The Internet Key Exchange

Kati Tuomainen
49066K
Helsinki University of Technology, Espoo

This paper will describe the Internet Key Exchange (IKE) protocol, the key management protocol for the IP Security Protocol (IPsec). It will discuss the need for a key management protocol for IPsec and the properties required from a key management protocol. The paper will focus on the basics of IKE from the viewpoint of a regular user, not from the mathematical viewpoint, nor will it go into detail about the generation of keys. The paper will also discuss some of the proposed improvements to the original IKE protocol and give a short overview of the proposed successors to IKE.

1. INTRODUCTION

1.1 Internet Key Exchange (IKE) Background

Internet Key Exchange, or IKE, is a protocol for establishing security associations (SA) between parties communicating with each other using the IP Security protocol (IPsec). A security association is a one way "connection" providing security to the traffic it is carrying [11].

IPsec is a protocol providing confidentiality, data integrity, and data source authentication to IP datagrams at the IP layer [11]. These services are provided with the Encapsulated Security Payload (ESP)-and Authentication header (AH) extension headers for the IP, which are defined in [10] and [9].

IKE, as well as IPsec, is a standard produced as a part of the work of the IP Security Protocol Working Group of IETF, or Internet Engineering Task Force. IETF is an open organization developing the architecture and operation of the Internet.

The IP Security Protocol Working Group is developing security services to the Internet Protocol. These services include authentication, integrity, access control, and confidentiality. The goal is to provide them with cryptographic security, develop IPsec so that the above mentioned services can be flexibly combined according to the client protocols needs and improve the documentation of the protocols that provide these services [8].

In this paper we discuss the background which led to the development of IKE and cover the basics of the protocol. We'll not go into the mathematics used in IKE, nor to the protocols using IKE, but rather provide an overview of the IKE protocol.

We will also look into the future development of IKE, give a short overview of IKEv2 and Just Fast Keying (JFK), another proposed successor to IKE, and see what other changes are needed in IKE to meet future needs.

In chapter 1 we introduce the Internet Key Exchange (IKE) protocol and some related concepts as well as give an overview and an outline of the article.

Chapter 2 goes more deeply into the backgrounds of IKE, discussing both the need for IKE and the structure of IKE.

Chapter 3 discusses the methods used in IKE to assure security, chapter 4 looks into the future of IKE. In chapter 5 we discuss the conclusions which can be made from this paper.

2. IKE BACKGROUND

In this chapter we discuss why IKE was designed, and introduce the different protocols used in IKE.

2.1 Why Is IKE Needed?

IPsec supports both manual and automated key management, but some services, like the anti-replay features of AH and ESP, or on-demand creation of SAs (creating SAs for services when needed) require automatic key management [11].

The anti-replay service, or partial sequence integrity service, helps to counter denial of service attacks. The anti-replay service is enabled by default, unless the receiver notifies the initiator during the SA negotiations, that it doesn't enable the service. When the anti-replay service is used, the transmitted Sequence Number must never cycle. That is, both parties must reset their counters by establishing a new SA and a new key before the transmission of the 2^{32}nd packet on an SA [9].

Because of the need for automated key management, IKE was designed. It negotiates and provides the authenticated keying material in a secure manner for the SAs [4]. IKE, however, is considered to be too complex, which is why there is an effort to find a replacement to

Author's contact information: khtuomai@cc.hut.fi

Published in *Internet Protocols for Mobile Computing - Seminar on Internetworking, Autumn 2002*, by Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory. The complete publication can be found at http://www.tml.hut.fi/Studies/T-110.551/2002/papers/December/index.html.
ISBN 951-22-6272-X - ISSN 1455-9749 - HUT - TML - TML-C9

it. The possible replacements are discussed in section 4.

2.2 The Structure of IKE

IKE is a hybrid protocol consisting of parts of ISAKMP, Oakley and SKEME [4]. In the following chapters we introduce these protocols and their use in IKE.

