Received: 12 March 2024 | Revised: 24 May 2024 | Accepted: 31 May 2024 | Published online: 10 June 2024

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2024, Vol. 00(00) 1–11
DOI: 10.47852/bonviewJCCE42022816

BON VIEW PUBLISHING

Design and Implementation of Discrete Field Arithmetic-Based Cylindrical Coil-Driven Crypto Framework for Cloud Data

Smitha Sasi[1], Srividya Bharadwaj Venkata Subbu[1], Premkumar Manoharan[2,*], Anju Vijayakumar Kulkarni[1] and Laith Abualigah[3,4]

[1]Department of Electronics and Telecommunication Engineering, Dayananda Sagar College of Engineering, India

[2]Department of Electrical and Electronics Engineering, Dayananda Sagar College of Engineering, India

[3]MEU Research Unit, Middle East University, Jordan

[4]Applied Science Research Center, Applied Science Private University, Jordan

Abstract: The ability to interpret and create code is known as cryptography and has been used to exchange information between peer parties securely. An encryption algorithm is a type of network security model that consists of designing and putting into practice cryptographic algorithms and the supporting framework to help secure data. This cryptosystem is intended for use with cloud data. Cloud security, usually called cloud computing security, is the technique of securing infrastructure, applications, and data stored in the cloud from cyber threats and attacks. This study proposes a new encryption method based on the (x, y) points generated by circles to protect data in the cloud. Using a map table, a new mapping technique is shown in this study to transfer a plain text value to a point on a predetermined circle over the finite field $GF(p^m)$. This mapping technique produces a high dispersion of various spots for recurrent intensity values while being highly quick, simple to use, and with low entropy for plain text input. The implementation of the encryption and decryption processes is extensively studied and analyzed. Security analysis is done after encryption is implemented to determine how resilient the suggested technique is against statistical threats. The results shown in this study prove the superiority of the proposed algorithm.

Keywords: cloud, cryptography, decryption, encryption, finite field

1. Introduction

It has been highlighted that cloud technology is employed in various architectures, services that combine other technologies, and software design methodologies [1]. Platform as a Service, Software as a Service, and Infrastructure as a Service are examples of cloud service models. Four cloud platform deployment types are necessary for public, private, community, and hybrid architecture solutions. Compared to conventional online computing or storage methods, cloud computing has advantages in terms of flexibility, accessibility, and capacity [2]. Nevertheless, various safety challenges, including (i) privacy and security concerns with cloud service providers and (ii) customer-related security issues, are connected to computational clouds. Intense computation on client devices with limited resources has been driven by and made possible by the astonishing growth of computational clouds. Smart mobiles can deliver data-heavy and computationally heavy apps primarily by exploiting the demand service paradigm of distant data centers. Yet, because of increasing concerns over data privacy and security, transferring private and sensitive data to faraway data centers is difficult. Several sorts of attacks on the robustness of the symmetric and public key cryptosystems have been put forth in the literature, for example, various fault analyses that attack and introduce flaws into the two structures to obtain the secret information [3].

This paper contributes to the security framework's design by creating a new encryption/decryption system based on a cylindrical coil. Additionally, it defines the essential aspects of the security framework used by the cloud computing industry. It would be desirable for those virtual servers and cloud service providers with equivalent security implementation needs. The intelligent algorithm in the framework enables faster computation with less bandwidth utilization, power consumption, and network delay. The system uses symmetrical encryption to enable trustworthy gateways and to give users trust. The major components of the proposed architecture include improved security and customer data privacy [4].

This paper contributes to the security framework's design by creating a novel encryption/decryption system based on a cylindrical coil. Additionally, it defines the essential aspects of the security framework used by the cloud computing industry. It would be desirable for those virtual servers and cloud service providers

*Corresponding author: Premkumar Manoharan, Department of Electrical and Electronics Engineering, Dayananda Sagar College of Engineering, India. Email: drpremkumar-eee@dayanandasagar.edu

with equivalent security implementation needs. The intelligent algorithm in the framework enables faster computation with less bandwidth utilization, power consumption, and network delay. The system uses symmetrical encryption to enable trustworthy gateways and to give users trust. The major components of the proposed architecture include improved security and customer data privacy.

The remainder of this paper is organized as follows: The literature review is described in Section 2. The functionality of the framework is described in Section 3. The testing surroundings are also covered in Section 3. The performance outcomes of both the current and suggested frameworks are presented in Section 4. The conclusion of this paper is described in Section 5.

