ASSIGNMENT NO	B3
TITLE	Implementation of Diffie-Hellman key exchange
PROBLEM STATEMENT/ DEFINITION	Implementation of Diffie-Hellman key exchange
OBJECTIVE	To understand how Diffie-Hellman key exchange algorithm works
OUTCOME	Understaning and implementation of key distribution algorithm
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/W- C++/JAVA//Python
REFERENCES	1. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491 2. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1 3. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 2nd Edition, 2008, ISBN: 978-0-07-064823-4 4. William Stallings, "Cryptography and network security principles and practices", Pearson, 6th Edition, ISBN: 978-93-325-1877-3 5. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill, ISBN, 007070208X, 9780070702080
STEPS	1. **Global Public Elements** q ; prime number α ; α < q and it is primitive root of q 2. **USER A KEY GENERATION** Select Private key X_A $X_A<q$ Calculation of Public key Y_A $Y_A=\alpha^{XA} \mod q$ 3. **USER B KEY GENERATION**

	Select Private key X_B $X_B<q$ Calculation of Public key Y_B $Y_B=\alpha^{XB} \bmod q$ 4. **Calculation of Secret Key by A** $k=(Y_B)^{XA} \bmod q$ 5. **Calculation of Secret Key by B** $k=(Y_A)^{XB} \bmod q$
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment No. 3. Problem Definition 4. Learning Objective 5. Learning Outcome 6. Concepts Related Theory 7. Algorithm 8. Test Cases 9. Conclusion/Analysis

Pr-requisites: Discrete mathematics and any programming language C++/Java/Python.

Concepts Related Theory:

Diffie Hellman (DH) key exchange algorithm is a method for securely exchanging cryptographic keys over a public communications channel. Keys are not actually exchanged – they are jointly derived. It is named after their inventors Whitfield Diffie and Martin Hellman.

Working of Diffie-Hellman Algorithm:

1. In Public key encryption schemes are secure only if authenticity of the public key is assured.

2. Diffie-Hellman key exchange is a simple public key algorithm.
3. The protocol enables 2 users to establish a secret key using a public key scheme based on discrete algorithms.
4. The protocol is secure only if the authenticity of the 2 participants can be established.
5. There are 2 publicly known numbers :A prime number q and a*n integer α that is a primitive root of q.*

Note:*Premitive root of a prime number P is one, whose powers module P generate all the images from 1 to P-1*

For example:

2 is a primitive root mod 5, because for every number a relatively prime to 5, there is an integer z such that 2z≡a.
All the numbers relatively prime to 5 are 1, 2, 3, 4, and each of these (mod 5) is itself (for instance 2 (mod 5) = 2):

- 20=1, 1 (mod 5)=1, so 20≡1
- 21=2, 2 (mod 5)=2, so 21≡2
- 23=8, 8 (mod 5)=3, so 23≡3
- 22=4, 4 (mod 5)=4, so 22≡4.

4 is not a primitive root mod 5, because for every number relatively prime to 5 (again, 1, 2, 3, 4) there is not a power of 4 that is congruent. Powers of 4 (mod 5) are only congruent to 1 or 4. There is no power of 4 that is congruent to 2 or 3:

- 40=1, 1(mod5)=1
- 41=4, 4(mod5)=4
- 42=16, 16(mod5)=1
- 43=64, 64(mod5)=4

6. Suppose users A and B wish to exchange the key.

User A selects a random integer $X_A < q$ and

computes $Y_A = \alpha^{XA} \ mod\ q$

7. User B independently selects a random integer $XB < q$ and

compute $Y_B = \alpha^{XB} \ mod\ q$

8. Each side keeps X value private and makes Y value available publicly to the other side

user A computes the key as:

$k=(Y_B)^{XA} \ mod \ q$

User B computes the key as :

$k=(Y_A)^{XB} \ mod \ q$

The calculations produce identical results :

$k=(Y_B)^{XA} \ mod \ q$–>calculated by user A

$=(\alpha^{XB} \ mod \ q)^{XA} \ mod \ q=(\alpha^{XB})^{XA} \ (mod \ q)$–>

By rules of modular arithmetic$=\alpha^{XBXA} \ mod \ q$

$= (\alpha^{XA})^{XB} \ mod \ q$

$k=(\alpha^{XA} \ mod \ q)^{XB} \ mod \ q$

- **Diffie Hellman key Exchange Algorithm**

6. $k=(Y_A)^{XB} \ mod \ q$ -> same as calculated by B
7. **Global Public Elements**

   q ; prime number

   α ; α < q and it is primitive root of q
8. **USER A KEY GENERATION**

   Select Private key X_A $X_A<q$

   Calculation of Public key Y_A $Y_A=\alpha^{XA} \ mod \ q$
9. **USER B KEY GENERATION**

   Select Private key X_B $X_B<q$

   Calculation of Public key Y_B $Y_B=\alpha^{XB} \ mod \ q$
10. **Calculation of Secret Key by A**

