Enhanced diffie-hellman algorithm for reliable key exchange

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042015 doi:10.1088/1757-899X/263/4/042015

Enhanced diffie-hellman algorithm for reliable key exchange

Aryan, Chaithanya Kumar and Durai Raj Vincent P M

School of Information Technology and Engineering, VIT University, Vellore-632014, Tamil Nadu, India.

E-mail: aryaann44@gmail.com

Abstract. The Diffie -Hellman is one of the first public-key procedure and is a certain way of exchanging the cryptographic keys securely. This concept was introduced by Ralph Markel and it is named after Whitfield Diffie and Martin Hellman. Sender and Receiver make a common secret key in Diffie-Hellman algorithm and then they start communicating with each other over the public channel which is known to everyone. A number of internet services are secured by Diffie –Hellman. In Public key cryptosystem, the sender has to trust while receiving the public key of the receiver and vice-versa and this is the challenge of public key cryptosystem. Man-in-the-Middle attack is very much possible on the existing Diffie-Hellman algorithm. In man-in-the-middle attack, the attacker exists in the public channel, the attacker receives the public key of both sender and receiver and sends public keys to sender and receiver which is generated by his own. This is how man-in-the-middle attack is possible on Diffie- Hellman algorithm. Denial of service attack is another attack which is found common on Diffie-Hellman. In this attack, the attacker tries to stop the communication happening between sender and receiver and attacker can do this by deleting messages or by confusing the parties with miscommunication. Some more attacks like Insider attack, Outsider attack, etc are possible on Diffie-Hellman. To reduce the possibility of attacks on Diffie-Hellman algorithm, we have enhanced the Diffie-Hellman algorithm to a next level. In this paper, we are extending the Diffie –Hellman algorithm by using the concept of the Diffie -Hellman algorithm to get a stronger secret key and that secret key is further exchanged between the sender and the receiver so that for each message, a new secret shared key would be generated. The second secret key will be generated by taking primitive root of the first secret key.

1. Introduction

Modern cryptography is used in Computer and Communication. Modern cryptography uses binary-bit sequence. Study of cryptosystem is called Cryptology. Cryptography and Cryptanalysis are the two branches of Cryptology. Breaking and getting the information part comes under Cryptanalysis. Cryptography provides information security by giving Confidentiality, Data Integrity, Authentication, and Non-repudiation to services. A cryptosystem is the execution of cryptographic techniques. A cryptosystem is made up of plain text, encryption algorithm, cipher text, decryption algorithm, the encryption key and decryption key. There are two basic types of a cryptosystem, symmetric key cryptosystem and asymmetric key cryptosystem.

In Symmetric key cryptosystem, the same key is used for encryption and decryption. Key establishment and Trust Issue are the two main challenges in symmetric key cryptosystem. In

Asymmetric key cryptosystem, different key is used for encryption and decryption. Here, the keys are mathematically related to each other. Asymmetric key cryptosystem is also called as Public key cryptosystem. The Public-key cryptography comes in the picture along with the Diffie-Hellman algorithm. Public-Key Cryptography means simply Asymmetric Cryptography which uses public and private keys to encrypt and decrypt the message. This concept was used by Diffie and Hellman to exchange the secret key for sending and receiving the messages. One of the most critical problem in cryptography is exchanging the key between two communicating devices. It was not about establishing a shared-secret key, but it was about to do it in such a way that anyone who is there at the communication between the devices do not find out the key. Diffie- Hellman algorithm was first creditor was Ralph Merkle and this algorithm is named after Whitfield Diffie and Martin Hellman. This algorithm makes the key exchange secure over a public channel.[1][2] The Diffie – Hellman is used for public key cryptography, SSL, SSH, PGP and other PKI systems. Many web services uses Diffie–Hellman for reliable communication and for securing purpose.[3] The most amazing thing in Diffie-Hellman key exchange is the communication between sender and receiver will happen over the public channel and for attacker now it's becoming possible. Some attacks which is possible on the Diffie-Hellman algorithm: man-in-middle attack, plain-text attack, logjam attack, etc. Logjam attack is a new type of attack found on the Diffie-Hellman key-exchange protocol which is used in TLS.
We have proposed an upgraded Diffie-Hellman algorithm for more secure and reliable key exchange and for reliable information exchange between the sender and the receiver.

