Volume 3, Issue 6, June 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Secure Key Exchange Algorithm - Mathematical Approach

Sakthi Nathiarasan A , Yuvaraj K

B.Tech Student

Department of Information Technology

Sri Krishna College of Engg and Tech, Coimbatore, India

Abstract— Secure key distribution is an important concern related to cryptography. The first published and most widely used key exchange Algorithm is the Diffie–Hellman Key Exchange Algorithm. The purpose of this Algorithm is to enable two users to securely exchange a key that can be then used for subsequent encryption of messages. However, this Algorithm suffers from Man-in-the-middle Attack, in which an opponent can perform Eavesdropping and Replay attacks, in which an opponent can replay the message later .Other Algorithms as like ECMQV (Elliptic Curve Menezes Qo Vanstone), SPDH (Secure Plain Diffie-Hellman) can handle this problem but is far more complex and slower because these Algorithms are three-pass Algorithms whereas the Diffie–Hellman Algorithm is a simple two-pass Algorithm. And the proposed key exchange Algorithm is a mathematical model based on logarithms and exponents to prevent Man-in-the-middle attacks and timestamps to prevent Replay attacks. The proposed Algorithm is simple, flexible and making both hardware and software implementation easier.

Keywords— Diffie-Hellman, SPDH, ECMQV, Cypher, Plaintext

I. INTRODUCTION

In cryptography, secure communication between two entities can be achieved by confidentiality, authentication, data integrity, access control and non-repudiation. Confidentiality ensures protection of data from unauthorized disclosure. Authentication assures that the data is coming from original sender. Access control prevents unauthorized use of resource. Integrity assures that data received are exactly as sent by an authorized entity. Nonrepudiation protects from denial of service. Cryptographic systems can be divided into Symmetric key systems and Asymmetric key systems. Symmetric key systems are used for encryption and decryption of a message that should be kept secret. The encryption and decryption is done using a shared secret key. Asymmetric key cryptography is used to transport the shared secret key in a safe manner. Asymmetric key cryptography must provide a way to solve key transport over an unsecure channel. This problem is partly solved by the Diffie–Hellman key exchange algorithm, which makes it possible to obtain privacy in the key exchange. The 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. In the Diffie–Hellman Algorithm the public key is used on both sides to calculate the shared secret. The problem is that the Diffie–Hellman Algorithm is vulnerable to Man-in-the-middle attacks and Replay Attacks. However Algorithms as like ECMQV (Elliptic Curve Menezes Qo Vanstone), SPDH (Secure Plain Diffie-Hellman) can overcome these problems. but these Algorithms are complex and three-pass making both software and hardware implementation harder. In the following process of key exchange we will try to introduce a new Algorithm which overcomes Man-in-the-middle Attack and all sort of Replay Attacks .The proposed Algorithm can send keys safely and the probability of being hacked the key can be reduced.

II. BASIC DEFINITION

A. *Cryptography*

Cryptography is the science of information security. It is the practice of techniques for secure communication in the presence of opponents.it applies various techniques like encryption, decryption, digital signatures and so on to protect the data.

B. *Cypher*

A Cypher (also called cipher) is an algorithm for performing encryption or decryption—algorithm to convert plain text into cipher text.

C. *Plaintext*

This is the original intelligible message or data that is fed into the encryption algorithm as input.

D. *Ciphertext*

This is the transformed form of plaintext. The exact transformation of plaintext into ciphertext depends on encryption algorithm and secret key used.

E. *Secret Key*

F. *It is also an input to the encryption algorithm.it should be exchanged secretly between the entities involved in communication.*

G. Symmetric Key Cryptography

It is a form of cryptosystem in which encryption and decryption are performed using the same key.it is also called conventional encryption. most of the symmetric key algorithms performs substitution or permutation for transformation of plaintext into cipher text. The most famous symmetric key algorithms are DES(Data Encryption Standard) and AES (Advanced Encryption Standard). symmetric key algorithms suffers from brute-force and cryptanalysis attacks.

H. Asymmetric Key Cryptography

It is a form of cryptosystem in which encryption and decryption are performed using different keys- one a public key and one a private key.it is also known as public-key cryptography.in asymmetric cryptosystems each entity have a public-private key pair. A public key system is so constructed that calculation of one key (the 'private key') is computationally infeasible from the other (the 'public key'), even though they are necessarily related. private keys are always kept secret .asymmetric algorithms can be used for confidentiality, authentication or both. The famous Asymmetric key algorithms are Diffie-Hellman , RSA(Rivest ,Shamir and Adleman),ECC(Elliptic-Curve Cryptography*).*

