Journal of Engineering and Applied Sciences 12 (Special Issue 7): 8229-8239, 2017
ISSN: 1816-949X

A Comparative Evaluation of a Classical and Quantum Key Exchange System for use in a Web Application Security

Alharith A. Abdullah and Mustafa T. Mohammed
College of Information Technology, University of Babylon, Babil, Iraq

Abstract: In the web application security a public key cryptography play important role to provide a secure method for exchanging secret keys online. There are many key agreement algorithms, one of the most common key exchange algorithm is Diffie-Hellman which provides highly secure key exchange between communicating parties. In addition, the Quantum Key Distribution (QKD) is a new key agreement method that uses the laws of quantum mechanics to exchange secret key securely. This study presents a comparative evaluation of possible classical and quantum key exchange systems for implementation in a web application security. We applied each of the key exchange algorithms to produce a shared random secret key between two parties to use to encrypt and decrypt messages through one of the symmetric encryption algorithms and then compares them in terms of features offered, ephemeral key, security, authentication, attacks, encryption/decryption efficiency, and key generation requirements.

Key words: Cryptography, quantum cryptography, web application security, key agreement algorithm, Diffie-Hellman, quantum key distribution, BB84 protocol

INTRODUCTION

It is very important to witness the increase that the world is experiencing in the use of internet, therefore, it is very necessary to provide a confidentiality and privacy to the channels that are transmitting and receiving data in the World Wide Web (WWW).

A third party or an attacker's main duties is to obtain information and intercept the data that is transmitted through the main public network. Protecting sensitive data are required to provide a security mechanism. One of the security mechanisms is encryption whereby the encrypted data makes it very difficult for an attacker to access sources of the code or access confidential information that is transmitted in the network (Stallings, 2006). There are many cryptography algorithms and protocols to generate encrypted data. But there are two main manners for the algorithms which are used to generate encrypted data to any information based on a key. To generate keys in the network there are to processes which are termed symmetric key encryption algorithm and asymmetric key encryption algorithm. Symmetric key encryption algorithms are those cryptographic algorithms that simultaneously use one cryptographic key to encrypt and decrypt the data. Asymmetric key encryption algorithm or public-key encryption algorithms are those algorithms that use different keys for encryption and decryption. The key distribution mechanism is very important to the symmetric and asymmetric encryption algorithm because any existence of weakness in the distribution will make the intercept key easy. The purpose of key distribution mechanism is to provide secure procedures for handling cryptographic keying materials. Most modern cryptographic systems that used key distribution mechanism have high security because the algorithm included complex computation and need long time to break it. This means that current cryptographic systems will become more vulnerable as the speeds and powers of computers continue to increase especially with quantum computers. The key distribution mechanism is important between two parties who may not have ever communicated previously, so that, they can encrypt their communications. Historically, Diffie and Hellman (1976) discovered a universal algorithm which is now known as the Diffie-Hellman (DH) (Steiner *et al*., 1996). This algorithm is used in many secure connectivity protocols on the internet. DH is a method for securely exchanging a shared secret between two parties, over an untrusted network. An important concept in this algorithm is the Quantum Key Distribution (QKD) which is a key exchange that exploits certain properties of quantum physics to ensure its security and to exchange the secret key (Gisin *et al*., 2002). In this study, we will describe the two of key distribution mechanism in details which are Diffie-Hellman (DH) algorithm and quantum key

Corresponding Author: Alharith A. Abdullah, College of Information Technology, University of Babylon, Babil, Iraq

distribution (BB84) protocol where we will implement these algorithms with one of the symmetric encryption algorithm. We will select a simplified data encryption standard algorithm. Then we will make comparison on these two effective algorithms based on some user-defined parameters.

Diffie-Hellman key distribution algorithm: Diffie-Hellman key exchange is one of the cryptographic algorithms that uses the asymmetric cryptographic key and creates an insecure connection between the two parties and the transmitted key through a channel. The process of encryption works by transferring the secret key between two parties to encrypt the symmetric key which tends to be vulnerable to specific type of attack known as the man in the middle (Diffie and Hellman, 1976). In encryption operations and computer security, the man in the middle is identified as an attack that threatens the confidentiality of data between the two parties who think they exchange confidential information with each other. Below is a simple illustration demonstrating the implementation of the key exchange process in Diffie-Hellman algorithm.

