IKE
Internet Key Exchange

Pekka Riikonen
priikone@iki.fi

1 Introduction

The Internet Key Exchange (IKE) protocol is the main part of the IPSEC implementation, and is used to negotiate secret key material between two parties, called the initiator and responder. The key material is the result of the protocol, and is used to create the Security Associations (SAs) that define how the traffic between the two hosts are to be protected. The IKE also provides mutual authentication by authenticating both of the parties to each other; Both of the parties need to provide a digital signature in the key exchange protocol that the other party will verify. A successful verification means that the other party is authenticated. In order to be able to verify the signature also the public key (certificate) needs to be trusted, and verified. If certificates are not available, the trust can be gained also with a pre-shared-key.

2 IKE Modes

IKE has several modes which define how the actual key exchange procedure is to be done. Two of the most common modes are Main Mode and Aggressive Mode. There are also other modes like Base Mode and New Group Mode, but they are seldom used, and vendors usually does not support them at all.

The difference of Main Mode (MM) and Aggressive Mode (AM) is in length of the procedure. The AM can be completed with only 3 packets, where MM takes 6 packets to complete. Which ever mode is chosen these modes are called Phase-1 modes in IKE.

There is also a Phase-2 mode, called the Quick Mode (QM). The Phase-2 supports only one mode and it is called the Quick Mode. The Quick Mode takes 3 packets to complete.

3 Phase-1 Mode

The IKE negotiation always starts by executing the Phase-1 of the protocol. This section describes the procedure when performing Main Mode. The exam-

ples assume that digital signatures (certificates) are used in authentication.

The purpose of the Phase-1 is to authenticate the peers to each other, and provide protection for the upcoming Phase-2 negotiation. The Phase-1 is used to exchange proposals, vendor specific information, certificates, and it is also used to perform the mutual authentication, which is the main purpose of the Phase-1 negotiation. The result of the Phase-1 can be called in many ways: Phase-1 SA, ISAKMP SA, IKE SA, etc. They all mean the same thing. The Phase-1 SA is also used to protect the actual Phase-2 negotiation, described subsequently. The Phase-1 SA can also be used to protect other informational notifications that may be sent in IKE (such as SA delete notifications).

The Phase-1 Main Mode is performed as follows:

```
First & second packets:

  Initiator                                  Responder
-------------                              -------------
  VID, SA       ------------------------------>              (1)
      <------------------------------      VID, SA           (2)
```

In the first IKE packet the initiator may send zero (0) or more Vendor ID Payloads (VID), that each include a Vendor ID. The Vendor IDs are usually used to tell the responder what kind of extensions the initiator supports. IKE does not provide any other mechanism to tell what extensions initiator supports. The actual data of the Vendor ID is usually a message digest of some ASCII text.

The SA payload is mandatory and it is used to list the security properties the initiator supports. It includes the ciphers, hash algorithms, key lengths, life times, and other information. It is possible to send only one (1) SA payload in Phase-1.

The responder may also send zero (0) or more VID payloads in its reply to the initiator. The responder must also include SA payload in its reply. The SA payload responder sends includes the security properties it selected from the initiator's security property list (the SA payload).

These packets are not encrypted, since there are no key to encrypt them with.

```
Third & fourth packets:

  Initiator                                  Responder
-------------                              -------------
 KE, NONCE, CR     --------------------------->              (3)
      <----------------------------     KE, NONCE, CR       (4)
```

The IKE protocol is based on the Diffie-Hellman key exchange algorithm, which was the first ever invented algorithm that use public key cryptography (in 1974). The third packet is used to exchange the Diffie-Hellman public keys inside a Key Exchange (KE) payload. The Diffie-Hellman public keys are created

automatically every time the Phase-1 negotiation is performed, and they are destroyed automatically after the Phase-1 SA is destroyed.

The responder also sends its Diffie-Hellman public key in the fourth packet to the initiator.

There is also a NONCE payload that is sent in third and fourth packet which is used (with other information) in the IKE to compute the secret data for the Phase-1 SA. The NONCE payload includes random data from random number generator.

The CR payload is a Certificate Request payload and is used to request for certificates by a specific CA. The CR payload includes the name of the CA for which it would like to receive the remote's end entity certificate (peer certificate). If empty CR payload is received it means that it requests any certificate from any CA. The CR payload is usually sent in the third and fourth packets, but it can be sent also in first and second packet. In our example they are sent in third and fourth packet.

