload (see Subsection 6.1).

4 Protocol for carrying Authentication for Network Access (PANA)

This section briefly introduces the draft PANA protocol [4], a link-layer agnostic transport for EAP to enable client-to-network access authentication. PANA runs between a PaC (PANA Client) and a PAA (PANA Authentication Agent) situated in the access network, where the PAA may optionally be a client of an AAA infrastructure. PANA carries any authentication mechanism that can be specified as an EAP method (e.g. EAP/IKEv2), and can be used on any link that supports IP. The header of every PANA packet contains two sequence numbers to provide ordered delivery of EAP messages: one transmitted sequence number (tseq), and one received sequence number (rseq). The payload of any PANA message consists of zero or more Attribute Value Pairs (AVPs), e.g. an optional cookie AVP, used for making an initial handshake robust against 'blind DoS attacks' [4], a MAC AVP, protecting the integrity of a PANA message, or an EAP AVP, which transports an EAP payload.

Two important features of PANA, namely the security association (SA) and re-authentication, are now described. Once the EAP method has completed, a session key (e.g. the EAP/IKEv2 *MSK*) is shared by the PaC and the PAA. The session key is provided to the PaC as part of the EAP key exchange process, and the PAA can obtain the session key from the EAP server via the AAA infrastructure (if used). PANA SA establishment based on the EAP session key is required where no physical or link layer security is available. Two types of re-authentication (or fast reconnection) are supported by PANA. The first type enters the chosen EAP method (e.g. EAP/IKEv2) at the authentication phase, where the initial handshake phase can be omitted. If there is an existing PANA SA, PANA_auth messages carrying the EAP fast reconnection process can be protected with a MAC AVP. The second type is based on a single protected PANA message exchange without entering the authentication phase. If there is an existing PANA SA, both PaC and PAA can send a PANA_reauth_request to the communicating peer and expect the peer to return a PANA_reauth_answer, where both messages are protected with a MAC AVP.

5 PANA/IKEv2 Authentication Procedure

The PANA/IKEv2 mechanism proposed here involves three functional entities, namely the *PaC* (also referred to here as the *client*, *user* or *subscriber*), the *PAA* (or *authenticating party*) and the *EAP server*. The *PaC* is associated with a network device and a set of credentials that are used to prove the PaC identity for network access. The *PAA* authenticates the credentials provided by the PaC and grants network access. In the context of this article, the *EAP server* is assumed to be implemented on the AAA server. The PAA is thus an AAA client that communicates with the user's EAP server through an AAA protocol supporting EAP (e.g. Diameter EAP [15]) and key wrap (e.g. Diameter CMS [16], where this involves encrypting a content-encryption key using a key encrypting key).

PANA/IKEv2 also involves a further entity, namely the EP (Enforcement Point), which may be co-located with the PAA, which applies per-packet enforcement policies (i.e. filters) to the traffic of the PaC's devices.

