Benha Journal of Applied Sciences (BJAS) **print: ISSN 2356–9751**
Vol. (10) Issue (4) (2025), (199- 207) **online: ISSN 2356–976x**
https://bjas.journals.ekb.eg/
engineering sciences

Math application in smart contracts

Hala S.Omar[1], Wageda I.Elsobky[1], Tamer O.Diab[2] and Mohamed A.Elsisy[1]
[1]Basic Engineering Sciences Dept., Benha Faculty of Engineering, Benha University, Benha Egypt.
[2] Electrical Engineering Dept., Benha Faculty of Engineering, Benha University, Benha Egypt.
E-mail: hala.saeed@bhit.bu.edu.eg

Abstract
Smart contracts are blockchain-based algorithms that activate when specific conditions are fulfilled. They streamline the execution of agreements, allowing both parties to trust the outcome instantly without needing intermediaries or experiencing delays. To ensure secure and verified contract execution, cryptographic methods such as hash functions and digital signatures are used. Additionally, mathematical approaches like mathematical proofs and finite state machines are applied in designing and assessing smart contracts to guarantee their proper functionality. This paper explores the mathematical foundations of smart contracts, highlighting how they rely on mathematics to ensure immutability, security, and enforceability. A key technique behind their encryption methods is the pseudo-random number generator, which is based on chaotic maps. These chaotic maps generate highly random patterns depending on the initial seed value through complex mathematical operations. This work provides an overview of how chaotic maps are implemented in smart contracts. Additionally, the results obtained from these chaotic maps are presented showing that these maps achieve high performance in digital signature algorithms.

Keywords: Smart contracts; Chaotic maps; Digital signatures; Block chain; Encryption.

1. Introduction

Smart contracts are algorithms built on blockchain technology that activate when certain conditions are met [1]. They simplify the execution of agreements, ensuring that both parties can immediately trust the outcome without needing a middleman or facing delays [2]. Smart contracts can also be integrated with various payment systems and digital transactions, including cryptocurrencies like Ethereum and Bitcoin [3]. They can trigger the next action in a sequence once specific conditions are satisfied. Since the data within smart contracts is encrypted and stored on a shared ledger, the risk of data loss is nearly nonexistent [4]. Ethereum is the most popular platform for smart contracts, utilizing the Solidity programming language, which was developed by the Ethereum community to create smart contract applications that run on the Ethereum Virtual Machine (EVM) [5]. The concept of smart contracts was first introduced by Nick Szabo in 1994. Szabo, a cryptographer and legal scholar known for his work on digital currency, envisioned smart contracts long before the necessary digital infrastructure or distributed ledger technology was available, which limited their development and adoption at the time [6]. In 1996, Ian Grigg and Gary Howland expanded on this idea through their research on the Ricardo payment system for asset transfers, introducing the concept of Ricardian Contracts, which served as a foundation for modern smart contracts [7]. In 2008, a decentralized ledger system on a blockchain network was used to develop Bitcoin, the first cryptocurrency [8]. This technology paved the way for creating smart contract software, which encodes contract terms directly onto the blockchain. Smart contracts function using simple "if/when...then..." statements programmed into the blockchain. Once specific conditions are met and verified, a network of computers executes the agreed-upon actions, such as transferring funds, registering a vehicle, sending notifications, or issuing tickets [9]. After completing the transaction, the blockchain is updated, ensuring the deal is finalized and that the results are visible only to authorized participants. The mathematics behind smart contracts is crucial for guaranteeing their security, immutability, and enforceability [10,11].

Smart contracts use cryptographic methods, such as digital signatures and hash functions, to secure and verify the execution of agreements [12]. A digital signature is a mathematical technique used to ensure the integrity and authenticity of a message, software, or digital document. It serves as a digital equivalent of a handwritten signature or a stamped seal but offers much greater security. Digital signatures also help prevent hacking and impersonation in online communications. To further enhance the integrity of digital signatures, chaotic maps—mathematical systems known for their unpredictability—can be utilized [13].

