International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Improvement of Diffie-Hellman Key Exchange Algorithm

Gurshid

Netaji Subhas Institute of Technology, Delhi, New Delhi-110078, India

Abstract: ***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. The purpose of this paper is to propose an algorithm which is an improvement over the Diffie-Hellman key exchange. The algorithm is based on using arithmetic and logarithmic calculations for transmission of the shared session keys which enable users to securely exchange keys which further can be used for later encryptions. 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 its 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 algorithm uses simple arithmetic and logarithmic equations to generate and exchange keys over an insecure network.***

Keywords: Key Exchange, Diffie-Hellman Protocol, Security, Time Complexity

1. Introduction

It is reasonable to assume that people in any civilization, anywhere in this world tried to conceal information in written form as soon as writing was developed. This is probably the first and primitive form of encryption but is only one half of cryptography, the other half is the ability to recreate the original message from its concealed form. Cryptography is not hiding a message, so that no one can find it, but rather to leave the message in public in such a way that no one except the intended recipient understands the message.

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

The first recorded use of cryptography for correspondence was by the Spartans who (as early as 600 BC) employed a cipher device called “scytale” to send secret communications between military commanders.

The scytale consisted of a wooden baton wrapped with a piece of parchment inscribed with the message. Once unwrapped the parchment shrunk and appeared to contain some incomprehensible marks; however, when wrapped around another baton of identical dimensions the original text appears.

In the past military uses of cryptography were the main motivations behind the study of cryptography. It was a secret endeavor, mostly undertaken by big governments, who could hide all the efforts and create smokescreens necessary to hide a lot of people, activities and active researchers. In those days most of the cryptosystems were private or symmetric key cryptosystems.

In this two users select a key in advance, which is their private key, then they use the key in a private key cryptosystem to communicate data over the public channel. Military establishments and diplomatic offices normally have staffs, procedures and protocols in place to handle this key selection by two users and ways to change these keys periodically. Secret or private key cryptography is still the backbone of modern day cryptography, but it falls short of today’s needs.

In 1976, Whitfield Diffie and Martin Hellman who were influenced by the work of Ralph Merkle on public key distribution, proposed an algorithm for key exchange which uses exponentiation in a finite field. Today,

Diffie-Hellman is used in a variety of protocols and services. It is used in interactive transactions, rather than a batch transfer from a sender to a receiver. The algorithm is used when data is encrypted on the Web using either SSL or TLS and in VPN. So its security is of utmost importance. However, like other security algorithm it is vulnerable to various attacks like plaintext attacks, man-in the middle attacks etc. So we propose a modification of the original algorithm so as make it more tolerant and secure by using a random parameter.

2. Diffie-Hellman Algorithm

To implement Diffie-Hellman, the two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive whole numbers p and q, such that p is a prime number and q is a generator of p. The generator q is a number that, when raised to positive whole-number powers less than p, never produces the same result for any two such whole numbers. The value of p may be large but the value of q is usually small.

Once Alice and Bob have agreed on p and q in private, they choose positive whole-number personal keys a and b, both

Licensed Under Creative Commons Attribution CC BY

less than the prime-number modulus p. Neither user divulges their personal key to anyone; ideally they memorize these numbers and do not write them down or store them anywhere. Next, Alice and Bob compute public keys A1 and B1 based on their personal keys according to the formulas.

$$A1 = q^a \bmod p \quad \text{and} \quad B1 = q^b \bmod p$$

The two users can share their public keys a* and b* over a communications medium assumed to be insecure, such as the Internet or a corporate wide area network (WAN). From these public keys, a number x can be generated by either user on the basis of their own personal keys.
Alice computes x using the formula: $A2 = (B1)^a \bmod p$
Bob computes x using the formula: $B2 = (A1)^b \bmod p$

The value of x turns out to be the same according to either of the above two formulas. However, the personal keys a and b, which are critical in the calculation of x, have not been transmitted over a public medium. Because it is a large and apparently random number, a potential hacker has almost no chance of correctly guessing x, even with the help of a powerful computer to conduct millions of trials. The two users can therefore, in theory, communicate privately over a public medium with an encryption method of their choice using the decryption key A2 or B2.

The most serious limitation of Diffie-Hellman Algorithm in its basic or "pure" form is its time complexity. Since, it is 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. Modification to Diffie-Hellman Algorithm

3.1 Description

Here too, first we select a prime number 'p';
Alice then selects a random natural number 'a' as its private key. Its public key is calculated as $(p / \log_{10} a) \bmod (p+1)$.

