Malaysian Journal of Computing, 7 (2): 1139-1149, 2022

eISSN: 2600-8238 online

APPLICATION OF LAMPORT DIGITAL SIGNATURE SCHEME INTO THE STATION-TO-STATION PROTOCOL

Md Nizam Udin[1*], Farah Azaliney Mohd Amin[2], Nor Ainaa Mat Abu[3], Siti Nurfazliana Mohamad Sarif[4] and Intan Nur Athirah Binti Mohammad Zuki[5]

Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA Cawangan Negeri Sembilan, Kampus Seremban
70300 Seremban, Negeri Sembilan, Malaysia
[1]nizam1558@uitm.edu.my, [2]farah525@uitm.edu.my, [3]ainaa.ma27@gmail.com,*
[4]fazliana027@gmail.com, [5]intanathirahzuki@gmail.com

ABSTRACT

In cryptography, the key exchange protocol is very important before starting secure communication. Both parties will use an asymmetric key algorithm to exchange their keys for encryption and decryption. One of the methods to securely exchange the keys is Diffie-Hellman Key Exchange (DHKE) protocol. DHKE allows both parties to exchange their keys over the insecure public channel safely. However, DHKE protocol does not authenticate the message, making it easily exposed to third-party interruptions like Man-in-the-Middle (MitM) attack. Therefore, Station to Station (STS) protocols was introduced after modifying the DHKE protocol and adding authentication elements. Thus, this paper proposes to provide the authentication of the STS protocol by using Lamport Digital Signature Scheme. Lamport Digital Signature Scheme, also known as Lamport one-time signature scheme, gives very strong security because it can be built from any cryptographically secure one-way function and usually uses a cryptographic hash function. The results from the study are STS protocol algorithms that provide a verification scheme using the Lamport Digital Signature Scheme. As a result, both parties will obtain a common key for encryption and decryption, in which both parties play a role by using their respective digital signature for verification within the STS of this protocol.

Keywords: *Lamport Digital Signature Scheme, Station to Station Protocol, Diffie-Hellman Key Exchange, Man in the Middle Attack and One-way Function, Public Key Cryptography*

Received for review: 20-02-2022; Accepted: 19-09-2022; Published: 01-10-2022
DOI: 10.24191/mjoc.v7i2.17181

1. Introduction

Cryptography allows the parties to secure the privacy of the information they send to each other while guaranteeing the integrity and authenticity of communication (Bellare & Rogaway, 2005). Therefore, it plays a very important role in private data communication. Cryptography has been widely used daily—for example, the secureness of credit card information when online shopping and privacy in online banking. Bellare and Rogaway (2005) stated that this study involves a few branches in mathematics like number theory, group theory and computational complexity theory.

A few basic terms in cryptography are plaintext, ciphertext, encryption algorithm, decryption algorithm and secret key. Cryptography was divided into two branches: private key

cryptography and public key cryptography. Private key cryptography is about sharing secret keys between two parties for encryption and decryption. Meanwhile, public key cryptography involves the generation of public keys and secret keys (Paar & Pelzl, 2010). Most of its applications are key establishment or key distribution and digital signature. Gómez Pardo (2013) mentioned that public key encryption is more convenient than private key encryption in this modern communication.

Key establishment or key distribution is a central part of cryptography in which two communication parties will compute and exchange the private key for the decryption (Burmester & Desmedt, 1995). Whitfield Diffie and Martin Hellman proposed the Diffie-Hellman Key Exchange (DHKE) protocol in their published paper entitled "*New Direction in Cryptography"* in 1976. DHKE is also the protocol that allows sharing a common secret key between two parties over an insecure communication channel. It is one of the cryptographic schemes whose security depends on computational intractability in solving the Discrete Logarithm Problem (Paar & Pelzl, 2010).

However, the communication may be exposed to a threat like a Man-in-the-Middle attack as the authentication of the communication entities is not provided in DHKE (Nan Li, 2010). Man-in-the-middle (MitM) attack is an eavesdropping attack done by a third party in communication. They may aim to gain confidential information such as personal information or password or sabotage or spy on the victim (Swinhoe, 2019). This study will use the Station-to-Station (STS) protocol to overcome the drawback of the DHKE protocol. Diffie-van Oorschot-Wiener introduced STS as a discrete logarithm-based key agreement scheme (Hankerson *et al.*, 2004). STS provides security to DHKE protocol by providing digital signatures to authenticate in DHKE.