2. Literature Review

A research initiative launched by the International Business Machines (IBM) Company in the late 1960s produced the cipher known as LUCIFER, which later became the basis for the Data Encryption Standard (DES). The National Bureau of Standards requested a new national encryption standard, and the modified version of LUCIFER was presented as a potential solution. Finally, the DES was adopted in 1977. The Feistel block cipher is the foundation of DES. Horst Feistel, an IBM cryptography specialist, created this block cipher in the early 1970s. It comprises several rounds, including exclusive OR operations, bit shuffles, and nonlinear substitutions (S-boxes). A plain text communication is organized into the 64-bit blocks needed for input once it has been received to be encrypted. The last block is padded if the message's bit count cannot be evenly divided by 64. For the full 64-bit block of data, DES performs an initial permutation. Next, it is divided into two 32-bit sub-blocks, L_i and R_i, which are processed through 16 rounds (the subscript i in L_i and R_i indicates the current round). The effects of increasing the number of rounds, which are all equivalent, are dual: the algorithms' security is raised while their temporal efficiency decreases. There is no doubt that these are two opposing results; hence, a compromise is required. The choice of 16 for the DES key size was likely made to ensure that there would be no correlation between the plain text or key and the cipher text. The 32-bit L_i and R_i output values are switched after the 16th round to produce the pre-output. A function that is the precise inverse of the initial permutation is used to permute this [R16, L16] concatenation. The 64-bit cipher text results from this last permutation.

Advanced Encryption Standard (AES) is a symmetric block cipher technique that supports data blocks larger than 128-bits. The technique uses a 128-bit input block of plain text and outputs 128-bits of cipher text (encrypted) data [5]. The United States has made AES the national DES. SO has accepted AES as a global DES. AES has endured the test of time and has proven incredibly resilient to attack [6]. AES can be used in a variety of contexts, including Cipher-Block Chaining, Electronic Code Books (ECB), Output FeedBack, CounTeR, and Cipher FeedBack. These techniques all succeed in converting plain text data into cipher text data. Each of these operating modes also contrasts with various security strengths. Since the decryption process also depends on the block connections, the chaining and feedback modes produce linkages from one cryptographic block to another, which suggests that if a block is broken or lost, decryption is impacted. Each cryptographic block is independently enciphered and decoded in the ECB mode. The encipherment or decipherment of a block is ultimately completely free of other blocks. Similar towards how counter mode does not use any linkage between blocks, cryptographic operations can be carried out concurrently in counter mode. ECB mode does not hide any patterns in the data since, while utilizing ECB, indistinguishable plain text blocks are encrypted into indistinguishable cipher text blocks. The cryptography community believes it is extremely weak and should not be utilized. The primary drawback of this technique is that the user must know how to contact the entity with whom they are sharing data to obtain the key. Asymmetric algorithms like Rivest-Shamir-Adleman (RSA) are frequently used to encrypt and send separate copies of symmetric encryption keys. Compared to DES, it requires more processing and communication rounds.

Both the source and the receiver have a key pair – a public key and a private key – in asymmetric key algorithms or public key algorithms. The public key is available and generated by the sender and accessed by the recipient, whereas the private key is generated by the receiver and is kept hidden. The ability to obtain knowledge of the decryption key from knowledge of the encryption (public) key is essential for an asymmetric system to function. For instance, the popularity and marketing of the RSA public key system, which is currently in use, hinges on how difficult it is to generate factors for large prime integers. As there is no need for a previous secret key exchange because the public key is secure, the costs associated with security research and development are reduced. However, RSA blends public and symmetric key methods (hybrid encryption) to increase efficiency. The communicating resources employ a public key exchange to create a shared key (the traffic or session encryption key). This shared key is then used with a symmetric algorithm to offer data confidentiality.

Dual RSA is a family of variants that can be used when two instances of RSA are needed, reducing the amount of space needed to store the keys [7]. Blind signatures and authentication/secrecy were the two dual RSA applications that were suggested in this study. New RSA variants have been proposed by Pradeep et al. [7], in which two distinct RSA key pairs with the same public and private exponents are produced as a result of the key generation techniques. A factorization technique was proposed by Sivakumar et al. [8] to speed up the RSA algorithm, which is used to find the factor of a positive integer N. This study focused on the Fermat method factorization of all trivial and nontrivial integer values, which requires fewer steps when factorizing the RSA modulus. In contrast to RSA, the Modified RSA Encryption Algorithm introduced by Sivakumar et al. [8] is secure. The intractability hypothesis is dependent on the factoring issue as well as the decisional composite residual assumptions.