2 The Diffie-Hellman Algorithm

The Diffie-Hellman algorithm uses mathematics i.e. modular arithmetic and discrete logarithm for making a common key for both sender and receiver using the communication channel where sender and receiver choose a common prime number p and q as it's primitive root, where $q<p$.[4]
There are many attacks possible on Diffie-Hellman algorithm like known plaintext attack, man-in-the-middle attack, Insider attack, Outsider attack, etc. The known plaintext attack uses plain text and cipher text from the public channel to get access of the secret key. In Man-in-the-middle attack, attacker keeps the public keys of both the sender and receiver and send them fake public keys.
Fig.1 reflects a clear picture of the Diffie-Hellman algorithm. Here, Ram and Sita agree on a prime number p, q as its prime number, over a public channel which is known to everyone and attacker Ravan is also there on public channel. Now, Ram and Sita chooses their own so called private key which is 'a' and 'b' respectively.
Now, for exchanging the key, Ram will calculate his public key which is $A=q^a \bmod p$ and will exchange it with the public key of Sita which is $B= q^b \bmod p$. Now, for getting the shared secret key, Ram and Sita will calculate private key and they find that they got a shared-secret key $S=B^a \bmod p= A^b \bmod p$. The Diffie-Hellman algorithm gives a shared-secret key by taking the primitive root of the prime number and finding the public keys and exchanging it over the public channel and by using modular arithmetic.