2.2.1 *ISAKMP.* ISAKMP, the Internet Security Association and Key Management Protocol, is a protocol that defines the procedures and packet formats needed in establishing, negotiating, modifying and deleting SAs. The formats and procedures defined by ISAKMP provide a consistent framework for transferring key and authentication data independent from the key generation techniques, encryption algorithms or authentication mechanisms used [16].

In IKE ISAKMP is used as the framework for the exchanges discussed in section 3, and IKE exchanges use the standard ISAKMP payload syntax, attribute encoding, timeouts and retransmissions of messages, and informational messages, or messages which notify the receiver, when, for example, a signature verification or decryption was unsuccessful [4].

2.2.2 *Oakley.* Oakley is a key exchange protocol by which two authenticated parties can agree on secure and secret keying material. It is designed to be a compatible component of the ISAKMP protocol, and it supports Perfect Forward Secrecy, user-defined abstract group structures, key updates, and incorporation of keys distributed via out-of-band mechanisms [17].

Perfect Forward Secrecy (PFS) means, that the key exchange mechanism protects short-lived keys from compromise even after the long-lived keys have been exposed. Or in other words the compromise of a single key will permit access only to data protected by that single key. To achieve PFS, the key used to protect a data transmission mustn't be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material mustn't be used to derive any more keys [15; 4].

Oakley allows the users to flexibly choose the features that are best suited to their security and performance requirements. The users can choose from different options for distributing keys, which of the anti-clogging and perfect forward secrecy features to use, and the algorithm on which the authentication is based. The different combinations of these options form different modes [17].

IKE implements only the subset of Oakley necessary for reaching its goals, and the implementation is not dependent on Oakley. There are four groups for doing the Diffie-Hellman exchange used in IKE that are based on Oakley [4].

2.2.3 *SKEME.* SKEME is also a key exchange protocol like Oakley, and likewise flexible and scalable. There are four modes in SKEME: a basic mode providing a public key based exchange and Perfect Forward Secrecy through the Diffie-Hellman exchange, a public key based exchange without the Diffie-Hellman exchange, a key exchange based on a previously shared key and providing PFS, and a re-keying mechanism based on symmetric key techniques [15].

The Diffie-Hellman exchange means the exchange of a secret key obtained with the Diffie-Hellman key agreement protocol, which is explained, for example, in [18].

As with Oakley, IKE doesn't implement the entire SKEME protocol, nor is IKE dependent on SKEME. From SKEME the IKE protocol implements the method of public key encryption for authentication and its concept of fast re-keying using an exchange of nonces[1] [4].

3. METHODS USED IN IKE FOR ASSURING SECURITY

In IKE different exchanges are represented as modes which operate in one of two phases [4]. Below we discuss the different phases and modes based on their definition in [4].

3.1 Phases

In phase 1 two ISAKMP peers establish a secure, authenticated channel over which they communicate thus creating an ISAKMP Security Association (SA). An ISAKMP SA is bidirectional and once it is established, either party may initiate an exchange. A phase 1 negotiation may be used for more than one phase 2 negotiations.

In the ISAKMP SA negotiations of phase 1 the parties must agree on the encryption algorithm, hash algorithm, authentication method, and the information about a group over which to do the Diffie-Hellman exchange. They can also agree on a pseudo-random function, but if the don't negotiate on it, the HMAC version of the negotiated hash algorithm is used as a pseudo-random function.

HMAC, or Keyed-Hashing for Message Authentication, is a mechanism for message authentication using cryptographic hash functions, which can be used with any iterated cryptographic hash function. HMAC is defined in [14].

In phase 2 SA:s are negotiated on behalf of services in need of key material and/or parameter negotiation. Because a phase 2 SA is negotiated on behalf of a service, IKE doesn't place any requirements on what must be agreed upon during the negotiation, because that depends on the service for which the SA is negotiated. A phase 2 negotiation can request multiple Security Associations in one negotiation.

3.2 Modes

The modes used in IKE are called Main Mode, Aggressive Mode, Quick Mode, and New Group Mode. All the exchanges of IKE have a fixed number of messages depending on the mode used, and exchange types can't be switched during an exchange.