Alice and Bob agree on two numbers "p" and "g" where "p" is "a" large prime number and "g" is called the base or generator. Alice picks a secret number "a" and for the other side Bob picks a secret number "b". Alice and Bob compute their public values $A = g^a \bmod p$, $B = g^b \bmod p$, respectively. Alice and Bob exchange their public values to compute $K = B^a \bmod p$ in Alice's side and $K = A^b \bmod p$ in Bob's side. Finally, based on the laws of algebra Alice's key is the same as Bob's key so Alice and Bob know and share the secret key K (Garzia, 2013). Figure 1 illustrates the implementation of the key exchange process in Diffie-Hellman algorithm.

Quantum key distribution key distribution algorithm: One of the most famous technologies currently known is the quantum key distribution feature that provides the secret key distribution in a secure and confidential manner between the two parties, referred to as Alice and Bob. To allow them to exchange information and important data across the network and knowledge of the eavesdropper who is referred to as the attacker. It also allows the QKD for the exchange of basic materials between the parties referred to earlier and in a very secure manner and also provides important advantages of the quantitative transfer process (Cai and Scarani, 2009). The most important protocol for quantum distribution is the BB84 protocol (Bennett, 1992; Bennett and Brassard, 2004). This protocol is used to send encrypted series of numbers from

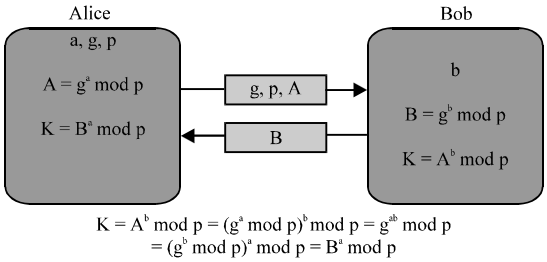

Fig. 1: Diffie-Hellman key distribution algorithm

Table 1: Coding scheme for BB84 protocol

Bit	$\oplus$	$\otimes$
0	$\|0\rangle = \alpha_{00}$	$\|+\rangle = \alpha_{10}$
1	$\|1\rangle = \alpha_{01}$	$\|-\rangle = \alpha_{11}$

single photons called "quantum exchange" then opens discussion to choose the communication medium between the parties (the key stage of the agreement). The transmission of single polarized photons is utilized by BB84 protocol as the qubit state. We have four polarizations photons that are clustered collectively based on two different non-orthogonal bases. Base $\oplus$ represents the two non-orthogonal basis of the horizontal polarization (0°) and vertical polarization (90°) and the intuitive notation in this research is represented through the base states by using $|0\rangle$ and $|1\rangle$ where $\oplus = \{|0\rangle, |1\rangle\}$. The base $\oplus$ represents the diagonal polarization (45°) and (135°) and the intuitive notation in this research is represented through the base states by using $|+\rangle$ and $|-\rangle$ with $|+\rangle = 1/\sqrt{2}(|0\rangle+|1\rangle)$ and $|-\rangle = 1/\sqrt{2}(|0\rangle-|1\rangle)$ where $\otimes = \{|+\rangle, |-\rangle\}$. The coding scheme for BB84 protocol is defined in Table 1.

The explanation of BB84 protocol is as follows. Alice chooses a random string of bits $d = \{0, 1\}^n$ and a random string of bases b $\{\oplus, \otimes\}^n$ where n>N. Alice prepares a photon in quantum state α_{ij} for each bit d_i in d and b_i in b as in Table 1 and sends it to Bob via. quantum channel. With respect to either $\oplus$ or $\otimes$, chosen at random, Bob's measurements produce a string $d \in \{0, 1\}^n$ while his choices of bases from $b \in \{0, 1\}^n$. Alice via. the classical channel sends the value of b_i to Bob. Bob reacts to Alice by stating whether he used the same basis for measurement. Both d_i and d_i are rejected if $d_i \neq d_i$. Alice selects a random subset of the remaining bits in and reveals their values to Bob via. the classical channel. If the result of Bob's measurements for any of these bits does not match the values disclosed, the attacker is detected and communication is aborted. The string of bits remaining in once the bits disclosed are removed is the common secret key, $k = \{0, 1\}^N$ (Bennett and Brassard, 2004).

