Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Review of Diffie–Hellman key Exchange

Sunita
Asst. Professor(CSE & IT Dept.)
B.P.S Mahila Vishwavidhalaya
Khanpur Kalan, Sonepat, India

Neeraj Goyat , Annu Malik
Network Security
B.P.S Mahila Vishwavidhalaya
Khanpur Kalan, Sonepat ,India

Abstract— Deffie-hellman key exchange algorithm is an asymmetric key cryptosystem. It is designed for only exchanging the secret keys. The purpose of this algorithm is to enable the two entities, who want to communicate, to jointly establish the shared secret key, also called session key, over an insecure communication channel, example internet, to exchange the data without having to remember or store the session key. Once both the party agrees on the key to be used, they need to use another symmetric key encryption algorithm for actual encryption or decryption of message or data to be send. This key exchange scheme is neither used for encryption or decryption nor for digital signature.

Keywords— Session key, asymmetric, digital signature, cryptosystem, encryption.

I. INTRODUCTION

The Diffie-Hellman key exchange protocol is a cryptographic protocol that was developed by Whitfield Diffie and Martin Hellman in 1976, although it was later alleged 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 .It was the first published public key algorithm in the ground-breaking paper "New Directions in Cryptography" that define the public key cryptography.

It can be called with various names:-

- Diffie-Hellman key agreement
- Diffie-Hellman key establishment
- Diffie-Hellman key negotiation
- Exponential key exchange
- Diffie–Hellman protocol
- Diffie–Hellman handshake

This algorithm is an anonymous i.e. non-authenticated, key-agreement protocol that provides the basis for a variety of authenticated protocols which is used to provide perfect forward secrecy in Transport Layer Security's short-lived modes. It is the example of the key exchange implemented within the field of cryptography .The motive of this protocol is to enable two users that have no prior knowledge of each other to securely exchange a secret value over an insecure channel (i.e. not protected from the interception but is protected from modification) and then agree on the secret key, if both the party computes the same value for the key. And that key will be used for the encryption of the message using a symmetric key cipher. Firstly both the parties agree on a non secret value i.e. public key which may be certified so that the parties can be authenticated and there may be a combination of these attributes. This algorithm is only used and limited to exchange the secret values. This algorithm was followed shortly afterwards by RSA, another implementation of public key cryptography using asymmetric algorithms. This algoritm is based on the difficulty of computing discrete logarithms of large numbers. There are no known successful attack strategies.

Diffie–Hellman establishes a shared secret that can be used for secret communications by exchanging data over a public network. The following diagram illustrates the general idea of that how the key will be exchanged by using shape instead of a very large number. The key part of the process is that Ayog and Eila exchange their secret shape in a mix only. Firstly they will agree on a common shape which is non secret i.e. a rectangle. And then they have their own private shape i.e. a secret shape, after that they will share the figure which is the mixture of secret and non secret shapes. Finally this generates an identical key that is mathematically difficult (impossible for modern supercomputers to do in a reasonable amount of time) to reverse for another party that might have been listening in on them. Ayog and Eila now use this common secret to encrypt and decrypt their sent and received data. Note that the rectangle shape is already agreed by Ayog and Eila:

