Automatic Configuration Method for the IKE Protocol based on X.509*

Zhen Zhao, Kwang Sun Ko, and Young Ik Eom
School of Information and Communication Engineering, Sungkyunkwan University
email: {capzz, rilla91, yieom}@ece.skku.ac.kr

Abstract

The Internet Key Exchange (IKE) protocol is most widely used as a security key exchange protocol on the Internet. Security policies used by the IKE protocol must be configured in advance, however the complex options and manual settings cause inconvenience. This paper proposes an automatic configuration method for the IKE protocol based on X.509 certificate. Security policies are embedded in the certificate, read, and added into the IKE configuration file by a negotiation assistant module in order to achieve automatic IKE configuration. Our proposed method reduces the complexity of configuration process and improves the adaptability of the IKE protocol.

1. Introduction

The IKE protocol uses two-phase negotiation to establish a security association (SA), derive a secure key, renew the key material, and send the error messages or status information. However the IKE configuration and security policy maintenance have to be done manually [1]. This is an obstacle for IKE management.

We propose an automatic configuration method for the IKE protocol based on X.509. A negotiation assistant module is designed to process security policies embedded X.509 certificate in order to achieve automatic configuration for the IKE protocol and efficient security policy management.

In the second section we give a review of the IKE protocol, outline the problems of the current system, and introduce the X.509 certificate infrastructure. Our proposed method is presented in section 3 with detailed description. We also discuss the implementation foreground of this method. And the conclusion comes in the last.

2. Background

We introduce the key point of the IKE protocol in this section, and show the existing problem with IKE configuration process. We also give a review to the X.509 certificate.

* **This research is supported by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy(MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-S1-10M**

2.1The IKE protocol

The IKE protocol is a hybrid protocol. It consists with the Internet Security Association and Key Management Protocol (ISAKMP), the secure key exchange protocol OAKLEY, and SKEME. The IKE protocol is created on the framework that defined by the ISAKMP protocol, continues to use the OAKLEY protocol's security key exchange mode and the SKEME protocol's key renewing and sharing technique [1].

The purpose of the IKE protocol is to allow hosts to exchange information required for secure communication. Particularly, the cryptographic keys which are used for encoding authentication information, performing payload encryption, and the SA for IPSec are negotiated by the IKE protocol.

The IKE protocol operates in two phases, for each phase, it performs the following operations:

<Table 1> Operations of the IKE 2 phases

Phase	Operation
1	Authenticates and protects the identities of the IPSec hosts
	Negotiates a matching IKE SA policy between hosts to protect the IKE exchange
	Performs an authenticated Diffie-Hellman exchange of having matching shared secret keys.
	Set up a secure channel to negotiate IKE phase 2 parameters
2	Negotiate IPSec SA parameters protected by an existing IKE SA
	Establishes IPSec SAs
	Periodically renegotiates IPSec SAs
	Optionally performs an additional Diffie-Hellman exchange

The IKE negotiation is in fact of making agreement for security policies that used in the IKE SA or IPSec SA. For phase 1 and phase 2, there are some different types of parameters of security policies need to be negotiated by both hosts (Table 2).

<Table 2 > Security policy parameters

Phase	Parameters
1	Encryption algorithm
	Hash algorithm
	Authentication method
	Diffie-Hellman group identifier
	Security association's lifetime
	Other necessary attributes
2	AH sub-proposals
	ESP sub-proposals
	IP Compose sub-proposals
	Other necessary attributes

In current IKE implementations, security policies must be configured manually. The parameters that compose security policies should be considered one by one on each host. As shown in Table 2, there are several single parameters and several composed proposals need to be configured. It causes inconvenience of using IKE.

2.2 X.509 certificate

A public key certificate is a data structure that binds public key values to subjects [2]. The use of digital certificates ensures the confidentiality of transmission, the integrity of data exchange, and the non-repudiation of information and validation of the entities. Figure 1 shows the structure of X.509 certificate.

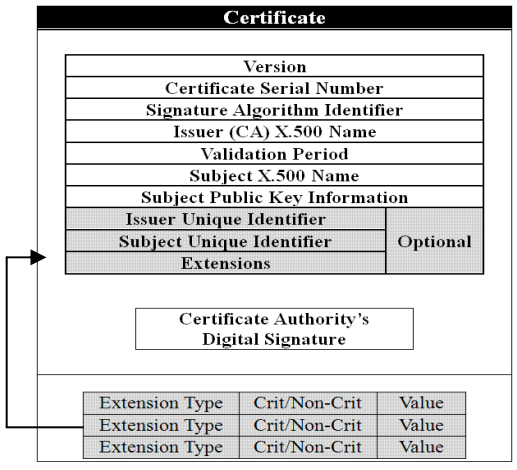

Fig.1 X.509 v3 certificate structure

We use the Extensions field to embed security policies into the certificate. X.509 v3 allows the use of private extensions to carry information that is unique to the subject [3]. Also, the X.509 certificate supports the hierarchical trust, in which is inherited, this characteristic helps to make use of the certificate in a hierarchical topology computer network. The method will be introduced in the later section.