III.PREVIOUS WORKS

A. Diffie-Hellman Key Exchange

The first published public-key algorithm by Diffie and Hellman that defined public-key cryptography and is generally referred to as Diffie-Hellman Key Exchange. The purpose of this algorithm is to enable two users to securely exchange a key that can be used for subsequent encryption of messages. The algorithm itself is limited to exchange of secret keys. The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. we can define discrete logarithm in the following way. First ,we define a primitive root of a prime number ‘p’ as one whose powers modulo ‘p’ generate all integers from 1 to p-1. And if ‘a’ is a primitive root of the prime number ‘p’ then the numbers

a mod p , (a^2) mod p……….(a^p-1) mod p

are distinct and consists of integers from 1 to p-1 in some permutation. for any integer ‘b’ and a primitive root of a prime number ‘p’, we can find a unique exponent i such that

b= a^i (mod p)   where 0<=i<=(p-1)

the exponent ‘i’ is referred to as discrete logarithm of b for the base a mod p.

1) Algorithm Description

Suppose we have two people wishing to communicate: Alice and Bob

Step 1: Alice and Bob shares global public elements
A prime number ‘q’
An integer ‘g’ where g<q and primitive root of q.

Step 2: At Alice:
Selects private ‘a’ where a<q
then calculates public $A = g^a$ mod q and sends it to Bob.

Step 3: At Bob:
Selects private ‘b’ where b<q
then calculates public $B = g^b$ mod q and sends it to Alice.

Step 4:Calculation of secret key by Alice :
$S= B^a$ mod q

Step 5:Calculation of secret key by Bob :
$S= A^b$mod q

Step 6:Alice and Bob now share the secret key ‘S’ and encrypt the messages using ‘S’.

The security of Diffie-Hellman key exchange lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms. For larger primes, the later task is considered infeasible.

2) Man-in-the-middle Attack

Diffie-Hellman key Exchange is vulnerable to Man-in-the-middle Attack. Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows:

1. Darth prepares for the attack by generating two random private keys x1 and x2, and then computing the corresponding private keys y1 and y2.
2. Alice transmits ‘A’ to Bob.
3. Darth intercepts ‘A’ and transmits ‘y1’ to Bob. Darth also calculates $k2= A^{\mathbf{x2}}$mod q.
4. Bob receives ‘y1’ and calculates $k1=y1^{\mathbf{b}}$mod q.
5. Bob transmits ‘B’ to Alice.
6. Darth intercepts ‘B’ and transmits ‘y2’ to Bob. Darth also calculates $k1= B^{\mathbf{x1}}$mod q.
7. Alice receives ‘y2’ and calculates $k2=y2^{\mathbf{a}}$mod q.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key ‘k1’ and Alice and Darth share secret key ‘k2’in which all future communications are intercepted by Darth, which leads to Man-in-the-middle Attack.

B. The ECMQV Algorithm

The Man-in-the-middle problem can be solved with a three-pass protocol as for instance the ECMQV (Elliptic Curve Menezes Qo Vanstone) algorithm. This protocol is using long-term static key pairs. Furthermore it is required to calculate two key pairs for each participant in the communication path. As the protocol is a three-pass algorithm it requires three communication paths to obtain a shared secret.

The algorithm has the following calculating sequences:

1. Bob calculates A,RB (ephemeral key pair) which is sent to Alice
2. Alice sends B,RA, tA = MACk1(2,B,A,RA,RB) to Bob
3. Bob sends tB = MACk1(3,A,B,RB,RA) to Alice.

This algorithm suffers from serious drawbacks like highly complex to implement, far heavier and requires extra network bandwidth.

C. Secure Plain Diffie-Hellman Algorithm

Secure Plain Diffie–Hellman algorithm is a two-pass algorithm based on HMAC.it is an improvement of existing Diffie-Hellman algorithm. It consists of Authentication Center (AuC) and participants.

Input: A basepoint P, private key k, a private key xAES, a private key yHMAC, a personal ID pid, Timestamp ts

Participants: A, B and Authentication Center (AuC)

Output: A shared secret G

Algorithm:

```
1. A calculates  P(A)k
2. A calculates HMAC(AES(PA(A)k + pidA + ts)) using xAES(A) and
   yHMAC(A).
3. AuC Checks (2) (Recalculates HMAC and checks that ts is
   newer than listed and reliable)
4. AuC unpack HMAC(AES(PA(A)k + pidA + ts)) using xAES(A) and
   yHMAC(A)
5. AuC recalculates HMAC(AES(PA(A)k + pidA + ts)) using xAES(B)
   and yHMAC(B)
6. B Checks (5) (Recalculates HMAC and checks that ts is newer
   than listed and reliable)
7. B unpack (6) using xAES(B) and yHMAC(B)
8. B calculates P(B)k and stores P(A)k
9. B calculates HMAC(AES(P(B)k+pidB+ts)) using xAES(B) and
   yHMAC(B)
10. AuC checks (9) (Recalculates HMAC and checks that ts is newer
    than listed and reliable)
11. AuC unpack HMAC(AES(P(B)k+pidB+ts)) using xAES(B) and
    yHMAC(B)
12. AuC recalculates HMAC(AES(P(B)k + pidB + ts)) using xAES(A)
    and  yHMAC(A)
13. A Checks (12) (Recalculates HMAC and checks that ts is newer
    than listed and reliable)
14. A unpack (13) using xAES(A) and yHMAC(A) and stores P(B)k
15. Common shared secret is P(A)k P(B)k
```