MATERIALS AND METHODS

Practical implementations: In this part of the study, we implement the steps of Diffie Hellman algorithm and the BB84 protocol, to exchange key and then use this shared secret key with one of the symmetric encryption algorithms and see the results. We surly preform the implementation of DH algorithm and BB84 protocol with Java programming language and then we apply the shared secret key of DH algorithm and BB84 protocol with Simplified Data Encryption Standard S-DES that represents the symmetric encryption algorithm and see the result and the differences between the quantum cryptography and the classic cryptography. First of all, we implement the DH algorithm as shown in Fig. 2.

We notice at the end of the algorithm we get the secret key that exchanges it between the two parties. After generating the secret key of the DH algorithm. We can now start implementing the symmetric key algorithm S-DES step by step.

Step 1; Enter 8 bit binary plain message: The binary message consists of entering 8 binary bits and uses it with the secret key that is generated from the DH protocol with limitation that it must be more than 10 binary numbers to complete the work of the S-DES algorithm. Subsequently, we insert the message and take the secret key from the DH algorithm. Now the S-DES algorithm starts working on the key step-by-step, analysing the key according to the methods of it. At first the secret key will enter into a set of transformations including permutation, expanding initialization, shifting one position to the left LS-1 and shifting two positions to the left LS-2. The permutation occurs first on the whole key that contains 10 binary bits and second permutation will occur on the 10 binary key but the output of it will contain 8 bits which indicates the loss of 2 binary numbers in the first and second permutation. Now we divide the 8 binary numbers into two groups equally. Then, using the two groups unilaterally, we apply permutation and then expand to the 4 keys to simplify the work with it. when we are done with expanding we make the initialization on the key and start shifting LS-1 and LS-2 to get the new secret key with secure and encrypted message (Fig. 3).

Step 2; Enter the binary cipher message: The work of this step is simply like the first step but the message will be taken from the result of the first step and using the DH

```
 ENTER THE FIRST PRIME NUMBER
881
 ENTER THE SECOND PRIME NUMBER
953
person A : ENTER YOUR SECRET NUMBER
877
person B : ENTER YOUR SECRET NUMBER
941
A's SECRET KEY :663
B's SECRET KEY :663
A's SECRET KEY :1010010111
B's SECRET KEY :1010010111
```

Fig. 2: Implementation of DH algorithm

algorithm key to make the transformation to get the first massage that we entered statically. This gives us a mechanism to implement the S-DES algorithm with the DH algorithm and obtain a result of the implementation of the protocol and the algorithm together as well as acquire a new random secret key and the security of the channel between Alice and Bob who share the secret key together (Fig. 4).

Now, we implement the BB84 protocol to exchange secret key. The implementation of the protocol begins when we insert the digital number into the end of the user Alice and then the implementation starts using the specified angles and polarized photons to obtain a binary number extracted from the input number (Fig. 5). Then we have a button to send the Qubits to Bob (Fig. 6). Bob receives the binary number sent by Alice. A special button is shown for filtering or measuring (Fig. 7). We click on it after the measurement process of the Qubits performs a random inverse process with specific angles and polarized photons to extract and convert the binary number sent to ensure the security between the parties (Fig. 8).

The last step is to extract the secret key by clicking on the button called the basic matrix exchange. This button generates the secret key and exchanges it between the two parties The new secret key is shown with the percentage of the average key length (Fig. 9).

