ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Internet Key Exchange Aggressive mode negotiations using Cookie and Nonce Alternatives

Sushma Yalamanchili*
Research Scholar, Dept. of Computer Science & Engg
Acharya Nagarjuna University
Andhra Pradesh, India
sushma_yalamanchili@yahoo.co.in

M. Kameswara Rao
Dept. of Computer Science, P.G.Centre
P.B.Siddhartha College, Vijayawada
Andhra Pradesh, India
kamesh.manchiraju@gmail.com

Ch. Smitha Chowdary
Dept. of Computer Science, P.G.Centre
P.B.Siddhartha College, Vijayawada
Andhra Pradesh, India
chsmithachowdary@gmail.com

Abstract: The Internet is constantly evolving with new technology, networks, applications and users that require different levels of security. It is therefore a requirement that security requirements be reassessed at frequent intervals of time by all stakeholders. Internet Security Protocol (IPSec) provides security at the Internet layer protocol. Internet Key Exchange (IKE) is optionally used to perform key exchange, authentication and formation of Security Associations between a pair of communicating parties. In this paper, we examine the impact of using costly computational operations involving large random and prime numbers such as nonces and cookies in IKE interactions in Aggressive mode methods. IKE Aggressive mode interactions for all four methods are proposed using alternatives for cookies and nonces which will reduce the computational overhead involved in these interactions.

Keywords: Internet Security; Internet Key Exchange; Aggressive mode; Cookies; Nonces; Authentication; Digital Signatures

I. INTRODUCTION

As the number of users, hosts and applications on the Internet grows, providing security to resources and mitigating attacks is highly challenging. The capabilities of security systems are unable to keep pace with the escalating security requirements. Hence network designers, system administrators, infrastructure providers, application developers and users have to periodically review their security systems and strategies. They should aim to safeguard resources and provide authorized access to them. Security provided on the Internet has a lot to do with Internet Security Protocol (IPSec) and its components.

II. INTERNET KEY EXCHANGE

Internet Key Exchange (IKE) [1, 2] is a key exchange protocol that can be used with IPSec for key exchange and creation of secure communication channels called Security Associations (SAs) [3] between two communicating entities. The SAs are defined by a set of cryptographic parameters. There are two phases in IKE with Phase 1 [1] operating in a more secure and hence more complex Main mode. Three message pairs are exchanged between the negotiating entities in each of the four Main mode methods. The other mode supported by Phase 1 is Aggressive mode that is quicker and involves only three messages. Aggressive mode messages do not protect the identities of the participating entities. Aggressive mode also supports the same four methods that Main mode does. The Phase 1 modes establish SAs which are determined by cryptographic parameters such as the Diffie-Hellman [4] group which includes the public values of the Diffie-Hellman key exchange, namely, a large prime number and a generator value, symmetric encryption algorithm to be used in Phase 2 negotiations and a keyed hash message authentication code [5]. The two negotiating entities get authenticated in Phase 1. Phase 2 operates in Quick mode and this creates unidirectional IPSec SAs between the same pair of entities.

There are four methods in Aggressive mode namely, Pre-shared key method, Digital Signature method, Public Key Encryption method and Revised Public Key Encryption method. Public key encryption is used to generate Diffie-Hellman keys [4] between the two entities by only exchanging a large prime, a generator and the respective half-keys in the clear. Both entities compute the same Diffie-Hellman full-key using these exchanged values. This paves the way for generating secrets and keys that are used as keying material for further IKE negotiations. Hash values are computed for use in authentication between the two communicating entities.

III. LITERATURE SURVEY

Virtual Private Networks (VPNs) [6] offer a viable alternative to companies that find leased lines for communication to be cost-prohibitive. In Aggressive mode, the identities of the two communicating parties are not encrypted and appear in the clear when used with VPNs. Security vulnerabilities of IKE Aggressive mode [7, 8, 9, 10] with particular reference to VPNs and password cracking are discussed extensively. Despite its vulnerabilities, Aggressive mode can be used with VPNs particularly to support remote access users and when at least one of the peers uses dynamic external Internet Protocol (IP) addresses. When the peer

devices in VPNs support Digital signatures, Aggressive mode is optional and not a requirement.