These packets are not encrypted, since there are no key to encrypt them with.

```
Fifth & sixth packets:

  Initiator                                  Responder
-------------                              -------------
 ID, CERT, SIG     ---------------------------->              (5)
      <----------------------------     ID, CERT, SIG        (6)
```

The last round of the Phase-1 is fully encrypted, since the key was computed after the third and fourth packets (the Phase-1 SA is created and Diffie-Hellman is computed). The last round is used to send Identification (ID) payload, zero (0) or more Certificate payloads (CERT), which each include one certificate (or CRL), and the Signature payload (SIG) which is the digital signature that the other party must verify.

The ID payload is used to tell the other party who you are, and it also can be used to make policy decisions and to find the certificate of the remote end. The ID may be IP address, FQDN, email address, or something similar.

The CERT payload is optional payload, but usually if it is not sent the result of the IKE is "Authentication Failed" error. The CERT payload includes the sender's end entity certificate, but it is also possible to send CRL inside a CERT payload. The CERT payload is optional because it is possible that the remote end has cached the public key locally, and does not need to receive the CERT payload in the negotiation. Usually implementations do not cache it locally and in this case failing to send CERT payload also causes failure of the IKE negotiation.

The SIG payload includes the digital signature computed with the private key of the corresponding public key (usually sent inside the CERT payload), and provides the authentication to the other party. When both of the parties successfully verify each other's SIG payloads they are then mutually authenticated. They use the public key found in the certificate (usually received in

CERT payload, or some other means (cached locally, fetched from LDAP, etc)) to verify the signature. Before verifying the signature they also verify the certificate of the remote end. They check whether they trust the issuer (Certification Authority, CA) of the certificate, and they verify that the certificate is valid (not revoked, etc).

After these packets are sent and the digital signatures are successfully verified the result of this Phase-1 negotiation is the Phase-1 SA, which can be used to protect other packets sent in the IKE, such as the packets of the Phase-2 negotiation. This also completes the Phase-1 negotiation successfully.

4 Phase-2 Mode

After the Phase-1 is successfully completed the Phase-2 negotiation can proceed. The purpose of the Phase-2 exchange is to provide, and refresh the key material that is used to create the Security Associations (SAs) to protect the actual IP traffic with IPSEC. The Phase-2 exchange is protected with the Phase-1 SA by encrypting the Phase-2 packets with the key material derived from the Phase-1. The Phase-2 also provides proposal list which defines the actual ciphers, HMACs, hash algorithms and other security properties that are used in the protection of the IP traffic. The proposal that was proposed in the Phase-1 is merely for protection of traffic under the Phase-1 SA (like the packets of the Phase-2), and not for the actual IP traffic. The Phase-2, also called the Quick Mode, is for the protection of the IP traffic.

The Phase-2 Quick Mode is performed as follows:

```
First, second and third packets:

  Initiator                                  Responder
-------------                               -------------
 SA, HASH, NONCE, IDi, IDr     ---------------->            (1) (7)
      <----------------------    SA, HASH, NONCE, IDi, IDr  (2) (8)
 HASH            ------------------------------>            (3) (9)
```

The Phase-2 Quick Mode is three (3) packets long and it includes several payloads which relates to the key generation. The HASH and NONCE payloads are the keying material which are exchanged, and they are used to create the new key pair. The NONCE payload includes always random data.

The SA payload is the Phase-2 proposal list which includes the ciphers, HMACs, hash algorithms, life times, key lengths, the IPSEC encapsulation mode (ESP, AH, etc) and other security properties. Note that it is possible to send more than one SA payloads in Phase-2, although usually only one is sent.

The ID payloads, marked here as IDi and IDr, for initiator's ID and responder's ID, respectively, are optional in Phase-2. Usually IKE implementations do send the ID payloads in Phase-2 since they can be easily used to make local policy decisions. However, as noted, they are not mandatory and can be omitted. The IDi is the initiator's ID, usually IP address or similar, and the IDr

is the responder's ID, usually IP address, IP range or IP subnet. Both of the initiator and responder usually use the ID payloads to search the local policy for matching connection. The ID payloads in the Phase-2 are also called as "proxy IDs", "pseudo IDs" or similar, since they do not necessary represent the actual negotiator (for example when Security Gateway (SGW) negotiates on behalf of some client).

