# A Protocol for Internet Key Exchange(IKE) using Public Encryption Key and Public Signature Key

***V.NagaLakshmi*[1], *I.Rameshbabu*[2]**

***[1]Department of Computer Science, Gandhi Institute of Technology and Management, Visakhapatnam***
***[2] Department of Computer Science, Acharya Nagarjuna University, Guntur, A. P India***

## SUMMARY

Internet Key Exchange (IKE) is a key exchange mode for Internet Security Association and Key Management Protocol (ISAKMP) and is used to securely exchange encryption keys as part of building a Virtual Private Network (VPN) tunnel [1]. IKE is defined in RFCs 2407, 2408 and RFC 2409 and uses Diffie Hellman key exchange to set up a shared session secret, from which cryptographic keys are derived [2]. Two basic methods are used to establish an authenticated key exchange in IKE, namely, the main mode and the aggressive mode. Each generates authenticated keying material from Diffie Hellman Key Exchange. IKE uses two chosen numbers called a nonce, and a cookie which are kept secret [3]. There are many limitations with these concepts of nonce and cookies, especially when they are very large [4]. In this paper we propose a protocol for the public encryption key, main mode, revised protocol. Instead of using nonce and a cookie, we propose to use a hash function of public encryption key and the signature key for generating a secret key, So that the limitations of using the nonce and cookies can be resolved. The proposed protocol uses Diffie Hellman key exchange.

Keywords: IKE, ISAKMP, nonce, cookie.

## 1. Introduction

A fundamental problem in cryptography is how to communicate securely over an insecure channel, which might be controlled by an adversary. It is common in this scenario for two parties to encrypt and authenticate their messages in order to protect the privacy and authenticity of these messages. One way of doing so is to use public-key encryption and signatures. Another way of addressing this problem is for users to first establish a common secret key via a key exchange protocol and then use this key to derive keys for symmetric encryption and message authentication

Schemes [5] In practice, one finds several flavors of key exchange protocol, each with its own benefits and drawbacks. Among the most popular is the 3-party 'Kerberos' authentication system [6]. Another is the 2-party SIGMA protocol [7] used as the basis for the signature-based modes of the Internet Key Exchange protocol.

## 2. The Protocols

IKE is specified by the Internet Society that references the Internet Security Association and Key Management Protocol (ISAKMP) and the Internet IP Security Domain of Interpretation (DOI) for ISAKMP. ISAKMP specifies the high-level, abstract syntax and semantics for certain types of key management protocols. Thus, while the letter "P" in the abbreviation "ISAKMP" means "protocol," ISAKMP specifies only a framework for key management protocols but not any implemental protocol since the specification lacks sufficient low-level details. IKE and DOI fill in the details and specify a set of implemental protocols that fit into the framework. Whereas IKE focuses mainly on the detailed protocol semantics, DOI focuses mainly on the detailed syntax and semantics of the information carried by the messages of the protocol [8]. The distinction between IKE and ISAKMP is very confusing. Probably the best way to think of it is that IKE is a profiling (i.e., defining fields, choosing options) of ISAKMP.

## 3. Internet Security Association Key Management Protocol (ISAKMP)

ISAKMP defines procedures and packet formats to establish, negotiate, modify and delete security associations. As part of SA establishment, ISAKMP defines payloads for exchanging key generation and authentication data. These payload formats provide a consistent framework independent of the specific key exchange protocol, the encryption algorithm and the authentication mechanism [9].

ISAKMP provides a way to

Manuscript received July 5, 2007
Manuscript revised July 25, 2007

a) Agree on which protocols, algorithms, and keys to use (negotiation services).
b) Manage those keys after they have been agreed upon (key management) and
c) Exchange those keys safely [10].

