iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** III **Month of publication:** March 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Performance study on Diffie Hellman Key Exchange Algorithm

K.Suganya[1], K.Ramya[2]

1. Associate Professor, Department of Software Engineering & IT [PG]

A.V.C College of Engineering, Mayiladuthurai

2. Assistant Professor& Head, Department of Science & Humanities

Kingston Engineering College, Katpadi, Vellore

Abstract— ***This paper introduces a security improvement that makes the Diffie-Hellman key agreement and encryption scheme more secure against attacks, such as the known plaintext attacks, it suggests the use of randomized parameter in both schemes, this will allows to produce a new shared secret key each time a communication session is build and to generate different encryption messages for all kinds of messages even for same message, thus making the Diffie-Hellman more secure compared with the basic version of the Diffie-Hellman.***

Keywords— ***DH, SSL, TLS, SSH, PKI, IETF***

I. INTRODUCTION

Diffie–Hellman key exchange (D–H) is a specific method of exchanging keys. It is one of the earliest practical examples of Key exchange implemented within the field of cryptography. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. It is a type of key exchange.

The scheme was first published by Whitfield Diffie and Martin Hellman in 1976, although it later emerged that it had been separately invented a few years earlier within GCHQ, the British signals intelligence agency, by Malcolm J. Williamson but was kept classified. In 2002, Hellman suggested the algorithm be called Diffie–Hellman–Merkle key exchange in recognition of Ralph Merkle's contribution to the invention of public-key cryptography (Hellman, 2002).

II.ALGORITHM DESCRIPTION

Diffie-Hellman key exchange, also called exponential key exchange, is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted, making the task of a would-be code breaker mathematically overwhelming.

ISSN: 2321-9653

The idea of public key cryptography was born as a result of two major challenges. The first of these was the problem of key distribution: if two people who have never met before are to communicate using digital systems as a medium, using conventional cryptography would mean that they must somehow agree on a common key that will be known to themselves and no one else. The other problem was the issue of signatures: this is a method of providing the recipient of a purely digital electronic message with a way of demonstrating to other people that it had come from a particular person, serving as a signature comparable to a written one on a letter.

The Diffie-Hellman Key Exchange Algorithm:

1. Global Public Elements: Prime number q; $\alpha < q$ and α is a primitive root of q.

2. User A Key Generation: User B Key Generation:

3. Select private XA $XA < q$

Select private XB $XB < q$

4. Calculate public YA

$$YA = \alpha\ XA\ mod\ q$$

Calculate public YB

$$YB = \alpha\ XB\ mod\ q$$

5. Calculation of Secret Key by User A:

$$K = (YB)\ XA\ mod\ q$$

Calculation of Secret Key by User B:

$$K = (YA)\ XB\ mod\ q$$

The result is that the two sides have exchanged a secret value. Furthermore, because XA and XB are private, an adversary only has the following ingredients to work with: q, a, YA, and YB. Thus, the adversary is forced to take a discrete logarithm to determine the key. For example, to determine the private key of user B, an adversary must compute $XB = dloga, q\ (YB)$. The adversary can then calculate the key K in the same manner as user B calculates it. The security of the Diffie-Hellman key exchange lies in the fact that, while it is relatively easy to calculate exponentials modulo q prime, it is very difficult to calculate discrete logarithms. For large primes, the latter task is considered infeasible.

Sender Side

1. Xa < q (user can select any random number less than q)
2. Ya = a Xa mod q (Ya is a public key of sender)
3. K = Yb Xa mod q (where Yb is a public key of receiver and K is a private key)
4. pow = 2K
5. pow = pow + q

Encrypt every letter of plain text using pow.

Receiver Side

1. Xb < q (user can select any random number less than q)

ISSN: 2321-9653

2. Yb = a Xb mod q (Yb is a public key of receiver)

3. K = Ya Xb mod q (where Ya is a public key of sender

and K is a private key)

4. pow = 2K

5. pow = pow + q

Decrypt every letter of Cipher text using pow.

