*Lab Practices-III Fourth Year Computer Engineering*

# Information and Cyber Security

**Assignment No.7**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| W  (4) | C  (4) | D  (4) | V  (4) | T  (4) | Total  (20) | Sign/Remark |
|  |  |  |  |  |  |  |

**7.1 Title: Implementation of Diffie-Hellman key Exchange (DH)**

# Problem Definition:

Implementation of Diffie-Hellman key Exchange (DH)

# Prerequisite:

Basics of Computer Networking and Python

# Software Requirements:

Python 3

# Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

# Learning Objectives:

Learn Diffie-Hellman key Exchange (DH)

# Outcomes:

After completion of this assignment students are able to understand the Diffie-Hellman key Exchange

# Theory Concepts:

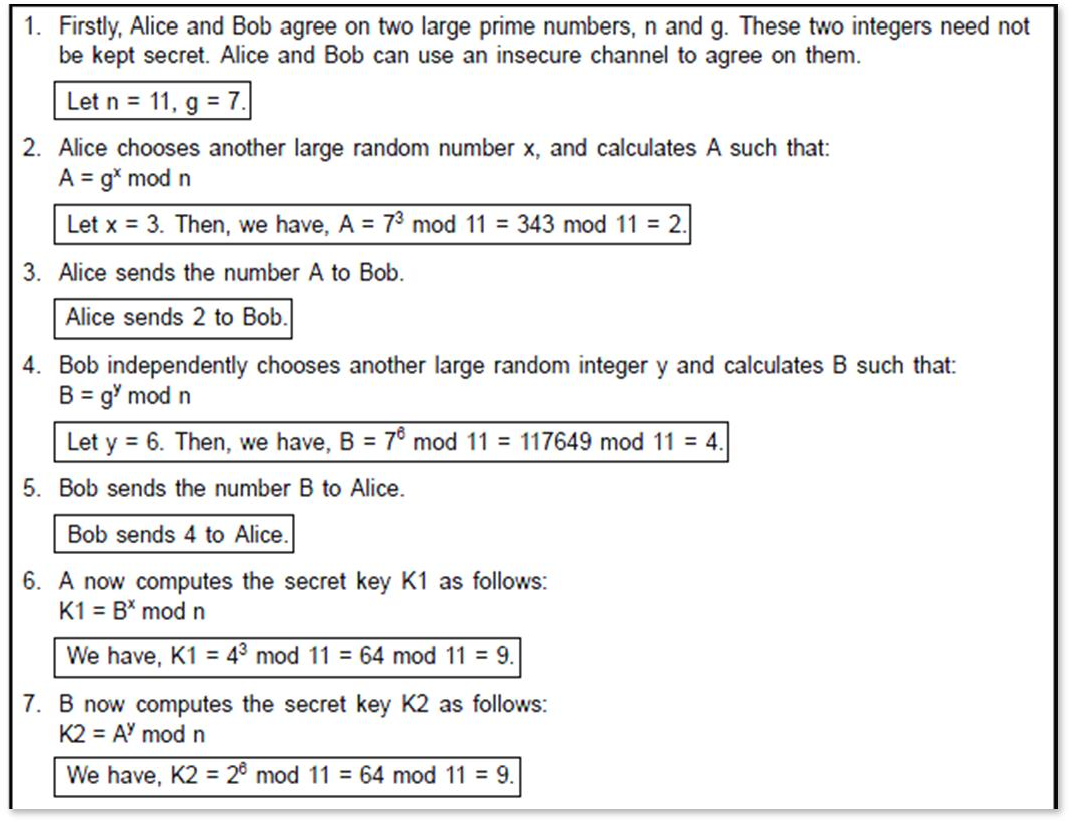
* + 1. **Diffie-Hellman key Exchange (DH)**

In the mid- 1970’s, Whitefield Diffie, a student at the Stanford University met with Martin Hellman, his professor &the two began to think about it. After some research & complicated mathematical analysis, they came up with the idea of AKC. Many experts believe that this development is the first & perhaps the only truly revolutionary concept in the history of cryptography.

# Silent Features of Diffie-Hellman key Exchange (DH)

1. Developed to address shortfalls of *key distribution* in symmetric key distribution.
2. A *key exchange algorithm*, not an encryption algorithm
3. Allows two users to share a *secret key* securely over a public network
4. Once the key has been shared Then both parties can use it to encrypt and decrypt messages using symmetric cryptography
5. Algorithm is based on “difficulty of calculating discrete logarithms in a finite field”
6. These keys are mathematically related to each other.
7. ‘’Using the public key of users, the session key is generated without transmitting the private key of the users.’’