After generating the random secret key of the BB84 protocol, we start implementing the symmetric key algorithm S-DES with the same procedure that we implemented but now by using BB84. We implement the first the encryption of the S-DES algorithm as in Fig. 10 and the decryption of the S-DES algorithm in Fig. 11.

```
Enter 8-bit Plaintext : 11110000


YOUR KEY LENGTH IS : 10

Enter 10-bit Key : THIS KEY TAKEN FROM THE DIFFIE HELLMAN CLASSIC KEY ALGORITHEM


 Key Generation ...

---------------------------------------
Input Key :  - 1 0 1 0 0 1 0 1 1 1
After Permutation(P10) Key :  - 1 0 0 0 0 1 1 1 1 1
After LeftShift LS-1 Key :  - 0 0 0 0 1 1 1 1 1 1
Subkey K1 Generated :  - 1 0 1 0 1 1 1 1
After LeftShift LS-2 Key :  - 0 0 1 0 0 1 1 1 1 1
Subkey K2 Generated :  - 1 1 1 0 1 0 1 1

---------------------------------------
Plaintext array :  - 1 1 1 1 0 0 0 0

---------------------------------------
Initial Permutaion(IP) :  - 1 0 1 1 1 0 0 0

---------------------------------------
First Round LH :  - 1 0 1 1
First Round RH:  - 1 0 0 0
EXPANSION/PERMUTATION on RH :  - 0 1 0 0 0 0 0 1
XOR With Key :  - 1 1 1 0 1 1 1 0
S-Box S0:  - 1 1
S-Box S1:  - 0 0
Output of mappingF :  - 1 0 0 1
After First Round :  - 0 0 1 0 1 0 0 0

---------------------------------------
After Switch Function :  - 1 0 0 0 0 0 1 0

---------------------------------------
Second Round LH :  - 1 0 0 0
Second Round RH:  - 0 0 1 0
EXPANSION/PERMUTATION on RH :  - 0 0 0 1 0 1 0 0
XOR With Key :  - 1 1 1 1 1 1 1 1
S-Box S0:  - 1 0
S-Box S1:  - 1 1
Output of mappingF :  - 0 1 1 1
After Second Round :  - 1 1 1 1 0 0 1 0

---------------------------------------
After Inverse IP (Result) :  - 1 1 1 0 1 1 0 0

---------------------------------------
```

Fig. 3: S-DES algorithm implementation encryption part based on DH algorithm

```
 Decryption
Enter 8-bit Ciphertext : 11101100


Enter 10-bit Key : THIS KEY TAKEN FROM THE DIFFIE HELLMAN CLASSIC KEY ALGORITHEM


 Key Generation ...

----------------------------------------

 For decryption Two Sub-keys will be used in reverse order

----------------------------------------

Input Key :  - 1 0 1 0 0 1 0 1 1 1
After Permutation(P10) Key :  - 1 0 0 0 0 1 1 1 1 1
After LeftShift LS-1 Key :  - 0 0 0 0 1 1 1 1 1 1
Subkey K1 Generated :  - 1 0 1 0 1 1 1 1
After LeftShift LS-2 Key :  - 0 0 1 0 0 1 1 1 1 1
Subkey K2 Generated :  - 1 1 1 0 1 0 1 1

----------------------------------------
Plaintext array :  - 1 1 1 0 1 1 0 0

----------------------------------------
Initial Permutaion(IP) :  - 1 1 1 1 0 0 1 0

----------------------------------------
First Round LH :  - 1 1 1 1
First Round RH:  - 0 0 1 0
EXPANSION/PERMUTATION on RH :  - 0 0 0 1 0 1 0 0
XOR With Key :  - 1 1 1 1 1 1 1 1
S-Box S0:  - 1 0
S-Box S1:  - 1 1
Output of mappingF :  - 0 1 1 1
After First Round :  - 1 0 0 0 0 0 1 0

----------------------------------------
After Switch Function :  - 0 0 1 0 1 0 0 0
----------------------------------------
Second Round LH :  - 0 0 1 0
Second Round RH:  - 1 0 0 0
EXPANSION/PERMUTATION on RH :  - 0 1 0 0 0 0 0 1
XOR With Key :  - 1 1 1 0 1 1 1 0
S-Box S0:  - 1 1
S-Box S1:  - 0 0
Output of mappingF :  - 1 0 0 1
After Second Round :  - 1 0 1 1 1 0 0 0

----------------------------------------
After Inverse IP (Result) :  - 1 1 1 1 0 0 0 0

----------------------------------------
THIS IS THE DIFFIE HELLMAN CLASSIC KEY IN SDES ALGORITHEM : 10
```

