Improving the Diffie-Hellman Key Exchange Algorithm by Proposing the Multiplicative Key Exchange Algorithm

Sharad Boni
Student, Information Technology Department
Sardar Patel Institute of Technology

Jaimik Bhatt
Student, Information Technology Department
Sardar Patel Institute of Technology

Santosh Bhat
Student, Information Technology Department
Sardar Patel Institute of Technology

ABSTRACT
Exchanging cryptographic keys has been a problem with respect to security. Whitfield Diffie and Martin Hellman proposed the Diffie-Hellman key exchange algorithm to overcome the problem. Since then, the concept of key exchange over an unsecured network has completely been revolutionized. The algorithm is based on using arithmetic calculations for transmission of the shared session keys. The purpose of this algorithm is to enable users to securely exchange keys which can be used for later encryptions. This ability to securely exchange session keys dynamically and publicly between a group of users has become the foundation for secure group applications such as distributed computing, distributed databases and conference calls. Man-in-the-middle attacks are better secured using the Diffie-Hellman algorithm. Over time, Diffie-Hellman algorithm has been altered several times by various authors. However, some limitations to the Diffie-Hellman algorithm still persist. One of the limitations of the Diffie-Hellman algorithm is that it is computationally intensive thereby increasing the time complexity when generating public keys. The proposed algorithm has similar grounds with the Diffie-Hellman algorithm, and a new technique is used for sharing session keys which overcome the time complexity limitation of the Diffie-Hellman algorithm. The proposed "Multiplicative Key Exchange Algorithm" uses simple arithmetic equations to generate and exchange keys over an insecure network.

General Terms
Security, Diffie-Hellman, Key Exchange Algorithm, Encryption, Decryption.

Keywords
Security; Diffie-Hellman; Key Exchange; Multiplicative; time complexity

1. INTRODUCTION
The world today has been transformed into a global marketplace where e-commerce and ERP have made the corporate world start thinking from scratch again. There are many benefits for industries to learn and adapt from latest technologies. But with advancements in technology comes the need for more safety measures to protect critical data from the unauthorized persons like an eavesdropper. Nonetheless, the chief transfer medium of significant information is the Internet and it is widely accepted as an unsafe way for information exchange, where bad guys are also present with the good guys. Hence, it is very important to figure out the loopholes in the security measures and countermeasures that can be taken in advance for the prevention data loss. Diverse countermeasures that are employed in the present era are cryptography, steganography, firewalls, access lists, proxy application gateways, security protocols like SSL, TLS, etc[4].

Cryptography is the stand-out security strategy that has evolved over decades, especially after the introduction and growth of computers.

Cryptography is broadly described as the art and science of scrambling data to prevent unauthorized access over unsecure channel transmission. Encryption is basically classified into three categories, the first being symmetric key cryptography, second, the asymmetric encoding and finally the hash function. Secret key or symmetric key cryptography was the first one to develop. It uses a single secret or private key for encrypting and decrypting data on sender's and receiver's end respectively. The decryption process at the receivers end is exactly opposite to the encryption process at the senders end. However, one major problem associated with it is the key distribution problem i.e. distributing the key to the receiving party over the internet. This is a major setback because if the key is detected by the eavesdropper, then the cipher text can be discovered easily [5].

This key distribution problem is symmetric encryption was addressed by the research project undertaken by Whitfield Diffie (research scholar) and Martin Hellman (professor) of M.I.T University, U.S.A in 1976. The algorithm developed by them is popularly known as "Diffie-Hellman" algorithm, that is used for securely exchanging a shared secret between two parties, in real-time, over an un-trusted network[6].

A shared secret is important between two parties who may not have ever communicated previously so that they can encrypt their communications. It is basically a key agreement protocol that maintains secrecy between two parties for key exchange. Key agreement is a method in which the device communicating in the network establishes a shared secret between them without exchanging any secret data. In this method the devices that need to establish shared secret between them exchange their public keys. Both the devices on receiving the other device's public key perform key generation operation using its private key to obtain the shared secret [7]. Due to its superior security attribute, it is widely used and has changed a lot with the pace of time. Several researchers have modified the Diffie-Hellman algorithm and used it over many security protocols like Secure Socket Layer (SSL), Internet Protocol Security(IPSec), etc.[8]. The next section describes the Diffie-Hellman algorithm.