Main Mode is a basic phase 1 method for establishing an authenticated key exchange and it must be implemented in an IKE implementation. Main Mode is an instantiation of the ISAKMP Identity Protect Exchange. The first two messages of Main Mode negotiate policy, the next two exchange Diffie-Hellman public values and

[1] a nonce is a parameter varying with time, for example a timestamp

auxiliary data, for example nonces, necessary for the exchange, and the last two messages authenticate the Diffie-Hellman Exchange.

Aggressive Mode, also a phase 1 method, in its turn, is an instantiation of the ISAKMP Aggressive Exchange. In Aggressive Mode, the first two messages negotiate policy, exchange Diffie-Hellman public values and auxiliary data necessary for the exchange, as well as identities. The second message also authenticates the responder. The initiator is authenticated in the third message, which also provides a proof of participation in the exchange. This final message is not sent in an ISAKMP SA, because that allows each party to postpone the exponentiation, if they wish to, until negotiation of this exchange is complete.

Aggressive Mode is another method for establishing an authenticated key exchange, but unlike Main Mode, it is not required in an implementation of IKE, only recommended. In Aggressive Mode the SA negotiation is limited, because due to the wy the messages are constructed, the Diffie-Hellman group can't be negotiated, and the choice of authentication method may also limit the attributes that are negotiable. Therefore, in situations, where rich attribute negotiation is required, it might be advisable to use Main Mode.

Quick Mode, a phase 2 mode, is in itself not a complete exchange, since it is bound to a phase 1 exchange, but its implementation is required. The information exchanged with Quick Mode must be protected by an ISAKMP SA (negotiated in phase 1) or, in other words, apart from the the ISAKMP header all the payloads should be encrypted.

Quick Mode is used as part of the phase 2 SA negotiation process to derive keying material and negotiate a shared policy for those SAs, which are not ISAKMP SAs. It is also a mechanism to refresh keying material.

New Group Mode is a mechanism for defining private groups for Diffie-Hellman exchanges, and its implementation is recommended. Before the New Group Mode can be used an ISAKMP SA must be established, because though New Group Mode isn't a phase 2 exchange, it can only follow a phase 1 negotiation to guarantee, that the new group defined with it stays private.

New groups may also be directly negotiated in the SA proposal with Main Mode. Then the component parts required for the creation of the group are passed as SA attributes. But using New Group Mode the nature of the group can be hidden. Then only the group identifier is passed as cleartext during the phase 1 negotiation.

3.3 Authentication Methods Used in IKE

The authentication methods used in IKE include digital signatures, public key encryption, and pre-shared key. They can be used with either Main Mode or Aggressive Mode. If not otherwise noted, the following is based on [4].

3.3.1 *Digital Signatures.* Digital signature is an encrypted hash of the message send. The hash is a fixed-size string obtained from the message with a hash function (see [18] for more). The hash is encrypted using the private key of the sender, or with a key the parties have agreed on earlier. The recipient can then decrypt the signature with the senders public key (or the prearranged key) and check that the hash obtained from the message matches the hash obtained by decrypting the signature.

When digital signatures are used in IKE, the auxiliary information exchanged in the second messages is nonces. The exchange is authenticated by signing a mutually obtainable hash with the digital signature algorithm negotiated in the second messages.

3.3.2 *Public Key Encryption.* When public key encryption is used, the initiator must already have the responder's public key. If the responder has multiple public keys, a hash of the key the initiator is using to encrypt the information is passed as a part of the third message, so the responder knows which private key to use for the decryption.

In IKE, using public key encryption means that encrypted nonces are passed as the auxiliary information (of the second messages), as well as the encrypted identities of the parties. The encryption is done with the receivers public key. Each party is then able to reconstruct a hash, thus proving that the other party decrypted the nonce, and the reconstructed hash authenticates the exchange.

In Aggressive Mode authentication with public key encryption also enables identity protection.

There is also another revised public key authentication mode, which retains the advantages of authentication using public key encryption but does it using less public key operations. In this mode, the nonce is still encrypted using the public key of the recipient, but the identities (and the key, if sent) are encrypted using the symmetric encryption algorithm, negotiated in the SA negotiations, with a key derived from the nonce.