A digital signature scheme is one of the schemes that assure the authenticity of the digital message. It works like a handwritten signature, proving to the receiver that the message was from the sender who generated it (Sako, 2011). One of the methods to construct digital signatures is using a one-time signature scheme (OTS). Note that this method has a limit to sign and deficiency on the length of signature and size of the public and private key (Zaverucha *et al.*, 2010). The OTS is known as the Lamport Signature Scheme. Leslie B. Lamport invented the Lamport signature scheme in 1979, and the process involved three steps: key generation, signing and verification. In a key generation, a key pair, a private key and a public key, were generated (Zentai, 2020). In this paper, the Lamport signature scheme will be implemented into the STS protocol to overcome the authentication problem in DHKE. As a result, the algorithm for Lamport Digital Signature Scheme with STS protocol will be shown.

2. Literature Review

Lamport's signature scheme is a one-time signature scheme proposed by Lamport and Rabin (Thanalakshmi *et al.*, 2022). This scheme is evaluated based on the idea of executing public keys to secure the keys by using a one-way function (Chang & Yeh, 2005). There are many benefits of one-time signatures, such as one-way functions that can be implemented using fast hash functions such as Secure Hash Algorithm (SHA-2) or Message Digest (MD5). However, although this scheme is quite fast, it tends to be onerous when it needs to authenticate multiple messages as additional data is required to generate and verify each new message.

A new scheme that will generalise the Lamport signature scheme has been suggested. In addition, an effective solution has been proposed for signing a long letter to make the proposed scheme more realistic and accurate. They suggested a new scheme to increase the size of Lamport's one-time signature. Key generation, signing, and verification will all be part of the new process (Chang & Yeh, 2005). The drawback of this research is that the suggested signs of a very long message are inaccurate. Therefore, the researcher needs to strengthen the problem

by hashing the message before signing. According to Zentai (2020), the algorithm of the Lamport signature scheme is not secure when generating multiple signatures with the same key, but it works when some modifications are added.

The Diffie-Hellman algorithm will employ modular arithmetic and discrete logarithm to generate a shared key for both sender and receiver, utilising a communication channel in which both sender and receiver choose a common prime integer, *p,* and primitive root, *q* (Aryan *et al.*, 2017). However, many attacks are possible to happen in the basics of the Diffie- Hellman algorithm.

Kallam (2015) stated that the basics of DHKE protocol are not secure against a Man-in-the-Middle (MitM) attack. In general, an authentication process will be expected to guarantee that at whatever point Alice sends a message to Bob, the beneficiary must be Bob and not Eve (attacker). It is very important and generally the norm to discard the keys after use so that there will be no long-term keys that can be revealed to bring issues or problems later.

Diffie *et al.* (1992) stated that the STS protocol consists of Diffie-Hellman key establishment followed by an exchange of authentication signatures. For the key establishment, they assumed the parameters used, such as the primitive element α and the specification of a particular cyclic group, are permanent and known to all users. They also assumed that Alice knew Bob's genuine public key and vice versa. In their research, they concluded that the exchanged exponentials are digitally signed and retransmitted during the STS protocol. According to Krishna Kumar *et al.* (2012), Eve, an eavesdropper, will not be able to change or modify the original exponentials without initiating a failure during Alice and Bob's key agreement. They also show how to defeat MitM attacks by using the STS protocol.

3. Methodology

3.1 Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange is a cryptographic protocol that enables two parties to create a shared secret key to communicate over an insecure communications channel. Furthermore, the diffie-Hellman protocol allows two parties to securely share a session key, which can then be used to encrypt messages symmetrically. Besides, the Diffie-Hellman protocol also calculates a session key based on mutually decided parameters shared in the earlier phase. A key agreement protocol is the protocol type (Kallam, 2015).