Anti-clogging is a technique used to avoid a scenario where an entity is flooded with requests or messages that virtually leaves the entity unable to function normally or worse to incapacitate it altogether [11]. In particular, anti-clogging minimizes the number of computationally intensive public key operations by using anti-clogging tokens. Cookies [12] which are typically 8-byte pseudo random numbers are used in IKE modes to minimize Denial-of-Service (DoS) attacks [11]. They are computed on each side by computing the message digest of the source and destination Internet Protocol (IP) addresses, a random number, date and time [13]. They help avoid replay attacks due to the time-related information that is incorporated into them. Nonces are very large random numbers that are used in the future key generation process [13]. Perlman et al. analyzed IPSec and IKE [14, 15] and suggested that using cookies and nonces results in computational overhead on the communicating peers. She also suggested that stateless cookies be explored in place of state-preserving cookies. Neuman and Stubblebine suggested that timestamps be used in the place of cookies [16]. An alternative to the use of cookies and nonces in IKE Main mode Revised Public Key Encryption method was proposed by NagaLakshmi and Rameshbabu [17]. In that work, they discussed several existing IKE Phase 1 methods in both modes and presented an authentication mechanism for IKE Main mode Revised Public Key Encryption method that involves the use of Public keys and Digital Signatures. They advocate the replacement of nonces and cookies with smaller and less computationally intensive alternatives and propose an authentication mechanism that is applied to Main mode Revised Public key encryption method.

Another alternative to existing IKE Phase 1 authentication methods is presented by Yalamanchili and Sambasiva Rao [18] that uses Digital Signatures and Public Key Certificates. In that work, they propose an authentication technique based on Dual Signature and dual hash that is applied to all four methods of IKE Phase 1 Main mode. Techniques for Message generation and recovery are also proposed in that work.

An analysis of the security aspects of IKE versions 1 and 2 is presented by Cremers [19] which does not detect any vulnerabilities in the key generation process with specific reference to secrecy. However several weaknesses in the authentication process are identified.

IV. PROPOSED AGGRESSIVE MODE NEGOTIATIONS

We extend that work and propose the interactions for the four IKE Aggressive mode methods using public key encryption and digital signatures. In the Aggressive mode negotiations, we replace a nonce with the hash value of the private key of the message initiator. We refer to this as the nonce equivalent. A cookie is replaced with the hash value of the Public key of the message. We refer to this as the cookie equivalent. These values are used in computing the secret SKEYID and the keys SKEYID_d, SKEYID_a, and SKEYID_e [1]. The hash values, Hashi and Hashr are computed using them.

A. Notation:

In each of the four proposed Aggressive mode method interactions described below, we use the subscripts i and r to refer to the parameters of the initiator and recipient respectively. The parameters that are used in the Aggressive mode interactions are as follows: ID refers to the Identity payload, SIG refers to the signature value which is signed, PU{X} refers to the encrypted value of X using the Public key, PR{X} refers to the encrypted value of X using the private key, Hash[X] refers to the hash value of X, K{X} refers to the encrypted value of X using key K and prf refers to the pseudo random function. When a prf has not been defined, a Hash Message Authentication Code (HMAC) function may be used in its place. SA_i refers to the security association payload containing the cryptographic parameter set that is proposed by the initiator while SA_r refers to the cryptographic parameter set that will be used during the IKE interactions. DHK_i and DHK_r refer to the Diffie-Hellman halfkeys of the initiator and recipient respectively. DHK_{ir} refers to the Diffie-Hellman fullkey which is the same as DHK_{ri}. $PSKey_i$ and $PSKey_r$ refer to the pre-shared keys of the initiator and the recipient respectively.

B. Proposed Aggressive Mode Pre-Shared Key Method:

In this method (Table 1), a pre-shared key is shared by the two communicating peers prior to the start of the Aggressive mode negotiations through an out-of-band channel. The first message in the original pre-shared key method includes the IKE header, cookie, security association proposal, Diffie-Hellman halfkey, nonce, identity and an optional certificate belonging to the Initiator. In our proposed method, we replace the initiator cookie with the hash value of the public key of the recipient. The nonce value is replaced with an encrypted hash value of the private key of the initiator using the public key of the recipient.