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

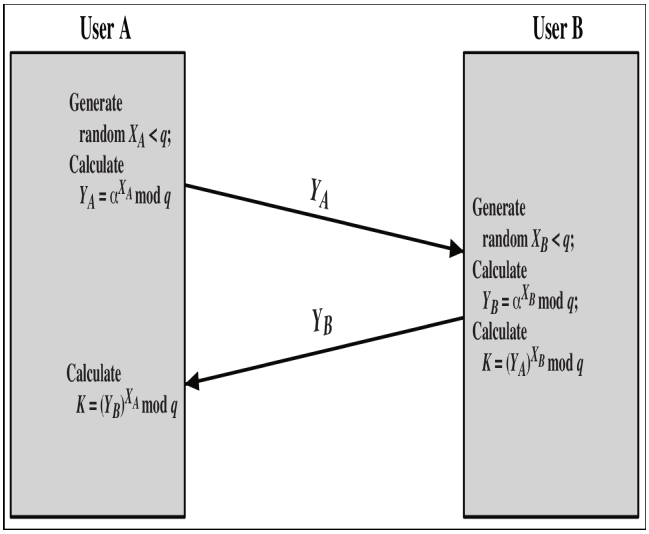
User A Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret Key by User A
$K = (Y_B)^{X_A} \bmod q$

Calculation of Secret Key by User B
$K = (Y_A)^{X_B} \bmod q$

Diffi Hellman key exchange algorithm



Key Exchange method

III.LIMITATIONS OF DH ALGORITHM

1. There is no identity of the parties involved in the exchange.

2. It is easily susceptible to man-in-the-middle attacks. A third party C, can exchange keys with both A and B, and can listen to the communication between A and B.

3.The algorithm is computationally intensive. Each multiplication varies as the square of n, which must be very large. The number of multiplications required by the exponentiation increases with increasing values of the exponent, x or y in this case.

4. The computational nature of the algorithm could be used in a denial-of-service attack very easily.

5.The algorithm cannot be used to encrypt messages.

ISSN: 2321-9653

6. There is also a lack of authentication.

IV. EXAMPLE OF DH ALGORITHM

Ex:1

For q = 7 check that 2 is not a primitive root of 7

and 3 is a primitive root of 7;

• Let *q = 7* and *a = 3* is publicly known numbers in

DH algorithm;

• Let XA = 4 and XB = 3 be private keys of A and B,

respectively;

• Then $YA=3^4 \bmod 7 = 4$

$YB= 3^3 \bmod 7 = 6$

• Common secret key $k = 6^4 \bmod 7 = 1$

Ex:2

Pick p = 13, a prime number.

Pick g = 2, a generator for Z13.

Alice :

Pick a random x = 3.

Compute X = gx mod p = 23 mod 13 = 8.

Bob :

Pick a random y = 7.

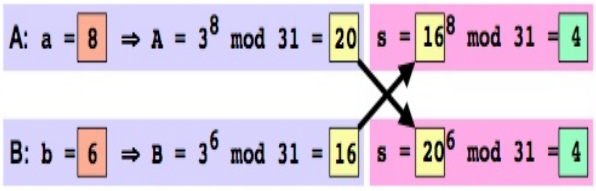
Compute Y = gy mod p = 27 mod 13 = 11.

Alice computes: K1 = Y x mod p = 113 mod 13 = 5.

Bob computes: K2 = Xy mod p = 87 mod 13 = 5.

⇒ K1 = K2 = 5.

Another diagrammatical Example



V.ADVANTAGES & DISADVANTAGES

Its advantages are the security factors with respect

to the fact that solving the discrete logarithm is very challenging, and that the shared key (i.e. the secret) is never itself transmitted over the channel.

Nonetheless, the algorithm has its share of drawbacks including the fact that there are expensive exponential operations involved, and the algorithm cannot be used to encrypt messages - it can be used for establishing a secret key

only. There is also a lack of authentication.

Another advantage of the Diffie–Hellman Algorithm is that, it is a lightweight two-pass protocol with only a public key

transport from participant A to participant B and again from B to A

Diffie-Hellman is currently used in many protocols, namely:

– Secure Sockets Layer (SSL)

- Transport Layer Security (TLS)

– Secure Shell (SSH)

– Internet Protocol Security (IPSec)

– Public Key Infrastructure (PKI)

VI. USES OF SECURE INTERNET PROTOCOLS IN DH

The Diffie-Hellman protocol has been applied to many security protocols including the *Security Sockets Layer (SSL)*, *secure shell (SSH)*, and *IP Sec*.