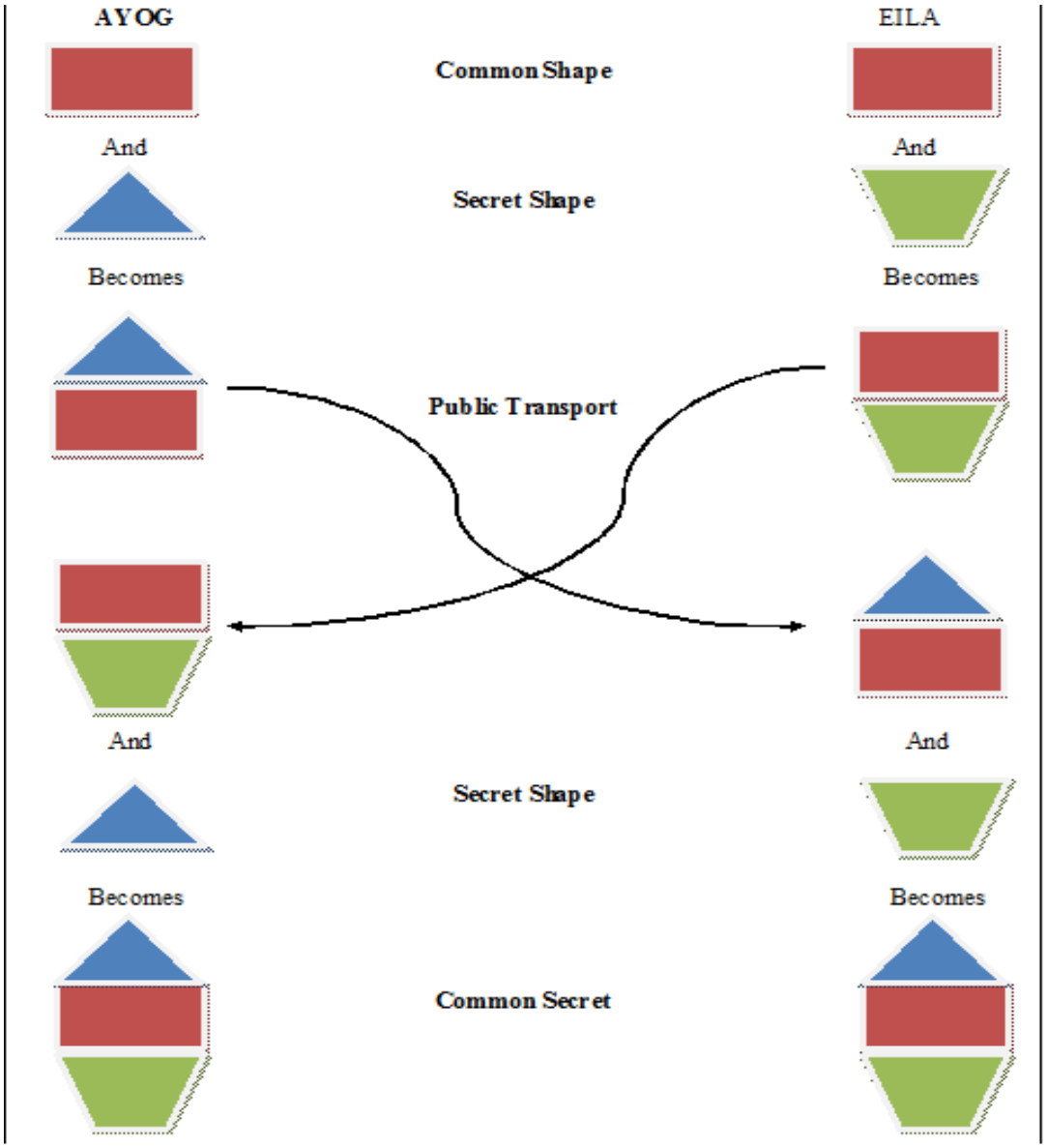


Fig (a)
General illustration of deffie-hellman key exchange

II. METHODOLOGY

Description of the algorithm:-

Now we will assume that Ayog and Eila want to agree upon a key to be used for encryption/decryption messages that would be exchanged between them. Following are the steps:-

- Finally, Ayog and Eila agree on two publicly known numbers: a prime number m and an integer n that is a prime root of m.
- Ayog selects a random integer $\beta<m$ and calculates $\Phi = n^{\beta} \bmod m$.
- Ayog sends the number Φ to Eila.
- Eila selects a random integer $\gamma<m$ and calculates $\Psi = n^{\gamma} \bmod m$
- Eila sends the number Ψ to Ayog.
- Ayog computes the key as $\kappa=\Psi^{\beta} \bmod m$.
- Eila computes the key as $\kappa=\Psi^{\gamma} \bmod m$.
- These two calculations produce the identical values. And the result is that the two sides have exchanged a secret value.