ISAKMP makes a distinction between "key exchange" and "key management" and considers the latter to be a superset of the former. Key exchange is mainly concerned with exchanging information to generate secret keys shared between two parties. ISAKMP requires a key exchange protocol to:

i) Generate a set of secret key(s) shared exclusively between the two parties
ii) Authenticate the identity of each party to the other. (Here, "authenticating identity" means authenticating the binding between a party's claimed identity and the pieces of information the party claims to have sent and received.)
iii) Ensure the set of secret keys generated by one protocol message exchange to be independent of key sets generated by other protocol message exchanges. (This means compromise of one key set does not lead to compromise of other sets. This property is usually known as perfect forward secrecy (PFS)) and
iv) Be scalable. Here scalability means that a key exchange protocol can be executed between any two parties within a very large population, even if the two parties do not share any secret a priori. This requirement, coupled with the requirement for authentication, implies the use of public key cryptography, and dependency on the public key infrastructure (PKIX) [8].

To establish an ISAKMP SA, the initiating node proposes five things:

1. An encryption algorithm (to protect data)
2. A hash algorithm (to reduce data for signing)
3. An authentication method (for signing data)
4. Information about a group over which a Diffie-Hellman exchange will be done.
5. A pseudo-random function (PRF) used for hashing certain values during the key exchange for verification purposes (this is optional, a hash algorithm may be used) [10].

## 4. Internet Key Exchange (IKE)

IKE (Internet Key Exchange) is a protocol for doing mutual authentication and establishing a shared secret key to create an IPSec SA. The specification of IKE is in three pieces; ISAKMP (Internet Security Association and Key Management Protocol, RFC 2408), IKE (RFC 2409) and the DOI (Domain of Interpretation, RFC 2407). The intention of IKE is to do mutual authentication using some sort of long term key (pre shared secret key, public signature-only key, or public encryption key), and to establish a session key [4]. All IKE communications consist of pairs of messages: a request and a response. The pair is called an "exchange". IKE message flow always consists of a request followed by a response. It is the responsibility of the requester to ensure reliability. If the response is not received within a timeout interval, the requester needs to retransmit the request (or abandon the connection) [11].

### 4.1 IKE Phases:

IKE performs all this communications in two phases. Phase 1 does mutual authentication and establishes session keys. It is based on identities such as names, and secrets such as public key pairs, or pre-shared secrets between two entities. Then using keys established in phase1, multiple phase-2 SAs between the same pair of entities can be established. The phase-1 exchange is known as the ISAKMP SA, or sometimes it is referred to as the IKE SA. An ESP or AHSA would be established through phase 2.

### 4.1.1 Phase 1 IKE

There are two types of phase-1 exchanges, called modes. Aggressive mode accomplishes mutual authentication and session key establishement in three messages. Main mode uses six messages, and has additional functionality, such as the ability to hide endpoint identifiers from eavesdroppers and additional flexibility in negotiating cryptographic algorithms. The first request/response of an IKE session (IKE_SA_INIT) negotiates security parameters for the IKE_SA, sends nonces, and sends Diffie-Hellman values. The second request/response (IKE_AUTH) transmits identities, proves knowledge of the secrets corresponding to the two identities, and sets up an SA for the first (and often only) AH and/or ESP CHILD_SA. Like this there are 8 variants of the phase 1 of IKE. That is because there are 4 authentication methods (Original public key encryption, revised public key encryption, public key signature, and pre-shared secret key encryption) and for each authentication method, a main mode protcol and an aggressive mode protocol.

#### i) Public Signature keys, Main Mode

In this mode, the two parties have public keys capable of doing signatures. Both endpoint identifiers are hidden from an eaves dropper. Messages 3 and 4 includes nonces and the Diffie Hellman private value . Depending on that they are calculating a shared secret key and that key can be used in messages 5 and 6 for encrypting the authentication message and the certificate. Fig 1 illustrates this protocol.