Fig. 4: S-DES algorithm implementation decryption part based on DH algorithm

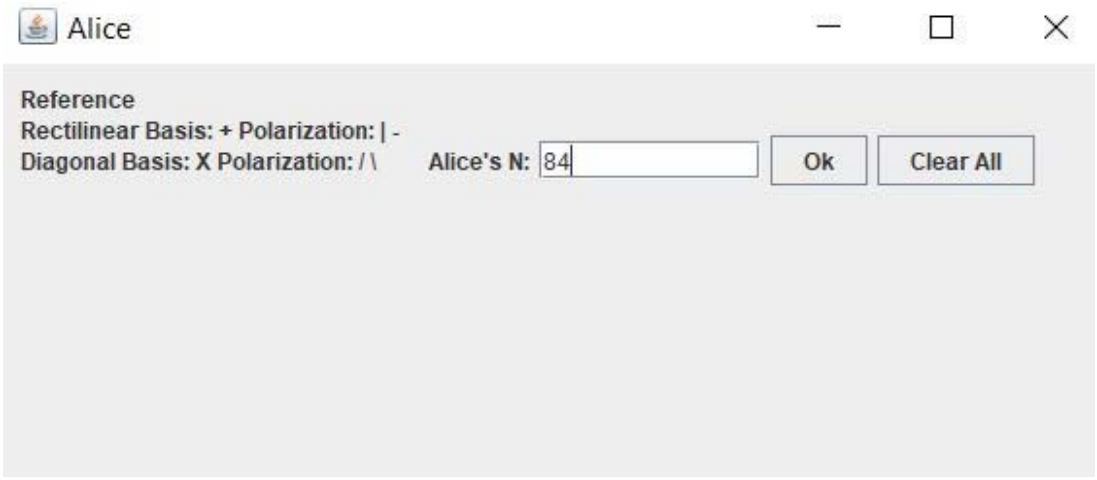

Fig. 5: Generate Alice's basis

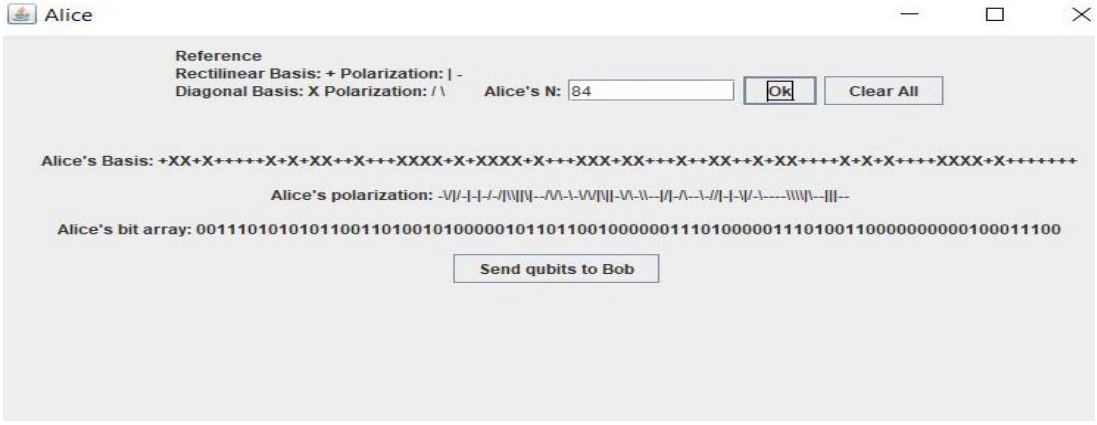

Fig. 6: Send the polarization to Bob

Fig. 7: Bob's measurement

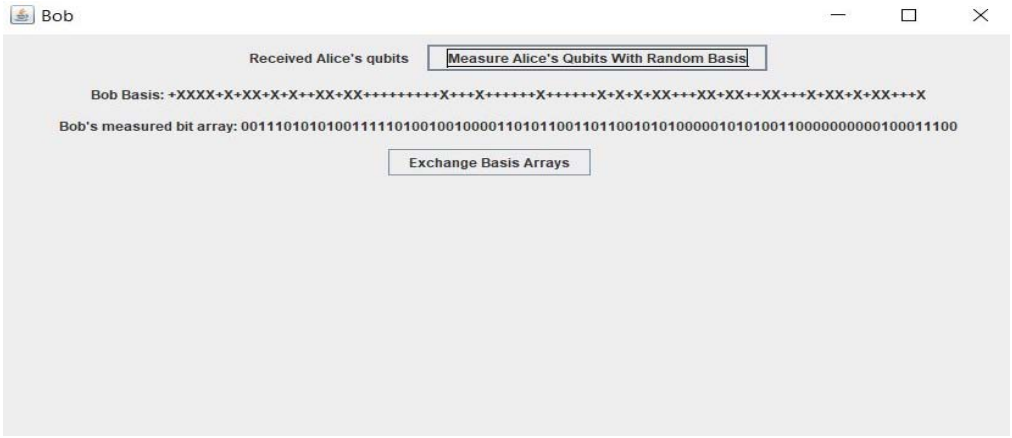

Fig. 8: Bob's measurement bit array

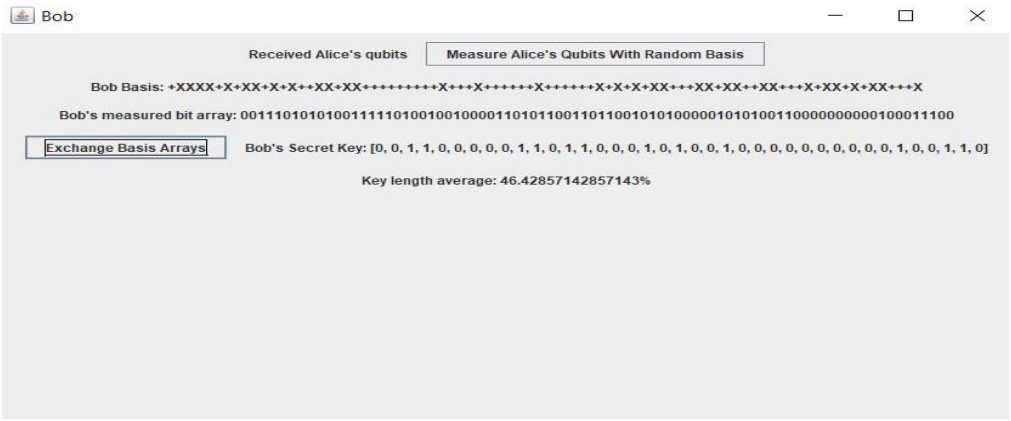

Fig. 9: Generate secret key based on BB84 protocol

RESULTS AND DISCUSSION

Algorithms comparison: Now, we can make a simple comparison between Diffie Hillman and BB84 by relying on several parameters.

Ephemeral keys: Generating ephemeral keys for the BB84 protocol is extremely difficult because of the randomization of generating secret key and applying the polarization theory on the input number. This will make the secret key hard to break from the intruder and more stable to establish secure network. Generating ephemeral keys for Diffie Hellman algorithm is more extremely easy when we compare it with BB84 protocol. Because of the key that is generated from the Diffie Hellman algorithm is not depended on the fully randomization (Fig. 10 and 11).

Security: BB84 is the first security protocol implementing QKD which makes it unconditionally secure against unlimited resources because it uses the idea of photon polarization as well as each bit is encoded with random photon polarization. So, based on quantum laws, the attacker cannot intercept the quantum channel between Alice and Bob, therefore, the communication channels used by Alice and Bob are trusted. The discrete logarithm and the prime length numbers used in the Diffie Hellman algorithm determine the value of security and confidentiality.

Key encoding: The BB84 protocol uses the quantum distribution key which is larger for coding. It relies on randomization and the use of polarized photons per bit. Its coding is larger compared to other classical systems. The Diffie Hellman algorithm uses public key and private key that is smaller to encode and it relies on the mathematical equation and discreet logarithm.