This is shown diagrammatically in fig(b).

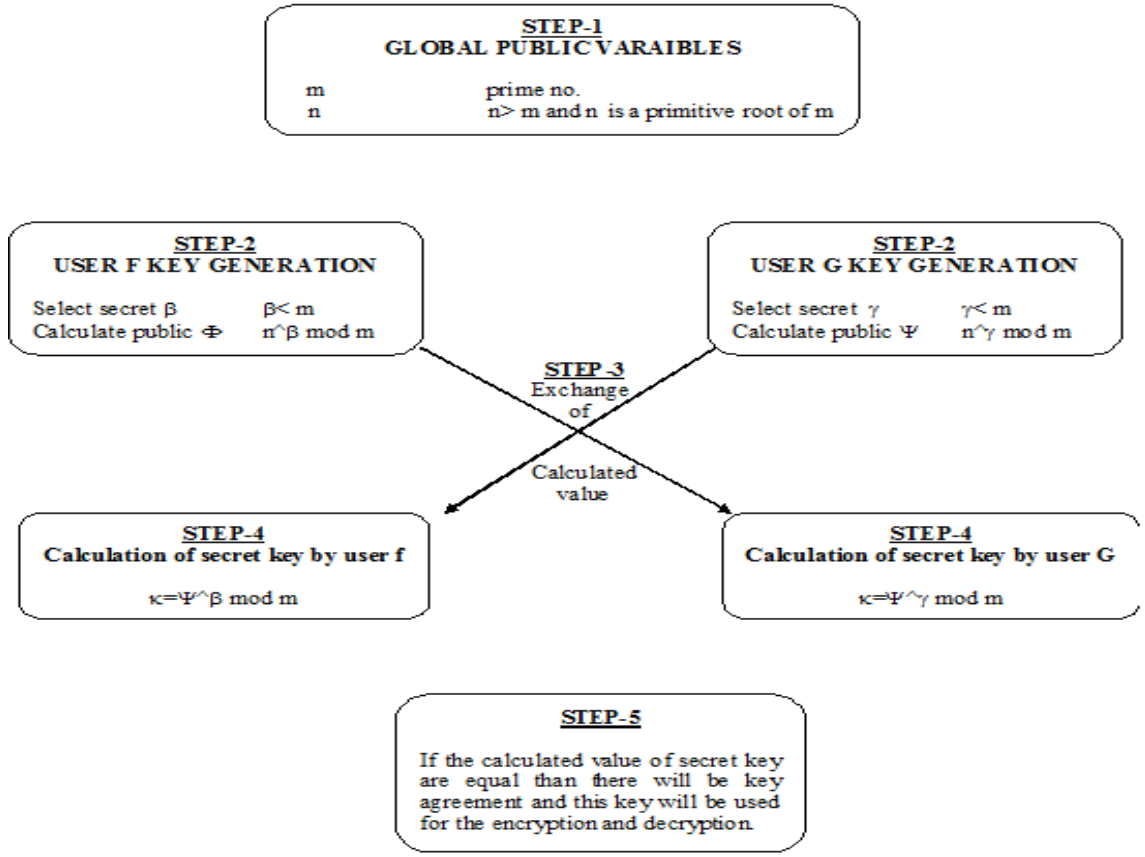


Fig (b)
Deffie-hellman key exchange Algorithm

III. EXAMPLE

Let us now take a simple example to prove that Diffie Hellman works in practical situation. Here we will take very small values for ease of understanding. But in real life ,these values are very large .The process of key agreement is as shown below:-