3. Automatic configuration method for the IKE

We adopt the mechanism to combine security policies with X.509 certificate, thereby reducing the burden of manual preparation of security policies, and improving the success rate of the IKE negotiation [4][5]. In this section, we specify the method, negotiation assistant module, and policy version validation, achieving the mechanism.

3.1 Negotiation assistant module

The negotiation assistant module is an extension of the IKE implementation. It takes the charge of communicating with the CA, parsing security policies in the certificate to maintain the IKE configuration file, managing the policy version validation messages.

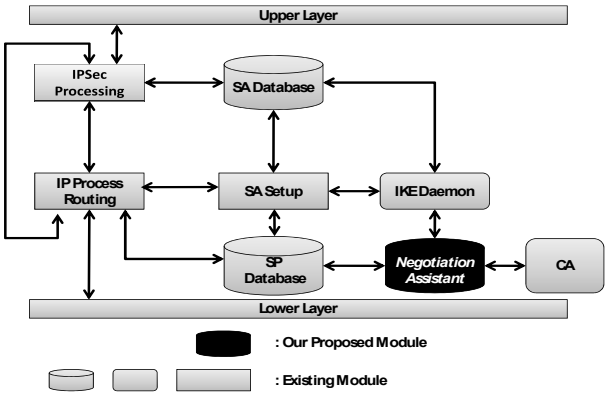

Fig.2 the negotiation assistant module in the IP packet process routing

The negotiation assistant module is the key of our proposed method. When the host needs a certificate for the IKE negotiation, the negotiation assistant module communicates with the CA directly, requests issuing the certificate, parses security policies that is embedded, and updates the Security Policy Database through the IKE daemon. Before starting the IKE negotiation, the negotiation assistant module will pre-contact with the

responder to validate the security policy version, this is done by examining the policy version validation message (which will be introduced in the next section) through the IKE daemon. We will introduce the process of issuing a security policy embedded certificate, and the policy version check mechanism in the next two sections.

3.2 Certificate and security policy

First, pre-proposed security policies will be prepared by the CA, and these security policies are issued as the extension of the certificate by the CA for each host. The process of issuing a certificate shown as follows

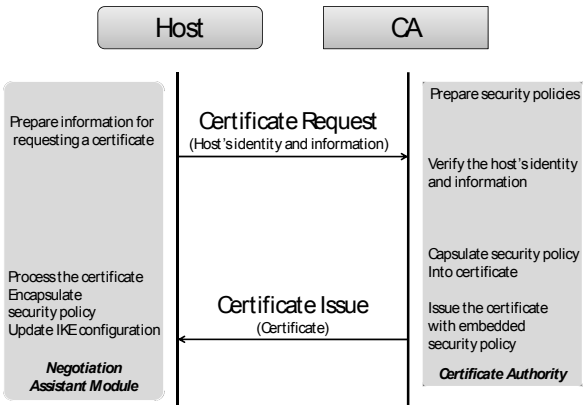

Fig. 3 Process of issuing a certificate

Before starting the IKE negotiation, in order to authenticate itself during the negotiation, the host sends a request of issuing a certificate to the CA. When the CA receives the host's request, a validation of the host's identity is performed. Then the CA generates the X.509 certificate with embedded security policies, and then publishes the certificate. As soon as the host gets its certificate, the negotiation assistant module parses the certificate, encapsulates security policies and imports them into the IKE configuration file.

By doing this, security policies are added into the IKE configuration file as many other specified security policies.

3.3 Policy version validation

Compared with current IKE negotiation process, our proposed method introduces the security policy version validation mechanism to guarantee that both hosts have the same version of the security policy to start the IKE negotiation. As shown in Figure 4, Host A, the initiator, sends its security policy information through PolicyVersionCheck message to Host B by negotiation assistant module. Host B's negotiation assistant module responses the message, make proper reaction, sends back PolicyVersionAnswer message.

