Implementation of certificate based authentication in IKEv2 protocol

Ana Kukec, Stjepan Groš, Vlado Glavinić
Faculty of Electrical and Computing Engineering
University of Zagreb
Unska bb, 10000 Zagreb, Croatia
E-Mail: {ana.kukec, stjepan.gros, vlado.glavinic}@fer.hr

Abstract.
IPsec is a security architecture for Internet, which is directly positioned on the top of the IP layer. The major part of IPsec consists of the Internet Key Exchange protocol, now in its version 2. IKEv2 offers authentication, authorization and key agreement services. One of the possible authentication mechanisms in this protocol is based on X509 certificates and the PKI infrastructure. As we are in the process of the IKEv2 protocol implementation, in this paper we describe experiences and design decisions taken during the implementation of the X509 certificate based authentication in the IKEv2 daemon. IPsec is a security architecture for Internet, which is directly positioned on the top of the IP layer. The major part of IPsec consists of the Internet Key Exchange protocol, now in its version 2. IKEv2 offers authentication, authorization and key agreement services. One of the possible authentication mechanisms in this protocol is based on X509 certificates and the PKI infrastructure. As we are in the process of the IKEv2 protocol implementation, in this paper we describe experiences and design decisions taken during the implementation of the X509 certificate based authentication in the IKEv2 daemon.

Keywords.
Security, IPsec, IKEv2, Certificates, PKI

1. Introduction

Internet is the ubiquitous network used for quite a number of different purposes, many of which require some form of security services, what includes confidentiality, integrity, authentication, and authorization. There are many protocols currently used for those purposes, acting with different goals on different layers of the 7 layer ISO/OSI model, and thus, different mechanisms for their implementation, e.g. [2][3][7]. One of the most popular approaches of providing security services is using the IPsec architecture on the top of the IP layer [3]. The IPsec architecture is currently in its second revision, being relatively recently defined and still gaining momentum. This revision is substantially improved as it heavily draws on experiences gained during several years of the deployment, use, and analysis of version 1. IPsec's importance stems from the fact that its implementation is mandatory for IPv6 compliant devices. Furthermore, IPsec is recognized and supported by many other standardization bodies, e.g. 3GPP for use in UMTS devices.

IPsec is actually a framework that defines the external behavior of IPsec compliant nodes. It also defines associated protocols allowing IPsec compliant nodes to transfer data and control messages: ESP [5] and optionally AH [4] for data transfer, as well as IKEv2 [6] for signaling purposes.

During the past year we have been working on an open source IKEv2 protocol implementation [8]. One of the most important feature of this implementation is the ability to use certificate based authentication. In this paper we present our experiences gained during this development process. Since not all details are defined by the relevant specifications, we also describe and justify the major design decisions.

This paper is organized as follows. First, in Section 2 we give an overview of the IKEv2 protocol and of some parts of the IPsec framework necessary to understand the certificate

based authentication. Then in Section 3 we provide an overview of certificate based authentication, describe the specifics of authentication in the IKEv2 protocol along with the design decision we had to make. Section 4 describes the implementation details, while Section 5 concludes the paper.

2. IPsec and IKEv2 overview

The IKEv2 protocol has two main purposes: (i) to establish an IKE security association (IKE SA) between itself and a given peer, and (ii) to establish one or more child security associations (CHILD SA). IKE SA is an encrypted and integrity protected tunnel between two IKEv2 protocol implementations, which is used for both signaling and for carrying out processes like mutual authentication and CHILD SA management (e.g. creation, rekeying, removal). CHILD SA is on the other hand used for transferring data between peers and it is hence the central part of the whole IPsec architecture.

In the rest of the paper abbreviations IKE and IKEv2 will be used interchangeably. Furthermore, the following terms will be used:

- *IKE initiator* and *IKE responder*. An initiator initiates the establishment of IKE SA and the first CHILD SA.
- *Peer*. The partner, i.e. the responder, if we are describing the behavior of the initiator, and v.v. the initiator, if we are describing the behavior of the responder.
- *Exchange*. An exchange consists of sending a request and receiving a response. The protocol is defined through a number of such exchanges.
- *Message*. Either a request or a response. A message has a header (denoted by HDR in the figures) that is never encrypted, followed by one or more payloads.
- *Payload*. The part of the message carrying user data, usually encrypted. There exist multiple payload types (e.g. SAi1, KEi, Ni, CERT, CERTREQ).