2. DIFFIE-HELLMAN ALGORITHM
The cryptographic private key is required for the transfer of data in asymmetric encryption. The crucial part in this type of encryption is the exchange of the encryption key from the sender to receiver without being intercepted by anyone in between. The Diffie-Hellman algorithm made the transfer or exchange of the same cryptographic key possible on both sides secretively. The Diffie-Hellman algorithm was the first public key algorithm that was published first in 1976. It was

the collaborative effort of Whitfield Diffie and Martin Hellman to establish a first practical method to share a secret over an unprotected channel. However, it is also believed that this method was first invented by Malcolm Williamson of U.K; however he did not publish it[1]. Though the Diffie Hellman algorithm is a bit time consuming, but it is the sheer strength of this algorithm which makes its use so popular in encryption key generation. The primary purpose of the algorithm is to enable users to securely exchange a key that can be used for ensuing encryption. This cryptographic problem guarantees that apart from user A and B no other participants can learn any information about the agreed value and also ensures A and B that their respective partner has actually calculated this value[2]. The process of Diffie Hellman algorithm is described as follows:

i. Both parties A and B agree upon two constants p and g. Where p is a prime number and g is the generator less than p.

ii. Both A and B choose their private keys a and b respectively such that they are random numbers and less than p.

iii. Let g^a mod p and g^b mod p be the public keys of A and B respectively.

iv. Then A and B exchange their public keys over a unsecure medium like the internet.

v. Then party A computes (g^b mod p) g^a mod p that is equal to g^{ba} mod p.

vi. Also party B computes (g^a mod p) g^b mod p that is equal to gab mod p.

vii. The shared secret key K is computed as

$$K = g^{ba} \bmod p = g^{ab} \bmod p.$$

The Diffie-Hellman algorithm states that it is computationally infeasible to determine the value of K by just observing the conversation and knowing the public keys.[3]

However, the Diffie-Hellman Algorithm still remains computationally intensive thereby increasing the time complexity when generating public keys which the proposed algorithm aims to resolve. Therefore, this paper makes a comparative study over Diffie-Hellman and the proposed algorithm approach with respect to time complexity.

3. THE PROPOSED "Mulplicative Key Exchange" ALGORITHM

In this paper a new public key cryptographic algorithm named "Multiplicative Key Exchange Algorithm" is proposed. It is different from the Diffie Hellman algorithm as it uses multiplication instead of exponential powers.

The algorithm is as follows:

i. Let 'g' be a prime number.

ii. Alice and Bob are two parties and 'g' is known to both the parties after they have agreed to a number.

iii. Alice thinks of a prime number 'a' and Bob thinks of a prime number 'b' then,

iv. A= g x a mod(g +1) and B= g x b mod(g +1) where A and B are intermediate keys.

v. Now, Alice and Bob exchange their intermediate keys A and B.

vi. So, Alice has the intermediate key with value B and Bob has intermediate key with value A.

vii. Finally, the common shared key is established as C = (B x g x a)mod(g +1) and C= (A x g x b)mod(g +1).

Refer Figure 1 below for complete illustration of the Mulplicative Key Exchange Algorithm.