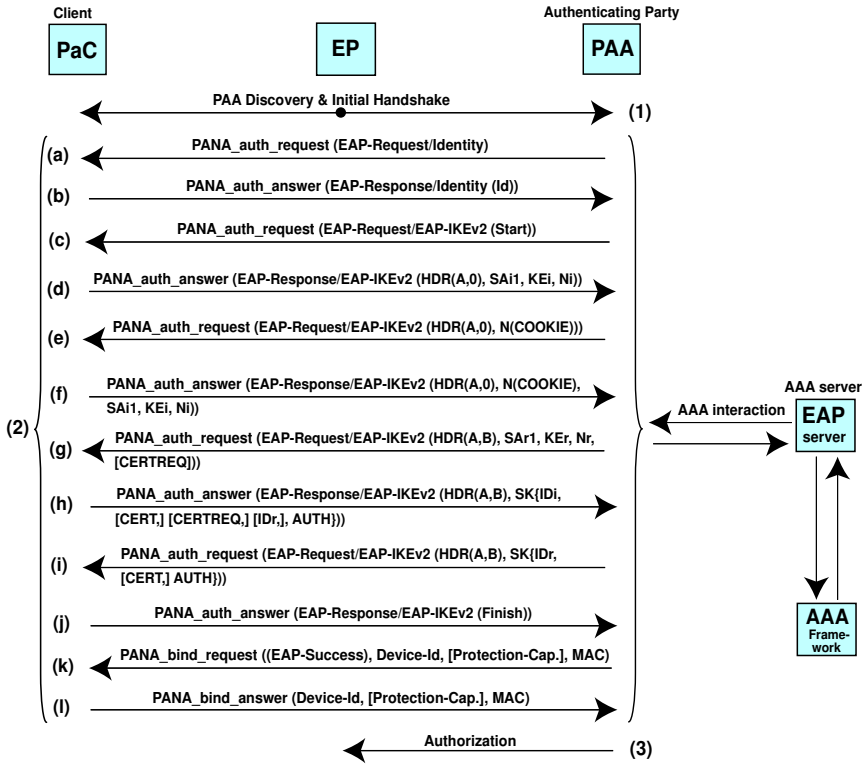

Fig. 2. PANA/IKEv2 full authentication procedure. The name of each message is shown, followed by the contents of the message in round brackets. Square brackets are used to denote optional fields.

Figure 2 shows the PANA/IKEv2 full authentication procedure, which has three main phases: (1) Discovery and initial handshake, (2) Authentication, and (3) Authorization. In the *Discovery* phase, an IP address for the PAA is identified, and a PANA/IKEv2 session is established between the PaC and the PAA, following the PANA model (see subsection 4.2 of [4]). In the *Authentication* phase, the main focus of this article and further explained below, EAP/IKEv2 messages encapsulated in PANA/IKEv2 are exchanged between the PaC and the PAA. At the end of this phase, a PANA SA is established, including the provision of a shared secret EAP/IKEv2 session key, called the 'Pre-Master-Secret' [6] or *KEYMAT*[8], exported as part of the EAP keying framework [17] for further key derivation; we call this the PANA/IKEv2 SA. During the *Authorization* phase, a separate protocol is used between the PAA and the EP to manage the PaC network access control. After this phase, the established PANA/IKEv2 session as well as the PANA/IKEv2 SA is deleted, following the PANA standard (see subsection 4.5 of [4]).

During the *Authentication* phase, the first PANA_auth_request message (a) issued by the PAA encapsulates an EAP-Request/Identity payload. The PaC responds (b) with a PANA_auth_answer, which carries an EAP-Response/Identity payload including the user identifier *Id*. After that, an EAP-Request/EAP-IKEv2 (Start) packet is transported in a PANA_auth_request (c). The PaC sends back (d) a PANA_auth_answer carrying the EAP-Request/EAP-IKEv2 payload, which contains *HDR*, *SAi1*, *KEi*, and also *Ni*, the random number chosen by the PaC. Next, we may optionally have an IKEv2 DoS protection round trip (e, f). The next PANA_auth_request message (g) issued by the PAA includes the EAP-Request/EAP-IKEv2 packet that contains *SAr1*, *KEr*, the random number *Nr* chosen by the PAA, and *CERTREQ*, an optional list of the PAA trust anchors. At this point, each party can derive the keying material for that IKE_SA. All but the *HDRs* of the messages that follow are encrypted and integrity protected.

On receipt of this message, the PaC sends back (h) a PANA_auth_answer message with its identity *IDi*, an *AUTH* value, and the following optional payloads: *CERT*[9], *CERTREQ*, and *IDr*, which enables the PaC to specify which of PAA's identities it wants to talk to. The notation $SK\{\dots\}$ here indicates that the content between brackets is encrypted and integrity protected. The PAA then sends a PANA_auth_request to assert its identity (*IDr*); this message also includes *AUTH* and optionally *CERT*(i). An EAP-Response/EAP-IKEv2 (Finish) packet is transported in a PANA_auth_answer (j).