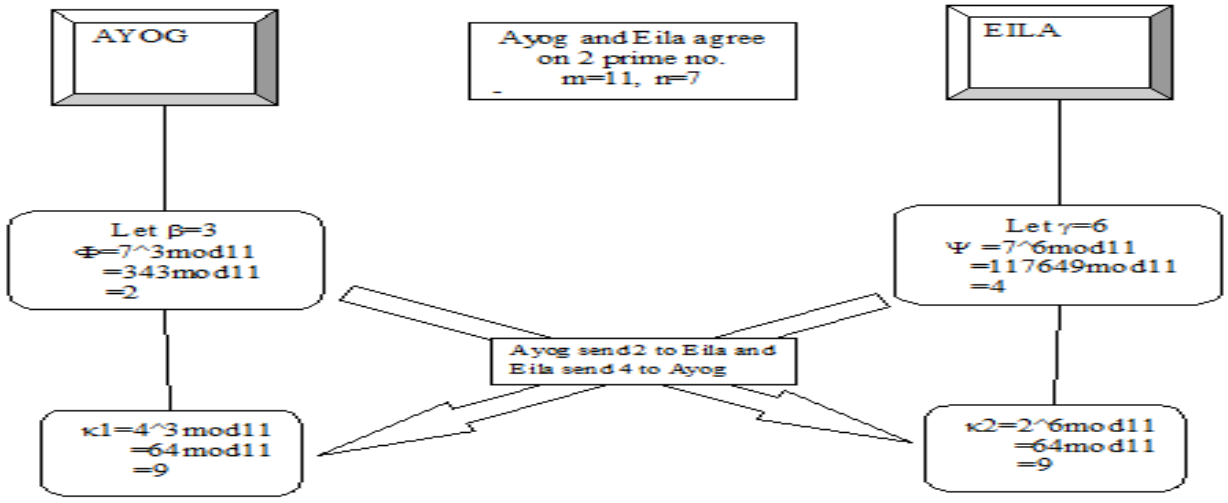


Fig (c)
Diffie-hellman key exchange example

In this example, k1 is actually equal to k2.This means that k1= k2 =9 is the symmetric key, which Ayog and Eila must keep secret and henceforth use for encrypting/decrypting their messages with.

IV. MAN IN MIDDLE ATTACK

The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack. In this attack, an opponent Yan intercepts Ayog's public value and sends her own public value to Eila. When Eila transmits his public value, Yan substitutes it with her own and sends it to Ayog. Yan and Ayog thus agree on one shared key and Yan and Eila agree on another shared key. After this exchange, Yan simply decrypts any messages sent out by Ayog or Eila, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party. This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants.

This problem is also called BUCKET BRIGADE ATTACK.