The two public numbers shared by both parties are large prime integer *p* and base *an* in this protocol. The complexity of estimating discrete logarithms contributes to the effectiveness of the Diffie-Hellman key exchange protocol. The Diffie-Hellman protocol was the first practical solution to the key distribution problem, allowing two parties who had never met in advance or exchanged keying material to establish a mutual secret by exchanging messages over an open channel. Using a symmetric key cypher, the key could be used to encrypt subsequent communications. The protection is based on the Diffie-Hellman problem's intractability and the major issues of computing discrete logarithms. "Alice" and "Bob" are the names of the two parties involved in the main exchange (Mishra & Kar, 2017).

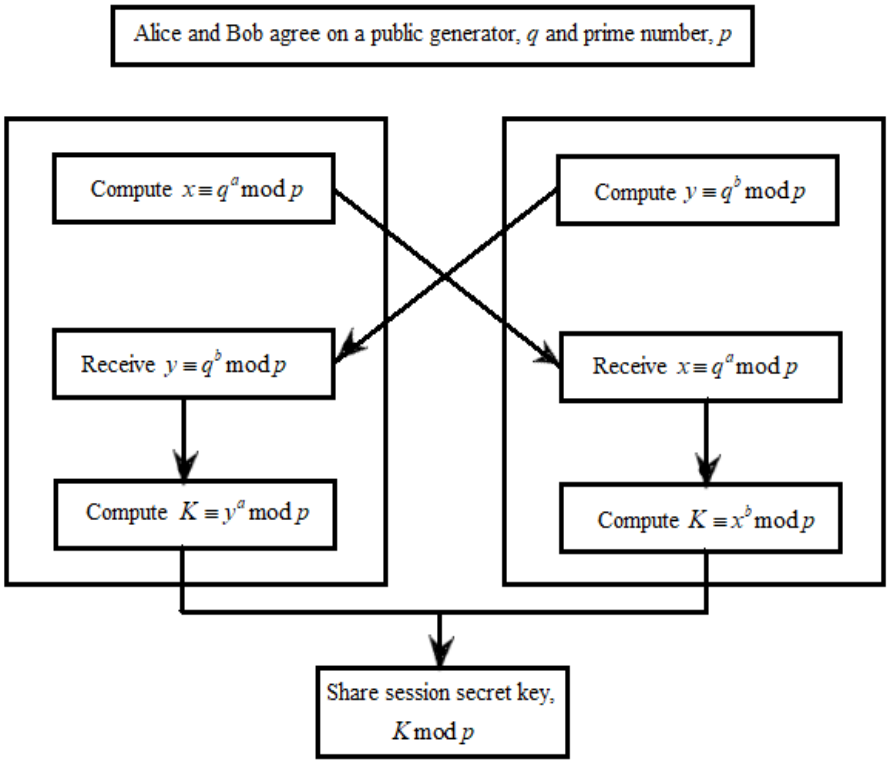

Figure 1. Diffie-Hellman Key Exchange

Figure 1 shows a simple protocol that makes use of the Diffie-Hellman calculation. The following are the functions of an algorithm.

1. Alice and Bob agree to choose the prime number p and q an integer that is a primitive root p.
2. Alice and Bob choose the private key $a < p$ and $b > p$, respectively.
3. Alice and Bob compute public key, $x \equiv q^a \bmod p$ and $y \equiv q^b \bmod p$.
4. Alice and Bob exchange their public key.
5. Alice receives Bob's public key, y. Bob receives Alice's public key, x.
6. Using the message that they received, they calculate the symmetric key. Alice and Bob compute symmetric keys $K \equiv y^a \bmod p$ and $K \equiv x^b \bmod p$ respectively.
7. Symmetric key, K is the session shared secret.

Alice and Bob have agreed upon a symmetric key as their secret key. These two calculations produce identical results:

$$\begin{aligned} K &\equiv y^a \bmod p \\ &\equiv \left(q^b \bmod p\right)^a \bmod p \\ &\equiv \left(q^b\right)^a \bmod p \end{aligned}$$

by the rules of modular arithmetic

$$\equiv q^{ba} \bmod p$$
$$\equiv \left(q^{a} \bmod p\right)^{b} \bmod p$$
$$\equiv \left(q^{a}\right)^{b} \bmod p$$
$$\equiv x^{b} \bmod p$$

3.2 Station-to-Station Protocol

The Station-to-Station protocol is a three-pass version of the classic Diffie-Hellman protocol that allows two parties to construct a shared secret key with the mutual entity and mutual explicit key authentication. Digital signatures are used in the STS. A communication's digital signature is a number based on a secret known only to the signer and the content of the message being signed. The RSA signature scheme is frequently used with the STS protocol. To use an RSA signature scheme, users must first establish public and private key pairs (Krishna Kumar *et al.*, 2012). The STS protocol is shown in Figure 2.