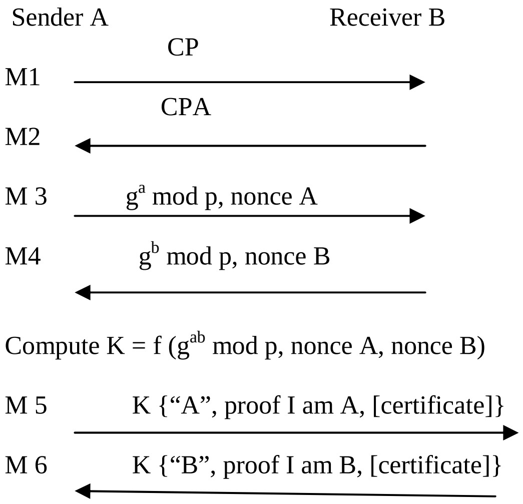
Sender A
Receiver B
CP
M1
CPA
M2
M 3
$g^a$ mod p, nonce A
M4
$g^b$ mod p, nonce B
Compute K = f ($g^{ab}$ mod p, nonce A, nonce B)
M 5
K {“A”, proof I am A, [certificate]}
M 6
K {“B”, proof I am B, [certificate]}

Figure 1: Public Signature keys, main mode

### ii) Public Signature keys, Aggressive Mode

In this mode the messages of 1, 3 and the proof of the identity of a person of main mode are combined. Likewise in protocols 2,4 and 7 the proof of the identity of the person has been combined and there is no generation of the shared secret key and the encryption of the message. That why the total messages in this are three only. Fig 2 illustrates this protocol.

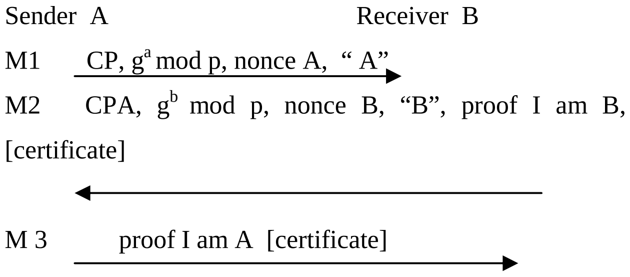
Sender A
Receiver B
M1
CP, $g^a$ mod p, nonce A, “ A”
M2
CPA, $g^b$ mod p, nonce B, “B”, proof I am B, [certificate]
M 3
proof I am A [certificate]

Figure 2: Public Signature keys, aggressive mode

### iii) Public Encryption Key, Main Mode, Original

Figure 3 illustrates this protocol. The two messages 1 and 2 of this are same as the previous one that is sending the crypto proposal by sender A and the acceptance of it by receiver B. The message 3 is the Diffie-Hellman value and the nonce of the sender which is encrypted with the receivers public key and the proof of a person encrypted with the receivers public key separately. The message 4 is vice versa. Then the shared secret key is computed. The 5 th and 6th messages are the authentication of the sender and the receiver encrypted with the shared secret key. The problem with this variant is that in message 3 there are two fields separately encrypted with receiver’s public key, and hence he/she needs to do private key operations to decrypt it. Likewise sender needs to do two private key operations to decrypt message 4. Another problem would occur if a nonce or a name were larger than the public key with which it is being encrypted.

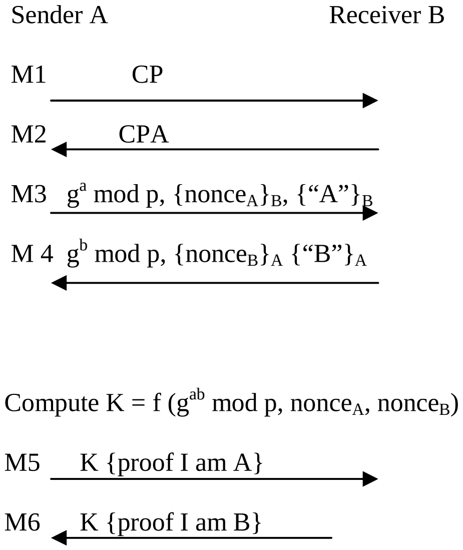
Sender A
Receiver B
M1
CP
M2
CPA
M3
$g^a$ mod p, $\{nonce_A\}_B$, $\{“A”\}_B$
M 4
$g^b$ mod p, $\{nonce_B\}_A$ $\{“B”\}_A$
Compute K = f ($g^{ab}$ mod p, $nonce_A$, $nonce_B$)
M5
K {proof I am A}
M6
K {proof I am B}

Figure 3: Public Encryption Key Main mode, Original protocol

### iv) Public Encryption Key, Aggressive Mode, Original

Figure 4 illustrates this protocol. This protocol is almost the same as the main mode version except that messages 1 and 2 are removed and receiver provides his/her proof in message 2 rather than as in main mode, doing it after receiver presents his/her proof.