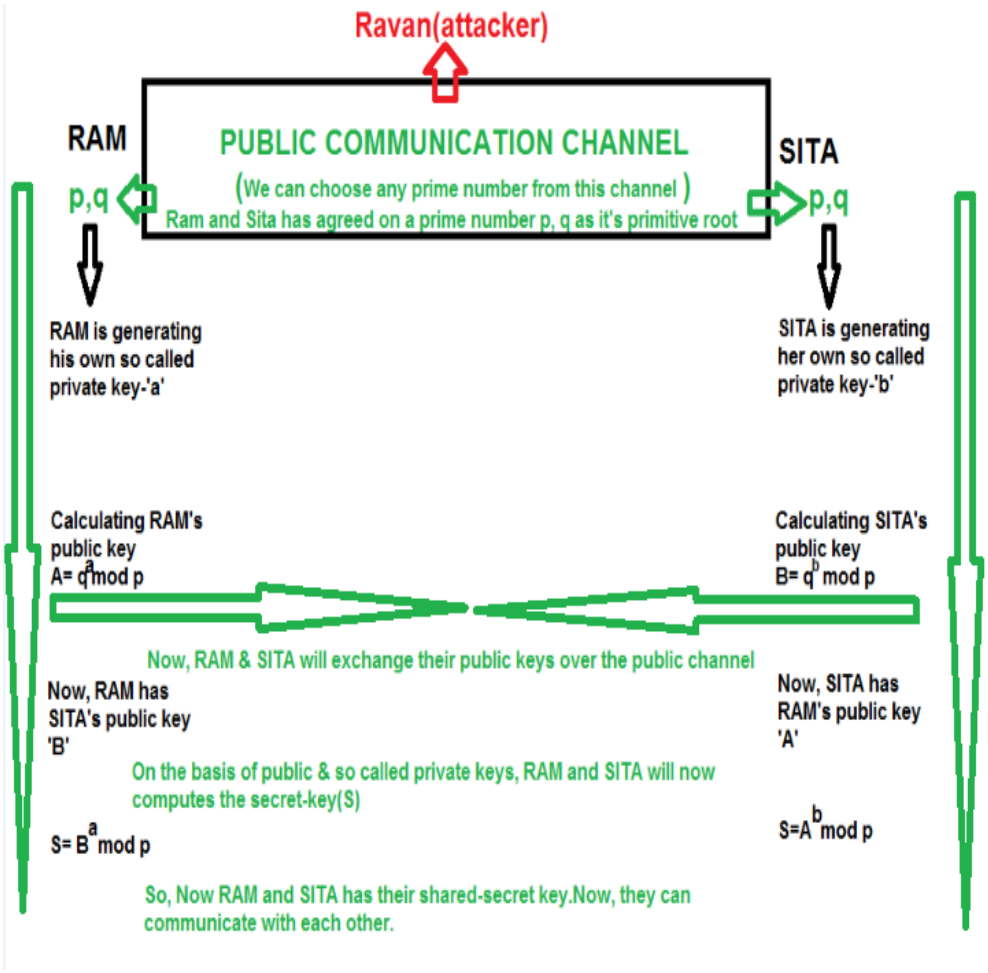

Fig. 1. The Diffie-Hellman key exchange.

THE DIFFIE-HELLMAN ALGORITHM:

1. Sender and Receiver agree on a prime number p, q as it's primitive root.
2. Sender and Receiver choose their so called private key 'a' and 'b' which is known to themselves only respectively.
3. Sender's public key A= q^a mod p.
4. Receiver's public key B= q^b mod p.
5. Sender and Receiver exchange their public key. Now Sender has B and Receiver has A.
6. Sender calculates B^a mod p= q^{ba} mod = S.
7. The receiver calculates A^b mod p= q^{ba} mod p= S.
8. Hence, the sender and receiver get 'S' as their shared secret key.

This is how the key exchange takes place in Diffie-Hellman algorithm. The parameter which is known to attacker is p , q, A, B as these parameters are exchanged on the public channel. So, to know the shared-secret key of the sender and the receiver, attacker would have to calculate the value of a and b which is known to only sender and receiver. So, it's tough for the attacker to get the secret key but not impossible.[5] Plain text, Man-in- Middle attack, logjam attack and many more attacks which have found on Diffie-Hellman algorithm which make it possible for an attacker.[6]

In this paper, Diffie-Hellman algorithm is extended further and a second shared-secret key is calculated which is multiplied with an arbitrary number and then exchanged between the sender and the receiver which makes this algorithm very strong. For the attacker, it will be very tough challenge for the attacker to find out the key encryption and decryption.

2.1 Mathematical-Background

Diffie- Hellman algorithm uses modular arithmetic and discrete logarithms.

PRIMITIVE ROOT:

Sender and Receiver choose a prime number p and q as it's primitive root.
[7] A number q is a primitive root of a number p if every number 'a' is coprimeto p is congruent to a power of q modulo p. That is, for every integer a coprimeto p, there is an integer a such that $q^a \equiv A \pmod p$. Such a is called the index or discrete logarithm of A to the base q modulo p.
For example, if p=7 and q=3 isits primitive root.
And $q^a \bmod p = 3^0 \bmod 7 = 1 \bmod 7 = 1$.
$= 3^1 \bmod 7 = 3 \bmod 7 = 3$.
$= 3^2 \bmod 7 = 9 \bmod 7 = 2$.
$= 3^3 \bmod 7 = 27 \bmod 7 = 6$.
The value of $q^a \bmod p$ is coming different every time when the value of 'a' is changing. 'a' belongs to whole number.

HOW THE SHARED SECRET KEY IS SAME FOR THE SENDER AND RECEIVER:

On the basis of public keys and so called private keys,
The shared-secret key for sender is $B^a \bmod p$.
Now,
$B^a \bmod p = (q^b \bmod p)^a \bmod p$
$= q^{ba} \bmod p$
$= (q^a \bmod p)^b \bmod p$
$= A^b \bmod p$
And, $A^b \bmod p$ is the secret key, Receiver is getting
So, This is how, the shared-secret key for the sender and the receiver is same.
These are the basic operations for calculating the shared-secret key. Modular arithmetic and Discrete-Maths is used in Diffie-Hellman algorithm. Public key is exchanged over the public channel in the Diffie-Hellman algorithm and using modular arithmetic secret is calculated.