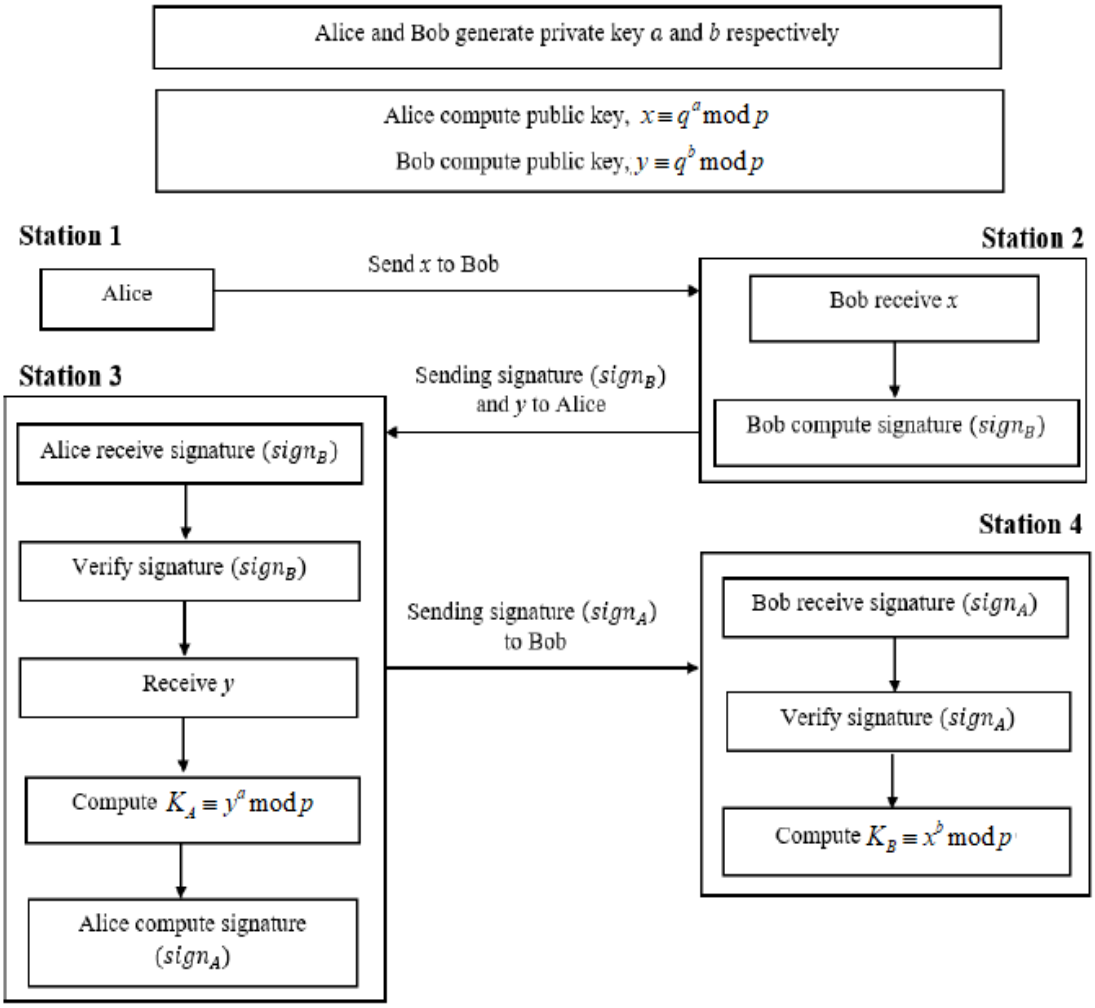

Figure 2. Station-to-Station protocol

The following is an algorithm for the Station-to-Station protocol:

Key Generation:

1. Alice generates a private key, a where $1 \le a \le p-2$.
2. Bob generates a private key, b where $1 \le b \le p-2$.
3. Alice compute public key, $x \equiv q^a \bmod p$.
4. Bob compute public key, $y \equiv q^b \bmod p$.