The recipient uses the Diffie-Hellman half-key that was sent by the initiator and computes the Diffie-Hellman fullkey. The nonce equivalent is obtained by decrypting the Nonce payload. This value along with the Diffie-Hellman keys and cookie equivalent that it received are used in computing SKEYID and the keys SKEYID_d, SKEYID_a and SKEYID_e.

Upon receiving the second message, the initiator is able to compute the Diffie-Hellman full key, keys and the hash values. By computing Hashr and matching it with the received Hashr value, the initiator is authenticated. In the third message, the initiator responds by sending the cookie equivalents of the initiator and the recipient along with Hashi value. The recipient is authenticated when the received Hashi value matches the computed Hashi at its end. Thus mutual authentication is achieved.

Our proposed method is still vulnerable to password cracking as the Identities are exchanged in the clear in messages 1 and 2. We therefore suggest a fix that sends an encrypted value of identity using the public key of the other peer as we are already using public key encryption to send the hash of private key of the message initiator as the nonce equivalent. This provides security against password cracking vulnerability of this method when used with VPNs.

Table I. Proposed Aggressive mode Pre-shared key method

M1	I→R	Hdr, Hash(PUr), 0, SAi, DHKi, PUr{Hash(PRi)}, IDi, [CERTi…]
M2	I←R	Hdr, Hash(PUr), Hash(PUi), SAr, DHKr, PUi{Hash(PRr)}, IDr, Hashr
M3	I→R	Hdr, Hash(PUr), Hash(PUi), Hashi
SKEYID = prf(preshared key, Hash(PRi) || Hash(PRr)) SKEYID_d = prf(SKEYID, DHKir || Hash(PUr) || Hash(PUi) || 0) SKEYID_a = prf(SKEYID, SKEYID_d || DHKir || Hash(PUr) || Hash(PUi) || 1) SKEYID_e = prf(SKEYID, SKEYID_a || DHKir || Hash(PUr) || Hash(PUi) || 2) Hashi = prf(SKEYID, DHKi || DHKr || Hash(PUr) || Hash(PUi) || PSKeyi || SAi || IDi) Hashr = = prf(SKEYID, DHKr || DHKi || Hash(PUi) || Hash(PUr) || PSKeyr, || SAr || IDr)		

C. Proposed Aggressive Mode Digital Signature Method;

The recipient uses the Diffie-Hellman half key and computes the fullkey. It also computes SKEYID and the keys along with the hash value Hashr. The Hashr value is signed using a pseudo-random function or HMAC version of the hash algorithm and sent to the initiator as SIGr. Other values sent are the Security Association accepted proposal, the Diffie-Hellman half key, the encrypted nonce equivalent using public key of the initiator, cookie equivalent and its identity. This allows the initiator to compute the Diffie-Hellman full key, keys and the hash values Hashi and Hashr. It extracts the value of Hashr from the signed SIGr and matches it with the computed Hashr value. In this manner the initiator gets authenticated.

Table II. Proposed Aggressive mode Digital Signature method

M1	I→R	Hdr, Hash(PUr), 0, SAi, DHKi, PUr{Hash(PRi)}, IDi, [CERTi…]
M2	I←R	Hdr, Hash(PUi), Hash(PUi), SAr, DHKr, PUr{Hash(PRr)}, IDr, [CERTr…], SIGr
M3	I→R	Hdr, Hash(PUr), Hash(PUi), SIGi
SKEYID = prf(Hash(PRi) || Hash(PRr), DHKir) SKEYID_d = prf(SKEYID, DHKir || Hash(PUr) || Hash(PUi) || 0) SKEYID_a = prf(SKEYID, SKEYID_d || DHKir || Hash(PUr) || Hash(PUi) || 1) SKEYID_e = prf(SKEYID, SKEYID_a || DHKir || Hash(PUr) || Hash(PUi) || 2) Hashi = prf(SKEYID, DHKi || DHKr || Hash(PUr) || Hash(PUi) || SAi || IDi) Hashr = prf(SKEYID, DHKr || DHKi || Hash(PUi) || Hash(PUr) || SAr || IDr)		

The initiator signs the computed Hashi value and sends it as SIGi along with the hash values of the public keys of the initiator and the recipient. When the recipient is able to match the computed Hashi value with the extracted Hashi value from SIGi, it too is authenticated. It should be noted that the hash values Hashi and Hashr in this message are signed as opposed to being sent in the clear as in the pre-shared key and public encryption methods.