It is assumed that the private k is generated by the participant, the private keys for A: xAES(A), yHMAC(A) and xAES(B), yHMAC(B) are delivered from authentication center (AuC) in the initial phase during setup. The personal ID is also generated by the participant and is well known by other participants. As it can be seen from this algorithm not even the AuC can calculate the common shared secret P(A)kP(B)k since only the participants themselves know the private key k. The key exchange is secured further. The HMAC algorithm solves the problem of malicious altering of the key exchange and will also prevent replay attacks. The message uniqueness is guaranteed by using Timestamp ts. The main drawback of this algorithm is the Implementation of HMAC which is tedious and is not flexible. several choices are there for key exchange with design simplicity which makes SPDH not popular.

IV. PROPOSED PROCESS

In our Proposed process we try to eliminate the Man-in-the-middle Attack and Replay Attacks with simple Mathematics based algorithm rather than using highly complex methods which increases design complexity, so our approach would be such that the middle man could not change the key. For his porpoise we introduce some techniques and timestamps to prevent Replay attacks. The proposed technique is as follows:

A. Algorithm Description

Suppose Krishna(K) wants to exchange key with Hari(H).

Both K and H use ‘e’ as a secret number as the base of log.

Step 1: K chooses a large prime number M and calculate K1=log e(M).

Step 2: H chooses a large prime number N and calculate K2=log e (N).

Step 3: K sends K1 and a Timestamp T1 to H,

Step 4: H sends K2 and a Timestamp T2 to K,

Step 5: K Verifies Timestamp ‘T2’ and calculates

Key S=K1+K2=log e (M) + log e (N) = log e (MN).

Step 6: H verifies Timestamp ‘T1’ and calculates

Key S=K1+K2=log e (N) + log e (M) = log e (NM)

Step 7: By the properties of logarithms log e (NM) =log e (MN).

Both K and H can check whether the key is being attacked or not by calculating as follows: e^log e (MN)=MN.

K calculates R1=MN/M If R1 is a prime number then key is not attacked. Similarly H calculate R2=MN/N. If R2 is a prime number then key is not attacked.

B. *Elimination of Man-in-the-middle Attack*

Note that both B and K use a secret number ‘e’ as the base of the log. If in the middle the key is attacked and the key is changed not necessarily the base will be ‘e’. As we can calculate R1=MN/M and R2=MN/N so we can easily catch the error.

C. *Elimination of Replay Attacks*

Replay Attacks are not all possible due to the use of timestamps. An opponent can replay a message but both ‘K’ and ‘H’ may check timestamps before calculating key in order to detect Replay Attacks.

V. CONCLUSION

Designing a Key exchange algorithm with 100% Accuracy is not at all possible . We considered Man-in-the-middle Attack and Replay Attacks while proposing the Algorithm. However we can’t say that Man-in-the-middle Attack cannot possible completely .because the base selected by the middle man can be same as ‘e’ unfortunately. Our Algorithm uses simple mathematical concepts making implementation easier as well as avoidance from common Attcaks.

REFERENCES

[1] Behour A. Forouzan, Sophia Chung Fegan “Data Communication and Networking”, Fourth Edition 2009.
[2] 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.
[3] 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.
[4] 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.
[5] 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.
[6] Francois J., Raymond A. “Seurity Issues in the Diffie-Hellman Key Agreement Protocol”, IEEE Trans. on Information Theory, pages 1–17 ,1992
[7] William Stallings “Cryptography and Network Security”, Fourth Edition 2006.
[8] Malek Jakob Kakish “ Security Improvments to the Diffie-Hellman Schemes” IJRRAS, volume 8,issue 1,july 2011.
[9] Francois J., Raymond A., “ Security Issues in the Diffie-Hellman Key Agreement Protocol”, IEEE Trans. on Information Theory, pages 1–17. 1998
[10] Benjamin Arazi, “Message Authenticaiton in Computationally Constrained Environments”,IEEE Trans. Mobile Computing ,Vol.8 ,No.7 July 2009.