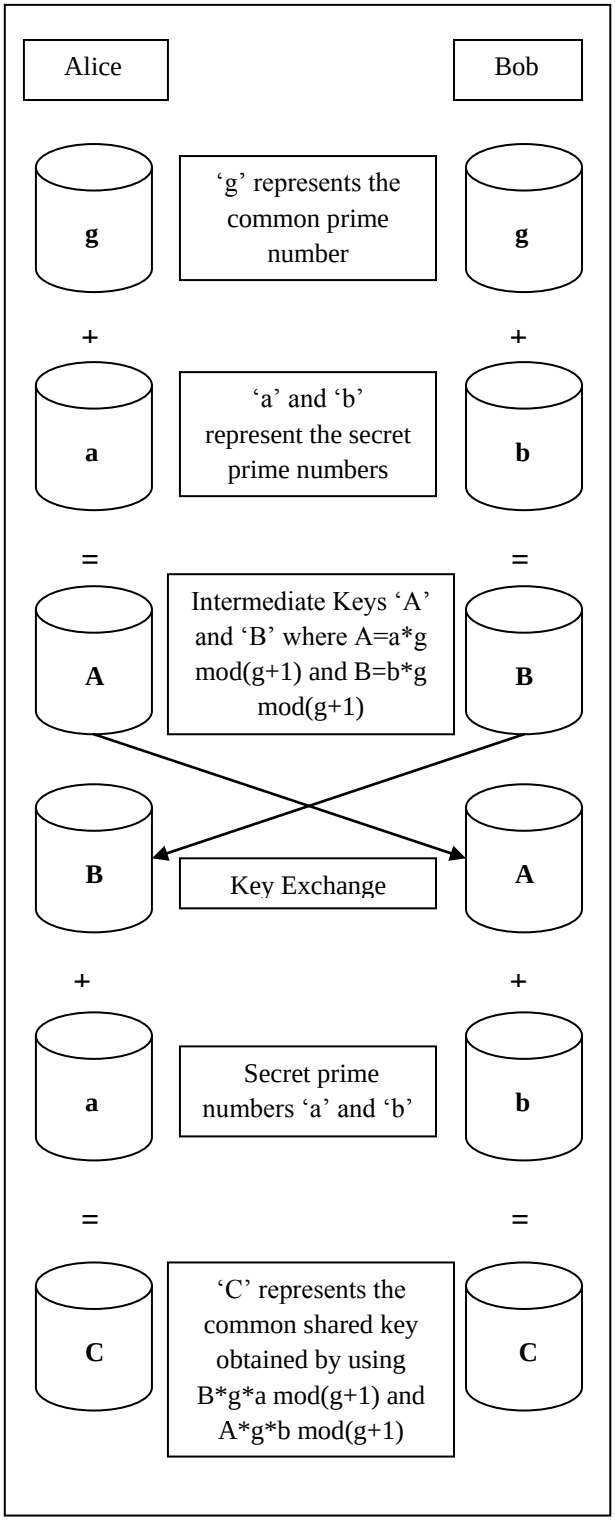

Figure 1.Illustration of Multiplicative Key Exchange Algorithm

4. SECRACY CHART

The chart described below shows what knowledge is being possessed by the users Alice, Bob and Eve, with some known values and some unknown values. Here Eve is an eavesdropper—she watches what is sent between Alice and Bob, but she does not alter the contents of their communications.

- g = common prime number, known to Alice, Bob, and Eve.
- a = private key known only to Alice.
- b = private key known only to Bob.
- A = public key of Alice, also known to Bob, and Eve.
- A = g x a mod (g+1)
- B = public key of Bob, also known to Alice, and Eve.
- B = g x b mod (g+1)
- C = common shared key

The table below shows the known and unknown parameters of Alice, Bob, Eve.

Table 1. Secrecy Chart

Alice	Bob	Eve
Unknown		
B	a	a, b, C
Known		
g = 3	g = 3	g = 3
a=2	b = 5	A = 2
A = 5 x 2 mod 4	B = 3 x 5 mod 4	B = 3
B = 3	A = 2	C =2 x 3 x a mod 4 =3 x 3 x b mod 4
C = B x g x a mod 4	C = A x g x b mod 4	
C = 3 x 3 x 2 mod 4 = 2	C = 2 x 3 x 5 mod 4 = 2	
C = 2	C = 2	

Since Eve cannot get the values of 'a' and 'b', so Eve is unable to get access to 'C', the common shared key.

Note: It would become difficult for Alice and for Bob to find out each other's private keys. If it is not difficult for Alice to solve for Bob's private key (or vice versa), Eve may simply replace her own private / public key pair, combine Bob's public key into her private key and produce a fake shared secret key, and solve for Bob's private key (and use that to solve for the shared secret key. Eve may attempt to choose a public / private key pair that will make it easy for her to solve for Bob's private key)[9].

This algorithm, like the Diffie-Hellman algorithm can be extended in cases where more than two parties are involved.