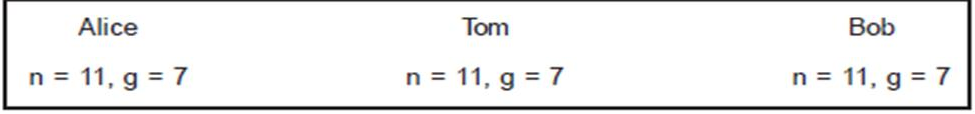
# Diffie-Hellman Key Exchange/Agreement Algorithm with Example



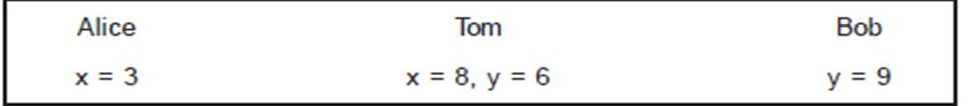
* + 1. **Diffie-Hellman Key exchange**

1. Public values:
   * large prime p, generator g (primitive root of p)
2. Alice has secret value x, Bob has secret y
3. Discrete logarithm problem: given x, g, and n, find A
4. A  B: gx (mod n)
5. B  A: gy (mod n)
6. Bob computes (gx)y = gxy(mod n)
7. Alice computes (gy)x = gxy (mod n)
8. Symmetric key= gxy (mod n)
   * 1. **Limitation:** Vulnerable to “man in the middle” attacks\*

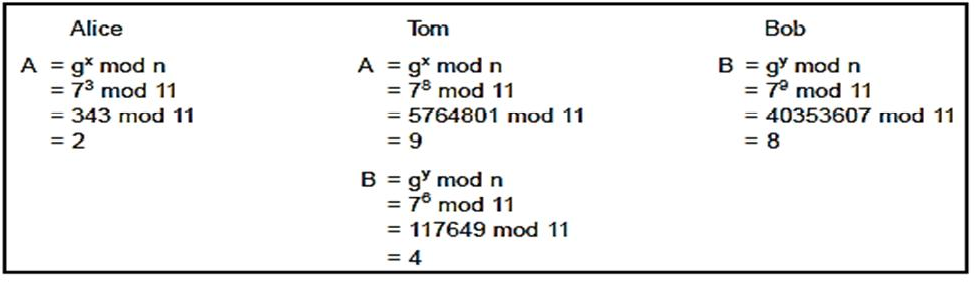
# Man-in-the-Middle Attack:



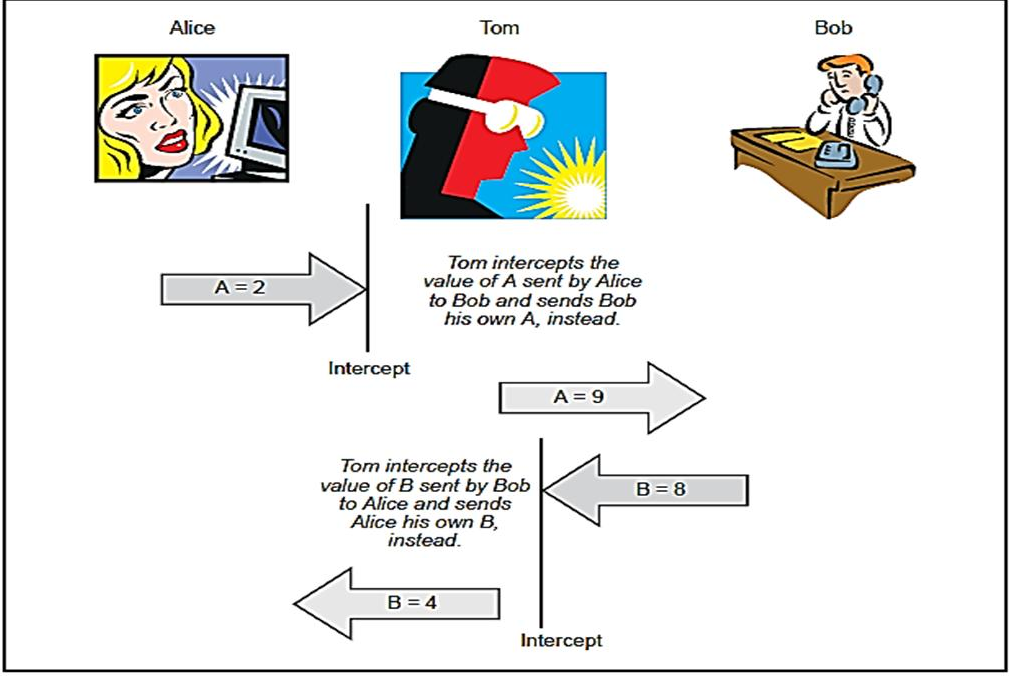
**Figure 7.1 Man-in-the-Middle Attack Part-I**