The first step in setting up IKE SA is to negotiate keys that will protect all subsequent exchanges, including authentication. Keys are established in a single exchange, called IKE SA INIT exchange (Figure 1). First the IKE

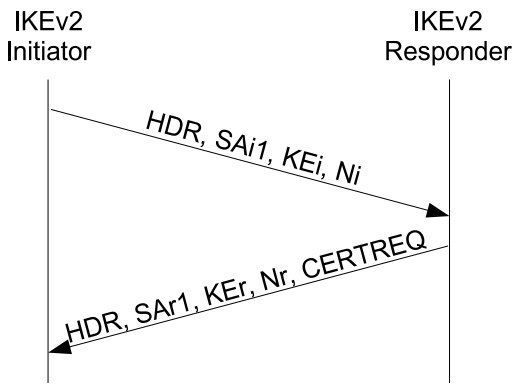

Fig. 1. IKE SA INIT exchange

Initiator sends its proposed encryption, authentication, integrity algorithms, and pseudo-random functions (SAi1). It furthermore sends the keying material for the Diffie-Hellman exchange (KEi) and some random value (Ni). Based on the received request, the responder replies with selected algorithms (SAr), it's own keying material (KEr), a random value (Nr) and the supported certification authorities.

Based on the IKE SA INIT exchange, both peers now have the shared secret that allows them to encrypt and integrity protect all further exchanges. The next exchange, called IKE AUTH (Figure 2), authenticates the peers and establishes the first CHILD SA. During this exchange the peers reveal their identities (IDi and IDr), the initiator presents its list of supported certification authorities (CERTREQ), and certificates (CERT payloads) and the authentication data (AUTH payload) are exchanged. The remaining payloads, which are exchanged in order to create the first CHILD SA include: proposals (SAi2), selected algorithms (SAr2), proposed traffic selectors (TSi, TSr) and narrowed traffic selectors (TSi, TSr flowing from responder to initiator). IKEv2 behaves as specified in [6].

There are three databases that provide the foundation for IPsec: the most important of them, with respect to certificate based authentication, is the Peer Authorization Database (PAD), while the other two are the Security Association Database (SAD) and the Security Policy Database (SPD), respectively. SAD and SPD are implemented in the kernel of almost all of the currently available operating systems

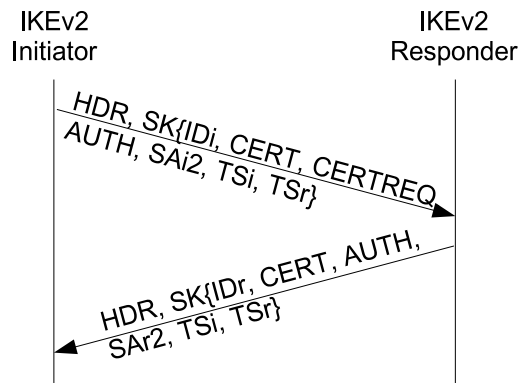

Fig. 2. IKE AUTH exchange

TABLE I
A STANDARD PAD ENTRY

Peer ID:	my.domain
Auth Data:	CA cert for my.domain
Revocation Info:	http://my.crl
SA Constraints:	*@my.domain

on the market, while PAD is a pseudo-database and exists primarily in the user space.

PAD is divided into two distinct parts: (i) an authentication part, and (ii) an authorization part. The former determines exactly which authentication method and authentication material has to be used when verifying the peer's authenticity, while the latter specifies exactly what the authenticated peer with a specific identity is allowed to do. The specification of PAD improves flexibility and offers standardized behavior of every IPsec compliant node. An example of a PAD entry is shown in the Table I.