This revised public key authentication mode adds minimal complexity and state, and saves two public key operations on each side. In this mode, the Key Exchange payload is also encrypted using the same derived key, which provides additional protection against cryptoanalysis of the Diffie-Hellman exchange.

3.3.3 *Pre-Shared Keys.* The use of a pre-shared key is also possible with IKE. When pre-shared key authentication is used with Main Mode the key can only be identified by the IP address of the parties, because the initiator must compute the hash of the message before it has processed the identifications. In Aggressive Mode you can use a wider range of identifiers of the pre-shared key. Aggressive Mode also lets two parties maintain multiple, different pre-shared keys and identify the correct key for a particular exchange.

4. FUTURE DEVELOPMENT

IKE is viewed to be, among other things, too complex, vulnerable to denial-of-service attacks, and too heavy, because of the high number of rounds performed. The complexity has, for example, lead to non-interoperational implementations [1]. Therefore the IPSEC working group has done a lot of work to find a replacement to IKE that would solve its problems.

The successor to IKE should in addition to being simpler, enabling authentication between two parties and establishing cryptographic keys for integrity and encryp-

tion of an IPsec SA, also include identity hiding, cheap and elegant rekeying, dead peer detection, plausible deniability, support for multiple services located at one IP address, negotiation of peer-dependent IPsec policies, and a two-phase structure enabling inexpensive creation of multiple SAs between the same two hosts or security gateways [6]. The two proposed successors are introduced in sections 4.1 and 4.2. For a more thorough discussion about their features and differences see [1; 5; 7].

In addition to trying to find a successor to IKE, the IPsec working group has also proposed other improvements to IKE. We'll explore them shortly in sections 4.3 and 4.4.

4.1 IKEv2

IKEv2, one of the proposed successors to IKE is based on IKEv1 (but it is not backwards compatible with it [6]). IKEv2 simplifies IKE by replacing all the possible phase 1 exchanges with a single exchange that is based on either public signature keys or shared secret keys. This exchange provides identity hiding, and it works in two round trips, while in IKEv1 all the identity hiding exchanges required three round trips [5].

Because IKEv2 allows the setup of an SA for ESP, AH, and/or IPcomp to be piggybacked on the initial IKE exchange, the latency for the setup of an IPsec SA is also reduced. The number of messages exchanged between the parties in the initial exchange is reduced to four, which also reduces the latency [5].

IKEv2 also improves the security and robustness of the protocol, because it allows the responder to be stateless until it can be sure that the initiator can receive at the claimed IP address and it can be authenticated. In IKEv2, all the messages are acknowledged and sequenced, which not only improves its robustness but also makes it possible to reduce the number of messages exchanged in phase 2 to two from the original three [5].

The protocol also simplifies IKE by:

(1) removing some fields from the IKE specification
(2) replacing the negotiation of cryptographic algorithms with proposals based on suites of algorithms.
(3) specifying required behavior under certain error conditions
(4) simplifying and clarifying the way shared state is maintained when there are network failures or Denial of Service attacks [5]
(5) improving and clarifying the documentation

4.2 Just Fast Keying

Just Fast Keying (JFK) is another proposed successor to IKE. It is not based on IKE, but is an entirely new proposal. By abandoning the approach used in IKE the developers of JFK claim to be able to define a better protocol. This introduction of the protocol is based on [1].

JFK was designed to be simple, efficient, secure, and resistant to Denial-of-Service attacks.

Simplicity makes the protocol more efficient, and makes it easier to both define as well as implement the protocol correctly. JFK achieves simplicity by using only one possible phase instead of the two phases of IKE, and fixing the number of rounds of an exchange to two. For efficiency and simplicity reasons JFK doesn't use negotiations.

Because Denial-of-Service attacks have increased in the Internet, resistance to them has become more important. JFK is resistant to both Memory-DoS and Computation-DoS attacks. Partially to help the protocol to resist these attacks, JFK only provides an imperfect form of PFS. This, in this case, means that the exponentials used in the Diffie-Hellman exchange may be reused as often as necessary [1; 7].