Secure Sockets Layer (SSL)

The SSL is the standard security technology developed by Netscape in 1994 to establish an encrypted link between a web server and a browser. This link ensures privacy and integrity of all data passed between the web server and browsers. SSL is used by millions of websites in the protection of their online transactions with their customers [9]. SSL is all about encryption. SSL uses certificates, private/public key exchange pairs and Diffie-Hellman key agreements to provide privacy (key exchange), authentication and integrity with *Message Authentication Code (MAC)*. This information is known as a *cipher suite* and exists within a *Public Key Infrastructure (PKI)*. SSL is useful for business/financial traffic, e.g. credit card transactions. SSL ensures confidentiality (it prevents eavesdropping), authenticity (the sender is really who he says he is), and integrity (the message has not been changed *en route*). It is possible that a user might not know SSL is used in the course of communication but they are likely to notice some blockages.

SSL/TLS is composed of two layers: the lower layer, called the *record protocol*, rides on TCP and manages the symmetric (private) cryptography so the communication is private and reliable. The upper layer is called the *handshake protocol* and it is in this layer that D-H is used. The handshake allows the server to authenticate itself to the client using public-key techniques, also called asymmetric encryption. It also allows the client and the server to cooperate in the creation of symmetric keys, which are used for rapid encryption and decryption. This implies that while communication is in progress the client and server exchange unencrypted handshake messages that include hellos and then information about which encryption, key exchange, and compression options they each accept and prefer [8]. During the SSL handshake, each computer generates a set of codes to encrypt information. From these codes, each computer creates two keys, one private and one public. Your computer keeps the private key secret, but sends out the public key to the other computer, which uses that key to encode subsequent messages so that only your computer can read them. The public key cannot, however, be used to decode the message; the decoding can only be done using the private key. These keys allow you and the other computer to lock and unlock information so that only the holder of the private key can read messages encrypted by the public key. Since only you and the other computer have a copy of your respective private keys, there is

no way for anybody else to intercept and decode your messages.

In SSL, the key exchange process uses D-H algorithm which is asymmetric (that is, public key) cryptography to ensure to each party that the other is who they say they are. After this exchange, keys are computed and the parties begin encrypting all traffic between them, using the computed keys and agreed upon methods.

Secure Shell (SSH)

SSH is a network security protocol very common for secure remote login on the Internet. The secure shell has come to replace the unsecured Telnet on the network and FTP on the system, mostly because both Telnet and FTP do not encrypt data, and instead send them in plaintext. SSH, on the other hand, can automatically encrypt, authenticate and compress transmitted data . The key exchange protocol itself is a component of the SSH as a whole, particularly responsible for parties agreeing upon the keys used by the various primitives later in the SSH protocol. This is the first stage of the SSH algorithm, and it happens before the establishment of session keys. The protocol proceeds in three stages. The first of these is the "*Hello*" phase, where the first identification is done. A list of supported algorithms is involved here after the first "*Hi*" message, and this list details the supported Diffie-Hellman key groups, among other things. The second stage sees the two parties agree upon a shared secret key *x*, which is done by an implementation of a Diffie-Hellman exchange. At the final stage, the shared secret key, session identifier and digest are used to generate the application keys. Currently, the "*diffie-hellman-group1-sha1*" method is practised in the key exchange, prescribing a fixed group on which all operations are performed. The key exchange is then signed with the host key to provide host authentication.

IP Security (IPSec)

IPSec (IP Security) is an extension of the *Internet Protocol (IP)*—it is a suite of protocols introduced by the *Internet Engineering Task Force (IETF)* to aid in configuring a communications channel between multiple machines. Operating at the IP layer of the seven-layer model, it does its job by authenticating and encrypting IP packets . Like the previous protocols, IPSec uses D-H and asymmetric cryptography to establish identities, preferred algorithms, and a shared secret. Before IPSec can begin encrypting the data stream, some preliminary information exchange is necessary. This is accomplished with the *Internet Key Exchange (IKE)* protocol. IKE uses DH to produce a shared secret via the usual mechanisms, and then authenticate each other; after that, the secret key is used for encryption purposes. This shared secret key is never exchanged over the insecure channel.