Finally the PAA encapsulates the EAP-Success packet in a PANA_bind_request message sent to the PaC (k), and receives back an acknowledge through a PANA_bind_answer (l). Both PANA_bind messages are protected by a MAC

[8] *KEYMAT* is derived from *Ni*, *Nr*, and a temporary key called *SK_d* using a pseudo random function. The key *SK_d* is taken from the bits output by another pseudo random function, using *SKEYSEED*, *Ni*, *Nr*, *SPIi*, and *SPIr* as inputs [7].

[9] If any CERT payloads are included, the first certificate provided will contain the public key required to verify the AUTH field. For symmetric techniques, *CERT* and *CERTREQ* payloads are not required in IKEv2 (see [7]).

AVP; they may optionally contain a Protection-Capability AVP to indicate if link-layer or network-layer encryption should be initiated after PANA/IKEv2. They are also used for binding device identifiers of the PaC and the PAA, via Device-Id AVP, to the PANA/IKEv2 SA established at the end of the authentication phase.

6 Security Analysis

In this section, security threats to the proposed PANA/IKEv2 protocol are considered. The security of the proposed PANA method is based on IKEv2 [7].

6.1 Man-in-the-Middle Attacks

Care has to be taken to avoid man-in-the-middle attacks arising when tunnelling is used, e.g. when using the Protected Extensible Authentication Protocol (PEAP) [18], or when EAP/IKEv2 is part of a sequence of EAP methods. Such vulnerabilities can arise (see, for example, Asokan et al. [13]) even when the authentication protocols used at the various 'levels' are in themselves secure (the man-in-the-middle attack described is taken into account in IKEv2). When such attacks are successfully carried out, the attacker acts as an intermediary between a PaC victim and a legitimate PAA. This allows the attacker to authenticate successfully to the PAA, as well as to obtain access to the network.

As a solution to the problem, Asokan et al. [13] and Puthenkulam et al. [14] suggest to cryptographically bind the session keys of the two phases. This can be achieved by binding together the tunnel session key T (a typical example of T is the TLS master key derived in the TLS handshake of PEAP) and the $KEYMAT$ derived from the EAP/IKEv2 method, to generate an ultimate session key K. There are two ways to achieve the necessary binding between $KEYMAT$ and K. In the first method the binding is established directly by taking $KEYMAT$ in addition to T as input to the computation of the session key K. This provides *implicit authentication* of the PaC. The second method is to make use of a cryptographic check value to verify that the PaC who is in possession of T is also in possession of $KEYMAT$. This second type of binding provides *explicit authentication* of the PaC.

In addition to authentication based on secret key or public key techniques, IKEv2 supports authentication using EAP [9] legacy mechanisms. Using PANA/IKEv2 in these scenarios leads to an outer EAP/IKEv2 exchange transporting an inner EAP legacy method, such as the example provided by Tschofenig and Kroeselberg [6], where EAP/IKEv2 encapsulates an EAP/SIM [19] message flow. For inner EAP legacy methods that create a shared key as a side effect of authentication (e.g. the MSK derived from EAP/SIM), that shared key must be used by both the PaC and PAA to generate an $AUTH$ payload.

Even when tunnelling, an EAP sequence of methods, or EAP legacy mechanisms are not used with PANA/IKEv2, user data need to be integrity protected on physically insecure networks to avoid man-in-the-middle attacks and session hijacking.

6.2 Identity Confidentiality and Integrity Protection

In PANA/IKEv2, a large number of identities are required due to nesting of authentication methods, and due to multiple uses of identifiers for routing (i.e. authentication end point indication). The identifier types and their requirements for confidentiality and integrity protection are as follows.

The identifier *Id*, used in the first round trip of the PANA/IKEv2 authentication phase (*b*), indicates where the EAP messages terminate; it is not used to identify the PaC, and thus it does not allow the adversary to learn the identity of the PaC. The identifiers *IDi* and *IDr* are used respectively to identify the PaC and PAA; *IDi* can be associated with a user identifier (e.g. an email address), and *IDr* can be a fully-qualified domain name (FQDN). Both identifiers are of importance for the PANA/IKEv2 Authorization phase (3), and are thus encrypted and integrity protected by PANA/IKEv2.