JFK has two variants, JFKi and JFKr. JFKi provides active identity protection for the initiator but none for the responder, JFKr provides active identity protection for the responder and passive identity protection for the initiator. JFKi also contains an optional signature.

As mentioned earlier, JFK has four exchange messages:

(1) the first message is sent by the initiator and it contains Diffie-Hellman exponentials.
(2) in the second message contains the responders exponentials and the authentication of the initiator (or a rejection message, if the initiators exponentials aren't acceptable)
(3) in the third message the initiator echoes content from the responders message, including the authenticator
(4) the fourth message contains responders application specific data and a signature on both nonces, both exponentials, and the Initiator's identity, encrypted with a key derived from the two nonces (or it can reject the third message, but no rejection message is sent)

In JFKr, the parties obtain a shared encryption key before sending their identities encrypted with this key.

4.3 NAT Traversal

NAT traversal means the ability to cross a firewall (or a machine) that handles the Network Address Translation (NAT) for the network behind it. In NAT, the IP addresses of the network behind the firewall aren't passed to the Internet, but the firewall maps the local addresses to a global IP address (usually its own), and then unmaps the incoming packets to the correct local IP address. See [3] for more about NATs. The following overview is based on [12].

To make NAT traversal possible, both parties need to know there is a NAT between them and support NAT traversal. The recipient must also be able to process IKE packets with source port different from 500 (because the NAT floats ports), and reply back to the source address from the packet. Also, when the original responder is doing rekeying etc., it must send the packets from the same port and IP address it used the last time the IKE SA was used.

NAT traversal support is communicated using vendor strings in the first two messages of phase 1, and both parties need to send and recieve one.

The presence of a NAT is detected during the phase 1 negotiations. This is done using the NAT-D payload. It detects the NAT and its location. The location of the

NAT is important, because you need to know that the keepalives come from behind the NAT.

The detection is done by sending the hashes of IP address and port number of both source and destination addresses from each party to another. When both parties calculate those hashes and get the same result, there is no NAT in between. If the hashes do not match, there is a NAT between the parties.

The decision of using the NAT-Traversal is left to Quick Mode, and is negotiated inside the SA payloads of it.

In Quick Mode, both ends can (optionally) send the original source addresses of the packets (in transport mode, see [11]) to the other end, giving the other end a possibility to fix the TCP/IP checksum field after the NAT transform. There is no use in sending the source addresses (and they should not be sent) in the UDP-Encapsulated-Tunnel mode, because it encapsulates an IP header inside the UDP-Encapsulated packet.

4.4 Other Improvements

There have also been other minor improvements to IKE. They have added more ECC groups [2] and MODP Diffie-Hellman groups [13] over which to do the key exchange. These new groups are also stronger (harder to crack with attacks).

5. CONCLUSIONS

This paper gives an overview of IKE and discusses its future development.

The IPsec working group has done a lot of work on the security protocols in use in the Internet. Their next major decision is to find IKE a successor.

The importance of an automated key management protocol has grown because the use of services providing secure communication channels has increased. But today the devices using these services often have, for example, limited memory capacities. Therefore it is important to find a more efficient key management protocol to succeed IKE.

Both JFK and IKE are viable options to be successors to IKE. They are in many respects similar to each other though they were developed from a slightly different perspective. For example, both protocols choose a Diffie-Hellman exponential and compute the Diffie-Hellman shared secret. Although similar, they also differ, for instance in the way they prevent different attacks, provide PFS, and in the way they agree on keys, such as how many rounds are used and how much computation is needed. But these differences don't clearly show that one candidate would be a better choice than the other, and the choice comes down to which features are more important than others.

The existence of these two proposals does not mean that the choice of the successor to IKE is limited to them alone or that these proposals couldn't be improved. If the IPsec working group feels that neither of them satisfy the requirements for the successor of IKE, they can also discard both JFK and IKEv2 and design a new protocol. However, the successor to IKE will most likely be either IKEv2 or JFK.