The authentication part of the PAD entry contains:

- *Peer's ID*. The peer's identifier that it uses to represent himself. There are multiple ID types (e.g. e-mail address, FQDN, IPv4/IPv6 address and Distinguished Name).
- *Authentication data*. Data to be used for authenticating a peer. There are multiple possible types of authentication, and thus multiple types of authentication data, e.g. shared secrets, the end-entity certificate, the certification authority certificate.
- *Information about the revocation material*. OCSP server and public key or pointer to the Certificate Revocation List.

The authorization part of the PAD entry contains *addresses* or *symbolic names* (IDs) that the peer is allowed to use when establishing a designated CHILD SAs.

The PAD entry could store some additional information, e.g. whether to perform the matching between a received ID payload and the corresponding PKIX attribute from the certificate [9].

PAD actually represents the link between IKEv2 and SPD. E.g., when a responder is being authenticated by an initiator, the following three steps are performed:

1) *The authentication.* After the receipt of the responder's identity and the authentication material, the initiator verifies the received authentication material against authentication data from the PAD entry that matches the responder's identity.
2) *The authorization.* In case of the successful verification of the responder's authentication material, the initiator checks in the PAD entry if the responder is allowed to create the designated CHILD SAs. This process is called the SA constraining.
3) *The SPD lookup.* After the authentication (step 1) and authorization (step 2), the initiator finally performs the security policy lookup for the responder. The SPD lookup is called the secure SPD lookup if it includes the check whether the IKEv2 ID matches the identity from the certificate Subject field or certificate SubjectAltName extensions.

3. The certificate based authentication

Besides authentication based on pre-shared keys, the RSA authentication is the most often used authentication method in IKEv2. There are different types of RSA authentication based on different kinds of credentials, e.g. raw RSA keys, DNS Signed Key, PGP Certificate, Kerberos Token, X509 Certificate - Signature, Hash and URL of X509 Certificate. All of the previously mentioned credentials can be exchanged either through out-of-band means or within the CERTREQ/CERT payloads (Figure 1 and 2).

A CERT payload can contain only one X509 certificate, while a CERTREQ payload contains hashes of public keys of multiple CAs. In most cases each peer will have only one X509 certificate, but there are scenarios that require more CERT payloads, e.g. when sending either the certificate chain or the certificate bundle. The certificate chain consists of the end-entity certificate followed by the intermediate CA certificates and the root certificate. The certificate bundle is an ASN.1 sequence of certificates, either as end-entity certificates or as the certificate chain. An example of the certificate bundle is the X509 Certificate - Signature stored in one CERT payload and the X.509 Certificate - Attribute in another, where the X.509 Certificate - Attribute is exchanged to obtain additional authorization information.

Despite of the importance and the advantages of the RSA authentication, the IKEv2 documentation [3][6] does not describe the RSA authentication in sufficient detail for an implementation, what can lead to interoperability problems. Hence, the most important issues and our design decisions are described in the following section.

4. Implementation details

In this section we both describe and justify the implementation of the following parts of the RSA authentication based on X509 certificates:

- CA certificates selection for CERTREQ payload creation,
- determination of the authentication method the peer will use,
- certificates selection for CERT payload creation.
- CERT payload validation.

4.1. The CERTREQ payload creation

This issue is related to the use of the PAD database. IKEv2 documentation [3][6] describes the contents of PAD entries and specifies that each PAD entry is selected based on the peer identity. However there are certain actions that need to be done before it is possible to select the appropriate PAD entry because they are to be performed before the peer reveals its identity in the IKE AUTH message. An example of such an action is the selection of the CERTREQ payload contents. The CERTREQ payload serves as the information to the peer about the sender's preferred certificates. For successful authentication, the CERTREQ payload has to contain at least one CA certificate available to both the initiator and the responder. An improperly created CERTREQ payload leads to the IKE cross-certification problem, which occurs when PAD databases of both peers do not share common CA certificate. Differences in some of the fields of a single CA certificate are allowed, but such certificates must have an identical public key as otherwise it would be impossible to achieve a successful mutual authentication in cases when the CERT payload is formed by basing on the received CERTREQ payload.