Strength: The BB84 protocol is more powerful and secure than other systems in the field of encryption and security. Because it sends a Qubit which in turn is very complex

```
Enter 8-bit Plaintext : 11111111


YOUR KEY LENGTH IS : 39

Enter 10-bit Key : THIS KEY TAKEN FROM BB84 ALGORITHEM QK


 Key Generation ...

---------------------------------------
Input Key :  - 0 0 1 1 0 0 0 0 0 1
After Permutation(P10) Key :  - 1 0 0 0 1 1 0 0 0 0
After LeftShift LS-1 Key :  - 0 0 0 1 1 0 0 0 0 1
Subkey K1 Generated :  - 0 0 0 1 0 1 1 0
After LeftShift LS-2 Key :  - 0 1 1 0 0 0 0 1 0 0
Subkey K2 Generated :  - 0 1 0 0 1 0 0 0

---------------------------------------
Plaintext array :  - 1 1 1 1 1 1 1 1

---------------------------------------
Initial Permutaion(IP) :  - 1 1 1 1 1 1 1 1

---------------------------------------
First Round LH :  - 1 1 1 1
First Round RH:  - 1 1 1 1
EXPANSION/PERMUTATION on RH :  - 1 1 1 1 1 1 1 1
XOR With Key :  - 1 1 1 0 1 0 0 1
S-Box S0:  - 1 1
S-Box S1:  - 1 0
Output of mappingF :  - 1 0 1 1
After First Round :  - 0 1 0 0 1 1 1 1

---------------------------------------
After Switch Function :  - 1 1 1 1 0 1 0 0

---------------------------------------
Second Round LH :  - 1 1 1 1
Second Round RH:  - 0 1 0 0
EXPANSION/PERMUTATION on RH :  - 0 0 1 0 1 0 0 0
XOR With Key :  - 0 1 1 0 0 0 0 0
S-Box S0:  - 1 0
S-Box S1:  - 0 0
Output of mappingF :  - 0 0 0 1
After Second Round :  - 1 1 1 0 0 1 0 0

---------------------------------------
After Inverse IP (Result) :  - 0 1 1 0 0 1 0 1

---------------------------------------
```

Fig. 10: S-DES Algorithm Implementation Encryption Part based on BB84 Protocol

```
 Decryption
Enter 8-bit Ciphertext : 01100101

Enter 10-bit Key : THIS KEY TAKEN FROM BB84 ALGORITHEM QK

 Key Generation ...

----------------------------------------

 For decryption Two Sub-keys will be used in reverse order

----------------------------------------

Input Key :  - 0 0 1 1 0 0 0 0 0 1
After Permutation(P10) Key :  - 1 0 0 0 1 1 0 0 0 0
After LeftShift LS-1 Key :  - 0 0 0 1 1 0 0 0 0 1
Subkey K1 Generated :  - 0 0 0 1 0 1 1 0
After LeftShift LS-2 Key :  - 0 1 1 0 0 0 0 1 0 0
Subkey K2 Generated :  - 0 1 0 0 1 0 0 0

----------------------------------------
Plaintext array :  - 0 1 1 0 0 1 0 1

----------------------------------------
Initial Permutaion(IP) :  - 1 1 1 0 0 1 0 0

----------------------------------------
First Round LH :  - 1 1 1 0
First Round RH:  - 0 1 0 0
EXPANSION/PERMUTATION on RH :  - 0 0 1 0 1 0 0 0
XOR With Key :  - 0 1 1 0 0 0 0 0
S-Box S0:  - 1 0
S-Box S1:  - 0 0
Output of mappingF :  - 0 0 0 1
After First Round :  - 1 1 1 1 0 1 0 0

----------------------------------------
After Switch Function :  - 0 1 0 0 1 1 1 1

Second Round LH :  - 0 1 0 0
Second Round RH:  - 1 1 1 1
EXPANSION/PERMUTATION on RH :  - 1 1 1 1 1 1 1 1
XOR With Key :  - 1 1 1 0 1 0 0 1
S-Box S0:  - 1 1
S-Box S1:  - 1 0
Output of mappingF :  - 1 0 1 1
After Second Round :  - 1 1 1 1 1 1 1 1

---------------------------------------
After Inverse IP (Result) :  - 1 1 1 1 1 1 1 1

---------------------------------------
THIS IS THE BB84 QK IN SDES ALGORITHEM : 0011000001101100010100100000000000100110
```

Fig. 11: S-DES Algorithm Implementation Decryption Part based on BB84 Protocol.

and very difficult to break than series of bits. The transmission is done via the polarized photons that are determined for each bit. One of the most important points of its power is the theory of no cloning theorem (Lindblad, 1999) through which the protocol derives its power and the power of its own network. If the attacker tries to enter or clone the information or data sent, the protocol immediately disconnects and warns both parties of the existence of an attacker or an intruder on the network.

The Diffie Hellman algorithm is less powerful than BB84. This is because it relies on simple and uncomplicated mathematical equations for the randomness used in the BB84. Another point of weakness of the network in Diffie Hellman is the possibility of cloning and tapping on the network without the knowledge of either parties and the possibility of taking information and important data from the network.