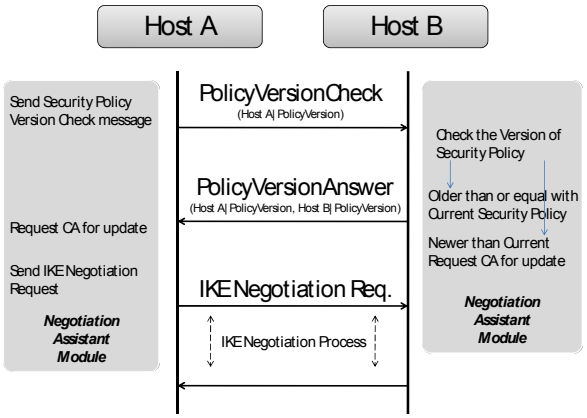

Fig.4 Policy version validation

The message PolicyVersionCheck (Host ID|PolicyVersion), Host ID and PolicyVersion here denote the initiator. PolicyVersionAnswer(Host ID|PolicyVersion, Host ID|PolicyVersion), the first one is the initiator's ID and parameter, while the second one belongs to the responder. Through the two messages, hosts exchange security policy version information, negotiation assistant module compares the information. We assign a number to each policy as its version, and appoint that the larger of the number the newer of the policy. For an example:

Host A | Policy Version > Host B | Policy Version

It means that Host A's policy is newer that Host B's. In order to start negotiation, Host B must update its policy with CA immediately.

There are three conditions and corresponding actions are taken place:

- Host A | Policy Version > Host B | Policy Version
  Host B contacts with CA to update certificate

- Host A | Policy Version < Host B | Policy Version
  Host A contacts with CA to update certificate

- Host A | Policy Version = Host B |Policy Version
  Host A Sends IKE negotiation request to Host B

The PolicyVersionCheck() message is send periodically until the last situation is satisfied that the hosts' security policy version are the same in the PolicyVersionAnswer() message. When Host A and Host B have the same policy, the traditional IKE negotiation can start. As security policies have been pre-matched, the negotiation is expected to be success.

4. IPSec local networks

Information security has become one of the most concerned issues. Our proposal has a positive meaning for solving the security problem of small scope network environment.

The use of IPSec can provide end-to-end data encryption, and efficiently improve the security of data transmission, ensure data integrity. For LAN environment, plaintext packets are easily sniffed by malicious users. It may cause many security problems. IPSec is a good solution, but the configuration is complex. In order to avoid the complexity of configure IPSec, we provide our method in the following assumption:

We assume that in a not fully trusted LAN environment [3], hosts know each other, however the encryption of the data has high requirements, and the end-to-end IPSec connection is needed. It is shown in Figure 5:

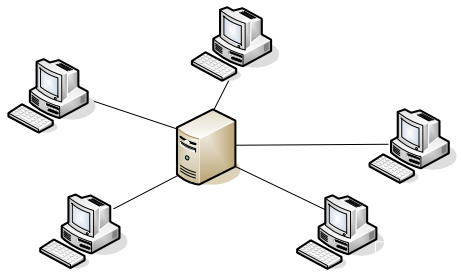

Fig.5 Local LAN configuration

The LAN is constituted by a number of host and a CA server. Host requires the CA server to issue a certificate before establishing a secure connection. When the certificate is obtained, the IPSec connection is established through proposed method. What the system administrators need to do is only CA server maintenance and develops the pre-proposed security policies. The IKE configuration of each host is completed by the negotiation assistant module. This method eliminates the process of user manually setting up security policies, whole auto-configuration.

In IPv6 environment, it is possible to realize of any end-to-end communication. For the enterprise-level applications, we have assumed a number of different LAN exist in spatial at the same time, each LAN is constructed as we proposed. A root CA server takes charge of overall situation contemporarily, and manages the universal security policies. With the characteristics of the X.509 certificate infrastructure, by inheriting the root certificate, it is possible to achieve automatic end-to-end IPSec connection among different LANs.

5. Conclusion and future work

By using the X.509, the proposed method reduces the complexity of configuration of security policies in a certain extent, improves the adaptability of the IKE protocol. We also explore the possibility of automatically updating security policies by certificate. In the future research work, we will focus on the design of the detail structure and implementation of the negotiation assistant module.

Reference

[1] D. Harkins and D. Carrel, RFC 2409, *Internet Key Exchange (IKE)*, Nov., 1998.

[2] R. Housley, W. Polk, W. Ford, and D. Solo, RFC3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Apr., 2002.

[3] C. R. Davis, *IPSec securing VPN*, Osborne/McGraw-Hill, 2001.

[4] J. Park, J. Lee, H. Lee, S. Park, and T. Polk, RFC4683, *Internet X.509 Public Key Infrastructure Subject Identification Method (SIM)*, Oct., 2006.

[5] M. Blaze, J. Ioannidis, and A. D. Keromytis, "*Trust Management for IPSec,"* ACM Transactions on Information and System Security, Vol. 5, No. 2, pp. 95-118, May, 2002.