Station 1

1. Alice sends her public key, x to Bob.

Station 2

1. Bob receives Alice's public key, x.
2. Bob computes his digital signature, $sign_B$.
3. Bob sending his digital signature, $sign_B$ and public key, y to Alice.

Station 3

1. Alice receives and verify the Bob's digital signature, $sign_B$.
2. If the signature valid, Alice will receive Bob's public key, y and compute session shared key, $K_{AB} \equiv y^a \bmod p$.
3. Alice computer her digital signature, $sign_A$
4. Alice sends her digital signature, $sign_A$ to Bob.

Station 4

1. Bob receives and verify the signature, $sign_A$.
2. If the signature valid, Bob compute $K_{BA} \equiv x^b \bmod p$.

3.3 Lamport Digital Signature

Lamport signatures are one-time signature schemes that allow the signer to sign his messages. This signature allows the verifier to verify messages that the signer constructed. The level of security of Lamport Signature is due to the hardness of inverting a cryptographic hash function (Zentai, 2020).

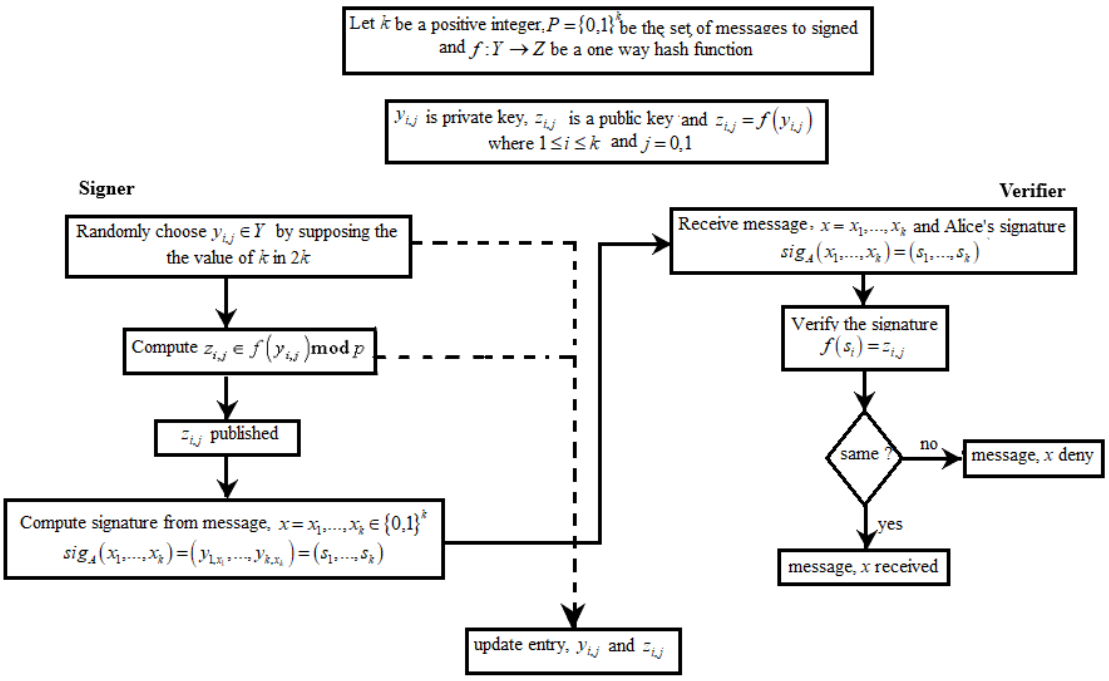

Figure 3. Lamport Signature Scheme

Figure 3 shows the process for Lamport's signature. There are three steps in the Lamport digital signature: key generation, signing and verification. In addition, the following are the functions of an algorithm.