Henri Poincaré was one of the early pioneers of chaos theory [14]. In the 1880s, while studying the three-body problem, he observed the existence of non-periodic orbits that neither consistently increase nor converge to a fixed point [14]. This insight laid the groundwork for the mathematical foundations of chaos theory, which heavily relies on the infinite recurrence of simple mathematical expressions. Chaotic maps, known for their

constantly changing behavior, have been applied in various fields, including economics, biology, robotics, and encryption [15]. They have been integrated into different encryption methods to enhance the security and performance of cryptographic protocols [16,17].

This research explores the application of chaotic maps in smart contracts. Section two provides an overview of chaotic maps, while section three explains digital signatures and highlights the most widely used schemes. In section four, a case study is presented along with its findings.

2. Chaotic maps

A chaotic map is defined as a growth function that exhibits distinct mathematical irregularities [18]. The mathematical foundation of chaos has largely developed through the iterative application of simple mathematical formulas. Chaotic maps continued to evolve and were increasingly applied across diverse fields, including robotics, biology, economics, and cryptography [19]. Significant research has focused on two primary types of chaotic systems: one-dimensional (1D) chaos and high-dimensional (HD) chaos. One-dimensional chaotic maps tend to generate sequences with lower randomness due to their moderate complexity and regularity, which may introduce security risks in visual encoding applications. In contrast, high-dimensional chaotic systems exhibit more unpredictable behavior, making them better suited for visual encoding due to their intricate structure and broader parameter range [20]. The unique properties of chaotic systems—such as determinism, ergodicity, and sensitivity to initial conditions—align closely with the confusion and diffusion principles essential for robust cryptographic frameworks. Therefore, developing new chaotic systems with enhanced chaotic behavior is crucial. This can be achieved by integrating two established one-dimensional chaotic maps to formulate a novel chaotic system with improved characteristics, including time evolution, bifurcation diagrams, and Lyapunov exponents [21].

3. Digital Signatures

Prominent algorithms such as ElGamal and Schnorr have demonstrated significant effectiveness, making them particularly well-suited for applications like smart contracts. These algorithms offer robust security protocols and deliver reliable performance [22].

3.1. Schnorr digital signature algorithm

Claus Schnorr expressed this idea in his unique terminology. This particular digital signature technique is recognized as one of the most ancient methods and is renowned for its straightforwardness. The security of this approach is dependent on the intricacy of certain discrete logarithm challenges, providing succinct and effective signatures. [23].

A. Choosing parameters:

It is commonly accepted that the discrete logarithm problem poses a challenge within the group of generators, G, of prime order, q, where each participant in the signature scheme selects a generator, g. Schnorr signatures are usually associated with this group. All parties consent to utilize the encoded hash function $H:\{0,1\}^* \rightarrow \mathbb{Z}_q$ where $\mathbb{Z}_q$represents the integers from 0 to $q-1$.

B. Generation of Key:

From $\mathbb{Z}_q$, a confidential signing key, u, is selected. The public key for verification is defined as $t = g^u \, mod \, q$.

C. Signing:

To generate a signature for a message, M:

- A random integer l is chosen from the specified range. Define a parameter w such that it:
$$w = g^l \quad (1)$$
- Subsequently, locate an element z such that:

$$z = H(w||M) \quad (2)$$

The display showcases a bit string that represents the concatenation symbol, ||.

- Suppose
$$s = l - uz \quad (3)$$

Where s represents the value of the signature.

- The combination of two distinct signatures is (s, z).

D. Verification

- Given a parameter w_v,
$$w_v = g^s t^z \quad (4)$$
- Let's assume
$$z_v = H(w_v||M) \quad (5)$$
- The signature will be deemed authenticated only if z_v is equal to z.

3.2. Elgamal digital signature algorithm

The Elgamal signature scheme was built upon the challenge of computing discrete logarithms. It was initially proposed by Taher Elgamal in 1985 [24].

A. Key generation

The key generation process consists of two steps. The first step involves selecting components that can be shared with other system users, while the second step involves computing a unique key pair for a specific user.

B. Parameter generation

- A key length N is chosen.
- A prime number q, with a length of N -bits, is selected.
- A cryptographic hash function H is chosen, with an output length of L bits. If L is greater than N, only the

leftmost bits of the hash output are processed.

- A generator $g < q$ of the multiplicative group of integers z_q^* modulo q is also selected as a component of the scheme.
- These components can be distributed among members of the system.

C. Individual keys

Are produced by utilizing a group of elements. To determine the key pair for each user:

- An arbitrary integer u is chosen from $\{1, \dots, q-2\}$.
- Calculate

$$t = g^u mod q \quad (6)$$

u is the secret key and t is the public key.

D. Signing

To create a message sign,

- A random integer, l, is chosen from a set of numbers, $\{2, \dots, q-2\}$, that are coprime to $q-1$.
- Then, a parameter w is calculated using the formula:

$$w = g^l mod q \quad (7)$$

- After that, the signature value s is estimated using the formula:

$$s = (H(m) - uw)l^{-1} mod(q-1) \quad (8)$$

- If s is not coprime to l, a new random l must be selected.
- The resulting signature is represented by the values (w, s).

E. Verification

- Involves three steps: first, check if $0 < w < q$.
- Second, check if $0 < s < q-1$.
- Third, the signature is valid only if this condition is satisfied.

$$g^{H(m)} \equiv t^w w^s mod q \quad (9)$$

4. Case Study

The objective of this case study is to improve the generation process of the secret signing key by utilizing the randomness inherent in a chaotic map to produce an extended private key sequence. The employed map is called the improved Logistic map and its formula is shown in Equation (10). The secret signing key, represented as u, is generated as a key sequence through the application of this chaotic map. The public verification key is calculated using the formula $t = g^u \mathrm{mod} q$. The procedures for signing and verification will adhere to the guidelines specified in sections 3.1 and 3.2. The steps of the algorithm are detailed below.

$$\boldsymbol{f}(x_n, a, b) = a x_n (\mathbf{1} - x_n) + b(\mathbf{1} + x_n)\boldsymbol{tan}(x_n) \quad (10)$$

$$x_{n+1} = \boldsymbol{f}(x_n, a, b) \times alpha - floor(\boldsymbol{f}(x_n, a, b) \times alpha) \quad (11)$$

Where a, b are the parameters of the map and x_0 is the initial condition.

The procedures of the new algorithm

Input: , g and l.

Output: The private key, , and a signature, s,for each private key.

a. Specify the parameters and initial conditions for the map.
b. Create a sequence of random iterations by applying the equation of the map.
c. Transform every random iteration into a 256-bit integer.
d. Produce a signature,s, for each private key, u.
e. Validate the authenticity of each signature.

4. Results and comparisons

The proposed algorithm in this case study utilizes chaotic maps to generate a more substantial private key sequence of size (2^{256}), enhancing randomness. This enhancement leads to a notable reduction in the time needed for both signature creation and verification.

Table 1 illustrates the repetition of the chaotic map for 100,000 iterations to produce the Schnorr and Elgamal signatures.

Additionally, the output size, the range of selection for q, and the specific value of q for each of the 100,000 iterations are presented in Table 1.

Table (1): Initialization values for the suggested algorithm for 100,000 iterations

Digital Signature Algorithm	Number Of Iterations	Output size	Initial Conditions	Value of q for each iteration	The range of q
Schnorr	100000	256-bit	$\boldsymbol{x}_0$ = 0.123456789, $\boldsymbol{a} \in (0, 10]$ $\boldsymbol{b} \in (0, 10]$ $\boldsymbol{alpha}$ = 12,345	q_1=37838519299144215460477230614169467374818347418841	$[10^{40}, 10^{100}]$
				q_2=45605771518754487477020504444837692942651741288277	
				q_3=81555865417961455191655538681095292674326374977290	
				q_4=17171201836499379579357259559085371009029990789872	
				q_5=54567233715420374473419797772206902871348710817837	
				q_6=25615894402767293233988367708202876924761665189624	
				q_7=64325487424887924303748155712063463755214711939611	
				q_8=84407796470872036023341619503488177284523786196504	
				q_9=41210815292157139293583535885818872697638361538358	
				q_{10}=89498038745373640609979803346960363323048500143068	
				⋮	
				q_{100000}=37687416849708531286150724726705474798359142833926	
Elgamal	100000	256 - bit	$\boldsymbol{x}_0$ = 0.123456789, $\boldsymbol{a} \in (0, 10]$ $b \in (0, 10]$ $alpha$ = 12,345	q_1=94386111575787764224221332618945170297648991 26728	$[10^{40}, 10^{100}]$
				q_2=98447602401025588702203603789523953220624638 646090	
				q_3=12243035621668846397525876022965792820633958 366482	
				q_4=81937176076204255282620201013767517038623233 540281	
				q_5=24849184848000949368322773305371548808145548 480314	
				q_6=31837895476622092686807813375683937723822174 566767	
				q_7=80543671396912181568771482391301192586089653 713619	
				q_8=89736374911650516240559576070356021322723137 497757	
				q_9=66704279985350843095794453263124789785263234 705122	
				q_{10}=93635953299778679035102884090020261424677588783557	
				⋮	
				q_{100000}=78055305029681268861714823608524251794408352873387	

Table 2 presents a portion of the results obtained from the algorithm after 100,000 iterations, using "Hello World" as the input message. The table also includes the corresponding value of q for each symbol.

Table (2): Part of the outcomes of the algorithm with the corresponding value of q for 100,000 messages tests.

Digital Signature Algorithm	Value of q for each iteration	Part of the output Sign With "Hello World" Input Message
Schnorr	q_1=37838519299144215460477230614169467374818347418841	The Sign is 490491433680900083315679314962517659055602855 55062
	q_2=45605771518754487477020504444837692942651741288277	The Sign is 6340627147004310742740016144718746820883791266856
	q_3=81555865417961455191655538681095292674326374977290	The Sign is 3637626485458599629193127734056561078036252256843 4
	q_4=17171201836499379579357259559085371009029990789872	The Sign is 7017060755726336173096235508906202444755655719944 2
	⋮	⋮
	q_{100000}=5456723371542037447341979777220690287134871081783 7	The Sign is 7967526644239137076068984570582664079145487943941 3
Elgamal	q_1=94386111575787764224221332618945170297648991267 28	The Sign is 560739736506652022256755059150132831138265756 7808
	q_2=98447602401025588702203603789523953220624638646090	The Sign is 2968285273302568121090373452463437187570252919864
	q_3=12243035621668846397525876022965792820633958366482	The Sign is 6466814605889207228998662041667669965672032296 66
	q_4=81937176076204255282620201013767517038623233540281	The Sign is 5753770772171483264812908285233423320120139423434
	⋮	⋮
	q_{100000}=24849184848000949368322773305371548808145548480314	The Sign is 4116706447239202510787649824973256825404582141716

Table 3 provides a comprehensive comparison of the results achieved by the algorithms in the case study in the case study against those derived from the traditional methods and frameworks outlined in Reference [25]. The table includes the value of q along with its corresponding signature for each algorithm

Table (3): Comparison of the outcomes obtained from the proposed algorithms and the conventional one for a total of 100,000 message tests.

Algorithm	Value of q	Part of the output Sign With "Hello World " Input Message	Time of signing(s)	Time of verification(s)	Private key space
Traditional Schnorr	q =180093932517718015563512479	The Sign is 1385336809344136	0.00016991869	0.000360910165	2^{160}

	974759603595626 85724938	1926531609259335 5834324938597991 61			
Schnorr in the case study	q_1=378385192991 442154604772306 141694673748183 47418841	The Sign is 4904914336809000 8331567931496251 7659055602855550 62	0.000080486	0.0000402130	2^{256}
	q_2=456057715187 544874770205044 448376929426517 41288277	The Sign is 6340627147004310 7427400161447187 4682088379126685 6			
	⋮	⋮			
	q_{100000}=54567233 715420374473419 797772206902871 348710817837	The Sign is 7967526644239137 0760689845705826 6407914548794394 13			
Traditional Elgamal	q=9570912735511 125118191297110 423014443816399 7194025	The Sign is 2227462216482128 6088056768358453 2607100399932357 0504	0.00034725805187	0.000673826930259 3	2^{160}
Elgamal in the case study	q_1=943861115757 877642242213326 189451702976489 9126728	The Sign is 5607397365066520 2225675505915013 2831138265756780 8	0.00001671199	0.00005969873	2^{256}
	q_2=984476024010 255887022036037 895239532206246 38646090	The Sign is 2968285273302568 1210903734524634 3718757025291986 4			
	⋮	⋮			
	q_{100000}=24849184 848000949368322 773305371548808 145548480314	The Sign is 4116706447239202 5107876498249732 5682540458214171 6			
Ref [25] structure 1	q=3930506341241 022328695670345 5542737154290 4833	-	0.006212	0.006267	
Ref [25] structure 2	q=3930506341241 022328695670345 5542737154290 4833	-	0.007018	0.006651	
Ref [25] structure 3	q=1656029012219 503526789496619 9	-	0.003745	0.003939	
Ref [25] structure 4	q=1656029012219 503526789496619 9	-	0.004991	0.003964	

Additionally, Table 4 presents a comparison of the results from the algorithms proposed in the case study against another set of evaluations involving 100,000 messages, demonstrating that those algorithms achieve the fastest signing and verification times.

Table (4): A comparative analysis of the proposed algorithms against alternative algorithms designed for a 224-bit key length.

Algorithm	Signature time(ms)	Verification time(ms)	Total time(ms)
Schnorr in the case study	0.12905	0.20465	0.3337
Elgamal in the case study	0.13036	0.2187	0.34906
Schnorr Scheme	0.1310	1.4503	1.5813
Elgamal Scheme	0.4946	0.2075	0.7021
Ref [25] structure 1	1.3081	1.4480	2.7561
Ref [25] structure 2	1.3456	1.4634	2.809
Ref [25] structure 3	0.3924	0.2609	0.6533
Ref [25] structure 4	0.5075	0.2538	0.7613
Ref [26]	3,5000	5,2200	40,200
Ref [27]	-	-	4465.38
Ref [28]	-	-	8508.74
Ref [29]	-	-	2344.23
Ref [30]	-	-	1515.03
Ref [31]	-	-	10.31
Ref [32]	-	-	912.19
Ref [33]	-	-	7.29
Ref [34]	-	-	29.570
Ref [35]	0.97	0.97	1.97

In Tables 5 and Table 6, a comparison of Schnorr and Elgamal digital signatures, utilizing the chaotic system and other architectures for a total of 100,000 messages is presented. The data indicates that the innovative algorithms are well-suited for hardware implementation, with low computational complexity.

Table (5): Schnorr Digital signature based on the new chaotic system and other structures for 100,000 messages.

	Length of characters	Signing time (ms)	Verification time (ms)
Test 1	208	0.3291466	0.434756
Test 2	416	0.312684	0.321200
Test 3	624	0.331325	0.431452
Test 4	832	0.389957	0.35785
Test 5	1040	0.3857896	0.41423
Test 6	1248	0.3395874	0.30142
Test 7	1456	0.34123	0.46874
Test 8	1660	0.300012	0.339874
Test 9	1868	0.398574	0.42347
Test 10	2076	0.391235	0.3010255

Table (6): Elgamal Digital signature based on the new chaotic system for 100,000 messages.

	Length of characters	Signing time (ms)	Verification time (ms)
Test1	208	0.33512	0.43475
Test 2	416	0.310124	0.3256889
Test 3	624	0.3000014	0.403254
Test 4	832	0.327854	0.359658
Test 5	1040	0.309347	0.429988
Test 6	1248	0.391257	0.389874
Test 7	1456	0.359987	0.410254
Test 8	1660	0.334785	0.369987
Test 9	1868	0.382557	0.43214
Test 10	2076	0.333254	0.3200124

5. Conclusion

Mathematics plays a crucial role in smart contracts, particularly through cryptographic techniques like digital signatures. These techniques help ensure the security, immutability, and enforceability of contracts. Digital signatures are used to sign contracts, with the Schnorr and Elgamal schemes being among the most commonly utilized in this context. This study demonstrates how integrating chaotic maps with these digital signature algorithms can enhance security by expanding the secret key's key space and introducing unpredictable chaotic behavior. Future research could focus on developing new chaotic systems to strengthen digital signature schemes, improving their performance and security within smart contracts.

References

[1] Zheng, Z., Xie, S., Dai, H. N., Chen, W., Chen, X., Weng, J., & Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems, 105, 475-491.

[2] Ante, L. (2021). Smart contracts on the blockchain–A bibliometric analysis and review. Telematics and Informatics, 57, 101519.

[3] Hewa, T. M., Hu, Y., Liyanage, M., Kanhare, S. S., & Ylianttila, M. (2021). Survey on blockchain-based smart contracts: Technical aspects and future research. IEEE Access, 9, 87643-87662.

[4] John, K., Kogan, L., & Saleh, F. (2023). Smart contracts and decentralized finance. Annual Review of Financial Economics, 15, 523-542.

[5] Kirli, D., Couraud, B., Robu, V., Salgado-Bravo, M., Norbu, S., Andoni, M., ... & Kiprakis, A. (2022). Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. Renewable and Sustainable Energy Reviews, 158, 112013.

[6] Metcalfe, W. (2020). Ethereum, smart contracts, DApps. Blockchain and Crypt Currency, 77.

[7] H.Saeed, Elsisi, M. A., Diab, T. O., El Sobky, W. I., Abdel-Wahed, M. S., & Mahmoud, A. K. (2023, July). Famous Digital Signatures Used In Smart Contracts. In 2023 International Telecommunications Conference (ITC-Egypt) (pp. 649-656). IEEE.

[8] Balcerzak, A. P., Nica, E., Rogalska, E., Poliak, M., Klieštik, T., & Sabie, O. M. (2022). Blockchain technology and smart contracts in decentralized governance systems. Administrative Sciences, 12(3), 96.

[9] Fauziah, Z., Latifah, H., Omar, X., Khoirunisa, A., & Millah, S. (2020). Application of blockchain technology in smart contracts: A systematic literature review. Aptisi Transactions on Technopreneurship (ATT), 2(2), 160-166.

[10] Oliva, G. A., Hassan, A. E., & Jiang, Z. M. (2020). An exploratory study of smart contracts in the Ethereum blockchain platform. Empirical Software Engineering, 25, 1864-1904.

[11] Xu, Y., Chong, H. Y., & Chi, M. (2021). A review of smart contracts applications in various industries: a procurement perspective. Advances in Civil Engineering, 2021, 1-25.

[12]Vora, J., DevMurari, P., Tanwar, S., Tyagi, S., Kumar, N., & Obaidat, M. S. (2018, July). Blind signatures based secured e-healthcare system. In 2018 International conference on computer, information and telecommunication systems (CITS) (pp. 1-5). IEEE.

[13]Fang, W., Chen, W., Zhang, W., Pei, J., Gao, W., & Wang, G. (2020). Digital signature scheme for information non-repudiation in blockchain: a state of the art review. EURASIP Journal on Wireless Communications and Networking, 2020(1), 1-15.

[14]Sobti, R., & Geetha, G. (2012). Cryptographic hash functions: a review. International Journal of Computer Science Issues (IJCSI), 9(2), 461.

[15]Lakshmanan, T., & Madheswaran, M. (2012). A novel secure hash algorithm for public key digital signature schemes. Int. Arab J. Inf. Technol., 9(3), 262-267.

[16]Monrat, A. A., Schelén, O., & Andersson, K. (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access, 7, 117134-117151.

[17]Basha, S. J., Veesam, V. S., Ammannamma, T., Navudu, S., & Subrahmanyam, M. V. V. S. (2021, February). Security Enhancement of Digital Signatures for Blockchain using EdDSA Algorithm. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 274-278). IEEE.

[18]Naik, R. B., & Singh, U. A review on applications of chaotic maps in pseudo-random number generators and encryption. Annals of Data Science. 2024. vol 11(1), pp.25-50.

[19]Wang, Xingyuan; Zhao, Jianfeng. An improved key agreement protocol based on chaos. Commun. Nonlinear Sci. Numer. Simul. 2012. vol 15 (12), pp.4052–4057.

[20]Chen, A., & Chen, Y. Existence of solutions to anti-periodic boundary value problem for

nonlinear fractional differential equations with impulses. Advances in Difference Equations. 2011, 1-17.

[21]Z. A. Abduljabbar et al.. Provably Secure and Fast Color Image Encryption Algorithm Based on S-Boxes and Hyperchaotic Map. in IEEE Access. 2022. vol. 10, pp. 26257-26270, doi: 10.1109/ACCESS.2022.3151174.

[22]Ouannas, A.; Khennaoui, A.A.; Momani, S.; Pham, V.T.; El-Khazali, R. Hidden attractors in a new fractional-order discrete system: Chaos, complexity, entropy, and control. Chin. Phys. B. 2020. vol. 29, pp. 050504.

[23]Khennaoui, A.A.; Ouannas, A.; Boulaaras, S.; Pham, V.T.; Azar, A.T. A fractional map with hidden attractors: Chaos and control. Eur. Phys. J. Spec. Top. 2020, vol. 229, pp.1083–1093.

[24]Hala Saeed, Hossam E. Ahmed, Tamer O. Diab, Hossam L. Zayed, Hany Nasry Zaky, and Wageda I.Elsobky. Evaluation of the Most Suitable Hyperchaotic Map in S-Box Design Used in Image Encryption. International Journal of Multidisciplinary Research and Publications (IJMRAP). 2022. Vol. 5(4), pp. 176-182.

[25]EL-MELIGY, N. E., EL-SOBKY, W. I., MOHRA, A. S., HASSAN, A. Y., & DIAB, T. O. NEW HIDING TECHNIQUE IN DIGITAL SIGNATURE BASED ON ZIGZAG TRANSFORM AND CHAOTIC MAPS. Journal of Jilin University (Engineering and Technology Edition). 2023. vol. 42(9), pp. 1671-5497.

[26]P. Kuppuswamy. A New Efficient Digital Signature Scheme Algorithm based on Block cipher. IOSR J. Comput. Eng. 2012. vol. 7, no. 1, pp.47–52, doi: 10.9790/0661-0714752.

[27]H. Cui, R. H. Deng, J. K. Liu, X. Yi, and Y. Li. Server-Aided attribute-based signature with revocation for resource-constrained Industrial-Internet-of-Things devices. IEEE Trans. Ind. Informat. 2018. vol. 14, no. 8, pp. 3724–3732.

[28]C. Esposito, A. Castiglione, F. Palmieri, and A. D. Santis. Integrity for an event notification within the industrial Internet of Things by using group signatures. IEEE Trans. Ind. Informat. 2018. vol. 14, no. 8, pp. 3669–3678.

[29]L. Shen, J. Ma, X. Liu, F. Wei, and M. Miao. A secure and efficient ID-Based aggregate signature scheme for wireless sensor networks. IEEE Internet Things J. 2017. vol. 4, no. 2, pp. 546–554.

[30]M. A. Mughal, X. Luo, A. Ullah, S. Ullah, and Z. Mahmood. A lightweight digital signature-based security scheme for humancentered Internet of Things. IEEE Access. 2018. vol. 6, pp. 31630–31643.

[31]G. K. Verma, B. B. Singh, N. Kumar, M. S. Obaidat, D. He, and H. Singh. An efficient and provable certificate-based proxy signature scheme for IIoT environment. Inf. Sci. 2020. vol. 518, pp. 142–156.

[32]J. H. Seo. Efficient digital signatures from RSA without random oracles. Inf. Sci. 2020. vol. 512, pp. 471–480.

[33]Meshram, C., Obaidat, M. S., Tembhurne, J. V., Shende, S. W., Kalare, K. W., & Meshram, S. G. A lightweight provably secure digital short-signature technique using extended chaotic maps for human-centered IoT systems. IEEE Systems Journal. 2020. vol. 15(4), pp.5507-5515.

[34]Lakshmanan, T., & Madheswaran, M. A novel secure hash algorithm for public key digital signature schemes. Int. Arab J. Inf. Technol. 2012. vol. 9(3), pp.262-267.

[35]Ullah, S. S., Ullah, I., Khattak, H., Khan, M. A., Adnan, M., Hussain, S., & Khattak, M. A. K. A lightweight identity-based signature scheme for mitigation of content poisoning attack in named data networking with internet of things. IEEE Access. 2020. vol. 8, pp.98910-98928.