5. EXPERIMENTATION

Below are the values tested for the Diffie-Hellman as well as Multiplicative algorithm. The computation time recorded is in nano-seconds .The system configuration used for the testing comprises of Intel core i3 - 4005U 1.7GHz,4 GB RAM , 64-bit Windows 10 OS. See the table below.

Table 2. Results

g	P	A	b	Diffie-Hellman	Multiplicative
5	2	2713	447019	1035425	477625
5	2	2713	101342203111	1366881	677182
5	2	2713	639264958483	1533454	922525
5	2	2713	343579	239084	82713
5	2	4451	9421129	324815	62186
5	2	5297	44204857	376738	73053
5	2	6551	78306203	423226	67620
5	2	8719	964034563	360437	60978
5	2	9413	5738314027	431678	79694

6. PERFORMANCE ANALYSIS

As seen from the Results table above the multiplicative algorithm requires less computation time than Diffie-Hellman algorithm. Also the multiplicative algorithm is approximately ten times faster than Diffie-Hellman algorithm. The graph below depicts the comparison of the two algorithms.

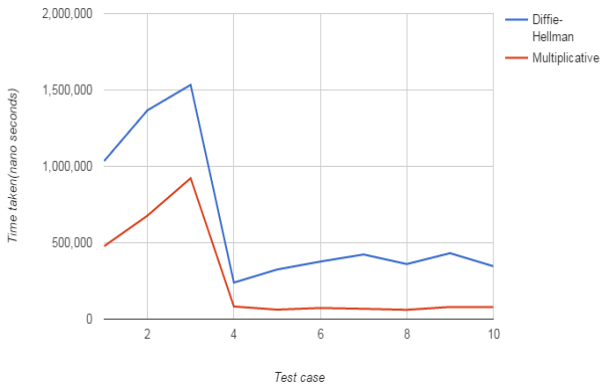

Figure 2. Comaprison analysis of two algorithms

7. CONCLUSION AND FUTURE WORK

This paper proposed a novel algorithm to counter the shortcomings, i.e. the huge time taken to compute the original key by the two communicating parties, of Diffie-Hellman algorithm. Experimental results show that the proposed Multiplicative algorithm can effectively lower the computation time as compared to the Diffie-Hellman Algorithm .This algorithm can find implementations in the

area where speed of generation of keys is more important than a subtle decrease in security and where the devices are not high end or have lower end configurations. In future, the proposed system can be modified in order to provide a better security mechanism and also the particular fields where it can be applied.

8. ACKNOWLEDGMENTS

Sincere thanks to the experts who have contributed towards development of the Diffie-Hellman key exchange algorithm.

9. REFERENCES

[1] Y. Amir, Y.Kim & C. Nita-Rotaru, " Secure communication using contributory key agreement", IEEE Transactions on Parallel and Distributed systems, pp. 468-480, 2009.

[2] International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Web Site: www.ijettcs.org Email: editor@ijettcs.org Volume 4, Issue 1, January-February 2015 ISSN 2278-6856

[3] Akhil Kaushik & Satvika, "Extended Diffie-Hellman Algorithm for Key Exchange and Management.", Special Issue: Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM 2013.

[4] A. J. Menezes, P. C. V. Oorschot, & S. A. Vanstone, "Handbook of Applied Cryptography", 5th edn., CRC Press Inc., USA, 2001.

[5] C. Bissell & A.K. Vladimir, "Pioneer of the sampling theorem, cryptography, optimal detection, planetary mapping". [History of Communications], IEEE Communications Magazine, Vol. 47, No.10, pp. 24 - 32, Oct 2009.

[6] RFC 2631, Diffie-Hellman Key Agreement Method, June 1999, Available at http://tools.ietf.org/html/rfc2631

[7] D. Wallner, E. Harder, & R. Agee, "Key management for multicast: Issues and architectures", Internet Draft (Work in progress), July 1998.

[8] S. B. Wilson & A. Menezes, "Entity authentication and authenticated transport protocols employing asymmetric techniques", SPRINGER 1997.

[9] *Buchanan, Bill*, "Diffie-Hellman Example in ASP.NET", *Bill's Security Tips*, retrieved 2015-08-27.