Step 1: Key generation

i. Let k be a positive integer and let $P=\{0,1\}^k$. Suppose $f: Y \to Z$ is one-way function.

ii. The key K consists of the $2k$ $z_{i,j}$'s and the $2k$ $z_{i,j}$'s i,j's. The $y_{i,j}$'s is the private key while $z_{i,j}$'s is the public key.

iii. Signer randomly chooses $y_{i,j} \in Y$ where $1 \le i \le k$ *and* $j=0,1$ by supposing the value of K in $2k$.

iv. Let $z_{i,j}=f(y_{i,j})$, $1 \le i \le k$ *and* $j=0,1$ to compute the public key $z_{i,j}$.

Step 2: Sign

i. Signer wants to sign the message, $x=x_1,...,x_k \in \{0,1\}^k$.

ii. For $K=(y_{i,j},z_{i,j}:1 \le i \le k, j=0,1)$, $sig_A(x_1,...,x_k)=(y_{1,x_1},...,y_{k,x_k})=(s_1,...,s_k)$.

iii. Signer sends the message $x=(x_1,...,x_k)$ along with his signature, $sig_A(x_1,...,x_k)=(s_1,...,s_k)$ to verifier.

Step 3: Verification

i. Verifiers receive the signer's messages, $x=(x_1,...,x_k)$ and signature, $sig_A(x_1,...,x_k)=(s_1,...,s_k)$.

ii. A signature $(s_1,...,s_k)$ on the message $(x_1,...,x_k)$ is verified as follows: $ver_A((x_1,...,x_k),(s_1,...,s_k))=true \leftrightarrow f(s_i)=z_{i,j}, 1\le i\le k$.

iii. If the signature is verified, Bob will receive the message from Alice.

The Lamport digital signature scheme is not the most realistic digital signature method since user must produce a new match of keys for every message means the user need to generate $y_{i,j}$ and $z_{i,j}$ again every time to sign a new message. Moreover, the previous key must be destroyed and cannot be used again.

4. Result and Discussion

4.1 Application of Lamport Digital Signature Scheme to Station-to-Station (STS) protocol

The Lamport digital signature Scheme will be applied to the Station-to-Station (STS) protocol in this project to provide authentication elements to it.