The transport of inner EAP legacy methods by PANA/IKEv2 adds further identifiers: the inner EAP identifier (i.e. an *NAI* [20]), and a separate identifier for the selected EAP legacy method (e.g. an *IMSI* [19]). These identifiers are also encrypted and integrity protected by the PANA/IKEv2 SA up to the PANA/IKEv2 endpoint.

In summary, PANA/IKEv2 includes identity confidentiality and integrity protection support, which protects the privacy of the PaC and PAA identities against *passive* (e.g. eavesdropping) and *active* attackers (e.g. impersonation of the access network).

6.3 Mutual Authentication

PANA/IKEv2 provides mutual authentication via the IKEv2 mechanisms. The PaC believes that the PAA is authentic because the network sent a correct *IDr* name, which corresponds to the input used to generate the $AUTH$ value. The PAA believes that the PaC is genuine because the received *IDi* matches the input used to compute the $AUTH$ value. Moreover, PANA/IKEv2 validates the EAP AVP exchanges through its PANA message validity check scheme (Section 4.1.6 of [4]).

6.4 Key Derivation

PANA/IKEv2 supports session key derivation through the EAP/IKEv2 method. It is good security practice to use different keys for different applications. To export an IKEv2 session key as part of an EAP keying framework [17], Tschofenig and Kroeselberg [6] suggest deriving another session key for use with EAP, referred to as the 'Pre-Master-Secret'. They reuse the IKEv2 key derivation function, specified in Section 2.17 of [7], to export a freshly generated $KEYMAT$ as a 'Pre-Master-Secret' for further EAP/IKEV2 key derivation.

6.5 Service Theft and Dictionary Attacks

PANA/IKEv2 does not specify any mechanism for preventing service theft. Therefore an attacker can gain unauthorized access to the network by stealing service from another user, spoofing both the IP and MAC addresses of a legitimate PaC to gain unauthorized access. In a non-shared medium, service theft can be prevented by simple IP address and MAC address filters. In shared links, filters are not sufficient to prevent service theft as they can easily be spoofed (as described by Parthasarathy [21]). A recent draft [22] describes how an IPsec[10] SA can be established to secure the link between the PaC and the EP, which can be used to prevent service theft in the access network.

Because PANA/IKEv2 is not a password protocol, it is not vulnerable to dictionary or social engineering attacks, assuming that the pre-shared secret or the key used for digital signature are not derived from a weak password, name, or other low entropy source.

6.6 Perfect Forward Secrecy, Brute-Force Attacks and Generation of Random Numbers

PANA/IKEv2 generates IKEv2 keying material using an ephemeral Diffie-Hellman exchange, in order to gain the property of "perfect forward secrecy" [7]. Support of this property requires that, when a connection is closed, each endpoint forgets not only the keys used by the connection but any data that could be used to recompute those keys.

The Diffie-Hellman exchange must be based on one of the groups defined in [7], where all but the first of the groups (which is only present for historical reasons) offers security against any computationally feasible brute force attack. It is assumed that all Diffie-Hellman exponents are erased from memory after use.

In the context of the PANA/IKEv2 SA, four cryptographic algorithms are negotiated: an encryption algorithm, an integrity protection algorithm, a Diffie-Hellman group, and a pseudo-random function (prf). The prf is used for the construction of keying material for all of the cryptographic algorithms used. The strength of all IKEv2 keys is limited by the size of the output of the negotiated prf function. For this reason, a prf whose output is shorter than 128 bits (e.g. a CBC-MAC derived using a 64-bit block cipher) shall never be used with the PANA/IKEv2 protocol. Finally, a PANA/IKEv2 implementation also needs to use a good source of randomness to generate the random numbers (nonces) required in the protocol[11].