VII .Future of DH Algorithm

The cryptographic security standards used in public-key infrastructures, RSA and Diffie-Hellman, were introduced in the 1970s. And although they haven't been cracked, their time could be running out. That's one reason the National Security Agency wants to move to elliptic-curve cryptography (ECC) for cybersecurity. ECC, a complex mathematical algorithm used to secure data in transit, may replace Diffie-Hellman because it can provide much greater security at a smaller key size. ECC takes less computational time and can be used to secure information on smaller machines, including cell phones, smart cards and wireless devices.

Although Diffie-Hellman is a public-key algorithm, experts say it don't scale well for the future. At this point it is stated that Diffie-Hellman keys shorter than 900 bits are not secure enough. To make Diffie-Hellman keys, which now can go to 1,024 bits, secure for the next 10 to 20 years, organizations would have to expand to key lengths of at least 2,048 bits, according to Stephen Kent, chief scientist at BBN Technologies. Eventually, key sizes would need to expand to 4,096 bits. Scientists from the NIST's security technology group assume, that it is highly possible, that Diffie-Hellman will be broken within a decade or so

VIII . CONCLUSION

Designing a Key exchange algorithm with 100% Accuracy is not at all possible. . Our Algorithm uses simple mathematical concepts making implementation easier as well as avoidance from common Attacks. Security improvement is beneficial because Diffie Hellman Algorithm is the basis of several security standards and services on the internet, and if the security of the Diffie Hellman algorithm is compromised, such systems will collapse.Diffie Hellman key exchange approach for key distribution appears to be one of the preferred methods used in practice today.

REFERENCES

1. Rescorla, E., Diffie-Hellman Key Agreement Method, RFC 2631, IETF Network Working Group, http://www.ietf.org/rfc/rfc2631.txt

2. RSA Laboratories, RSA Laboratories' FAQ About Today's Cryptography, Version4.1, RSA Security Inc., 2000, http://www.rsa.com/rsalabs/faq/index.html

3. Costas Christoyannis, "What is Diffie-Hellman", http://www.hack.gr/users/dij/crypto/overview/diffie.html

4. Levy, Benjamin, "Diffie-Hellman Method for Key Agreement", http://apocalypse.org/pub/u/seven/diffie.html

5. RSA Laboratories, PKCS #3: Diffie-Hellman Key-Agreement Standard, Version 1.4.Revised November 1, 1993, http://www.rsalabs.com/pkcs/pkcs-3/index.html

6. Behour A. Forouzan, Sophia Chung Fegan ―Data Communication and Networking, Fourth Edition 2009.

7. Priyanka Goyal, Sahil Batra, and Ajit Singh. ―A literature review of security attack in mobile ad-hoc networks‖. International Journal of Computer Applications, 9(12):11–15, November 2010.

8. Hai Huang and Zhenfu Cao. ―An ID-based authenticated key exchange protocol based on bilinear Diffie–Hellman problem. Department of Computer Science and Engineering, Shanghai Jiaotong University, ASIACCS, 2009.

9. Jooyoung Lee and Je Hong Park. ―Authenticated key exchange secure under the computational Diffie–Hellman assumption.The Attached Institute of Electronics and Telecommunications Research Institute, Korea, IACR, 2008.

10.ElGamal T. ―A public-key cryptosystem and a signature scheme based on discrete logarithms‖, IEEE Transactions on Information Theory, volume 31, pages 469-472. 1985.