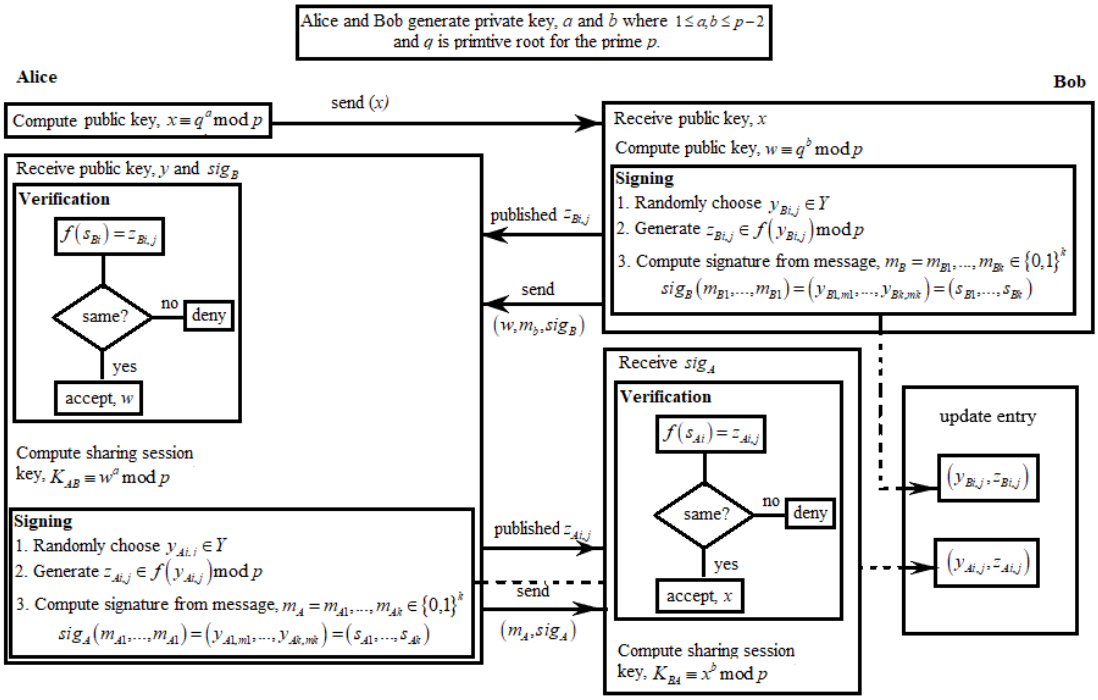

Figure 4. Implementation of STS Protocol by using Lamport Signature

Figure 4 illustrates the steps in adding a Lamport digital signature into the STS protocol. To start the protocol, let us assume there are two parties. Alice and Bob want to send their

respective public key, *x* and *w,* to each other securely so that they can generate a sharing session key, *K*. The algorithm to implement STS by using the Lamport signature is as follows.

1. Alice and Bob generate private key, *a* and *b* where $1 \le a,b \le p-2$ and they agree to choose *q* as the primitive root for the prime *p*.
2. Alice will compute public key, $x \equiv q^a \bmod p$ and sending it to Bob.
3. Bob receives Alice's public key, *x*. Bob compute his public key, $w \equiv q^b \bmod p$.
4. Bob randomly chooses $y_{Bi,j} \in Y$ where $1 \le i \le k$ and $j \in \{0,1\}$.
5. Bob generates $z_{Bi,j} \in f\left(y_{Bi,j}\right) \bmod p$.
6. Bob compute signature from the message $m_B = m_{B1},...m_{Bk} \in \{0,1\}^k$ where $sig_B\left(m_{B1},...,m_{Bk}\right) = \left(y_{B1,m1},...,y_{Bi,mk}\right) = \left(s_{B1},...,s_{B2}\right)$. Bob sends $\left(w, m_B, sig_B\right)$ to Alice.
7. Alice receives Bob's message and signature, $\left(w,m,sig_B\right)$.
8. Alice uses $sig_B = \left(s_{B1},...,s_{Bk}\right)$ to verify Bob's signature by using $f\left(s_{Bi}\right) = z_{Bi}$. If the same, then Bob's public key is accepted. If not, then the message is denied.
9. Alice randomly chooses $y_{Ai,j} \in Y$ where $1 \le i \le k$ and $j \in \{0,1\}$.
10. Alice generates $z_{Ai,j} \in f\left(y_{Ai,j}\right) \bmod p$.
11. Alice compute signature from the message $m_A = m_{A1},...m_{Ak} \in \{0,1\}^k$ where $sig_A\left(m_{A1},...,m_{Ak}\right) = \left(y_{A1,m1},...,y_{Ai,mk}\right) = \left(s_{A1},...,s_{A2}\right)$. Ali sends $\left(m_A, sig_A\right)$ to Bob.
12. Bob receives Alice's signature, $\left(m_A,sig_A\right)$.
13. Bob uses $sig_A = \left(s_{A1},...,s_{Bk}\right)$ to verify Alice's signature by using $f\left(s_{Ai}\right) = z_{Ai}$. If the same, then Alice's public key is accepted. If not, then the message is denied.
14. Alice and Bob compute session sharing key, $K_{AB} \equiv w^a \bmod p \equiv x^b \bmod p$.

4.2 The security of Station-to-station protocol using Lamport digital signature.

Lamport digital signature scheme is a one-way signature scheme in which the private key and public key will be updated or regenerated after previous communication is done. Hence, the attacker has a problem analysing for breaking the keys. Furthermore, the length of the key is equal to the length message, making the protocol hard to break. Therefore, the security of the STS protocol is well maintained.

5. Conclusion

From the result and discussion, it can be concluded that the application of the Lamport digital signature scheme in STS protocol is valid as the signature of both parties were proved. Thus, a new algorithm of the STS protocol by using the Lamport digital signature scheme. It is more secure than other signatures like RSA, Rabin, ElGamal and ECDSA since they can have the following duplicate signature key selection property. Meanwhile, the Lamport Digital signature scheme is a one-time signature scheme. The previous key created cannot be used

again for the next communication. The algorithm proposed will provide the authentication in the DHKE protocol, which signifies that both objectives in this study were achieved. So, the DHKE protocol can be used widely and safely as the authenticity of both communication parties has been secured. The authentication in the DHKE protocol is very important to prevent third-party interruption in communication such as MitM attack. The authenticity of information secures the integrity of the message and guarantees the message comes from the genuine sender, and only the intended receiver can read the message.