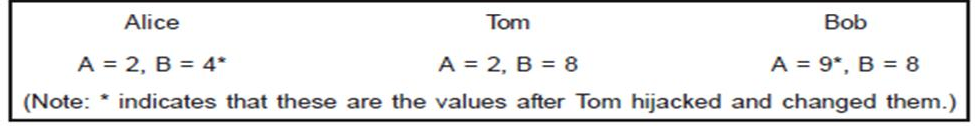
**Figure 7.2 Man-in-the-Middle Attack Part-II**



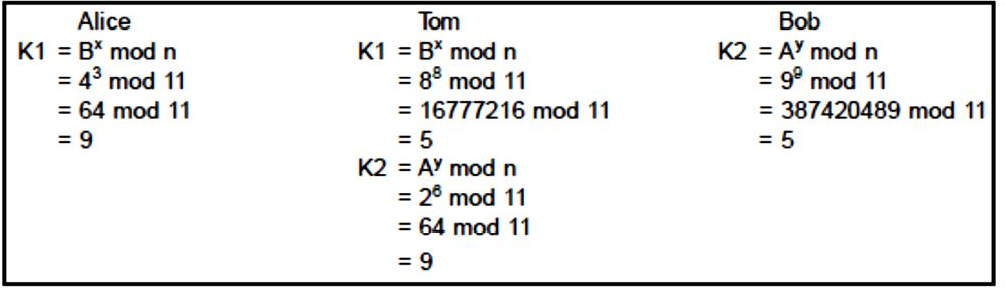
**Figure 7.3 Man-in-the-Middle Attack Part-III**



**Figure 7.4 Man-in-the-Middle Attack Part-IV**

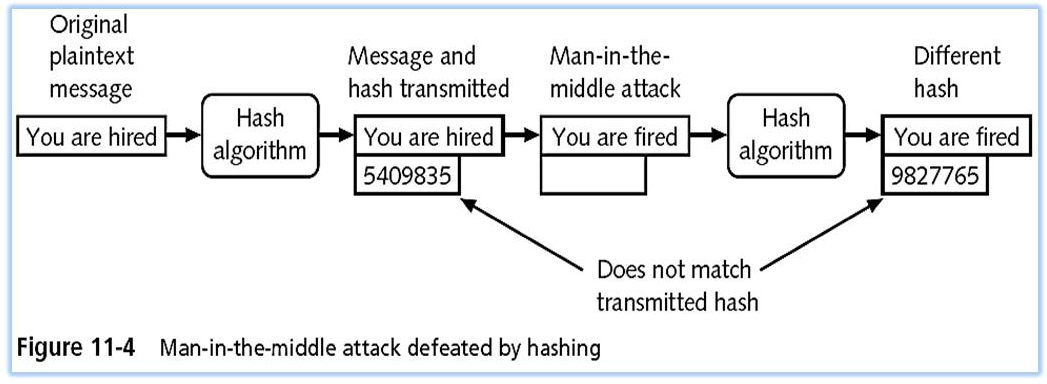


**Figure 7.5 Man-in-the-Middle Attack Part-V**



**Figure 7.6 Man-in-the-Middle Attack Part-VI**

* + 1. **Preventing a Man-in-the-Middle Attack with Hashing**



**Conclusion:** Thus we have studied and implement Diffie-Hellmen key exchange algorithm and how to prevent Man-in-the-Middle Attack

# Oral Questions

1. Explain “Diffie-Hellmen key exchange algorithm with suitable example
2. What is Man in the middle attack?
3. How to Preventing a Man-in Middle Attack?