D. Proposed Aggressive mode Public Key Encryption Method:

In the proposed Public encryption key method (Table 3), the hash value of the public encryption key of the recipient is used in place of the cookie and the encrypted hash value of the initiator's private key using the recipient's public key is used in place of the nonce. The Diffie-Hellman half key value and Identity are encrypted with the public key of the initiator. These values along with the security association proposal are sent to the recipient.

Table III. Proposed Aggressive mode Public key Encryption method

M1	I→R	Hdr, Hash(PUr), 0, SAi, PUr{Hash(PRi)}, PUr{DHKi, IDi, [CERTi…]}
M2	I←R	Hdr, Hash(PUr), Hash(PUi), SAr, PUi{Hash(PRi)), PUi{DHKr, IDr, [CERTr…]}, Hashr
M3	I→R	Hdr, Hash(PUr), Hash(PUi), Hashi
SKEYID, SKEYID_d, SKEYID_a, SKEYID_e, Hashi and Hashr are computed as in Digital Signature method.		

The recipient extracts the encrypted values that it has received from the initiator and is able to compute the Diffie-Hellman fullkey, keys and its hash value Hashr. The recipient follows a similar procedure in generating the equivalent of its cookie and nonce and sends the usual SA accepted proposal, encrypted Diffie-Hellman halfkey, encrypted Identity and its hash value Hashr to the initiator. Mutual authentication is achieved as in the previous two methods.

E. Proposed Aggressive mode Revised Public Key Encryption method:

In this method (Table 4), we aim to reduce the number of public key operations that take place in the Public key encryption method. This is achieved by only encrypting the equivalent of its nonce value, namely, the hash value of the initiator's private key, with the recipient's public key. The Diffie-Hellman half key and the Identity are encrypted using a new key called Kei. This is computed as the hash of the concatenated hash values of the initiator's private key hash and the recipient's public key hash. All other parameters and procedures remain the same. The recipient follows a similar procedure that results in the authentication of the initiator. At the end of message 3, the recipient is authenticated.

Table IV. Proposed Aggressive mode Revised Public key encryption method

M 1	I →R	Kei = Hash(Hash(PRi) || Hash(PUr)) Hdr, Hash(PRi), 0, SAi, PUr{Hash(PRi)}, Kei{DHKi, IDi, [CERTi…]}
M 2	I ←R	Ker = Hash(Hash(PRr) || Hash(PUi)) Hdr, Hash(PRi), Hash(PRr), SAr, PUi(Hash(PRi)), Ker{DHKr, IDr, [CERTr…]}, Hashr
M 3	I →R	Hdr, Hash(PUr), Hash(PUi), Hashi
SKEYID, SKEYID_d, SKEYID_a, SKEYID_e, HASHi and HASHr computed as in Digital Signature method.		

V. CONCLUSION

In our proposed work, we have eliminated the use of cookies and nonces in interactions for all four methods of IKE Aggressive mode. We replace cookies and nonces with small hash values involving public and private keys of the initiator and recipient. This results in reduced computation as cookies and nonces involve complex and heavy computation in generating them and in computing the secret, keys and hash values on each side that use them. We have applied this concept in Aggressive mode methods which already have less overhead as they only involve three messages.

Cookies are usually used for anti-clogging purposes. They combat Denial of Service and replay attacks [12]. We can achieve that functionality by using timers at the initiator end that will detect an aborted sequence of interactions or a lost message. This will abort partial SAs. Alternatively the initiator can scan all connection requests initiated by a given entity and only preserve the one that appears to be active and suspend or delete all others. Since cookies also serve to deflect replay attacks, we suggest addition of time-variant information to the cookie equivalent to achieve the same protection from such attacks. Timestamp information may also be included in the nonce equivalent of hash of private key of the message initiator to serve as ancillary information that may be used in generating keys.