6.7 Integrity, Replay Protection and Confidentiality

The protection of signaling packet exchanges through the PANA/IKEv2 SA prevents an opponent from acting as a man-in-the-middle adversary, from session

[10] http://www.ietf.org/html.charters/ipsec-charter.html
[11] See [23] for details on generating random numbers for security applications.

hijacking, from injecting packets, from replaying messages, and from modifying the content of the exchanged messages. Also, as with all PANA methods, in PANA/IKEv2 an integrity object is defined, supporting data-origin authentication, replay protection based on sequence numbers, and integrity protection based on a keyed message digest.

Moreover, in PANA/IKEv2 all but the headers of the IKEv2 messages that follow the Diffie-Hellman exchange are encrypted and integrity protected. The recipients must verify that all signatures and MACs are computed correctly, and that the *ID* names correspond to the keys used to generate the *AUTH* payload. The use of nonces guarantees liveliness during an exchange, and also protects against replay attacks.

6.8 Negotiation Attacks and Fast Reconnection

EAP method downgrading attacks might be possible because PANA/IKEv2 does not protect the EAP method negotiation, especially if the user employs the EAP/IKEv2 identifier with other EAP methods. Nevertheless, the EAP document [5] describes a method of avoiding attacks that negotiate the least secure EAP method from among a set. If a particular peer needs to make use of different EAP authentication methods, then distinct identifiers should be employed, each of which identifies exactly one authentication method. In any case, some protection against such an attack can be offered by repeating the list of supported EAP methods protected with the PANA/IKEv2 SA.

PANA/IKEv2 does not support EAP/IKEv2 protocol version negotiation, but supports cipher suite negotiation through IKEv2.

In line with Section 4, PANA/IKEv2 supports two types of fast reconnection. Since fast reconnection does not involve the entire AAA communication, it gives performance benefits.

6.9 Denial-of-service Attacks and Use of Cookies

PANA sequence numbers and cookies provide resistance against blind resource consumption DoS attacks, as described in [4]. But PANA does not protect the EAP/IKEv2 method exchange itself. Since in particular the PAA is not allowed to discard packets, and packets have to be stored or forwarded to an AAA infrastructure, the level of risk of DoS attacks largely depends on the chosen EAP/IKEv2 message flow.

The EAP/IKEv2 method offers an optional DoS protection capability inherited from IKEv2, which also uses cookies and keeps the responder stateless when it receives the first IKEv2 message. If DoS protection is required then an additional round trip is necessary.

It follows that in PANA/IKEv2 there can be at most two levels of cookies: PANA cookie and IKEv2 cookie. Since both cookies are optional, there are theoretically four possibilities:

a) Both PANA and IKEv2 cookies are used,

b) Only PANA cookies are used,
c) Only IKEv2 cookies are used,
d) Cookies are not used by either PANA or IKEv2.

Option *a*) is redundant, and option *d*) should only be employed when the access network is physically secure and there is no risk of DoS attacks.

The PANA/IKEv2 protocol also enables both the PaC and the PAA to transmit a tear-down message [4]. This message causes state removal, a stop to the accounting procedure, and removes the installed packet filters. Thus such a message needs to be protected to prevent an adversary from deleting state information and thereby causing DoS attacks. PANA/IKEV2 supports protected tear-down messages by using a MAC AVP, which neutralizes this threat.

7 Advantages and Disadvantages

In this section, the PANA/IKEv2 proposal is assessed with respect to how well it addresses security issues arising in future heterogeneous network access scenarios supporting ubiquitous mobility. The main advantages of PANA/IKEv2 in this context are as follows.