6. Acknowledgement

The authors would like to thank Universiti Teknologi MARA (UiTM) Negeri Sembilan, Seremban Campus for the facilities involved in making this research success.

References

Aryan, Kumar, C., & Durai Raj Vincent, P. M. (2017). Enhanced Diffie-Hellman algorithm for reliable key exchange. *IOP Conference Series: Materials Science and Engineering*, *263*, 042015. https://doi.org/10.1088/1757-899X/263/4/042015

Bellare, M., & Rogaway, P. (2005). Introduction to Modern Cryptography. http://www-cse.ucsd.edu/users/mihirhttp://www.cs.ucdavis.edu/∼rogaway

Burmester, M., & Desmedt, Y. G. (1996, April). Efficient and secure conference-key distribution. In *International Workshop on Security Protocols* (pp. 119-129). Springer, Berlin, Heidelberg.

Chang, M. H., & Yeh, Y. S. (2005). Improving Lamport one-time signature scheme. *Applied Mathematics and Computation*, *167*(1), 118–124. https://doi.org/10.1016/J.AMC.2004.06.108

Diffie, W., Van Oorschot, P. C., & Wiener, M. J. (1992). Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, *2*(2), 107–125. https://doi.org/10.1007/BF00124891

Gómez Pardo, J. L. (2013). *Introduction to Cryptography with Maple*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-32166-5

Hankerson, D., Menezes, A., & Vanstone, S. (2004). *Guide to Elliptic Curve Cryptography*. Springer Verlag New York.

Kallam, S. (2015). Diffie-Hellman: Key exchange and public key cryptosystems. *Master Degree of Science, Math and Computer Science, Department of India State University, USA*. http://cs.indstate.edu/~skallam/doc.pdf

Krishna Kumar, C., Jai Arul Jose, G., Sajeev, C., & Suyambulingom, C. (2012). Safety measures against man-in-the-middle attack in key exchange. *ARPN Journal of Engineering and Applied Sciences*, *7*(2), 243–246.

Lamport, L. (1979). Constructing digital signatures from a one way function.

Mishra, M. R., & Kar, J. (2017). A study on Diffie-Hellman Key Exchange Protocols. International Journal of Pure and Apllied Mathematics, 114(2). https://doi.org/10.12732/ijpam.v114i2.2

Nan Li (2010). Research on Diffie-Hellman key exchange protocol. *2010 2nd International Conference on Computer Engineering and Technology*, *4*, V4-634-V4-637. https://doi.org/10.1109/ICCET.2010.5485276

Paar, C., & Pelzl, J. (2010). Understanding Cryptography. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04101-3

Sako, K. (2011). Digital Signature Schemes. In H. C. A. van Tilborg & S. Jajodia (Eds.), *Encyclopedia of Cryptography and Security* (pp. 343–344). Springer US. https://doi.org/10.1007/978-1-4419-5906-5_17

Swinhoe, D. (2019). *What is a man-in-the-middle attack? How MitM attacks work and how to prevent them?* CSO Online. https://www.csoonline.com/article/3340117/what-is-a-man-in-the-middle-attack-how-mitm-attacks-work-and-how-to-prevent-them.html

Thanalakshmi P., Anitha R., Anbazhagan N., Park C., Joshi G. P., & Seo C. A. Hash-Based Quantum-Resistant Designated Verifier Signature Scheme. *Mathematics*. 2022; 10(10):1642. https://doi.org/10.3390/math10101642

Zaverucha, G. M., Stinson, D. R., & Cheriton, D. R. (2010). *Short One-Time Signatures*. https://eprint.iacr.org/2010/446.pdf

Zentai, D. (2020). On the efficiency of the Lamport Signature Scheme. *Land Forces Academy Review*, *25*(3), 275–280. https://doi.org/10.2478/raft-2020-0033