Two modifications were suggested by Khan et al. [9] to expedite RSA decryption. Batch Multi-Prime RSA quickens RSA decryption by combining multi-prime RSA with batch RSA. Depending on the multi-prime RSA and RSA-S2 system, Encrypt Assistant Multi-Prime RSA (EAMRSA) improves RSA decryption performance. By employing the RSA cryptosystem (EAMRSA), the authors developed a fast RSA decryption and signing technique and lowering modules and private exponents in modular exponentiation. One of the well-known public key algorithms is Elliptic Curve Cryptography (ECC), which Koblitz and Miller first suggested in 1985. ECC offers information privacy while communicating. The Elliptic Curve Discrete Logarithm Problem is the foundation for this scheme's security [9]. Elliptic Curve Cryptography (ECC) over GF(p) has been implemented efficiently and quickly on hardware by utilizing modified Jacobian coordinates. This process was accomplished by elliptic curve point operation equations [10]. By minimizing the number of times the multiplication method is performed, the approach solves the point-doubling issue and improves performance. However, a power analysis attack prompts the attacker to monitor power usage, which prompts an attack on the secret key.

Google recently implemented additional levels of encryption to safeguard data on its Google Cloud platform. Google uses the AES 128 and AES 256 encryption techniques to encrypt data at rest on its cloud infrastructure. Google separates user data into several segments and encrypts each segment with a different encryption key [11]. These encryption keys are only used in Google's central Key Management Service, wrapped around the data, and the newly created encryption key to provide additional security [12]. A unique key encrypts data when refurbished, not the existing one. The remaining pieces of data are unaffected if one chunk is hacked since each piece of data is encrypted with a different key. To ensure that only Google services operating in authority and having access at the time can decrypt each chunk, Google uses Access Control Lists [13]. This prevents unauthorized access, preserving data security and protection. Due to the fact that data chunks are disseminated globally, an attacker must both (1) find all of the locations of the various chunks that correspond to the data they want and (2) be aware of the encryption keys for every single piece of data. Data in an Amazon S3 area are repetitively stored across several locations. When data are corrupted, this redundancy helps with data recovery. Amazon S3 also uses versioning to track all the variations of each object in the bucket. Versioning allows us to recover from unforeseen user actions and program errors quickly. Data at rest, stored on discs at Amazon S3 data centers, are encrypted using 256-bit AES by Amazon's server-side encryption, which is identical to Google's [14]. Microsoft uses a shared responsibility strategy to protect data security and privacy on its Azure cloud platform.

3. Methodology

The proposed curve-based approach is based on a cylindrical helically coiled compression spring constructed of a length of wire or rod having a circular cross-section.

Figure 1 shows the cylindrical helically coiled spring comprises "$\boldsymbol{N}$-turns" each with a circular cross-section. These $\boldsymbol{N}$ circles of the spring are utilized in the proposed work for mapping the plain text. The proposed algorithm is explained elaborately by considering only a single circular cross-section of the "$\boldsymbol{N}$-turns" coil. A circle is a geometry composed of all points in a plane that are at a specific distance from the centre, or alternatively, it is the curve that a moving point in a plane draws so that its distance from the centre remains constant. Figure 2 shows the one circular cross-section of a cylindrical Helical coiled spring, and the simple text in the suggested technique is mapped to $(\boldsymbol{x},\boldsymbol{y})$ coordinates that are taken from each quadrant.

The parameters used in the algorithm are as follows:

- $\boldsymbol{R_c}$ denotes the radius of the circle (To be maintained as a secret)
- $\boldsymbol{P}$ denotes the prime value
- $(\boldsymbol{x},\boldsymbol{y})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the coordinates of all the possible points on the circle's perimeter
- $\boldsymbol{PU_c}$ denotes the public key
- $\boldsymbol{PR_c}$ denotes the private key

The step-by-step procedure of the proposed cryptographic algorithm is given as follows:

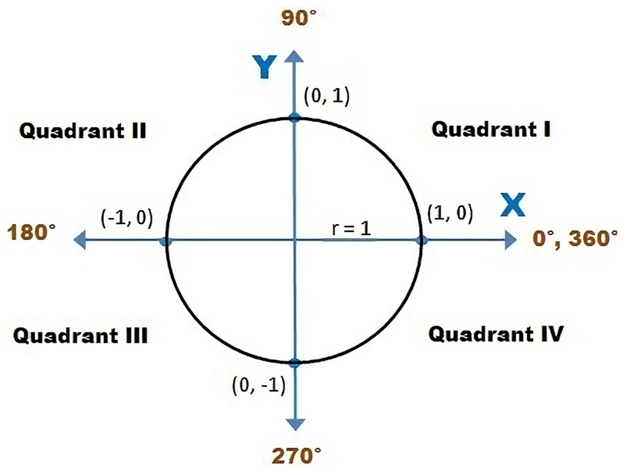
3.1. The cryptographic algorithm

Step 1: Mapping the plain text to the $(\mathbf{x},\mathbf{y})$ coordinates of the circle. Two different cases are considered during the process of mapping the plain text to the coordinates of the circle.

- **Case 1:** The points are generated using $\boldsymbol{x^2+y^2=R_c{}^2}$. The points thus obtained can lie in any of the four quadrants of the circle.