   $k=(Y_B)^{XA} \ mod \ q$
11. **Calculation of Secret Key by B**

   $k=(Y_A)^{XB} \ mod \ q$

10. The result is that two sides have exchanged a secret value.

11.Since X_A and X_B are private the other party can work only following ingredients:

q, α, X_A , X_B

Note: $YB = \alpha^{XB}$ mod a

$X_B = d \log \alpha, q(YB)$

$\uparrow$

Discrete Logarithm

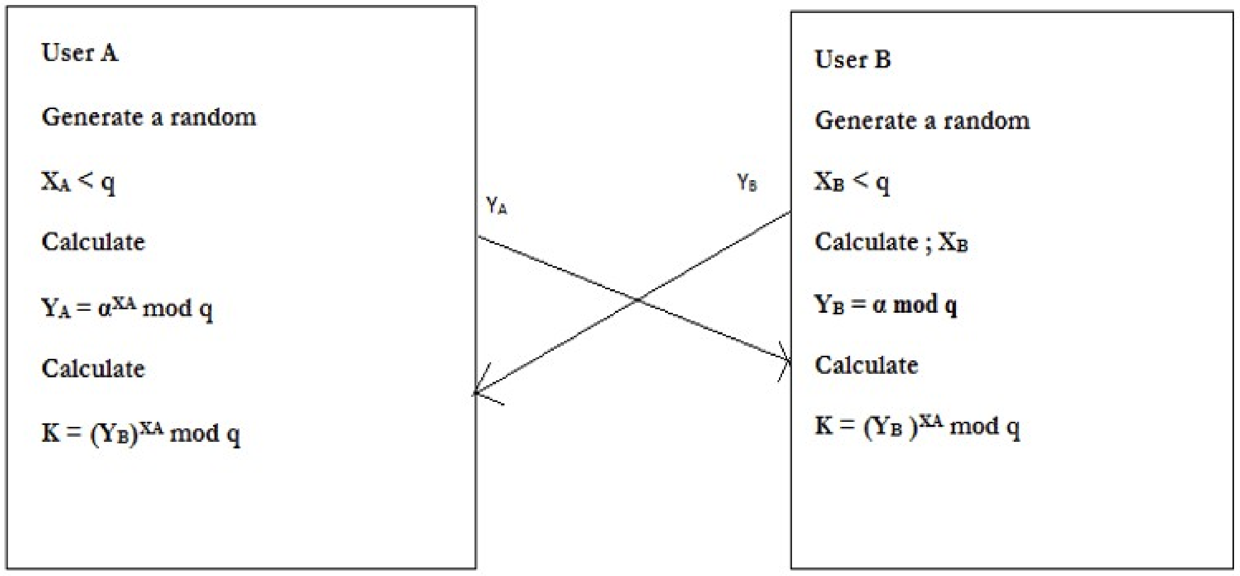
Discrete Logarithms:

Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups. If *G* is a multiplicative cyclic group and *g* is a generator of *G*, then from the definition of cyclic groups, we know every element *h* in *G* can be written as g^x for some *x*. The discrete logarithm to the base *g* of *h* in the group *G* is defined to be *x* . For example, if the group is Z_5^*, and the generator is 2, then the discrete logarithm of 1 is 4 because $2^4 \equiv 1 \bmod 5$.

The discrete logarithm problem is defined as: given a group *G*, a generator *g* of the group and an element *h* of *G*, to find the discrete logarithm to the base *g* of *h* in the group *G*. Discrete logarithm problem is not always hard. The hardness of finding discrete logarithms depends on the groups. For example, a popular choice of groups for discrete logarithm based crypto-systems is Z_p^* where p is a prime number. However, if $p-1$ is a product of small primes, then the Pohlig–Hellman algorithm can solve the discrete logarithm problem in this group very efficiently. That's why we always want *p* to be a safe prime when using Z_p^* as the basis of discrete logarithm based crypto-systems. A safe prime is a prime number which equals $2q+1$ where *q* is a large prime number. This guarantees that $p-1 = 2q$ has a large prime factor so that the Pohlig–Hellman algorithm cannot solve the discrete logarithm problem easily. Even *p* is a safe prime, there is a sub-exponential algorithm which is called the index calculus. That means *p* must be very large (usually at least 1024-bit) to make the crypto-systems safe.

12. The algorithm security lies on the fact that it is easy to calculate exponential modulo a prime, last difficult to calculate to calculate discrete logarithm.

Figure: D-H Key exchange algorithms



Example:

Consider q=353, α= 3 (3 is primitive root of 353)

A and B discrete private keys

$XA=97 and\ XB=223$

Each computes its public key

A computes $YA=3^{97}$ mod 353 =40

B computes $YB=3^{233}$ mod 353 = 248

After exchange of public keys, each can compute the common secret key

A computes K $=(YB)^{XA}\ mod\ 353=(248)^{97}\ mod\ 353=160$

B computes K $=(YA)^{XB}\ mod\ 353=(40)^{253}\ mod\ 353=160$

Uses of Diffie Hellman Algorithm

Aside from using the algorithm for generating public keys, there are some other places where DH Algorithm can be used:

- **Encryption:** Diffie Hellman key exchange algorithm can be used to do encryption, one of the first schemes to do it was ElGamal encryption. One modern example of it is called Integrated Encryption Scheme which provides security against chosen plain text and chosen clipboard attacks.

- **Password Authenticated Agreement:** When two parties share a password, a password-authenticated key agreement can be used to prevent the Man in the middle attack. This key Agreement can be in the form of Diffie-Hellman. Secure Remote Password Protocol is a good example that is based on this technique.
- **Forward Secrecy:** Forward secrecy based protocols can generate new key pairs for each new session, and they can automatically discard them at the end when the session is finished too. In these forward Secrecy protocols, more often than not, the Diffie Hellman key exchange is used.

Advantages of the Diffie Hellman Algorithm

- The sender and receiver don't need any prior knowledge of each other.
- Once the keys are exchanged, the communication of data can be done through an insecure channel.
- The sharing of the secret key is safe.

Disadvantages of the Diffie Hellman Algorithm

- The algorithm can not be used for any asymmetric key exchange.
- Similarly, it can not be used for signing digital signatures.
- Since it doesn't authenticate any party in the transmission, the Diffie Hellman key exchange is susceptible to a man-in-the-middle attack.

Test Cases: Observe the output for different values of q, α, X_A , X_B *respectively.*