Sender A Receiver B

M1 CP, $g^a$ mod p, $\{nonce_A\}_B$, $\{$" A"$\}_B$ →

M2 ← CPA, $g^b$ mod p, $\{nonce_B\}_A$, $\{$"B"$\}_A$, proof I am B

M 3 proof I am A →

Figure 4: Public Signature keys, aggressive mode, original protocol

### v) Public Encryption Key, Main Mode, Revised

This protocol is represented in Figure 5. The public encryption protocol was revised to require only a single private key operation on each side. This is done by encrypting with a secret key which is a function of the nonce, and the nonce is encrypted with the other side's public key( i.e., if sender A is sending the nonce A that will be encrypted with the B's public key and vice versa). Thus the other side uses its private key to retrieve the nonce, but then decrypts the other fields with a secret key. Even this protocol still has the problem[4].

Sender A Receiver B

M1 CP →

M2 ← CPA

$K_A$ = hash($nonce_A$,$cookie_B$)

M3 $\{nonce_A\}$B, $K_A\{g^a$ mod p)}, $K_A\{$"A"$\}$, [$K_A\{$A's certificate }] →

$K_B$ = hash($nonce_B$,$cookie_B$)

M 4 $\{nonce_B)\}$A, $K_B\{g^b$ mod p), $K_B\{$"B"$\}$, [$K_B\{$B's cert}] ←

K = f($g^{ab}$ mod p, $nonce_A$, $nonce_B$, $cookie_A$, $cookie_B$ )

M5 K {proof I am A} →

M 6 K {proof I am B} ←

Figure 5: Public Encryption Key, Main Mode, Revised protocol

### 5. Proposed protocol:

To overcome the limitations of the previous protocol we propose a protocol which uses the public encryption key and the public signature key. Figure 6 illustrates this protocol. In the public encryption key, Main mode, revised they are calculating the hash of nonce and the cookie of the sender. $K_A$ = hash (nonce A, cookie A). Instead of using that we propose to use the hash of the public signature key of the sender and the hash of the public encryption key of the receiver for calculating $K_A$. Actually the messages are 6 only but we are reducing the overhead of using the other values of nonce and cookies. The first two messages are sending and accepting crypto proposals. For sending the third message we are calculating the $K_A$ i.e., $K_A$ = ((hash($KR_A$), (hash ($KU_B$)). In this the hash of the private key of the sender and the hash of the public key of the receiver are taken and calculated the $K_A$. In the third message we are sending the hash of the private key of the receiver by encrypting it by using the receiver's public key. This can be used for calculating the KA by the receiver and it can be further used for encryption of the Diffie Hellman value and the identity of the receiver and the certificate of the receiver. The remaining messages can be sent as it is. We propose to extend this not only in the Public Encryption Keys, Main mode, revised protocol but for the other protocols also.

Sender A Receiver B

M1 CP →

M2 ← CPA

$K_A$ = (hash ($KR_A$), hash ($KU_B$))

M3 $\{hash(KR_A)\}_B$ , $K_A\{g^a$ mod p)}, $K_A\{$"A"$\}$, [$K_A\{$A's cert}] →

$K_B$ = (hash ($KR_B$), hash ($KU_A$))

M4 $\{hash(KR_B)\}_A$ , $K_B\{g^b$ mod p), $K_B\{$"B"$\}$, [$K_B\{$B's cert}] ←

K = f($g^{ab}$ mod p, hash($KR_A$), hash($KR_B$), $KU_A$, $KU_B$

M5 K{proof I am A} →

M 6 K{proof I am B} ←

Figure 6. Proposed protocol:
Public Encryption keys, Public Signature keys, Main mode

Note: M stands for the message, i.e M1 refers to message 1, M2 for message2 and so on. K stands for the Key.