Therefore, in our implementation we send CERTREQ payload hashes of all CA certificates stored in the PAD database. With this approach, all the authentication material is in the PAD database hence minimizing the possibility for IKEv2 cross-certification problems caused by improperly created CERTREQ payloads.

4.2. Selection of authentication method

The second issue is again the consequence of the fact that we cannot select an appropriate PAD entry before receiving the IKE AUTH message. Therefore, in the moment when the CERTREQ payload is created (second message in Figure 1), there is no way to know which authentication method the peer will use. Note that IKEv2 [6] specifies that both initiator and responder might use different authentication methods, but it does not specify how to achieve interoperability in such a scenario.

Therefore, in our implementation, we send the CERTREQ payload by default, even if the peer itself will not use RSA authentication. This way the peer is not limited to a restricted set of authentication methods to use. The purpose of the CERTREQ is strictly informational and if the CERTREQ payload is ignored it is not treated as an error. So, the receiver of the CERTREQ payload can either ignore this latter and use a different authentication method or it can use RSA authentication and reply with the CERT payload.

4.3. The CERT payload creation

The third issue is related to the process of selection of certificates to send in the CERT payload. Although IKEv2 does not require the CERTREQ payload to be present, we suggest sending the CERT payload only if we have previously received the CERTREQ payload. The rationale for this decision is: (i) optimal bandwidth usage, (ii) minimal possibility for IKEv2 cross-certification problems, and (iii) minimal information leakage. These benefits are provided by the CERT payload containing only a constrained set of certificates, which are selected basing on CA certificates from the CERTREQ payload.

It should be additionally noted that in the process of certificate selection the optimal bandwidth usage is essential, especially in the scenarios where peers exchange the certificate bundle or the certificate chain. Therefore we suggest that each implementation using CERTREQ/CERT payloads of the X.509 type Certificate - Signature should also use the Hash and URL type of the X509 certificate to shorten the size of the IKE AUTH message. In the case of the Hash and URL type of the X509 certificate, the CERT payload contains the SHA-1 hash of the X509 certificate, followed by the URL. After the receipt of such a CERT payload, the peer downloads the X509 certificate from the given URL and proceeds with the validation of the downloaded authentication material. The Hash and URL type of the X509 certificate gives the opportunity for IKEv2 messages to remain short. This is important since IKEv2 itself does not have a fragmentation mechanism and we cannot rely on the IP fragmentation mechanism, as it is liable to DoS attacks, some NATs/firewalls block the IP fragments, etc.

Although with the CERTREQ payload it is possible to minimize the possibility for IKEv2 cross-certification problems, they cannot be avoided completely. Therefore, besides CERT payload selection based on received CERTREQ payload, we also suggest the possibility for a peer to use the locally stored certificate exchanged either through previous authentication processes or through some out-of-band means. In such cases, neither CERTREQ nor CERT payloads are exchanged, hence there is no possibility for IKEv2 cross-certification problems to appear, as certificates are not selected based on the CERTREQ payload. In cases when such implementation receives the CERT payload, it ignores it and proceeds with the validation of locally stored certificates. The advantage of this solution is optimal bandwidth use as well as a high probability that a constrained number of peers have already been authenticated with their certificates being stored locally. If any IKEv2 cross-certification problems occur in CERTREQ payload based certificate selection, we propose that certificates be exchanged by some out-of-band means.

Based on the previous discussion, we could summarize the RSA X509 certificate authentication process as follows:

1) If an IKEv2 has a certificate stored locally then it does not send the CERTREQ payload hence the peer implementation does not respond with the CERT payload. If IKEv2 receives a CERT payload, it ignores it and proceeds with the validation of the locally stored certificate.
2) If an IKEv2 doesn't have a locally stored certificate, it sends the CERTREQ payload. The peer responds with the CERT payload which is based on the CERTREQ payload. While creating the certificate payload, the priority is as follows: first Hash and URL of the X509 certificates and then X509 Certificate - Signature. The peer should store each received certificate in the CERT payload and repeat step 1, i.e. in the process of reauthentication it should use the stored certificates.
3) If an IKEv2 cross-certification problem occurs, certificates should be exchanged using some out-of-band method, being followed by step 1.