3. Upgradation In The Diffie-Hellman Algorithm

Ram and Sitahave got a shared-secret key(S) in the Diffie-Hellman algorithm. Ram and Sita find 'e' as the primitive root of S as they have 'S' as a shared-secret key. Now, they will generate their own so called private key again as 'f' and 'g' respectively which is known to themselves only. Now, they will calculate their second public key $C = e^f \bmod S$ and $D = e^g \bmod S$ and they will exchange their second public keys. So, Ram has 'D' now and Sita has 'C'. On the basis of second public keys and so called private keys, Ram and Sita will now calculate their second shared-secret key(W) which is same to both.

MATHEMATICAL EXPLANATION OF SECOND SHARED SECRET KEY(W):

Ram's second shared-secret key: $W= D^f \bmod S$.

Sita's second shared-secret key: $W= C^g \bmod S$.

How are they equal?

$D^f \bmod S= (e^g \bmod S)^f \bmod S= e^{gf} \bmod S= (e^f \bmod S)^g \bmod S= C^g \bmod S = W$.

Now, Ram and Sita will take their respective random number 'h' and 'i' respectively and they will multiply it with their second shared-secret key (W) and form their respective private keys X= (W*h) and Y= (W*i) and they will exchange X and Y finally and this will be very hard for the attacker 'Ravan'.

UPGRADED DIFFIE-HELLMAN ALGORITHM:

1. Sender and Receiver agree on a prime number p, q as it's primitive root.
2. Sender and Receiver choose their so called private key 'a' and 'b' which is known to themselves only respectively.
3. Sender's public key $A= q^a \bmod p$.
4. Receiver's public key $B= q^b \bmod p$.
5. Sender and Receiver exchange their public key. Now Sender has B and Receiver has A.
6. Sender calculates $B^a \bmod p= q^{ba} \bmod = S$.
7. Receiver calculates $A^b \bmod p= q^{ba} \bmod p =S$.
8. Hence, the sender and receiver get 'S' as their shared secret key.
9. Now, Sender and Receiver take 'e' as the primitive root of 'S'.
10. Sender and Receiver generate their own so called private key 'f' and 'g' which is known to themselves only respectively.
11. Sender's second public key $C= e^f \bmod S$.
12. Receiver's second public key $D= e^g \bmod S$
13. Sender and Receiver exchange their second public key. Now Sender has D and Receiver has C.
14. Sender calculates $D^f \bmod S= e^{gf} \bmod S= W$.
15. Receiver calculates $C^g \bmod S= e^{fg} \bmod S= W$.
16. Hence, the sender and Receiver get 'W' as their second shared-secret key.
17. Sender and Receiver select their random number 'h' and 'i' respectively.
18. Sender calculates: X= (W*h) and Receiver calculates: Y= (W*i)
19. Sender and Receiver exchange X and Y finally.

The sender and receiver get a shared-secret key by using the existing Diffie-Hellman algorithm, now they find the primitive root of their shared-secret key. Using the Diffie-Hellman algorithm for the second time and sender and receiver will get a second-secret key. Now, they will take their own random number from the public channel and they will multiply their second shared-secret key with their own random number and they will exchange it for getting a new shared secret key for every message and even for the same message.