REFERENCES

[1] Aiello, W., Bellovin, S.M., Blaze, M., Canetti, R., Ioannidis, J., Keromytis, A.D, Reingold, O., Just Fast Keying (JFK). Online, referred to 5 October 2002. URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-jfk-04.txt

[2] Blake-Wilson, S., Brown, D., Poeluev, Y., Salter, M., Additional ECC Groups For IKE. Online. 23 July 2002, referred to 5 October 2002. URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ike-ecc-groups-04.txt

[3] Egevang, K., Francis, P., RFC 1631: The IP Network Address Translator (NAT). Online. May 1994, referred to 31 October 2002. URL: http://www.ietf.org/rfc/rfc1631.txt

[4] Harkins, D. and Carrel, D., RFC 2409: The Internet Key Exchange (IKE). Online. November 1998, referred to 5 October 2002. URL: http://www.ietf.org/rfc/rfc2409.txt

[5] Harkins, D., Kaufman, C., Kent, S., Kivinen, T., Perlman, R., Proposal for the IKEv2 Protocol. Online. April 2002, referred to 5 October 2002. URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-02.txt

[6] Harkins, D., Kaufman, C., Kivinen, T., Kent, S., Perlman, R., Design Rationale for IKEv2. Online. February 2002, referred to 5 October 2002. URL: http://www.ietf.org/internet-drafts/ draft-ietf-ipsec-ikev2-rationale-00.txt

[7] Hoffman, P (editor), Features of Proposed Successors to IKE Online. 31 May 2002, referred to 31 October 2002. URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-soi-features-01.txt

[8] The IP Security Protocol (ipsec) Working Group, IP Security Protocol (ipsec). Online. 1 October 2002, referred to 5 October 2002. URL: http://www.ietf.org/html.charters/ipsec-charter.html

[9] Kent, S., Atkinson, R., RFC 2402: IP Authentication Header. Online. November 1998, referred to 28 October 2002. URL: http://www.ietf.org/rfc/rfc2402.txt

[10] Kent, S., Atkinson, R., RFC 2406: IP Encapsulating Security Payload (ESP). Online. November 1998, referred to 28 October 2002. URL: http://www.ietf.org/rfc/rfc2406.txt

[11] Kent, S., Atkinson, R., RFC 2401: Security Architecture for the Internet Protocol. Online. November 1998, referred to 5 October 2002. URL: http://www.ietf.org/rfc/rfc2401.txt

[12] Kivinen, T., Huttunen, A., Swander, B., Volpe, V., Negotiation of NAT-Traversal in the IKE. Online. 24 June 2002, referred to 5 October 2002. URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-t-ike-03.txt

[13] Kivinen, T., Kojo, M., More MODP Diffie-Hellman groups for IKE. Online. 13 December 2001, referred to 5 October 2002. URL: http://www.ietf.org/internet-drafts/ draft-ietf-ipsec-ike-modp-groups-04.txt

[14] Krawczyk, H., Bellare, M., Canetti, R., RFC 2104: HMAC: Keyed-Hashing for Message Authentication. Online. February 1997, referred to 30 October 2002. URL: http://www.ietf.org/rfc/rfc2104.txt

[15] Krawczyk, H., SKEME: A Versatile Secure Key Exchange Mechanism for Internet. from IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security. Online, referred to 5 October 2002. URL: www.research.ibm.com/security/skeme.ps

[16] Maughan, D., Schertler, M., Schneider, M., Turner, J., RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP). Online. November 1998, referred to 5 October 2002. URL: http://www.ietf.org/rfc/rfc2408.txt

[17] Orman, H., RFC 2412: The OAKLEY Key Determination Protocol. Online. November 1998, referred to 5 October 2002. URL: http://www.ietf.org/rfc/rfc2412.txt

[18] RSA Laboratories, RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1. RSA Security Inc., 2000. Online, referred to 31 October 2002. URL: http://www.rsasecurity.com/rsalabs/faq/