- PANA/IKEv2 is implemented using PANA, a flexible and scalable network-layer access authentication protocol. Such a protocol is necessary when link-layer authentication mechanisms are either not available or unable to meet the overall requirements, or when multi-layer authentication is needed.
- PANA/IKEv2 also derives from IKEv2, which supports both symmetric and asymmetric mutual authentication, in addition to legacy authentication support, within a single protocol. Because of its greater flexibility, it seems likely that public key authentication will gain greater importance in future ubiquitous mobility access networks.
- PANA/IKEv2 is based on the EAP/IKEv2 method. This method enables the use of the existing IKEv2 infrastructure (e.g. the use of X.509 certificates [12]) in a number of new scenarios; it also enables use of IKEv2 in a transparent way. PANA/IKEv2 also includes identity confidentiality and integrity protection support, has the perfect forward secrecy property, and is not vulnerable to brute-force or dictionary attacks.
- The PANA/IKEv2 SA prevents man-in-the-middle attacks, session hijacking, packet injection, message replay, and content modification of the exchanged packets. The PANA/IKEv2 integrity object supports data-origin authentication, replay protection based on sequence numbers, and integrity protection. The use of nonces guarantees liveliness during an exchange, and also protects against replay attacks.
- PANA/IKEv2 provides ordered delivery of messages with sequence numbers, which along with cookies provides protection against blind DoS attacks. PANA/IKEv2 also offers an optional IKEv2 DoS protection capability.
- PANA/IKEv2 provides confidentiality and integrity protection of the IKEv2 payload, and includes IKEv2 cipher suite negotiation. PANA/IKEv2 also supports two types of fast reconnection, resulting in performance benefits.

The disadvantages of the proposed PANA/IKEv2 protocol are as follows:

- PANA/IKEv2 does not specify any mechanism for supporting EAP/IKEv2 version negotiation.
- PANA/IKEv2 does not specify any mechanism for preventing service theft. On the other hand, because PANA/IKEv2 is just a signalling protocol and does not carry user data traffic, in fact it does not have to formally specify any mechanism for preventing service theft. However, since EAP/IKEv2 has key derivation functionality, it is possible to establish a local IPsec tunnel to provide service theft prevention.

8 Further Work

The session key derivation procedure in the current version of PANA/IKEv2 depends heavily on the EAP/IKEv2 protocol. Therefore one interesting alternative may be to adopt one of the unified EAP session key derivation approaches for multiple applications currently being investigated (see, for example, Salowey and Eronen [24]), instead of adopting the existing scheme from EAP/IKEv2. An analogous scheme to PANA/IKEv2 would be to specify the GPRS GMM authentication protocol [25] as an EAP method (e.g. Buckley et al. [26]), enabling its use with PANA. Another interesting new application would be the transport of EAP Archie (see Walker and Housley [27]) by PANA.

9 Conclusions

"Heterogeneous network access control security" received the highest rating value in the list of open research issues for future mobile communication systems produced by the PAMPAS Project [1, p65]. In this paper, we have proposed the new PANA/IKEv2 protocol, in order to provide an IP compatible, flexible and scalable authentication method that allows a client to be authenticated using either symmetric or asymmetric techniques to an arbitrary access network.

The protocol is based on PANA, a network-layer access authentication protocol carrier, which communicates, via EAP, with an AAA infrastructure. PANA/IKEv2 is also based on EAP-IKEv2, which permits use of the IKEv2 infrastructure in future heterogeneous Internet access scenarios. PANA/IKEv2 prevents man-in-the-middle attacks, session hijacking, packet injection, message replay, content modification, and blind DoS attacks. It provides data-origin authentication, replay protection using sequence numbers and nonces, and integrity protection. As well as supporting identity and IKEv2 payload confidentiality, it allows IKEv2 cipher suite negotiation, and is not vulnerable to brute-force or dictionary attacks.

The performance gains arising from the two types of fast reconnection, the increase in flexibility provided by the public key based authentication option, and the benefits of security given by the PANA/IKEv2 SA make the PANA/IKEv2 scheme potentially attractive to all operators wishing to offer to their users heterogeneous Internet access in ubiquitous mobility networks.

10 Acknowledgements

The authors would like to acknowledge the many helpful insights and corrections provided by Hannes Tschofenig and Yoshihiro Ohba.