## 6. Conclusion

In this paper we propose a protocol for Internet Key Exchange. IKE (Internet Key Exchange) is a protocol for doing mutual authentication and establishing a shared secret key to create an IPSec SA. All IKE communications consist of pairs of messages: a request and a response. IKE performs all this communications in two phases. Phase 1 does mutual authentication and establishes session keys. It is based on identities such as names, and secrets such as public key pairs, or pre-shared secrets between two entities. To overcome the limitations of the Public Encryption key, Main Mode, revised protocol a protocol which uses the public encryption key and the public signature key is proposed. This can be extended to other protocols also.

## REFERENCES:

[1] Perlman, R. and Kaufman, C. "Key Exchange in IPSec: Analysis of IKE", IEEE Ineternet Computing, Nov/Dec 2000.

[2] Harkins,D., and Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, Nov 1998.

[3] Radia Perlman, "Analysis of the IPSec Key Exchange Standard "Sun Microsystems Laboratories, Charlie Kaufman, Iris Associates.

[4] **Kaufman**, Radia Perlman, Mike Speciner, "**Network Security** Private Communication in a Public World Charlie "Second edition, Chap 18, pages 444 - 460.

[5] M. Abdalla, P.-A. Fouque and D. Pointcheval "Password-based authenticated key xchange in the three-party setting" IEEE Proc.-Inf. Secur., Vol. 153, No. 1, March 2006.

[6] Steiner, J.G., Neuman, B.C., and Schiller, J.L.: 'Kerberos: an authentication service for open networks'. Proc. USENIX Winter Conference, Dallas, TX, 1988, pp. 191–202

[7] Krawczyk, H.: 'SIGMA: The ''SIGn-and-MAc'' approach to authenticated Diffie–Hellman and its use in the IKE protocols'. In Boneh, D. (Ed.), Proc. Advances in Cryptology—CRYPT

[8] P.C. Cheng "An architecture for the Internet Key Exchange Protocol", IBM Systems journal, Volume 40, Number 3, 2001.

[9] William Stallings " Cryptography and Network Security principles and practices" Pearson Education, Third, edition, Chap 16, page 508.

[10] Ken Camp "**ISAKMP**/Oakley" White Paper, Copyright, 1997-2001, www.ipadventures.com - ken@ipadventures.com

[11] C. Kaufman, Ed. PROPOSED STANDARD "Internet Key Exchange (IKEv2) Protocol**",** Network Working Group, Request for Comments: 4306 , Microsoft, Obsoletes: 2407, 2408, 2409 , December 2005 Category: Standards Track.

**Mrs. V.NagaLakshmi** received her MCA degree from Andhra University, Visakhapatnam, A.P., India in 1998 and joined as an Assistant Professor in the Department of Computer Science, Gandhi Institute of Technology and Management (GITAM) in 2000. She is pursuing her PhD from Acharya Nagarjuna University A.P., India. She presented /published papers in International conferences and journals. Her areas of interest include Cryptography and Network Security, Watermarking and RFID technologies. She is a life member of Computer Society of India.

**Dr I.Ramesh Babu** received his B.E., from University of Mysore in 1981, M.E from Andhra University in 1984 and PhD from Nagarjuna University in 1994. He joined as an Assistant Professor in the Department of Computer Science and joined as an Assistant Professor in Acharya Nagarjuna University, Guntur A.P., India, and became an Associate professor in the year 1994 and Professor in 2004. He held many positions in Acharya Nagurjuna University as Head, Director - Computer Centre, Chairman- Board of studies. His areas of interest include Image Processing, Computer Graphics, Cryptography and Network Security. He is a member of IEEE, CSI, ISTE, IETE, IGISS, Amateur Ham Radio (VU2 IJZ).