After the Phase-2 is completed by sending the last packet, the result of the Phase-2 is two Security Associations (SAs). One is for inbound traffic, the other is for outbound traffic. This also completes the IKE key exchange for basic key exchange.

5 Rekey

The rekey, or key re-generation is a process where new key material is created for protecting the IP traffic. The main cause of rekey in IPSEC is the expiration of the Security Association (SA) which is used to protect the traffic. The time when SA expires is dictated by the life time of the SA, which can be negotiated during the Phase-1 and Phase-2, for Phase-1 SA and Phase-2 SAs, respectively. The rekey is also performed because of security reasons. Longer the same key is used, more insecure the key becomes. Also the risk that an adversary is able to retrieve the old key is an cause of rekey. If the old key is compromised or about to become compromised the rekey is performed.

In IKE the rekey can be performed for both Phase-1 SA and Phase-2 SA. Since the Phase-1 SA is used to protect various notifications and the Phase-2 negotiation, if any of these needs to be sent the Phase-1 SA needs to exist. If it has expired it can be recreated by performing new Phase-1 negotiation.

The Phase-2 SA is used to protect the IP traffic and its rekey procedure is very important. The Phase-2 rekey can be performed either without PFS (Perfect Forward Secrecy) or with PFS.

The PFS is a security property which defines whether the new key is dependent of the old key. When PFS is selected the new key is not dependent of the old key in anyway. If the PFS is not selected the new key is derived from the old key. The PFS is important security property since the security of the past and future keys depends on one key. If the PFS is not selected, then new key is always dependent of some old key. If the old key is compromised at any time in the future, it is also possible that all new keys may be compromised. Therefore, an adversary needs to find out only one key to be able to find out all keys. On the other hand, when the PFS is selected the new key is never dependent of the old key, and compromising some old key (or some new key) in the future does not compromise any other key.

The downside of the PFS is that it consumes more computing power since it requires to do additional Diffie-Hellman key exchange. However, nowadays this should not be a problem or reason not to use PFS.

When the Phase-2 rekey is performed without PFS the rekey process simply performs the Phase-2 Quick Mode exchange, described earlier. The result of the

Phase-2 is new pair of inbound and outbound SAs which can be used to protect the IP traffic. In the generation of the new key some parts of the old key are used, and for this reason the new key becomes dependent of the old key.

When the Phase-2 rekey is performed with PFS the rekey process is a bit different from rekey without the PFS. When PFS is selected the Phase-2 Quick Mode negotiation includes Key Exchange (KE) payload in the first and second packets of Phase-2. The KE payload in Phase-2 has the same purpose as the KE payload in Phase-1. They exchange the Diffie-Hellman public keys and use the result of the Diffie-Hellman computation as the basis of the new key material.

After the new Phase-2 SA pair has been created the old SA pair is removed from the use by sending a SA delete notification for the old SAs. After the notification, all traffic is protected with the new SA pair.

6 IKE version 2 - The Future of IKE

The current IKE protocol version is the version 1 and was released as an IETF standard in the year 1998. Several Internet Drafts have existed to fix security issues found in the IKE specification, and to clear some of the issues related to implementation. The IETF effort to come up with better and simpler version of the IKE has resulted into development of the IKEv2. Currently one Internet Draft exists and is still in preliminary phase. The problem of the current IKE version is that the protocol is very complex and the specifications has been spread out to several RFCs which makes the implementation very difficult to do. The purpose of the IKEv2 is to remove the unnecessary parts of the IKE and bring the entire IKE specification into one document. For example, the Aggressive Mode is removed from the IKEv2, and the latency of the protocol is decreased by making the exchange (the old Main Mode) only 4 packets long, and the Quick Mode only 2 packets long.

7 Further Reading

Recommended reading is the IKE protocol specification to know more details about the internals of the IKE protocol. All IKE protocols specifications are freely available from the IETF (http://www.ietf.org/) website. The following RFCs describe the IKE protocol in detail:

1. RFC 2409 (http://www.ietf.org/rfc/rfc2409.txt)
2. RFC 2408 (http://www.ietf.org/rfc/rfc2408.txt)
3. RFC 2407 (http://www.ietf.org/rfc/rfc2407.txt)
4. RFC 2412 (http://www.ietf.org/rfc/rfc2412.txt)