ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

DESIGNING NEW TECHNIQUE IN DIGITAL SIGNATURE BASED ON GALOIS FIELD 2^n AND CHAOTIC MAPS

MUSTAFA HUSSEIN[1]*, ABD ELHADY MAHMOUD[2], SARA HAMDY[3], and WAGEDA ALSOBKY[4]

[1, 2, 3] Electrical Engineering, Benha University, Benha, Egypt.
[4] Basic Engineering Science, Benha University, Benha, Egypt.
Email: [1] mostafahosen042@gmail.com(*Corresponding Author), [2] abdoeng78@gmail.com,
[3] sara.hamdy@bhit.bu.edu.eg, [4] wageda.alsobky@bhit.bu.edu.eg

Abstract

Ensuring the utmost security, confidentiality, and integrity of digital communications has become an imperative requirement in today's world. This realization highlights the significance of employing Digital Signature Algorithms (DSA) in various online applications. DSA's true value lies in its ability to deliver secure digital signatures, assuring the verification of digital documents, messages, or transactions. This aspect holds paramount importance in critical domains such as online banking, e-commerce, digital contracts, and government services where safeguarding sensitive information is crucial. DSA encompasses diverse algorithms, including RSA, Elliptic Curve Cryptography (ECC), and Schnorr signatures, each possessing distinct strengths and weaknesses. RSA stands as one of the most prevalent DSA algorithms, although ECC is gaining popularity due to its smaller key size and faster performance. Moreover, Schnorr signatures are gaining attention due to their simplicity and efficiency. This paper introduces a novel Digital Signature algorithm scheme, incorporating robust elements like Hashing, Discrete Logarithm Problems (as seen in Elliptic Curve), and CHAOTIC maps for mapping, thus bolstering secrecy and enhancing security performance. The scheme aims to optimize speed and cost, offering a comparative analysis against other digital signature schemes such as RSA and the original ECDSA.

Keywords: Digital Signature Algorithm (DSA), RON RIVEST, ADI SHAMIR, and LEONARD ADLEMAN (RSA), Elliptic Curve Digital Signature Algorithm (ECDSA), Galois Field (GF), Elliptic Curve Cryptography (ECC), Digital Signature (DS), Elliptic Curve Discrete Logarithm Problem (ECDLP), Discrete Logarithm Problem (DLP), National Institute of Standards and Technology (NIST).

1. INTRODUCTION

Nowadays, the meaning of cryptography involves different concepts like utilizing multimedia and different images of data to be contained in secret messages transmitted and received between different parties especially agent and government, and there are many servers those its main job to store the stolen secret data. That belong to the World Wide Web users, to be encrypted mean the ability to resist hacking tries. And the cryptographic words is including whole of these concepts of encryption, hacking trials stopping and so on, but Cryptography word can be definitely explained as the science of making data which is transmitted and received between two parties is useless to the backdoors attackers, what push scientists to use asymmetric cryptography[24] type which is authentication to secure communication between the requiring parties, from these authentication ways the signature and while signatures are used to act like a contract between two parties to fulfil its constrains. Including proof to prove that signer is the signature's owner having something like an identity proof with the same signature. Signature validity, and non-repudiation and examples are too many for these

The Seybold REPORT
ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

Signature usage in life's daily applications involving Firstly communication over Internet using Emails, as during e-mails, trusting the email provider in factors of privacy and security isn't needed. The mail can be encrypted by signing after using the receiver's public key. By using this the sender can has the ability to know that there was no tampering. The only case for tampering to be happened is that the message wasn't delivered by the e-mail provider. Secondly Code contributions where many codes were written by programmers as open source to help others but some people can take these codes and attributing these codes to themselves as their codes. Whatever the original code implementers might be volunteers or paid for those contributions. The project's maintainers don't have enough time to check each and every contribution. So, they need trusted people. What leads to the need to signing each contribution by its maker. Thirdly Software updates where all of modern devices like Smart TV / Alexa / Fritz Box need updates. As the original manufacturer of these devices will want to assure that change or replacement weren't done in the update. So, a specific private-key belongs to the manufacturer's company will be shared with the device. And when there's update it can't be installed without the signature's checking of the company. Fourthly Cryptocurrencies while to prove the ownership of a bitcoin, asymmetric cryptography was used. At first, someone is become the coin's owner. Then, this owner is defined as the private key's owner, matching to a given public key. Fifthly Digital diplomas as during job's applying, there's a need for a proof of qualifications. Especially starting since remote working through internet, and this proof was written digitally. So, there is a need to DS [22].

And too many of applications as E-Governance, E-Learning, E-Shopping, E-Voting, etc. [23]. It's clear the importance of usage of Digital Signature Algorithm, but still there is a need for knowledge of its improvement historically to have the ability to improve its scheme. So, if a look was taken on Digital Signature Algorithm history it will be found that In 1982 United states government Planned to replace RIVEST and SHAMIR DSA by another Algorithm to prevent defects and provide more of data saving, and in 1991 National Institute of Standards and Technology (NIST) introduced the first version of DSA that was confirmed by the government at 1994 [1,2,3], this change faced a many objections because of difficulty to change most of work that depended on RSA to DSA , what make government stop changing to DSA till it was expired in 2013 however its strength, and still RSA has the same problems of need to too large prime key to achieve required level of security, while many of different applications depends on DSA including what cause need to larger memory, more processing steps and by default more time [4], what lead to thinking about new scheme that has advantages over RSA 's scheme, and overcome RSA 's scheme disadvantages and to be flexible replace it, this scheme based on mathematical problems, point mapping, DLP , it also enhance security performance of DSA 's scheme, and this achievement will be explained in the rest of paper paragraphs.

This paper is structured like the following: section 2 illustrates Digital Signature Algorithm, section 3 illustrates RIVEST SHAMIR and LEONARD ADLEMAN Digital Signature Algorithm, section 4 illustrates Elliptic Curve Digital Signature Algorithm over GF (2^m), section 5 illustrates Main Work, section 6 illustrates Conclusion).

The Seybold Report
ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

2. DIGITAL SIGNATURE ALGORITHM

DSA is an algorithm that depends on ECC over GF(P) field what based on DLP and difficult mathematical operations, it was proposed to overcome problems of RSA and standardized by NIST in 1994, it has advantages over RSA that it needs lower key size to achieve better security level what lead to lower memory need, lower processing steps and lower time and has disadvantages of taking full exponential time to solve ECDLP what will be solved in the new scheme, Also DSA divided into two parts creation and verification of signature[5-8].

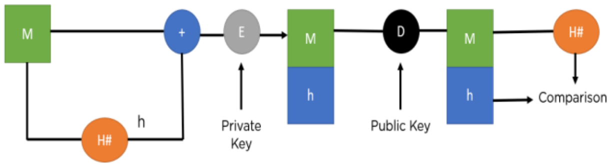


Figure 1: Dsa Scheme

DSA has many strength and weakness points as will be described in the following [15-17]

A. Strength-Points

1. Widely used and well-established algorithm.
2. Proven to be secure and efficient.
3. Generally considered to be a good choice for government and military applications.

B. Weakness-Points

1. Key management can be complex.
2. Strength depends on the size and quality of the prime numbers used.
3. Not as widely used as some other algorithms

Table 1: DSA Key Generation and Verification

Generation	Inputs	$\boldsymbol{k \geq 1024, m \geq 64 \;\&\; m < k,}$ $\boldsymbol{prime\ numbers\ p1, p2\ \ in}$ $\boldsymbol{where\ p1\ has\ size\ of m\ \ bits}$ $\boldsymbol{and\ p2 - 1\ is\ multiple\ of\ p1}.$
	Process	1. $Choose\ the\ message\ hash\ as\ \ sh$ 2. $Choose\ random\ integer\ x$ $private\ key.$ 3. $calculate\ g = sh^{\frac{p2-1}{p1}}\ mod(p2)$ 4. $if\ \ g = 1\ return\ to\ step\ 1.$ 5. $Calculate\ public\ key\ y = \ g^x mod p2.$ 6. $Choose\ random\ integer\ k$ $such\ that\ 0\ <\ k1\ <\ p2.$ 7. $calculate\ r\ =\ (g^{k1}\ mod\ p2)\ mod\ p1.$

The Seybold Report
ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

		8. $calculate\ s = \left(k1^{-1}(sh + xr)\right)modp1.$
	Outputs	$Signature\ is\ mix\ of\ r, s.$
Verification	Inputs	$r, s, sh, p1, p2$
	Process	1. $Calculate\ w\ as\ modular$ 2. $inverse\ of\ s\ regarding\ to\ p1$ 3. $Calculate\ u = sh * w\ mod\ p1.$ 4. $Calculate\ v = r * w\ mod\ p1.$ 5. $Calculate\ result = \left((g^u + y^v)mod\ p2\right) mod\ p1$
	Outputs	$if\ result = r\ then\ signature\ is\ verified.$

Figure 1 and Table 1 explained respectively the methodology of DSA and the steps of key generation and verification algorithms steps. Which is the base methodology for Digital signature schemes, in the next paragraph, one from most famous and most used schemes will be explained to understand its methodology and how it works to compare it with new modified scheme.

3. RIVEST SHAMIR AND LEONARD ADLEMAN DIGITAL SIGNATURE ALGORITHM

RSA considered as one of the most used algorithms in data securing during transmitting and receiving, it was invented by RON RIVEST , ADI SHAMIR , and LEONARD ADLEMAN after a lot of attempts in April 1977, and released in 1997 as a result to its secrecy, it's a one way algorithm not two ways as DSA , also it's based on large number factorizing problem what cause needs to the large size keys to achieve acceptable level of security, that was the reason to the need to supercomputing for large data and large keys [9,10], it's processed by keeping secret of the prime numbers and encrypting the messages by any user but only who has the prime number can decrypt the messages, and that is one from its weakness points as with the same usage of super computer for trial and error, these numbers can be discovered by the attackers. RSA has many strength and weakness points as will be described in the following [15, 16, 20, and 21].

A. Strength-points

1. Widely used and well-established algorithm.
2. Relatively easy to implement.
3. Proven to be secure if the key size is large enough.

B. Weakness-Points

1. Vulnerable to certain types of attacks, such as side channel attacks and chosen cipher text attacks.
2. Key management can be complex.
3. Can be slower than some other algorithms.

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

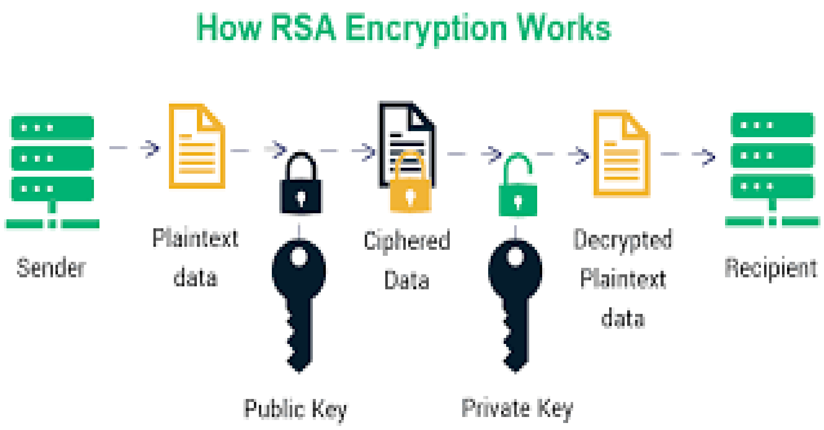


Figure 2: Rsa Scheme

Table 2: Rsa Digital Signature Algorithm

Generation	Inputs	**large prime numbers p1**
	Process	1. $Compute\ p_1 * p_2\ and\ hash\ the\ message$ ➔ n, H 2. $Compute\ private - key\ \ not\ factor\ of\ (p_1 - 1) * (p_2 - 1)$➔ $e.$ 3. $Compute\ private - key\ e^{-1}\ \%n.$➔ $d.$
	Outputs	$Signature\ \ H^d\%\ n$➔ S
Verification	Inputs	S, H,n.
	Process	Compute $s^e\%\ n$ ➔ $H1$
	Outputs	$If\ H\ ==\ H1, So\ the\ signature\ is$

Figure 2 and Table 2 explained the methodology and steps of RSA Scheme while being used in Digital Signature.

Where RSA is the most used scheme in Digital Signature Algorithms, then in the next paragraph the main scheme that will be modified to the new scheme will be explained, including its importance differences between it and RSA, its strength and weakness points.

4. ELLIPTIC CURVE OVER GF (2^m) DIGITAL SIGNATURE ALGORITHM

Elliptic curve cryptography is one of private-key cryptography and symmetric schemes algorithms, it's based on mathematical problems and finite fields, it was invented by NEAL KOBLITZ and Victor S. Miller in 1985, and started in wide spreading in 2004 [11-13], its strength being in the difficulty to solve the mathematical equation with many coefficients, it

The Seybold REPORT

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

doesn't need large keys with comparing to RSA to get equivalent level of security, but it takes full exponential time to solve ECDLP due to its dependability on large prime numbers when it's working over GF(P), but when it's working over GF(2^m) it depending on binary or polynomial operation to be in the same cycle of the finite field what leads to speeding the processing of the algorithm as machines languages are binary, what lead to faster processing [14,15].

Table 3: Ecdsa over Gf (2^m)

Generation	Inputs	order and base point ➔ O, BP.
	Process	1. Hashing the message ➔ H. 2. Choosing two random integers in limits of [1, O] one Of them is private-key ➔ K, private-key as PV. 3. Computing PV * BP Over GF (2^m) ➔ private-key as PK. 4. Computing K * BP ➔ K.X, K.Y. 5. R = K.X mod O. 6. Computing Modular Inverse of K* (H + PV*R) mod O ➔ S. 7. If R or S equal to zero return and reselect K.
	Outputs	Signature is combined of R, S.
Verification	Inputs	R, S, BP, H, PK, O.
	Process	1. If R, S negatives or non-integers ➔ signature isn't valid. 2. Computing Modular Inverse of S ➔ T. 3. Computing H*T mod O, R *T mod O ➔ L, Q. 4. Computing L* BP + Q * PV depending on addition and multiplication rules of GF(2^m) ➔ X, Y.
	Outputs	If X mod O not equals to R ➔ Signature isn't valid.

ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

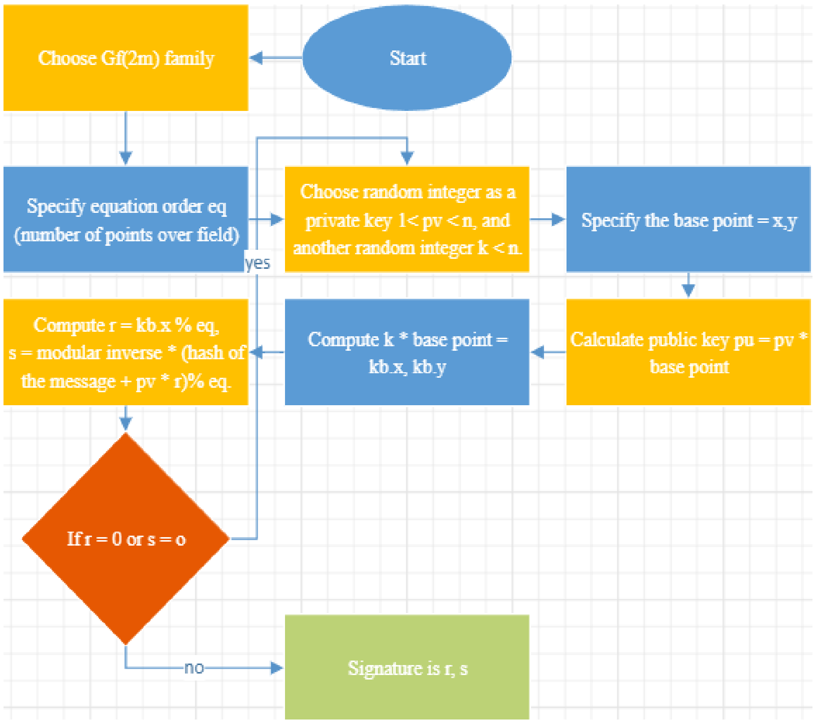


Figure 3: Key Generation Flowchart

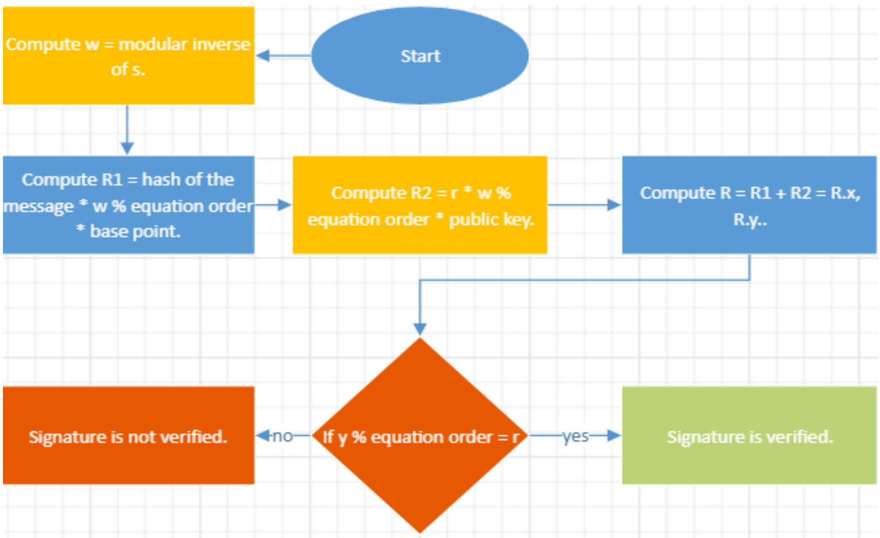


Figure 4: Key Verification Flowchart

Table 3, Figure 3 and Figure 4, explain the flow of steps of ECDSA over GF (2^m) including key generation and verification steps.

ECDSA has many strength and weakness points as will be described in the following [25, 26].

ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

Strength-points

1. More secure and efficient than DSA.
2. Widely used in modern applications.
3. Good choice for resource-constrained environments.

Weakness-Points

1. Requires careful selection of elliptic curves.
2. Security depends on the quality of the random number generator used.
3. Not as widely used as some other algorithms.

A. Comparison between ECDSA & RSA

From the explained paragraphs it can be deduced the difference between DSA that depends on ECC and the one that depends on RSA, as RSA need large size keys that leaded to need to more time and larger memory and supercomputing while securing large data, on the contrast ECC that uses lower keys what leaded to faster execution and need to lower memory than RSA [4, 13-16].

Table 4: Nist Recommended Security Bit Level Security

Bit Level	"RSA"	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

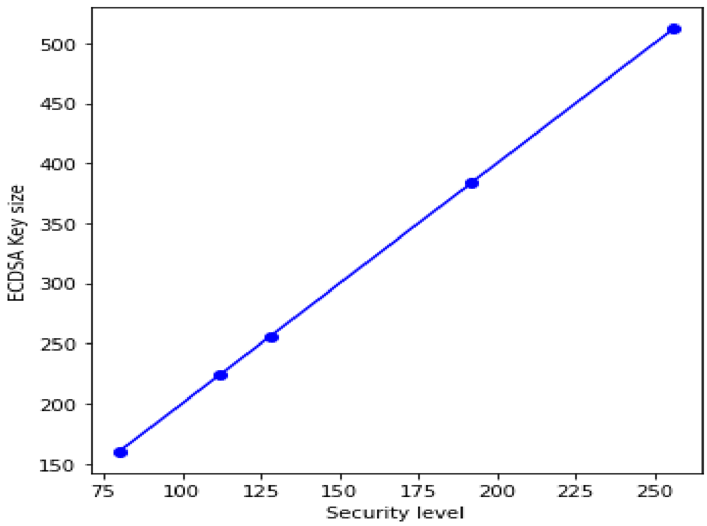


Figure 5: Ecdsa Key Size with Security Bit Level

ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

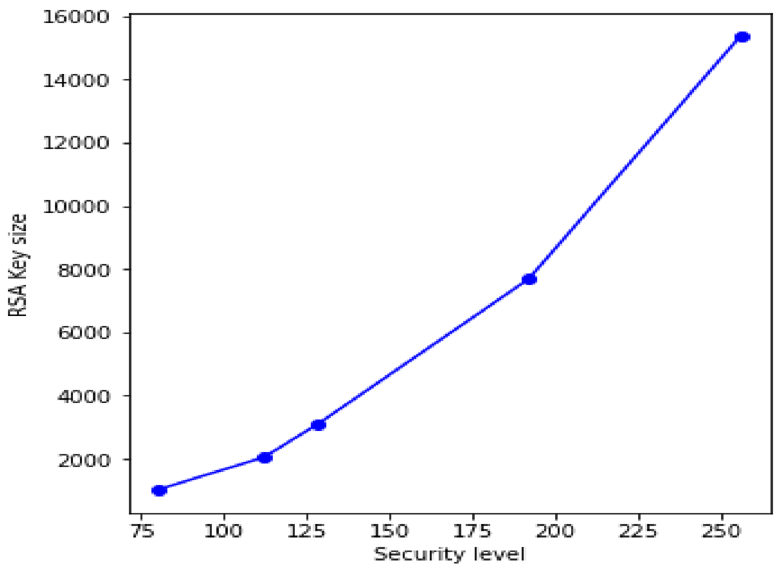


Figure 6: Rsa Key Size with Security Bit Level

Table 4 [16] , Figures 5 and 6 explained the dependability of both RSA and ECDSA security performance on key size which shows that how RSA needs very big keys to achieve good security performance which needs too much resources.

5. CONTRIBUTION

This paper Exploits the advantages of ECC at GF(2^m) and modified the algorithm by using CHAOTIC mappings to increase complexity of Digital Signature Algorithm and at the same time not increasing the size of the key needed in ECDSA so saving time due usage of ECDSA through GF(2^m), and enhancing security performance by using CHAOTIC maps as it will be explained in this paragraph where CHAOTIC maps are group of functions that map points between domains and used to increase equations complexity using different techniques like circular parameters, exponentiation, discrete and continuous timing, this was achieved through applying ECDSA steps these were explained in the previous paragraphs with changing the default equation to be dependable on GF(2^m) instead of Galois Prime Field what made the points cycle wider and faster than GF(P) on software processes using Parallel programming through Message Passing Interface and faster on hardware processes because of dependability on binary operations, and then a new concept was used in the new algorithm which is Message Passing Interface which made the new Algorithm even faster than ECDSA over GF(P),then CHAOTIC mapping was applied to transfer the points from one plan to another plan, so any attacker needs to change whole attacking protocol to has the ability to keep trying to break of system security, then with taking in consideration the results for signature verification through very complex mathematical equation, what make success in system security breaking mostly impossible, during the try to achieve this target there were many of problems including how to implement, how to compare, how to know it's the best solution for these problems, a python language used to implement coding of the so difficult implemented algorithm because python

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

uses interpreter which compile code more faster, and to compare results with other algorithms there was a need to implement each algorithm by python too to compare through the same environment including CPU cores and compiler, then it was tried to be attacked through the American standard steps of attack types which include try to forge message or hash or even finding one point through the field which all were failed because of the strength of the algorithm and perfect implementation.

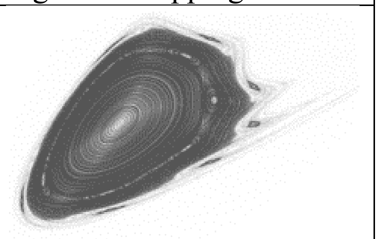
A. CHAOTIC maps

In this paragraph the second methodology after the methodology of Elliptic Curves over Gf (2^m) that the new scheme depended on will be explained where CHAOTIC Maps is the way of mapping or translating of a point from one domain to another domain which by applying in the new algorithm will increase the complexity of the algorithm as for searching for the translated point, there is a need to know in which domain that point is located.

Next part of paragraph will explain CHAOTIC mapping which also it has many types for example:

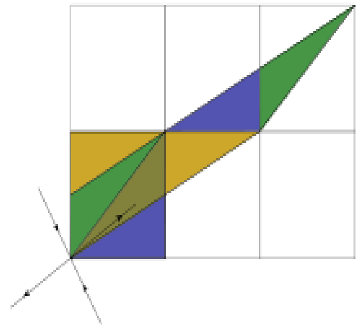
1. BOGANDOV maps

Table 5: BOGANDOV Mapping

Parameters	Equation	Figure of mapping
€ = 0 μ = 0 $k = 1.2$	$x_{n+1} = x_n + y_{n+1}$ $y_{n+1} = y_n + \epsilon y_n + kx_n(x_n - 1) + \mu x_n y_n$	

2. ARNOLD CAT maps

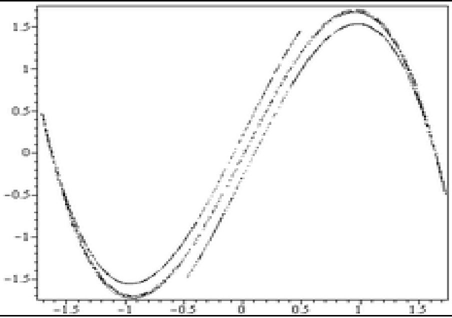
Table 6: ARNOLD CAT Mapping

Parameters	Equation	Figure of mapping
€ = 0 μ = 0 $k = 1.2$	$x_{n+1} = 2 * x_n + y_n$ $y_{n+1} = y_n + y_n$	

ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

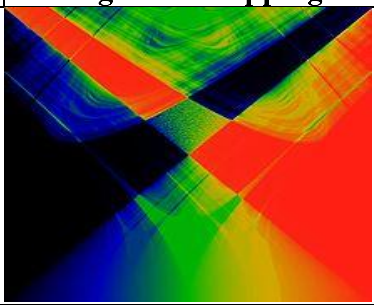
3. DUFFING maps

Table 7: DUFFING Mapping

Parameters	Equation	Figure of mapping
a = 2.75 b = 0.2	$x_{n+1} = y_n$ $y_{n+1} = -b * x_n + a * y_n - y_n^3$	

4. CIRCULAR maps

Table 8: Circular Mapping

Parameters	Equation	Figure of mapping
$\omega = 0.333$ $K = 4\pi$	$x_{n+1} = x_n + \omega + \frac{k}{2\pi}\sin(2 * \pi * x_n)$	

5. HENON maps

Table 9: HENON Mapping

Parameters	Equation	Figure of mapping
$A = 1.4$ $B = 0.3$	$x_{n+1} = 1 - A * x_n^2 + y_n$ $y_{n+1} = B * x_n$	

From Table 4 to table 8 explained little types of CHAOTIC maps where parameters represented constants, equations represented the equation of mapping and the form explained the domain shape where the equation of ECDSA over GF (2^m) will be mapped [15-21].

ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

B. Parallel Programming Using Message Passing Interface

In this paragraph the third methodology that the new scheme depended on will be explained where Message Passing Interface is used to help in communicating and synchronizing, the compiling of threads or processes between logical cores by of the CPU by provide possibility of how each process communicate with each other to run your threads faster, and that exactly what was used in this algorithm applying by assigning each part and each function to a separate core to run code in parallel.

A Dell Laptob with ram 16 Gb, and processor of 2.3 GHZ was used for the experiments.

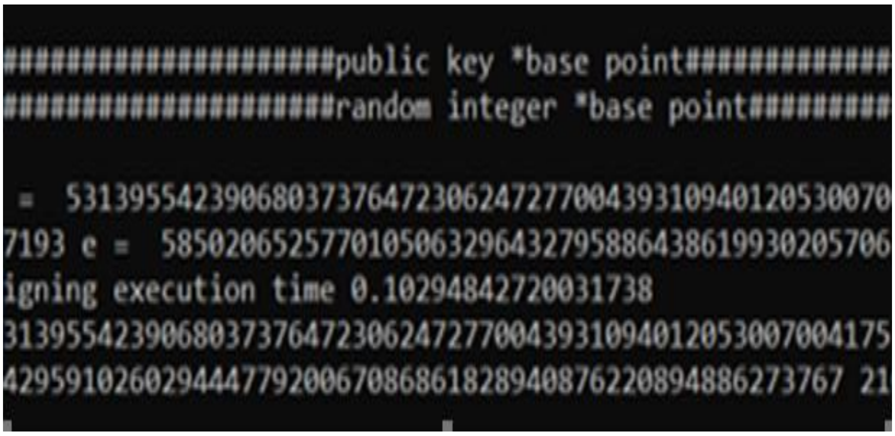


Figure 7: Screenshot for Coderunning after Usage of Message Passing Interface in Signing Algorithm

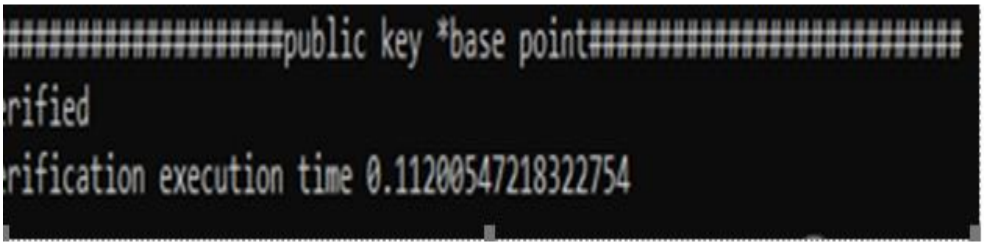


Figure 8*:* Screenshot for Code Running After Usage of Message Passing Interface In Verifying Algorithm

Figures 7 and 8 showed a screenshots for a real experiment applied.

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

Table 10: New Scheme's Flow of Steps

Generation	Inputs	order and base point ➔ O, BP.
	Process	1. Hashing the message ➔ H. 2. Choosing two random integers in limits of [1, O] One of them is private-key ➔ K, private-key as PV. 3. Computing PV * BP over GF (2^m) ➔ private-key as PK. 4. Computing K * BP ➔ K.X, K.Y. 5. Applying CHAOTIC map (HENON) on K.X, K.Y ➔ C.X, C.Y. 6. Computing round (C.Y) mod O ➔ R. 7. Computing Modular Inverse of K* (H + PV*R) Mod O ➔ S. 8. If R or S equal to zero return and reselect K.
	Outputs	Signature is combined of R, S.
Verification	Inputs	R, S, BP, H, PK, O.
	Process	1. If R, S negatives or non-integers ➔ signature isn't valid. 2. Computing Modular Inverse of S ➔ T. 3. Computing H*T mod O, R *T mod O ➔ L, Q. 4. Computing L* BP + Q * PV depending on addition and Multiplication rules of GF(2^m) ➔ X, Y. 5. Applying same CHAOTIC mapping on X, Y ➔ C.X, C.Y. 6. Computing round (C.Y) mod O ➔ Result.
	Outputs	If Result equals to R ➔ Signature is valid.

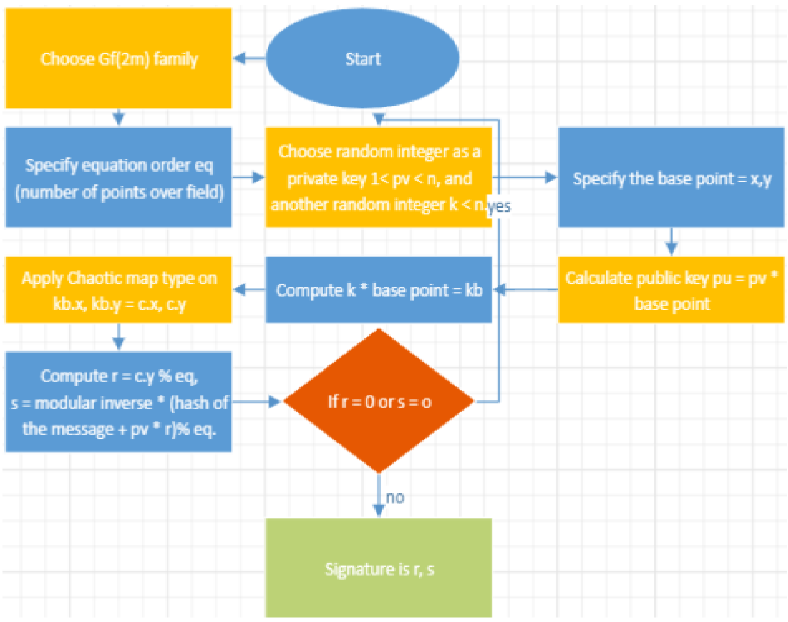


Figure 9: New Scheme's Key Generation Flowchart

ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

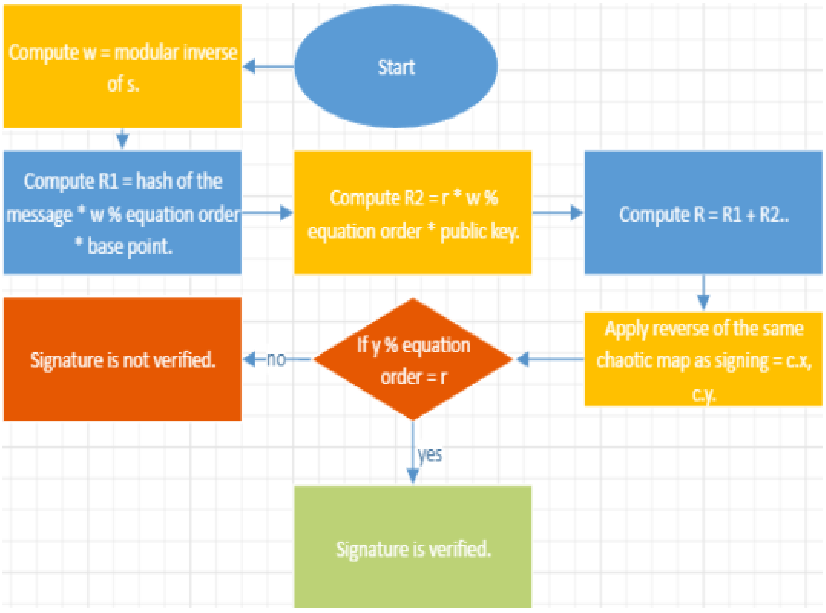
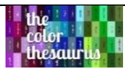


Figure 10: New Scheme's Key Verification Flowchart

Table 9, Figure 7 and Figure 8, explain the flow of steps of the new scheme including key generation and verification steps.

In the next table a result of the applied experiment to test the new scheme will be shown.

Table 11: Example for New Scheme Experiments

CHAOTIC type| Message-Total timing			
Inputs	n = 156927543384667019095894735580335045883120559545163053 3029 k = 1542725565216523985789236956265265265235675811949404040041 d = 1275552191113212300012030439187146164646146646466749494799 x = 0x36B3DAF8A23206F9C4F299D7B21A9C369137F2C84AE1AA0D a = 0x2866537B676752636A68F56554E12640276B649EF7526267 b = 0x64210519E59C80E70FA7E9AB72243049FEB8DEECC146B9B1 y = 0x765BE73433B3F95E332932E70EA245CA2418EA0EF98018FB		
Device	Dell, ram 16 Gb, CPU 2.3 GHZ		

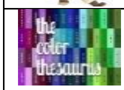
The Seybold Report
ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

ARNOLD CAT	0.2 s	0.2 s	0.2 s
Signature	r = 531395542390680373764723062472770043931094012053007004175 s = 52543198651898175299643170144161489921777490920946967 7193	r = 531395542390680373764723062472770043931094012053007004175 s = 121792156395451980977794798554290616551625800560726105386	r = 531395542390680373764723062472770043931094012053007004175 s = 867916513449366549461244070331995700072314597891317663873
CIRCULAR	0.2 s	0.2 s	0.2 s
Signature	r = 105033548811867532884676815334626076645083083562910377 5744 s = 56796694995308587792369094090367573039231793796464197 7970	r = 1050335488118675328846768153346260766450830835629103775744 s = 16432711982955610590505403801635144772616882931589840 6163	r = 1050335488118675328846768153346260766450830835629103775744 s = 9104514768834706743885033097940565312468576266464899 64650
HENON	0.2 s	0.2 s	0.2 s
Signature	r = 4969409451034625000479543890654190373433130981890289 82784 s = 5624380274797247208363273399945311299095625390681199 70610	r = 49694094510346250004795438906541903734331309818902898278 4 s = 158798197356194948817690437107206847243413430419376398803	r = 4969409451034625000479543890654190373433130981890289 82784 s = 904922554410109517301139708884911930764102227749967957290

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

Table 12: Comparison between Old Scheme of Key Size 192 Bit and New Scheme of 239 Bi

	ECDSA		The new scheme	
Parameters	n = 6277101735386680 763835789423176059 01376719477318284228408 1 k = 61405070670650010630650655656674055600061615565656 65656654 d = 5795647470673399383110260604924628771599304 46474 p = 6277101735386680763835789423207666416083908 700390324961279 x = 0x188DA80EB03090F67 CBF20EB43A18800F4FF0A FD82FF1012 a = 0xFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFEFFFFFF FFFFFFFFFC b = 0x64210519E59C80E70F A7E9AB72243049FEB8DEE CC146B9B1 y = 0x7192b95ffc8da78631011ed 6b24cdd573f977a11e794811		fhex= 0x800000000000000000000000000000 0000000000000000001000000001 n= 2208558830972980411979121875928648145578869937767132309367150412 07411783 k= 1712787255652165239672857892369562652652652356758119494040400416 70216363 d= 1456427555219115346513212300075341203043918714616464614664646674 94947990 a= 0x32010857077C5431123A46B808906 756F543423E8D27877578125778AC7 6 b= 0x790408F2EEDAF392B012EDEFB33 92F30F4327C0CA3F31FC383C422AA 8C16 x= 0x5894609CCECF9A92533F630DE71 3A958E96C97CCB8F5ABB5A688A23 8DEED y= 0x6DC2D9D0C94EBFB7D526BA6A6 1764175B99CB6011E2047F9F067293F 57F5	
Key size	192		239	
Execution time	Generat-ion	Verif-ication	Generat-ion	Verification
	0.15s	0.3s	0.1s	0.1s
	0.13s	0.3s	0.1s	0.1s
	0.13s	0.3s	0.1s	0.1s
Signature for	r= 334240353640598172939348833469460041559688182686935 1677613,		r= 452539588512284801421694235752731117665400284464849591780157974 21359928	

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

	s= 335691480652603752681250711538747207494196450687813 0591120	s= 7615506706188585469254970598203036598566265054549839825082199592507920
Signature for 	r= 334240353640598172939348833469460041559688182686935 1677613, s= 179194930724088500625401662245719959182150729477997 5390487	r= 4525395885122848014216942357527311176654002844648495917801579742 1359928 s= 1818316490090903811901160970950974953469349736898665484311920662 47966837
Signature for 	r= 334240353640598172939348833469460041559688182686935 1677613, s= 289032682010241430453871769243135495327700523673837 2811387	r= 4525395885122848014216942357527311176654002844648495917801579742 1359928 s= 1728602155118779336568925600802574939691220465108947737355307877 67717906

Tables 10, 11 explain the advantages of the new scheme GFECDSA over ECDSA while the new scheme is mostly faster than ECDSA while having bigger key what means stronger security performance due to higher complexity.

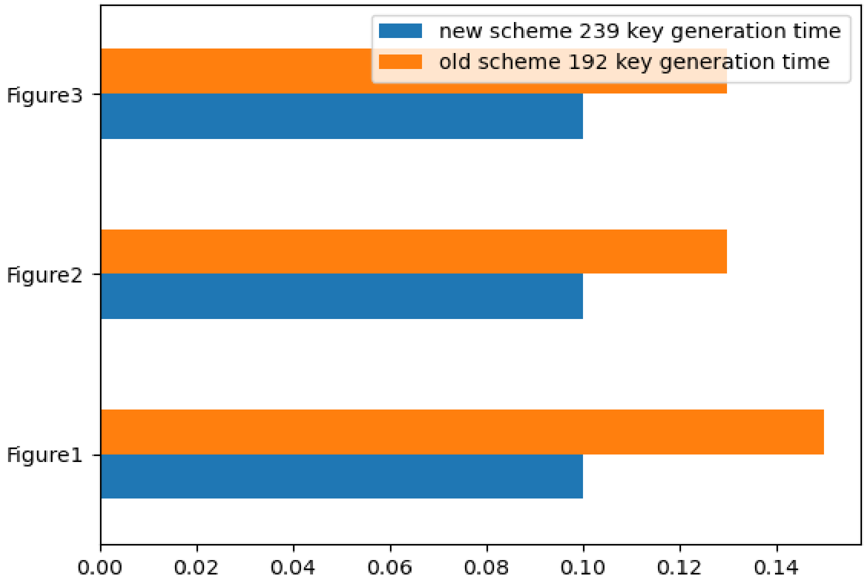


Figure 11: Comparison between Old and New Schemes Key Generation Time

The Seybold Report
ISSN 1533-9211
DOI: 10.5281/zenodo.8344314

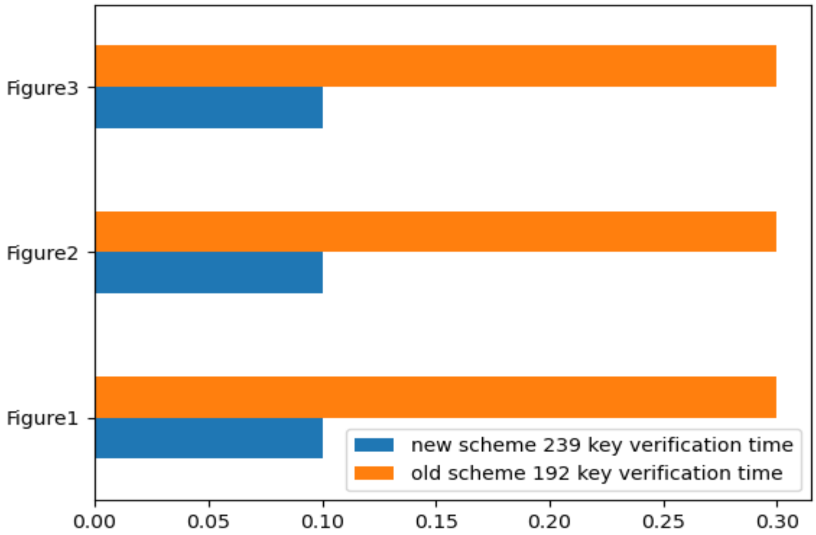


Figure 12: Comparison between Old and New Schemes Key Verification Time

Figure 11 & 12 explained the speed up that achieved from the original scheme to the new modified scheme despite the difference in the key size where the new scheme works on 239 bits while the original one works on 139 bits.

CONCLUSION

This paper managed to create new scheme based on advantages of speed of Elliptic Curve Digital Signature Algorithm over Galois Field 2^m which depends on binary field, and complexity of Elliptic Curve Discrete Logarithm Problem plus modulus features beside CHAOTIC maps to introduce secure scheme with the better timing than ECDSA using Python compiler and Message Passing Interface for Distributed systems, and the same key size so it's faster than RSA and need less processing and so lower cost, and these results were assured with software practical experiments, what will lead to big jump in the Digital Signature Algorithm improvement, and makes the road smooth in front the future research's work to increase the algorithm complexity and decrease the time of Algorithm execution via software to increase the scheme perfectness, a perfect implementation to a new scheme of ECDSA over GF(2^m) where the scheme was inherited from DSA scheme but the implementation was completely different as it worked to optimize the compilation and debugging of DSA code beside the difference and complexity of calculation the Modular Inverse on ECDSA over GF(2^m) through the binary field where there was no any sources describe or explain how to implement it clearly, so it took a lot of trials and errors, to lead to these accurate results which have these mentioned advantages which will help to try these new scheme in the applications of Communication, Code contribution, Online diplomas certificates, Cryptocurrencies, software updates and other many applications these have no finite and we need in our daily life.

The Seybold REPORT

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

References

1) Ryan, Keegan. "Hardware-backed heist: Extracting ECDSA keys from qualcomm's trustzone." Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019.

2) Jack Doerner; Yashvanth Kondi; Eysa Lee; Abhi Shelat et al. Fast threshold ECDSA with honest majority. Journal of Computer Security, 2022, 30.1: 167-196.

3) Lily CHEN; D. MOODY; LIU, Y. K. NIST post-quantum cryptography standardization. *Transition*, 2017, 800: 131A.

4) Fatma Mallouli; Aya Hellal; Nahla Sharief Saeed; Fatimah Abdulraheem Alzahrani, et al. A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms. In: 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom). IEEE, 2019. p. 173-176.

5) Huili Wang, et al. "Dynamic threshold ECDSA signature and application to asset custody in blockchain." Journal of Information Security and Applications 61 (2021): 102805.

6) Surender KUMAR; Vikram SINGH. A review of digital signature and hash function-based approach for secure routing in VANET. In: *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021. p. 1301-1305.

7) Erdem Alkim, et al. "The lattice-based digital signature scheme qTESLA." International Conference on Applied Cryptography and Network Security. Cham: Springer International Publishing, 2020.

8) Jean-Philippe Aumasson, Adrian Hamelink, and Omer Shlomovits. "A survey of ECDSA threshold signing." Cryptology ePrint Archive (2020).

9) Khalid EL MAKKAOUI; Abderrahim Beni-Hssane; Abdellah Ezzati; AnasEl-Ansari et al. Fast cloud- RSA scheme for promoting data confidentiality in the cloud computing. Procedia computer science, 2017, 113: 33-40.

10) Jianbing Ni, et al. "Identity-based provable data possession from RSA assumption for secure cloud storage." IEEE Transactions on Dependable and Secure Computing 19.3 (2020): 1753-1769.

11) Neal. KOBLITZ elliptical curve cryptosystems. Mathematics of computation 48.177 (1987): 203-209,204,205.

12) Lara-Nino, Carlos Andres, Arturo Diaz-Perez, and Miguel Morales-Sandoval. "Elliptic curve lightweight cryptography: A survey." IEEE Access 6 (2018): 72514-72550.

13) ANSI, X9. 62: private-key cryptography for the financial services industry: the elliptical curve Digital Signature Algorithm (ECDSA). Am. Nat'l Standards Inst (1999).

14) Taechan Kim, and Jinhyuck Jeong. "Extended tower number field sieve with application to finite fields of arbitrary composite extension degree." IACR International Workshop on Public Key Cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.

15) Nicolas Sklavos. Book Review: W. Stallings. Cryptography and Network Security: ISBN: 13: 978-0133354690. (2014): 49-50, p308, p431.

16) MAHTO; Dindayal Kumar Dilip YADAV. RSA and ECC: a comparative analysis. International journal of applied engineering research, 2017, 12.19: 9053-9061.

17) Wageda El Sobky, Sara Hamdy, and Mustafa Hussien Mohamed. "Elliptic curve digital signature algorithm challenges and development stages." Int. J. Innov. Technol. Exploring Eng. 10.10 (2021): 121-128.

18) Hany Nasry, Azhaar A. Abdallah, Alaa K. Farhan, Hossam E. Ahmed and Wageda I.El Sobky. Multi

The Seybold REPORT

ISSN 1533-9211

DOI: 10.5281/zenodo.8344314

CHAOTIC System to Generate Novel S-Box for Image Encryption. In: Journal of Physics: Conference Series. IOP Publishing, 2022. p. 012007.

19) Ashraf Shawky, Hend Ali, Wageda AlSobky, Tamer Omar. Efficient image encryption based on new substitution box using DNA coding and bent function. IEEE Access, 2022, 10: 66409-66429.

20) Wageda ALSOBKY; Hala SAEED; Ali N. ELWAKEIL. Different Types of Attacks on Block Ciphers. Int. J. Recent Technol. Eng., 2020, 9.3: 28-31.

21) Nada E. EL-MELIGY, Wageda AlSobky. A Novel Dynamic Mathematical Model Applied in Hash Function Based on DNA Algorithm and CHAOTIC Maps. Mathematics, 2022, 10.8: 1333.

22) Weidong Fang, et al. "Digital signature scheme for information non-repudiation in blockchain: a state-of-the-art review." EURASIP Journal on Wireless Communications and Networking 2020.1 (2020): 1-15.

23) Fan Ding, Yihong Long, and Peili Wu. "Study on secret sharing for SM2 digital signature and its application." 2018 14th International Conference on Computational Intelligence and Security (CIS). IEEE, 2018.

24) Gençoğlu, Muharrem Tuncay. Importance of cryptography in information security. IOSR J. Comput. Eng 21.1 (2019): 65-68.

25) A. Langley, "Edwards-curve Digital Signature Algorithm (EdDSA)," RFC 8032, 2017.

26) A. Langley, "Elliptic Curve Digital Signature Algorithm (ECDSA)," RFC 8422, 2018.