VI. REFERENCES

[1] G. Harkins, D. & Carrell, D., (1998) The Internet Key Exchange, RFC 2409. Available at ftp://ftp.isi.edu/in-notes/rfc2409.txt.

[2] Harkins, D. & Carrell, D., (2005) Internet Key Exchange (IKEv2) Protocol, RFC 4306. Available at ftp://ftp.isi.edu/in-notes/rfc4306.txt.

[3] Piscitello, D.M., (2002) "Security Parameters for Site-to-Site VPNs", Expert Editorial, Watch Guard Technologies. Available at http://www.corecom.com/external/livesecurity/secparams.htm.

[4] I Diffie, W. & Hellman, M.E., (1976) "New Directions in Cryptography", IEEE Transactions on Information Theory, vol. IT-22, November, pp 644–654.

[5] Piscitello, D.M., (2002) "Security Parameters for Site-to-Site VPNs", Expert Editorial, Watch Guard Technologies. Available at http://www.corecom.com/external/livesecurity/secparams.htm.

[6] Snader, J.C., (2005) *VPNs Illustrated: Tunnels, VPNs and IPsec*, Addison Wesley Professional, ISBN: 0-321-24544-X**.** Available at http://fengnet.com/book/.

[7] Thumann, M. & Rey, E., (2002) "PSK Cracking using IKE Aggressive Mode", Technical Report, Enno Rey, Netzwerke GmbH. Available at http://www.ernw.de/download/pskattack.pdf.

[8] Pilam, J., (1999) "Authentication Vulnerabilities in IKE and Xauth with Weak Preshared Secrets". Institute for Mathematics and its Applications. Available at http://www.ima.umn.edu/~pliam/xauth. Key fingerprint = AF19 FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46.

[9] Pitts, S., (2004) "VPN Aggressive Mode Pre-shared Key Brute Force Attack", Global Information Assurance Certification Paper, SANS Institute. Available at https://www.giac.org/paper/gcih/541/vpn-aggressive-mode-pre-shared-key-brute-force-attack/104625

[10] SecurityFocus Vulnerabilities, (1999) "IKE Aggressive Mode Shared Secret Hash Leakage Weakness", SecurityFocus. Available at http://www.securityfocus.com/bid/7423/info/.

[11] Malik, S., (2003) Advanced IPSec Algorithms and Protocols (2003), Session SEC-4010, Cisco Press Publications. Available at http://www.cisco.com/networkers/nw03/presos/docs/SEC-4010.pdf.

[12] Doraswamy, N. & Harkins, D., (2003) IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks, Second Edition, Prentice Hall, New Jersey, USA.

[13] Malik, S., (2002) *Network Security Principles and Practices*, Cisco Press Publications, November 15, ISBN: 1-58705-025-0.

[14] Perlman, R. & Kaufman, C., (2000) "Key Exchange in IPSec: Analysis in IKE", IEEE Internet Computing, Nov/Dec 2000.

[15] Perlman, R.J. & Kaufman, C., (2001) "Analysis of the IPSec Key Exchange standard", In: IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2001), 20-22 June 2001, Cambridge, MA, USA. Pp 150-156, IEEE Computer Society.

[16] Neuman, B. C. & Stubblebine, S. G., (1993) "A note on the use of timestamps as nonces", Operating Systems Review, Vol 27, issue 2, pp10–14.

[17] NagaLakshmi, V. & Rameshbabu, I., (2007) "A Protocol for Internet Key Exchange (IKE) using Public Encryption Key and Public Signature Key", International Journal of Computer Science and Network Security, Vol. 7, No.7, July, pp342-346.

[18] Yalamanchili, S. & Sambasiva Rao, K.V., (2011) "Authentication and Confidentiality in IKE using Dual Signature, Digital Enveloping and PGP", International Journal of Computational Intelligence and Information Security, vol 2, num 6, June 30.

[19] Cremers, C.J.F., (2011) **"**Key Exchange in IPSec revisited: Formal Analysis of IKEv1 and IKEv2", To appear in ESORICS 2011, September. Available at http://people.inf.ethz.ch/cremersc/downloads/download.php?file=papers/Cr2011-IKE.pdf.