**University of Stavanger**

**Assignment 2**

**Security and Vulnerability of the Networks**

Abolfazl Taleb zadeh

**October 2021**

**Abstract**

The second mandatory assignment is mainly about practicing key exchange capability using Diffie Hellman algorithm. Utilizing key exchange algorithms play a vital role in the world of cryptography particularly under the circumstances in which using a symmetric algorithm is urged. Therefore, a unique single key is needed to be exchanges among concerned parties. As far as transferring the keys without following the confidentiality principles will put a big question mark on the primary purpose of keeping the secrecy of the message, a well-trusted method is needed for sharing the credentials among all the parties within a communication. For implementing the Diffie Hellman algorithm, a class structure is preferred. In the class the arguments have been defined as the properties and different stages of the instruction is defined as methods which can be called in the proper order by the user. In addition, a pseudo-random number generator algorithm (CSPRNG) is used to fortify the enumerated key, which in this specific case the choice is B.B.S (Blum Blum Shub) algorithm. It can be observed that using B.B.S the calculated key gets significantly more secure and also the intended size of the key for being usable by the user-defined encryption algorithm which would be SDES, is delivered thoroughly. The beneficiary and efficiency of application of both Diffie Hellman and B.B.S is well verified and experimented in the second part of the assignment which aims to establish a safe mutual communication tool using the assumptions implemented in the class.

**I. Introduction**

Using symmetric encryption algorithms has always been known for having the burden of sharing the keys between the two sides of the communication. Diffie-Hellman, also known as DH, was the first algorithm ever used for sharing the keys of a symmetric encryption method which was published in 1976 by Whitfield Diffie and Martin Hellman. DH makes it possible for two parties of a mutual communication which use a symmetric encryption method to share a single key without any need of special equipment or predefined secret information. The most important downfall of this method is that none of the sides of the communication can not be sure about the authenticity of the public key that is obtained from the other side. In the other word there always is a scale of man in the middle attack risk [1]. A solution which is given for this problem is Public Key Infrastructure (PKI) which as stated in www.wikipedia.com, “have been proposed as a workaround for the problem of identity authentication. In their most usual implementation, each user applies to a “[certificate authority](https://en.wikipedia.org/wiki/Certificate_authority)” (CA), trusted by all parties, for a [digital certificate](https://en.wikipedia.org/wiki/Digital_certificate) which serves for other users as a non-tamperable authentication of identity”. (“Key Exchange” November 2014 section 2.3 public key infrastructure). There are also other methods like Password-Authentication Key Exchange (PAKE), Quantum Key Distribution (QKD) and etc. which we are not going to go any further into details. In this assignment since the generated password using DH is pretty simple, a pseudo-random number generator algorithm (PRNG) is being used to fortify the enumerate key using DH algorithm, which in this case B.B.S (Blum Blum Shub) algorithms is chosen. Pseudo-random number generators implement a value called seed to generate a random number which will be repeated in a specific range. There are multiple types of PRNGs which use different methods (e.g.RC4, NIST CTR\_DRBG, ANSI X9.17). in the first part of the assignment a secure key exchange instruction is practiced using DH with having the benefit of key extension capability using B.B.S algorithm. And in the second part this method has been implemented in real a mutual communication using two servers through different ports.

**II. Design and Implementation**

In this part the implementation method and the technique which is used in different part of each section will be discussed and analyzed in detail. there are eight steps in the first part of the program which will be argued in turns. The first part of the program is constructed in the form of a Python class because it will be used in other parts of the program. The class there are several public variables which would be known as the properties of the class after the instances are made from. These variables store the necessary values concerning different stages of the class. These variables include, a list which is called “primes” which holds a list of prime numbers that will be used for multiple purposes, a list called “z” which stored the values of a cyclic group of “p”, an integer variable called “f\_k\_dec” which keeps the decimal final key which is calculated by B.B.S algorithm, an integer variable named “p” which holds the prime value which is used for DH algorithm, integer variables “ya” and “yb” which store the public keys for the first and the second communication side also known as “alice” and “bob”, an integer variable called “k” which holds the value of the key enumerated by DH algorithm, integer variables “a” and “b” which store the private keys for both parties, an integer variable named “q\_times\_p” which holds the value of multiplication of two prime numbers which is used by B.B.S algorithm. The class also has an initiation method which receives a number “n” in the defining stage. This number assigns how big the chosen prime number and as the result the key should be.

**III. The Results**

**IV. Discussion**

**V. Conclusion**

**VI. Works Cited**

[1] https://en.wikipedia.org/wiki/Key\_exchange

[2] https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman\_key\_exchange

[3] https://www.youtube.com/watch?v=Yjrfm\_oRO0w

[4] https://www.youtube.com/watch?v=hp7bpkNL790

[5] https://dev.to/techparida/how-to-deploy-a-flask-app-on-heroku-heb

[6] https://www.youtube.com/watch?v=M-0qt6tdHzk&t=7s

[7] https://www.youtube.com/watch?v=9MHYHgh4jYc

[8] Cryptography and Network Security: Principles and Practice, Stallings, W.

[9] http://cs.wellesley.edu/~webdb/lectures/flask/flask.html

[10] https://www.youtube.com/watch?v=Yjrfm\_oRO0w&t=317s

[11] CS255 Stanford course textbook “Cryptography and Computer Security. Very basic number theory fact sheet”

[12]