Authentication: Efficient quantum secure direct communication scheme with authentication is presented which is approved based on quantum entanglement and polarized single photons (Horodecki *et al*., 2009). The Diffie Hellman algorithm key exchange establishes authentication key to provide two or more specified entities that have been connected over the secure network with a shared secret key which may use for cryptographic goal such as confidentiality or data integrity (Blake-Wilson and Menezes, 1998).

Attacks: The BB84 protocol is more powerful against multiple quantum attacks and the man in the middle attack (Khalaf, 2015; Abdullah, 2015). The BB84 protocol exploits the non-orthogonally of quantum states to decrease the information accessible and attacks.

One of the most aggressive attackers on the Diffie Hellman algorithm is the man in the middle who eavesdrops on the information sent and it is possible to withdraw the important data and confidential information without the knowledge of the two parties. This is a weak point in the algorithm and in the network in general.

CONCLUSION

Quantum cryptography is major achievement in the network security as well as using BB84 which is the most common protocol for QKD. Its uses the idea of photon polarization and the key consists of bits that will be transmitted as photon and each bit is encoded with a random polarization photon. The BB84 protocol is more secure and confidential in web application compared to the Diffie Hellman algorithm which relies on unconditional symmetric encryption. Generating secret key with BB84 protocol is based on the principle of randomization that make the network that run with BB84 more powerful and secure against breaking encryption and eavesdropping. The BB84 protocol uses non-cloning theorem and non-orthogonal when communicating over network. Entanglement assisted with convolutional coding exploit entanglement to encode stream of qubit. Importing classical convolutional coding theory produces high performance to quantum code. The implementation of the BB84 protocol with symmetric encryption algorithms generates new powerful secret key with randomization theory.

RECOMMENDATIONS

For future research in quantum cryptography, we suggest the use of the BB84 protocol with the network security protocols like SSL/TLS and IPS. Protocols improve the security of cloud computing and implementing them with the Internet of Thing (IOT) insures high performance, confidentiality, data integrity and high security.

REFERENCES

Abdullah, A.A., 2015. Modified quantum three pass protocol based on hybrid cryptosystem. Ph.D Thesis, Eastern Mediterranean University, Famagusta, Northern Cyprus.

Bennett, C.H. and G. Brassard, 2004. Quantum cryptography: Public key distribution and coin tossing. Theor. Comput. Sci., 560: 7-11.

Bennett, C.H., 1992. Quantum cryptography using any two nonorthogonal states. Phys. Rev. Lett., 68: 3121-3124.

Blake-Wilson, S. and A. Menezes, 1998. Authenticated Diffe-Hellman Key Agreement Protocols. In: Selected Areas in Cryptography, Tavares S. and H. Meijer (Eds.). Springer, Berlin, Germany, ISBN: 978-3-540-65894-8, pp: 339-361.

Cai, R.Y. and V. Scarani, 2009. Finite-key analysis for practical implementations of quantum key distribution. N. J. Phys., 11: 1-21.

Diffie, W. and M.E. Hellman, 1976. New directions in cryptography. IEEE Trans. Inform. Theory, 22: 644-654.

Garzia, F., 2013. Handbook of Communications Security. WIT Press, Ashurst, UK., ISBN:978-1-84564-768-1, Pages: 658.

Gisin, N., G. Ribordy, W. Tittel and H. Zbinden, 2002. Quantum cryptography. Rev. Mod. Phys., 74: 145-195.

Horodecki, R., P. Horodecki, M. Horodecki and K. Horodecki, 2009. Quantum entanglement. Rev. Mod. Phys., 81: 865-942.

Khalaf, R.Z., 2015. Quantum encryption algorithm based on modified BB84 and authentication DH algorithm. Ph.D Thesis, Eastern Mediterranean University, Famagusta, Northern Cyprus.

Lindblad, G., 1999. A general no-cloning theorem. Lett. Math. Phys., 47: 189-196.

Stallings, W., 2006. Cryptography and Network Security: Principles and Practices. 4th Edn., Pearson Education, Delhi, Indian, ISBN:978-81-7758-774-6, Pages: 673.

Steiner, M., G. Tsudik and M. Waidner, 1996. Diffie-Hellman key distribution extended to group communication. Proceedings of the 3rd ACM Conference on Computer and Communications Security, March 14-15, ACM Press, New Delhi, India, pp: 31-37.