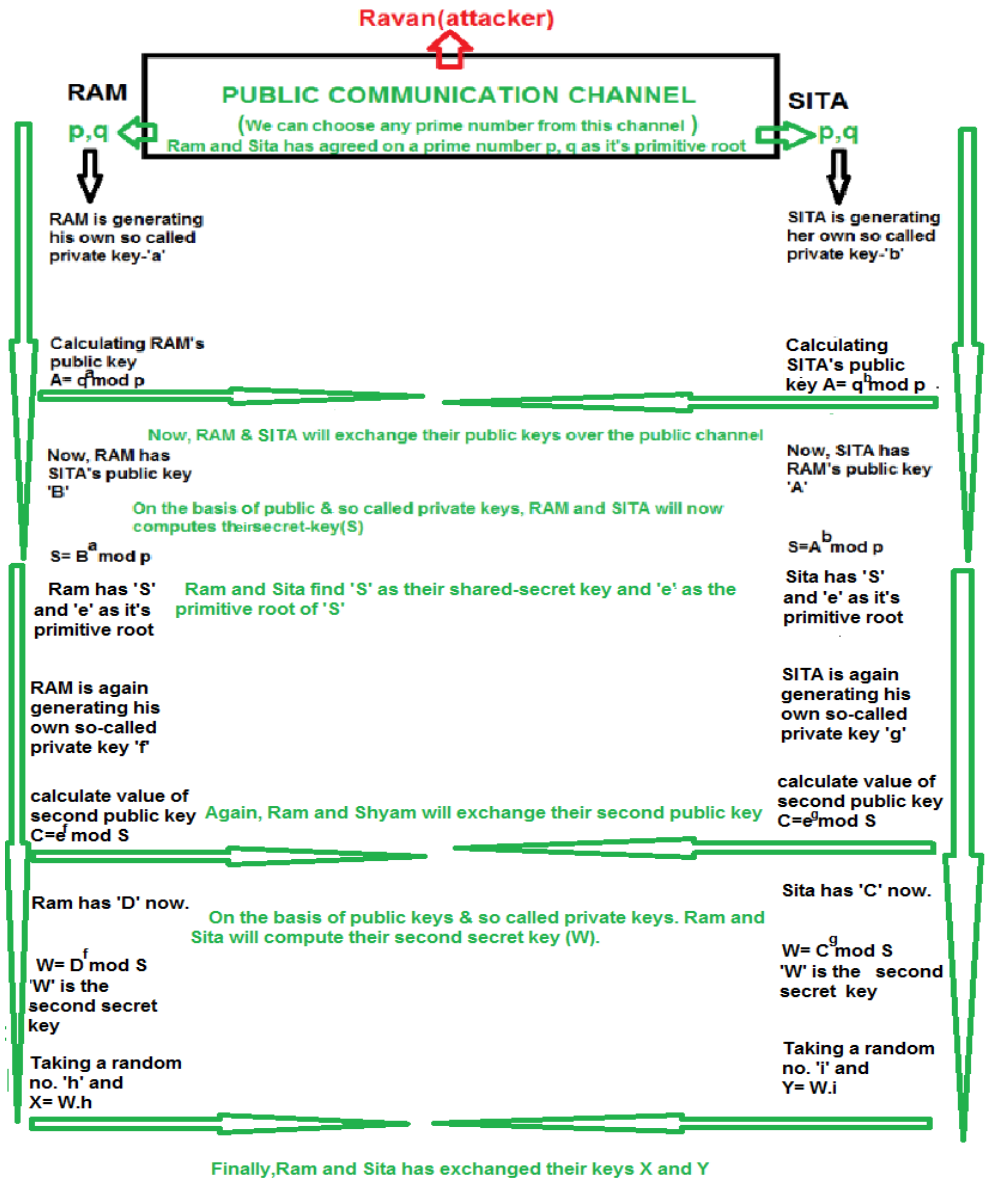

Fig. 2. Upgraded Diffie-Hellman key exchange for reliable key exchange.

The use of random number in the already strong second shared-secret key will enhance the security to a great extent against an attack like Plaintext, Man-in-Middle attack, logjam attack, etc. The generation of second secret keys every time for every message will make this system stronger as the attacker doesn't have any meaningful parameter and he is also unaware of the first shared secret key 'S'. So, Key can't be derived as we are generating the second secret key (W). Soit will be very difficult for attacker to find out the key.