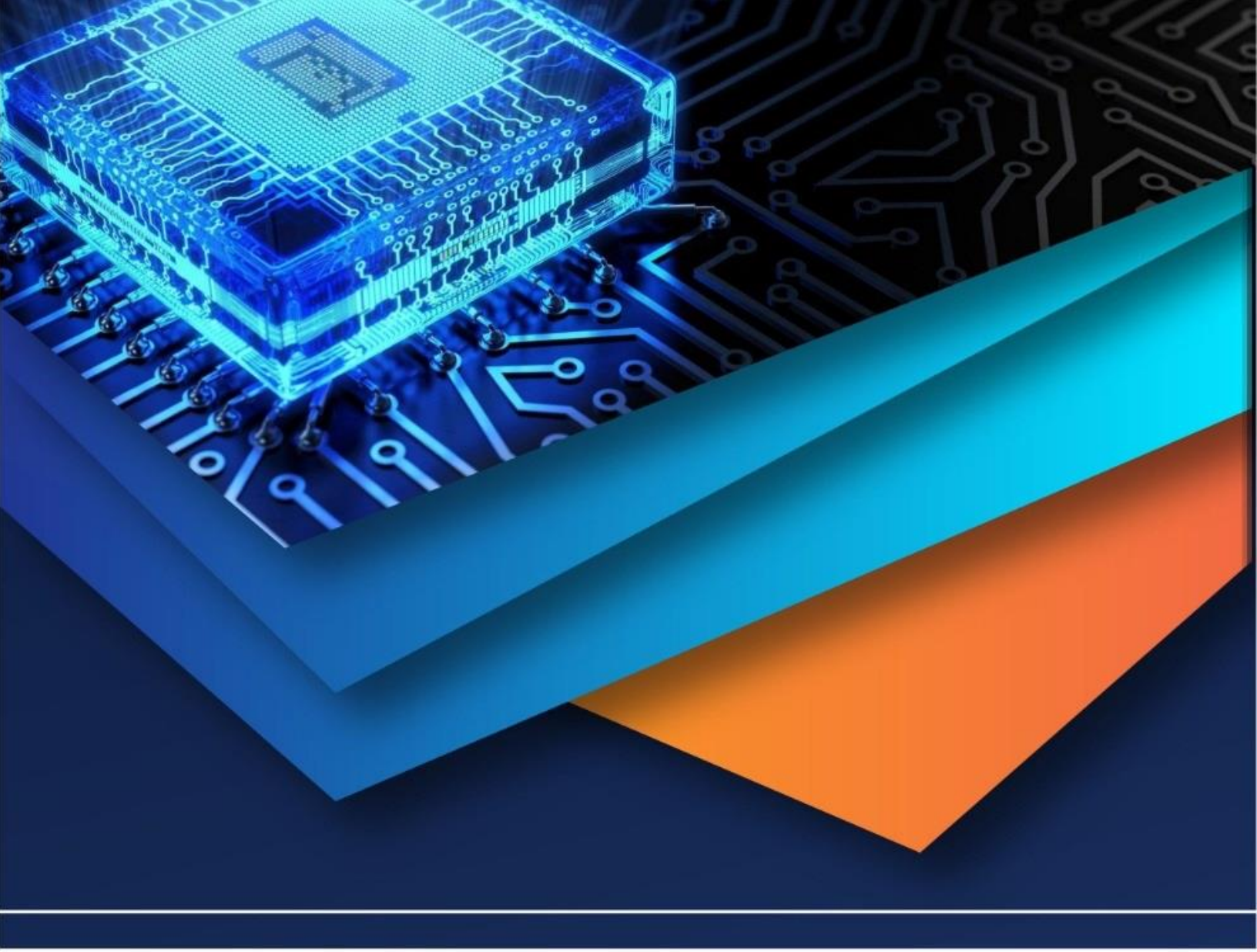
11. Francois J., Raymond A. ―Seurity Issues in the Diffie-Hellman Key Agreement Protocol‖, IEEE Trans. on Information Theory, pages 1–17 ,1992

12. William Stallings ―Cryptography and Network Security, Fourth Edition 2006.

13. Malek Jakob Kakish ― Security Improvments to the Diffie-Hellman Schemes‖ IJRRAS, volume 8,issue 1,july 2011.

14. Francois J., Raymond A., ― Security Issues in the Diffie-Hellman Key Agreement Protocol‖, IEEE Trans. on Information Theory, pages 1–17. 1998

15. Benjamin Arazi, ―Message Authenticaiton in Computationally Constrained Environments‖,IEEE Trans. Mobile Computing ,Vol.8 ,No.7 July 2009.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY