Emirates Journal for Engineering Research

Volume 28 | Issue 2 Article 3

8-14-2023

Efficient and secure digital signature algorithm (DSA)

Nissa Mehibel
university mh'amed bougara of boumerdes, n.mehibel@univ-boumerdes.dz

M'HAMED HAMADOUCHE
hamadouche-mhamed@hotmail.com

Recommended Citation
Mehibel, Nissa and HAMADOUCHE, M'HAMED (2023) "Efficient and secure digital signature algorithm (DSA)," *Emirates Journal for Engineering Research*: Vol. 28: Iss. 2, Article 3.
Available at: https://scholarworks.uaeu.ac.ae/ejer/vol28/iss2/3

Efficient and secure digital signature algorithm (DSA)

1. Introduction

Public key cryptography is a revolutionary method in the computer world proposed in 1975 by Diffie-Hellman [1]. It solves the problems encountered by secret key cryptography such as sharing the secret key in a public network and non-repudiation. The latter is ensured by a cryptographic mechanism called the digital signature.

The digital signature has a very important role in cryptography, in addition to ensuring non-repudiation, it also ensures the integrity of messages and the authentication of users. Its basic principle consists in generating a digital signature for each sender, after which the signature must be verified by the receiver to ensure that the message has not been altered by a third party after it has been signed.

The first digital signature algorithm was proposed in 1985 by ElGamal [2], the robustness of this protocol is based on the discrete logarithm problem (DLP). The DSA digital signature algorithm was proposed in 1994 by the US National Institute of Standards and Technology (NIST) and was specified in a US Government Federal Information Processing Standard (FIPS 186) called the Digital Signature Standard (DSS) [3], DSA is a variant of the ElGamal digital signature algorithm. In 1997, an attack was presented by Bellare et al [4] in which they showed that it is possible to recover the signer's secret key if the same random number is generated to sign two different messages. This attack is due to the use of an inadequate pseudo-random number generator.

Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of DSA pertaining to the ElGamal family of signature schemes [5]. The Elliptic Curve Digital Signature Algorithm (ECDSA) was first proposed in 1992 by Scott Vanstone [6]. It was accepted in 1998 as an ISO (International Standards Organisation) standard. It was also accepted in 1999 as an ANSI (American National Standards Institute) standard and in 2000 as IEEE (Institute of Electrical and Electronics Engineers) and NIST standards [7]. The robustness of ECDSA rest in the difficulty of solving the elliptic discrete logarithm problem (ECDLP). The advantage of elliptic curves in cryptography is that they provide a level of security that matches that of existing public key cryptosystems, while using a smaller key size and computation time [8]. However, ECDSA inherits the weakness of DSA when using a bad pseudo-random number generator. In 2006, Liao and Shen [9] proposed an improved ECDSA scheme by using two random numbers to generate a signature to overcome the weakness of ECDSA. In 2011, Junru [10] also proposed an improved scheme to reduce the computational cost of ECDSA while maintaining the security level of ECDSA. In 2016, Chande and Lee [11] showed that the algorithms proposed by Junru [10] are vulnerable to the repeated random number attack and proposed to improve Junru's [10] ECDSAs by using the same principle proposed by Liao and Shen [9] to reduce the probability of deriving the signer's secret key from an adversary. In 2019, Mehibel and Hamadouche [12] showed that the improvements of Liao and Shen [9] and Chande and Lee [11] have an anomaly (defect) in the signature verification phase due to missing parameters that are necessary for the receiver to be able to verify the validity of the signature. They also proposed improvements to the algorithms of Liao and Shen [9] and Chande and Lee [11] by introducing the parameters not considered by the latter while maintaining the security of the proposed schemes. In 2020, Zahhafi and Khadir [13] proposed a new digital signature scheme inspired by the DSA algorithm. Their method is an alternative to the classical DSA protocol if it is broken. The disadvantage of their algorithm is that the generation of the signature has three parameters instead of two for DSA, as well as the use of an additional modular exponentiation operation in both phases, generation and verification of the signature.

In this paper, we analyze the digital signature algorithm of Zahhafi and Khadir [13] and propose a new digital signature improvement.

The rest of the paper is organized as follows: in the section "Review of Zahhafi and Khadir DSA", we present the scheme of Zahhafi and Khadir [13] and its security analysis. The section "Proposed improvement" presents the improvement of the digital signature scheme of Zahhafi and Khadir [13]. The section "Security analysis and performance evaluation" deals with the security and performance analysis of the improved scheme.

2. Review of Zahhafi and Khadir DSA

Zahhafi and Khadir [13] have proposed a new digital signature scheme as an alternative to the classical DSA protocol if the latter is broken. Their contribution consists in using two random numbers in the signature generation phase in order to overcome the attack of derivation of the signer's private key in the case of using an inadequate pseudo-random number generator. However, the Zahhafi and Khadir [13] protocol is more computationally expensive than DSA and this is due to the use of an additional modular exponentiation operation in the signature generation and verification steps. In this section, we present and analyze the protocol of Zahhafi and Khadir [13].

2.1. Protocol DSA of Zahhafi and Khadir (2020)

The digital signature algorithm of Zahhafi and Khadir [13] consists of three steps: key generation, signature generation and signature verification.

<u>*Key generation phase*</u>

The signer selects two prime numbers p and q such that :
q divides $p - 1$, $2^{t-1} < q < 2^{t}$ with: $t \in \{160, 256, 384, 512\}$
$2^{L-1} < p < 2^{L}, 768 < L < 1024$ and L is a multiple of 64.

Then it selects a primitive root $a \bmod p$ and calculates $g = a^{\frac{p-1}{q}} \bmod p$.

The signer also selects an integer d such that $1 \leq d \leq q - 1$ and calculates $Q = g^{d} \bmod p$. Finally, he publishes.(p, q, g, Q) and keeps the parameter d secret as his private key.

<u>*Signature generation phase*</u>

The signer chooses two random numbers v and $w < q$ and calculates :
$V = g^{v} \bmod p$ and $W = (g^{w} \bmod p) \bmod q$
Then, he calculates the S= $w^{-1}(h(m) + d\,V + v\,W) \bmod q$
Where $0 < V < p$ and $S, W > 0$ and $h(m)$ est is the hash of message *m* such that$h(\,.\,)$ is a hash function.

The signer sends the parameters (S, V, W) comme as the signature of the message *m*.

<u>*Signature verification phase*</u>

To verify the signature (S, V, W) of the message *m* le the receiver has to perform the following :

- He first downloads the public parameters of the signer $(p, q, g, h(.), Q)$.
- He checks if $0 < V < p$ and $S, W > 0$. Otherwise, it rejects the signature.
- $u_1 = S^{-1}\,h(m) \bmod q$
- $u_2 = S^{-1}\,V \bmod q$
- $u_3 = S^{-1}\,W \bmod q$

- $Y = (g^{u_1} Q^{u_2} V^{u_3}) \, mod \, p) \, mod \, q$
- If $Y = W$ he receiver accepts the signature, otherwise it rejects it.

2.2. Weakness of Zahhafi and Khadir DSA

The DSA protocol proposed by Zahhafi and Kahdir [13] allows to remedy the signatory's private key derivation attack (repeated random number attack) if the same random number is used to sign two different messages (*m1* and *m2*) by using two random numbers (v and w) in the signature generation phase. However, if the signer uses the same pair (v and w) to sign two different messages (*m1* and *m2*) an adversary aware of this fact is able to find the random number w as follows:

Let S_1be the signature of message S_1 and $S_2$the signature of message m_2:

$$S_1 = w^{-1}(h(m_1) + d\,V + v\,W) \, mod \, q \qquad (1)$$

$$S_2 = w^{-1}(h(m_2) + d\,V + v\,W) \, mod \, q \qquad (2)$$

Subtracting formula (1) from formula (2), we obtain:

$$S_1 - S_2 = w^{-1}\big(h(m_1) - h(m_2)\big) \, mod \, q \qquad (3)$$

From formula (3) we can obtain the random number w as follows:

$$w = (S_1 - S_2)^{-1}\big(h(m_1) - h(m_2)\big) \, mod \, q \qquad (4)$$

Therefore, the random number w is not protected and can be easily calculated in case the same random numbers (v and w) are used to sign two different messages (*m1* and *m2*). Moreover, the parameters (v and w) are secret i.e. known only by the signer. Knowing the value of the secret key w compromises the security of the Zahhafi and Kahdir [13] protocol and makes it vulnerable.

Furthermore, the signer's private key can be expressed from formula (1) and (2) as follows:

$$d = V^{-1}(S_1 w - h(m_1) - v\,W) \, mod \, q \qquad (5)$$

$$d = V^{-1}(S_2 w - h(m_2) - v\,W) \, mod \, q \qquad (6)$$

Formula (5) and (6) show that all values are public to determine d except the value of v, if the latter is known the value of d can be computed from formula (5) or (6), as it has been pointed out in [14,15,16]. Therefore, the scheme of Zahhafi and Kahdir [13] creates a single instance to solve equation (5) and (6) which increases the risk of determining the value of the signer's private key by an adversary.

3. Proposed improvement

In this section, we present a new digital signature scheme to improve the Zahhafi and kahdir [13] protocol in order to overcome the weakness encountered in the latter. Our improved digital signature scheme is described below:

Key generation phase

This phase is the same as for the DSA protocol [17]. The signer first chooses the public parameters namely a prime number q of 160 bits and a prime number p of 1024 bits with the property$q \mid p - 1$, and chooses a primitive root $a \; mod \; p$ and calculates $g = a^{\frac{p-1}{q}} mod \; p$, then follows:

- He chooses a random number $d \in [1, q-1]$.
- He calculates $= g^{d} \; mod \; p$, Q is his public key..
- He publishes$(p, q, g, h(.), Q)$ and keeps the parameter *d* secret as his private key.

Signature generation phase

The signer performs the following operations to generate the signature of the message *m* :

- He chooses two random numbers v and $w \in [1, q-1]$
- He calculates $V = g^{v} \; mod \; p$ and $= g^{w} \; mod \; q$.
- He checks if $V = 0$ and $W = 0$ he must re-select other random numbers.
- He calculates $S = v^{-1}(w(h(m) + V) + d \; W) \; mod \; q$.
- The signature of the message *m* is (S, V, W).

Signature verification phase

To verify the signature (S, V, W) of the message *m* the receiver has to download the same parameters used by the signer $(p, q, g, h(.), Q)$ then he does the following:

- He checks if S,$W \in [1, q-1]$ and $V \in [1, p-1]$.
- He calculates $u_1 = S^{-1}(h(m) + V) \; mod \; q$ and $u_2 = S^{-1}W \; mod \; q$.
- He calculates $Y = W^{u_1}Q^{u_2} \; mod \; p$
- He accepts the signature only if Y$= V$.

Proof of verification of the signature

$$Y = W^{u_1} Q^{u_2} mod\ p$$

$$= g^{wS^{-1}(h(m) + V)} g^{S^{-1}dW} mod\ p$$

$$= g^{S^{-1}(w(h(m) + V) + d\,W)}\ \ mod\ p$$

$$= g^{v} mod\ p$$

4. Security analysis and performance evaluation

In this section, we analyze the security of the proposed scheme and show that our solution is efficient with respect to performance evaluation.

4.1. Security analysis

Here we discuss some possible attacks [2, 13,18 ,19 ,20] as well as the security properties satisfied by the proposed scheme.

Integrity

Integrity signifies that information cannot be accidentally or intentionally altered, modified by an adversary during transmission. The improved digital signature scheme provides integrity. If an attacker intercepting the data sent and modifying the message sent by the signer, he must be in possession of the private key "*d*" in order to be able to create a valid signature for the corrupted message. Otherwise the modified message is rejected as invalid by the receiver. To find the value of the private key "*d*" from the public key "*Q*" the attacker must solve the Discrete Logarithm Problem (DLP).

Authenticity

Authenticity means that the recipient can verify the identity of the sender or the origin of the source. With regard to the proposed scheme, the authenticity of the sender is ensured by signing the message, because the sender signs the message with his private key. If an attacker pretends to be a legitimate user in order to sign a message he must generate a valid signature, to do so he must be in possession of the private key of the legitimate user. Otherwise, the signature of the message generated by the attacker will be invalid and the recipient rejects the message in the signature verification phase. Moreover, to determine the private key of the legitimate user, the attacker will be faced with the discrete logarithm problem (DLP).

Non-repudiation

Non-repudiation means that an entity cannot subsequently deny the veracity of the originating person's signature or sending of a message, and the recipient has proof of the sender's identity. In our scheme, a message is electronically signed by the sender (the signer), and the message is sent with the value of his signature to the recipient. The sender cannot later repudiate having signed the message because the digital signature was created with his private key. The recipient verifies the signature of the message with the signer's public key. Once the signature is validated, the sender cannot deny having sent the messages containing the signature. Additionally, it is not possible to calculate the signer's private key from their public key and to resolve DLP in the event that a malicious attacker tries to impersonate a sender to a recipient. In this way, the proposed scheme provides reliable action to realize the non-repudiation service.

Unforgeability

Unforgeability indicates that only the sender of the message can generate a valid signature for a message. The proposed scheme provides the security attribute of unforgeability. If an attacker masquerades as an honest sender and forges a legal digital signature, it should be relatively easy to detect the forging or alteration by a authentic mechanism, since only the authentic sender can generate a valid signature that is verified by the signature verification phase. If an attacker wants to generate the forge signature, he needs the secret parameters *v* and *w*. However, to find the two random numbers *v* and *w* either the attacker must solve the equation $S = v^{-1}(w(\, h(m) + V) + d\, W\,)\, mod\; q$ with three unknown variables, or he will be supported at DLP by trying to find *v* and *w* from $V \ = \ g^{v}\; mod\; p$ and $W \ = \ g^{w}\; mod\; q$.

Key-only attacks

In this attack, the adversary only knows the public key of the signer. In the proposed scheme, if an attacker wishes to find the private key *d* of the signer from the public key Q where $Q = g^{d} modp$ he will have to solve the discrete logarithm problem.

key-signature attack

In this attack, the adversary only knows the signature of the message. In the proposed scheme, if an attacker wishes to find the private key *d* of the signer from the signature (S, V, W) of the message *m* he will have to solve the equation (7).

$$d = \ W^{-1}\big(Sv - w(\, h(m) + V)\big)modq \qquad (7)$$

Formula (7) is an equation with two unknowns; so that an attacker can determine the private key of the signer he must test all the possible values for the parameters v and w which is not easy for the attacker to propose a valid solution. In addition, the signer's secret key d has a unique possibility, therefore the attacker will never be sure of the correct value of d.

Repeated random number attack

This attack is due to the use of a bad pseudo-random number generator which generates the same random number to sign different messages. Assume that in the proposed scheme the same pair of random numbers ($v$and w) has been used to sign two different messages ($m_1$and m_2), we obtain:

$$S_1 = v^{-1}(w(h(m_1) + V) + dW) mod\ q \qquad (8)$$

$$S_2 = v^{-1}(w(h(m_2) + V) + dW) mod\ q \qquad (9)$$

Where S_1 is a signature of $m_1$and S_2 is a signature of m_2 , from the equation (8) and equation (9) we can obtain the following relation : $v = (S_1 - S_2)^{-1} w \left(h(m_1) - h(m_2)\right) mod\ q$

However, the adversary cannot recover neither the value of v nor the value of w.

Known-message attack

In this attack, the adversary collects a list of messages along with their valid signatures and tries to find the value of the secret key *d*. Assume in the proposed scheme that an attacker collects *n* valid signatures for *n* message, he obtain a set of *n* equations as follows :

$$\left.\begin{array}{l} S_1 = {v_1}^{-1}(w_1(h(m_1) + V_1) + dW_1) mod\ q \\ S_2 = {v_2}^{-1}(w_2(h(m_2) + V_2) + dW_2) mod\ q \\ \ldots\ldots \\ S_{n-1} = {v_{n-1}}^{-1}(w_{n-1}(h(m_{n-1}) + V_{n-1}) + dW_{n-1}) mod\ q \\ S_n = {v_n}^{-1}(w_n(h(m_n) + V_n) + dW_n) mod\ q \end{array}\right] \text{(E)}$$

The set (E) of equations obtained by the attacker contains *3*n* unknown parameters (d, v_i, w_i) wherev_i et w_i are sercret numbers chosen at random and $i \in \{1,2,3 \ldots n\}$.

However, he tests all the possible cases to find the values of the latter, which is extremely difficult and cannot be achieved in a reasonable time. Therefore, this attack is not effective.

Generic chosen-message attack

In this attack, the adversary chooses a list of messages before the signatures are seen; subsequently he obtains the valid signatures for the chosen messages from the signer. In addition, the messages chosen by the adversary are fixed and independent of the signer's public key. This attack is called "generic" because it does not depend on the signer's public key. Assume in the proposed scheme that the attacker chooses a list of messages (m_1, m_2,..m_n) and obtains their valid signatures [(S_1, V_1, W_1), ((S_2, V_2, W_2),…((S_n, V_n, W_n)] from the signer and tries to find the value of the latter's private key *d*. To do this, the attacker must solve the equations of the set (E), he can find several possibilities. However, the secret key d has a unique possibility. Therefore, without knowing the signer's public key$Q = g^d modp$, the attacker will never be sure of the correct value of the private key *d*. Therefore, this attack is not efficient.

Attacks for forging signatures

Forging a signature means producing a new signature by an opponent. Assume in the proposed scheme that an adversary wishes to produce a signature for a message *m*, he starts by choosing two parameters at random and tries to find the third as follows:

- If the attacker randomly sets the values of V and W and wishes to determine S, he must solve the discrete logarithm problem $x^t = W \, mod \, q$, such that $x = g^{h(m)+v} Q^w \, mod \, q$ and $t = \, s^{-1} \, mod \, q$.
- If the attacker randomly sets the values of W and S and wishes to determine V, he must solve the equation $(y + W)^t k = V \, mod \, p$ such that $y = g^{h(m)+V} \, mod \, p$,$k = Q^{w\,t} \, mod \, p$ and $t = \, S^{-1} \, mod \, q$. However, in order to calculate *y* the adversary will need the value of *V*, therefore, the adversary will not be able to solve the modular equation $(y + W)^t k = V \, mod \, p$ in order to determine *V*.
- If the attacker randomly sets the values of V and S and wishes to determine *W*, he must solve the equation $(V^S - Q^W)x = W \, mod \, q$ such that $x = g^{(h(m)+V)^{-1}} mod \, q$. However, there is no mathematical method to solve this modular equation.

4.2. Performance evaluation.

In this section, we study the performance of the proposed scheme by comparing it with the scheme of Zahhafi and kahdir [13] in terms of computational cost and security. The performance comparison is done by evaluating the achievement of the signature generation and verification phases of each algorithm. We use the following notation to analyze the performance of the scheme:

- T_{mul}time complexity for executing the modular multiplication
- T_{exp} time complexity for executing the modular exponentiation
- T_{add} time complexity for executing the modular addition
- T_h time complexity for executing the hash-function.
- T_{inv} time complexity for executing the modular inverse.

Table 1.Comparison in terms of computation cost.

Phases	DSA of Zahhafi and kahdir [13]	Proposed DSA
Phase one	$2T_{exp}$	$2T_{exp}$
Phase two	$2T_{exp}+ 3T_{mul} + T_{inv} + 2T_{add}+ T_h$	$2T_{exp}+ 3T_{mul} + T_{inv} + 2T_{add}+ T_h$
Phase three	$3T_{exp}+ 5T_{mul} + T_{inv} + T_h$	$2T_{exp}+ 3T_{mul} + T_{inv} + T_h$
Total	$5T_{exp}+ 8T_{mul} + 2T_{inv} + 2T_{add}+ 2T_h$	$4T_{exp}+ 6T_{mul} + 2T_{inv} + 2T_{add}+ 2T_h$

(Phase one: Key generation phase, Phase two: Signature generation phase, Phase three: Signature verification generation phase)

Table 1 shows the comparison between DSA of Zahhafi and kahdir [13] and our improvement in terms of computational cost. The number of operations to be performed in the phases: key generation and signature generation of the proposed DSA are the same as DSA of Zahhafi and kahdir [13]. However, in the signature verification phase, our DSA uses one less exponentiation operation and two less modular multiplication operations than DSA of Zahhafi and kahdir [13]. This allowed us to minimize the computational cost as the modular exponentiation operation is very expensive in terms of computational time estimated at 240 T_{mul} (T_{exp} = 240 T_{mul}) [14,21].

Table 2 illustrates the comparison between DSA of Zahhafi and kahdir [13] and our improvement in terms of security features. DSA of Zahhafi and kahdir [13] provides all the security features, however, it is less secure under repeated random number attack. As we demonstrated in Section 2, in the case where the same random numbers are used to sign two different messages, one of the random numbers suggested by Zahhafi and kahdir

[13] can be easily determined in order to the repeated random number attack, as this number is supposed to be a secret key secured by the DLP.

Table 2. Comparison in terms of security features.

Security features	DSA of Zahhafi and kahdir [13]	Proposed DSA
Integrity	Assured	Assured
Authenticity	Assured	Assured
Non-repudiation	Assured	Assured
Unforgeability	Assured	Assured
Key-only attacks	Secured	Secured
key-signature attack	-	Secured
Repeated random number attack	Unsecured	Secured
Known-message attack	Secured	Secured
Generic chosen-message attack	-	Secured
Attacks for forging signatures	Secured	Secured
Integrity	Assured	Assured
Authenticity	Assured	Assured
Non-repudiation	Assured	Assured

(- : not defined in the paper of Zahhafi and kahdir [13])

Therefore, our improvement solves the problem encountered in the DSA of Zahhafi and kahdir [13] which allowed us to have a secure DSA that provides all the security features of a digital signature algorithm. In addition, the proposed DSA offers better performance in terms of computational cost compared to the DSA of Zahhafi and kahdir [13].

5. Conclusion

Zahhafi and kahdir have recently introduced a new digital signature scheme based on DLP. Our analysis shows that their scheme is not secure. With our cryptanalysis, an adversary can launch the repeated random number attack and easily retrieve one of the random numbers which is supposed to be a secret key. To overcome the security issues, we propose a new digital signature scheme which is an improvement of Zahhafi and kahdir scheme. We analyze in detail the security of the proposed scheme. Our analysis indicates that the improved scheme should be able to provide all required security attributes. Moreover, we offer better performance in terms of computational cost, as the proposed scheme uses fewer operations: two less multiplication operations and one less exponentiation operation than the scheme of Zahhafi and kahdir.

References

[1] Diffie, W., and Hellman, M. (1976). New directions in cryptography.IEEE transactions on Information Theory, 22(6), 644-654.

[2] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, *31*(4), 469-472.

[3] Poulakis, D. (2011). Some lattice attacks on DSA and ECDSA. *Applicable Algebra in Engineering, Communication and Computing*, *22*(5-6), 347-358.

[4] Bellare, M., Goldwasser, S., and Micciancio, D. (1997, August). "Pseudo-random" number generation within cryptographic algorithms: The DDS case. In *Annual International Cryptology Conference* (pp. 277-291).Springer, Berlin, Heidelberg.

[5] Angel, J., Rahul, R., Ashokkumar, C., and Menezes, B. (2017, December). DSA signing key recovery with noisy side channels and variable error rates.In *International Conference on Cryptology in India* (pp. 147-165).Springer, Cham.

[6] Vanstone, S. (1992). Responses to NIST's proposal. *Communications of the ACM*, *35*(7), 50-52.

[7] Ajeena, R. K. K., and Yaqoob, S. J. (2017, April). The integer sub-decomposition method to improve the elliptic elgamal digital signature algorithm. In *2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT)* (pp. 14-20). IEEE.

[8] Mehibel, N., and Hamadouche, M. H. (2021). Authenticated secret session key using elliptic curve digital signature algorithm. *Security and Privacy*, *4*(2), e148.

[9] Liao, H. Z., and Shen, Y. Y. (2006). On the elliptic curve digital signature algorithm. *Tunghai Science*, *8*, 109-126.

[10] Junru, H. (2011, August). The improved elliptic curve digital signature algorithm.In *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology* (Vol. 1, pp. 257-259).IEEE.

[11] Chande, M.K and Lee, C. C. ‘An improvement of a elliptic curve digital signature algorithm’, International Journal of Internet Technology and Secured Transactions, Vol. 6, No.3 pp. 219 - 230 (2016).

[12] Mehibel, N., and Hamadouche, M. H. (2019). A new enhancement of elliptic curve digital signature algorithm. *Journal of Discrete Mathematical Sciences and Cryptography*, *23*(3), 743-757.

[13] Zahhafi, L., and Khadir, O. (2020). A DSA-like digital signature protocol. *Journal of Discrete Mathematical Sciences and Cryptography*, 1-12.

[14] Mehibel, N., and Hamadouche, M. H. (2021). Authenticated secret session key using elliptic curve digital signature algorithm. *Security and Privacy*, *4*(2), e148.

[15] Biswas, G. P. (2011). Establishment of authenticated secret session keys using digital signature standard. *Information Security Journal: A Global Perspective, 20*(1), 9-16.

[16] Nyberg, K., and Rueppel, R. A. (1994). Weaknesses in some recent key agreement protocols. *Electronics Letters, 30*(1), 26-27.

[17] National institute of standard and technology (NIST). FIPS Publication 186, DSA, Department of commerce, (1994).

[18] Goldwasser, S., Micali, S., and Rivest, R. L. (1988). A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on computing*, *17*(2), 281-308.

[19] Stallings, W. (2006). *Cryptography and network security, 4/E*. Pearson Education India.

[20] Stinson, D. R. (2005). *Cryptography: theory and practice*. Chapman and Hall/CRC.

[21]Tsai, C. H., and Su, P. C. (2015). Multi-document threshold signcryption scheme. *Security and Communication Networks*, *8*(13), 2244-2256.