4. Analysis and Result

Man-in-the-middle attack is done by keeping the public keys of sender and receiver by the attacker and sending fake public keys to them respectively. Here, the enhanced Diffie-Hellman algorithm secures the key as it's generating the second shared secret key and attacker is unaware of the idea of taking the primitive root of the first secret key.The known-plaintext attack uses plaintext and cipher text for retrieving the secret keys. So, this attack is very much possible on Diffie-Hellman algorithm.So, Known-plaintext attack is one of the most probable attacks in the original Diffie-Hellman algorithm. But in the upgraded Diffie-Hellman algorithm, the shared secret key is being generated for the second time i.e. second shared-secret key and attacker here is completely unaware of the idea, we are using of taking 'e' as the primitive root of first shared-secret key 'S' and then multiplying respective random number to the second shared-secret key and then exchanging it for generating key each time for every message or even for the same message. So, this will make our system very much secured.We are basically using the Diffie-Hellman algorithm for two times for generating a very strong shared-secret key and exchanging it by multiplying it with a random parameter so that it can generate a different key each time for the same message even.be very strong. The execution time of the upgraded algorithm is much greater than the original Diffie-Hellman algorithm. The difference between the two runtimes is very small.So, it is possible to introduce second-shared secret key and random parameters in the original Diffie-Hellman algorithm.

5. Conclusion

There are so many attacks possible on the original Diffie-Hellman algorithm. Some of themare : Known plain-text attack, Logjam attack, Man-in-Middle attack, etc. These attacks are possible on Diffie-Hellman. The proposed Diffie-Hellman algorithm will definitely make the existing algorithm very strong.In this paper, we have upgraded the Diffie-Hellman algorithm to a next level for the reliable key exchange. We have generated second shared secret key from the first shared secret key by taking the primitive root of the first shared-secret key and finally we are exchanging the keys by multiplying it with a random parameter for the generation of key each time for every message or even for the same message. So, this will make the algorithm stronger than before. This reduces the probability of most of the attacks like known-plaintext attack, etc.The existing algorithm is the making web services and many more standards and the upgraded Diffie-Hellman algorithm make the existing algorithm stronger.

References

[1] Vincent P M D R and Sathiyamoorthy E 2014 A Secured and Time Efficient Electronic Business Framework based on Public Key Cryptography in International Review on Computers and Software **9(10)** 1791-1798

[2] Koziel, Brian, et al. 2016 Neon-Sidh: efficient implementation of supersingular isogeny Diffie-Hellman key exchange protocol on ARM *International Conference on Cryptology and Network Security* Springer International Publishing

[3] Yao, Andrew C and Yunlei Zhao 2013 Method and structure for self-sealed joint proof-of-knowledge and diffie-hellman key-exchange protocols U.S. Patent No. 8,464,060

[4] Sheffer Y and Fluhrer S 2013 *Additional Diffie-Hellman Tests for the Internet Key Exchange Protocol Version 2 (IKEv2)*. No. RFC 6989.

[5] Durai Raj Vincent P M and Sathiyamoorthy E 2016 A Novel and efficient public key encryption algorithm *International Journal of Information and communication technology* **9(2)** 199-211

[6] Gowda and Shreyank N 2016 An advanced Diffie-Hellman approach to image steganography *Advanced Networks and Telecommunications Systems (ANTS), International Conference on IEEE*

[7] Tirthani, Neha and Ganesan R 2014 Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography *IACR Cryptology ePrint Archive* 49

[8] Mandal, Sayonnha and AbhishekParakh 2015 Implementing Diffie-Hellman key exchange using quantum EPR pairs *Proc. SPIE*. **9500**

[9] Harn, Lein and Changlu Lin 2014 Efficient group Diffie–Hellman key agreement protocols *Computers & Electrical Engineering* 40.6: 1972-1980.

[10] Cheng, Zhen and Jianhua Xiao 2015 Self-Assembled Architectures for Breaking Diffie-Hellman Key Exchange Algorithm *Journal of Computational and Theoretical Nanoscience* 12.2: 234-238.