Similarly, the Bob selects a random natural number 'b' as its private key. Its public key is calculated as
$(p / \log_{10} b) \bmod (p+1)$.

The public keys are then exchanged over a public channel.
The shared secret key is then calculated by both Alice and Bob using their own private key and the other's public key. This secret key is used to encrypt and decrypt the message.

The difference in this algorithm is that there is no need of any calculation involving q. This algorithm also calculates A2 and B2 in logarithmic time as oppose to exponential time in the diffie-hellman algorithm.

3.2 Algorithm

1) Let p = {P | P ∈ a prime number}.
p = Common Key.

2) User 1 :
- Select a = Private key
- $A1 = (p / \log_{10} a) \bmod (p+1)$

3) User 2 :
- Select b = Private Key
- $B1 = (p / \log_{10} b) \bmod (p+1)$
- A1, B1 = Public Key

4) Keys are exchanged.

5) User 1:
- $A2 = (B1 / \log_{10} a) \bmod (p+1)$

6) User B :
- $B2 = (A1 / \log_{10} b) \bmod (p+1)$

A2, B2 = Shared Secret between the two users

Both users have the same shared secret as A2=B2.

3.3 Experimentation

Below in the table are some values tested for the Diffie-Hellman as well as the improved algorithm. The computation time recorded is in nanoseconds. The system configuration used for the testing comprises of Intel core i3, 3.07GHz, 4 GB RAM, 64-bit Windows 10 OS. See the table below.

p	q	a	b	Diffie-Hellman Algorithm	Improved Algorithm
5	2	1234	7894	112563	61235
5	2	1234	2136540	1983145	222321
5	2	1234	11236547899	4896541	922131
7	6	6896	9213601	3103364	286414
9	8	28734	665423159	3796451	563652
11	9	88845	1123654789	4213546	774521

3.4 Analysis and Improvement

As seen from the table above the improved algorithm requires less computation time than Diffie-Hellman algorithm. Also the improved algorithm is more than 10 times faster than Diffie-Hellman algorithm. The graph below depicts the comparison between the two algorithms.

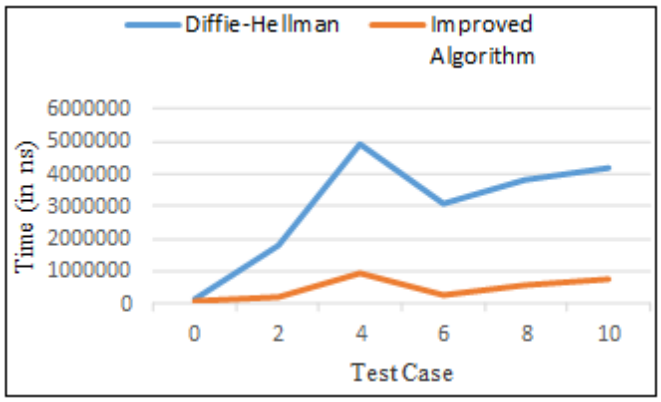

Figure 1: Comparison between both algorithms

Licensed Under Creative Commons Attribution CC BY

3.5 Conclusions

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 improved 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.

3.6 Acknowledgment

I would like to thank the experts who have contributed towards the development of Diffie-Hellman Algorithm.

References

[1] D. Wallner, E. Harder, & R. Agee, “Key management for multicast: Issues and architectures”, Internet Draft (Work in progress), July 1998
[2] A. J. Menezes, P. C. V. Oorschot, & S. A. Vanstone, “Handbook of Applied Cryptography”, 5th edn., CRC Press Inc., USA, 2001
[3] 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
[4] Y. Amir, Y.Kim & C. Nita-Rotaru, “ Secure communication using contributory key agreement”, IEEE Transactions on Parallel and Distributed systems, pp. 468-480, 2009
[5] Diffie W., Hellman M., 1976. “New directions in cryptography”, IEEE Transactions on Information Theory, volume 22, pages 644-654
[6] Diffie, Whitfield, and Martin E. Hellman, 1977. "Special feature exhaustive cryptanalysis of the NBS data encryption standard." Computer 10.6, pages: 74-84
[7] Diffie, W., & Hellman, M. E. (1979). Privacy and authentication: An introduction to cryptography. Proceedings of the IEEE, 67(3), pages 397-427
[8] Martin E. Hellman May 2002. An Overview of Public Key Cryptography, IEEE Communications Magazine, pages: 42–49

Licensed Under Creative Commons Attribution CC BY