4.4. The validation of CERT payload

After the receipt of the CERT payload, the peer is able to select the appropriate PAD entry that matches the identity received in the ID payload of the IKE AUTH message. Therefore the peer is able to validate the received credentials against the selected certification authorities (CAs). In the process of certificate validation

we used the OpenSSL library[12], also having to overcome some related problems.

Beside lack of documentation, there is another OpenSSL issue that arises in the IKEv2 surrounding: key OpenSSL functions for certificate validity check have to be carefully used. These functions verify a given X509 certificate against both CA certificates and the Certificate Revocation List (CRL). CA certificates can be stored either in the OpenSSL standard CA directory or in the PEM flat file, and each CA directory (or the PEM flat file) must contain only CA certificates from the specific PAD entry that will be used for authentication. This will provide secure verification based on the constrained set of the peer's CA certificates. Additionally, each CA directory or the PEM flat file should contain all of the CA certificates from the specific PAD entry, thus providing effective verification of each certificate in only one function call.

The successful validation of the received authentication material is followed by the SA constraining and the secure symbolic SPD lookup at the end of the authentication process.

5. Conclusion

The strength of RSA authentication is in its practical usage and different features provided by some of the popular principles of cryptographic material exchange, e.g. the Public Key Infrastructure (PKI) for the RSA authentication based on the X509 Certificate - Signature and Hash and URL of the X509 Certificate. There exist a vast number of scenarios for which IKEv2[6] requires the RSA authentication as the mandatory authentication method. In case of the initiator's extensible authentication protocol (EAP), the responder has to use RSA authentication. Unfortunately the respective documentation doesn't specify in enough detail parts of IKEv2 (i.e. those related to X509 certificate based RSA authentication), what can lead to interoperability problems and inefficient implementations.

The above problems can be solved either through the specification of currently unspecified parts of the IKEv2 protocol or through IKEv2 extensions substituting them. In this paper we suggest the former solution, i.e. additions to the currently unspecified parts of IKEv2 related to the RSA authentication based on X509 Certificate - Signature, and Hash and URL of the X509 Certificate types. At this moment it is the only possible solution as the latter one (IKEv2 extensions[11] standardization) will surely take some time to complete, in spite of being the only long-term solution. The deployment of an additional infrastructure such as PKI is expensive and creates new vulnerabilities.

Acknowledgment
This work has been carried out within projects 036-0361994-1995 Universal Middleware Platform for Intelligent e-Learning Systems funded by the Ministry of Science and Technology of the Republic of Croatia, and IKEv2 Step2 project funded by Siemens Networks d.d Zagreb.

References

[1] S. Kent, R. Atkinson, *Security Architecture for the Internet Protocol*, RFC2401, November 1998.
[2] T. Ylonen, C. Lonvick, Ed., *The Secure Shell (SSH) Protocol Architecture*, RFC4251, January 2006.
[3] S. Kent, K. Seo, *Security Architecture for the Internet Protocol*, RFC4301, December 2005.
[4] S. Kent, *IP Authentication Header*, RFC4302, December 2005
[5] S. Kent, *IP Encapsulating Security Payload (ESP)*, RFC4303, December 2005.
[6] C. Kaufman, Ed., *Internet Key Exchange (IKEv2) Protocol*, RFC4306, December 2005.
[7] T. Dierks, E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1*, RFC4306, April 2006.
[8] IKEv2 Daemon, http://ikev2.zemris.fer.hr. February 2007.
[9] Brian Korver, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX*, Work in Progress, draft-ietf-pki4ipsec-ikecert-profile-11, September 2006.
[10] Brian Korver, *Extension for EAP Authentication in IKEv2*, Work in Progress, draft-eronen-ipsec-ikev2-eap-auth-05, June 2006.
[11] Brian Korver, *EAP IKEv2 Method*, Work in Progress, draft-tschofenig-eap-ikev2-12, October 2006.
[12] OpenSSL library, http://www.openssl.org. February 2007.