References

1. C. Guenther. Pioneering advanced mobile privacy and security (PAMPAS) refined roadmap. Deliverable D03 IST-2001-37763, PAMPAS Project, http://www.pampas.eu.org/, February 2003.
2. Institute of Electrical and Electronics Engineers. *IEEE P802.1aa/D5-2003 DRAFT Standard for Local and Metropolitan Area Networks - Port Based Network Access Control - Amendment 1: Technical and Editorial Corrections*, February 2003.
3. W. Simpson. The point-to-point protocol (PPP). Request For Comments 1661 (STD 51), Internet Engineering Task Force, July 1994.
4. D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin. Protocol for carrying authentication for network access (PANA). Internet draft (work in progress), Internet Engineering Task Force, July 2003.
5. L. Blunk, J. Vollbrecht, B. Aboba, J. Carlson, and H. Levkowetz. Extensible authentication protocol (EAP). Internet draft (work in progress), Internet Engineering Task Force, June 2003.
6. H. Tschofenig and D. Kroeselberg. EAP IKEv2 method. Internet draft (work in progress), Internet Engineering Task Force, June 2003.
7. C. Kaufman (editor). Internet key exchange (IKEv2) protocol. Internet draft (work in progress), Internet Engineering Task Force, May 2003.
8. D. Harkins and D. Carrel. The Internet key exchange (IKE). Request For Comments 2409, Internet Engineering Task Force, November 1998.
9. L. Blunk and J. Vollbrecht. PPP extensible authentication protocol (EAP). Request For Comments 2284, Internet Engineering Task Force, March 1998.
10. R. Perlman. Understanding IKEv2: Tutorial, and rationale for decisions. Internet draft (work in progress), Internet Engineering Task Force, February 2003.
11. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, June 1976.
12. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. Request For Comments 3280, Internet Engineering Task Force, April 2002.
13. N. Asokan, V. Niemi, and K. Nyberg. Man-in-the-middle in tunnelled authentication. In *the Proceedings of the 11th International Workshop on Security Protocols*, Cambridge, UK, April 2003. To be published in the Springer-Verlag LNCS series.
14. J. Puthenkulam, V. Lortz, A. Palekar, D. Simon, and B. Aboba. The compound authentication binding problem. Internet draft (work in progress), Internet Engineering Task Force, October 2002.
15. T. Hiller and G. Zorn. Diameter extensible authentication protocol (EAP) application. Internet draft (work in progress), Internet Engineering Task Force, March 2003.
16. P. Calhoun, S. Farrell, and W. Bulley. Diameter CMS security application. Internet draft (work in progress), Internet Engineering Task Force, March 2002.

17. B. Aboba and D. Simon. EAP keying framework. Internet draft (work in progress), Internet Engineering Task Force, March 2003.
18. A. Palekar, D. Simon, G. Zorn, and S. Josefsson. Protected EAP protocol (PEAP). Internet draft (work in progress), Internet Engineering Task Force, March 2003.
19. H. Haverinen and J. Salowey. EAP SIM authentication. Internet draft (work in progress), Internet Engineering Task Force, February 2003.
20. B. Aboba and M. Beadles. The network access identifier. Request For Comments 2486, Internet Engineering Task Force, January 1999.
21. M. Parthasarathy. PANA threat analysis and security requirements. Internet draft (work in progress), Internet Engineering Task Force, April 2003.
22. M. Parthasarathy. Securing the first hop in PANA using IPsec. Internet draft (work in progress), Internet Engineering Task Force, May 2003.
23. D. Eastlake 3rd, S. Crocker, and J. Schiller. Randomness recommendations for security. Request For Comments 1750, Internet Engineering Task Force, December 1994.
24. J. Salowey and P. Eronen. EAP key derivation for multiple applications. Internet draft (work in progress), Internet Engineering Task Force, June 2003.
25. ETSI. *GSM Technical Specification GSM 04.08 (ETS 300 940): "Digital cellular telecommunication system (Phase 2+); Mobile radio interface layer 3 specification" (version 7.8.0)*. European Telecommunications Standards Institute, June 2000.
26. A. Buckley, P. Satarasinghe, V. Alperovich, J. Puthenkulam, J. Walker, and V. Lortz. EAP SIM GMM authentication. Internet draft (work in progress), Internet Engineering Task Force, August 2002.
27. J. Walker and R. Housley. The EAP Archie protocol. Internet draft (work in progress), Internet Engineering Task Force, February 2003.