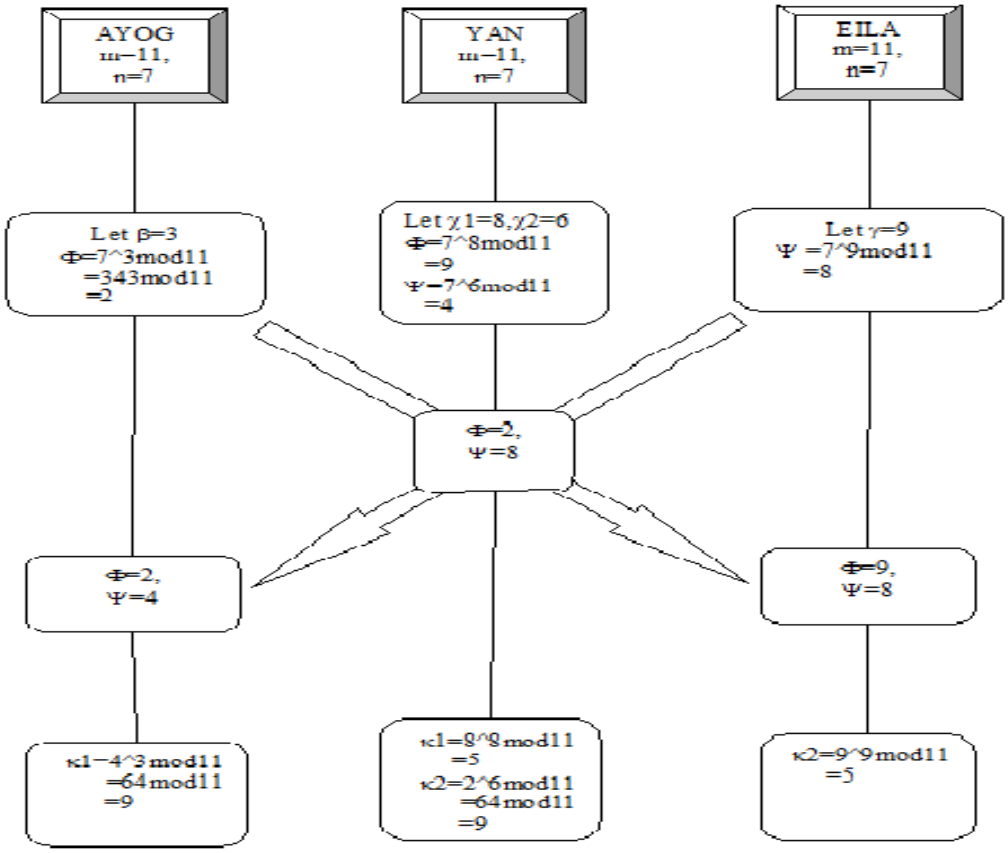
The attack proceeds as follows:-

- Yan prepares for the attack by generating two random private keys χ1and χ2,and then computing the corresponding public keys Φ and Ψ.
- Ayog will transmit Φ to Eila.
- Yan intercepts Φ and transmits χ1 to Eila. Yan also calculates k2=(Φ)^χ1 mod m.
- Eila receives χ1 and calculates k1=(χ1)^ γ mod m.
- Eila transmits Ψ to Ayog.
- Yan intercepts Ψ and transmits χ2 to Ayog. Yan also calculates k1=(Ψ)^χ1 mod m.
- Ayog receives χ1 and calculates k2=(χ2)^ γ mod m.

Now at this point, Ayog and Eila think that they share a private key, but unfortunately Ayog and Yan will share secret key k1 and Eila and Yan will share secret key k2. Whenever there will be communication between Ayog and Eila that is compromised in the following ways:-

- Ayog sends an encrypted message msg:E(k2,msg).
- Yan intercepts the encrypted message and decrypt it, to recover msg.
- Yan sends Eila E(k1,msg) or E(k1,msg'), where is any message. Here Yan can simply read the message and send the same message to Eila as it was send by the Ayog or Yan can change or modify the message and send the modified message to Eila.

This key exchange protocol is not suitable to such an attack because it does not authenticate the participants.



Fig(d)

V. SOLUTION

Possible solutions include the use of digital signatures and other protocol variants. And it follows with the authenticated version. The basic idea is as follows:-

Prior to execution of the protocol, the two parties Ayog and Eila each obtain a public/private key pair and a certificate for the public key. During the protocol, Ayog computes a signature on certain messages, covering the public value n^β

mod m. Eila proceeds in a similar way. Even though Yan is still able to intercept messages between Ayog and Eila, she cannot forge signatures without Ayog's private key and Bob's private key. Hence, the enhanced protocol defeats the man-in-the-middle attack. So, the basic version is susceptible to a man-in-middle attack, the authenticated version that usespublic key certificates is not.

VI. APPLICATION

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)

REFRENCES

[1]. William Stallings, "Cryptography and Network Security"
[2]. http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
[3]. http://www.rsa.com/rsalabs/node.asp?id=2248
[4]. http://searchsecurity.techtarget.com/definition/Diffie-Hellman-key-exchange