Figure 1
***N* turn variable diameter coil**

Figure 2
$(\boldsymbol{x},\boldsymbol{y})$ points of the circle considered along the 4 quadrants



- **Case 2:** The points are generated using $\boldsymbol{x^2+y^2=R_c{}^2 mod\,P}$.
- **Case 3:** The points are generated using the equation $\boldsymbol{x^2+y^2=R_c{}^2}\mathbf{mod\ GF}(\boldsymbol{p^m})$. The points thus obtained can lie in any of the four quadrants of the circle. Here $\boldsymbol{p}$ is binary.
- **Case 4:** The points are generated using the equation $\boldsymbol{x^2+y^2=R_c{}^2}\mathbf{mod\ GF}(\boldsymbol{p^m})$. The points thus obtained can lie in any of the four quadrants of the circle where Galois Field values are overternary.

Step 2 (Encryption): The mathematical relation between $\boldsymbol{R_c}$ and prime number $\boldsymbol{P}$ is given as follows:

$$\boldsymbol{R_c{}^2+2\times R_c=1mod\,P} \tag{1}$$

The public key $\boldsymbol{PUc}$ is generated based on the area of the circle. Hence, the mathematical relation for generating the public key is as follows:

$$\boldsymbol{PU_c=\pi\times R_c{}^2\ mod\,P=PU_c=\frac{22}{7}\times R_c{}^2\ mod\,P} \tag{2}$$

The multiplicative inverse of $\boldsymbol{7\,mod\,P}$ is required to be computed in order to determine the public key $\boldsymbol{PU_c}$. This substantiates the reason for not choosing the P value equal to 7. The Cipher text $(\boldsymbol{Cx}, \boldsymbol{Cy})$ is generated as follows:

$$\boldsymbol{Cx = (X)^{PU_c} mod\, P} \tag{3}$$

$$\boldsymbol{Cy = (Y)^{PU_c} mod\, P} \tag{4}$$

Step 3 (Decryption): The mathematical relation between the public key $\boldsymbol{PU_c}$ and the private key $\boldsymbol{PR_c}$ is as follows:

$$\boldsymbol{PUc * PRc = 1 mod(P-1)} \tag{5}$$

$$\boldsymbol{X = (Cx)^{PR_c} mod\, P} \tag{6}$$

$$\boldsymbol{Y = (Cy)^{PR_c} mod\, P} \tag{7}$$

The plain text $(\boldsymbol{X}, \boldsymbol{Y})$ is generated using Equations (6–7). The proposed algorithm is elaborated with an illustration as follows:

- Let us consider the radius of the circle $\boldsymbol{R_c = 5}$.
- According to the mathematical relation between $\boldsymbol{R_c}$ and prime number $\boldsymbol{P}$,

$$\boldsymbol{R_c^{\,2} + 2 \times R_c = 1 mod\, P}$$

$$\boldsymbol{5^2 + 2 \times 5 = 1 mod\, P}$$

$$\boldsymbol{25 + 10 = 1 mod\, P}$$

$$\boldsymbol{35 = 1 mod\, P}$$

The above equation is satisfied for $\boldsymbol{P = 17}$, i.e., $\boldsymbol{35 mod 17 = 1}$.

The pseudocode of the proposed strategy is given in *Algorithm*.

Algorithm
Pseudocode of the proposed algorithm

```
Sender:
  while(true) //Repeat forever
  canSend=true //It will allow the first frame to go.
  {
  WaitForEvent(); //sleep until the occurrence of an event
  if(Event(RequestToSend) AND canSend)
  {
  Initialize $x^2 + y^2 = Rc^2 mod\, P$
  Initialize $PUc = \pi * Rc^2 mod\, P = .PUc = 227 * Rc^2 mod\, P$
  $PUc * PRc = 1 mod(P-1)$
  Input the plain text value while the user Encrypt the value
  $X = (CX)^{PRc} \text{ mod } P\ Y = (Cy)^{PRc} mod\, P$
  canSend=false; //cannot send until the acknowledgement
  arrives.
  }
  WaitForEvent(); //sleep until the occurrence of an event
  if(Event(ArrivalNotification)) //indicates the arrival of the
  acknowledgement
  {
  ReceiveFrame(); //Means the ACK frame received
  canSend=true;
  } }
  Receiver:
  while(true) //means Repeat forever
  {
  WaitForEvent(); //sleep until the occurrence of an event
  if(Event(ArrivalNotification)) //indicates arrival of the data
  frame
  Deliver(data); //delivers the data to the network layer.
  SendFrame(); //Send the ACK frame
  }
```

4. Results and Discussion

A modest code is used to validate and assess the effectiveness of the suggested algorithm. By this test, researchers demonstrated that the suggested algorithm is preferable to all others and runs faster if implemented on hardware [15]. The period is active during key, encryption, and decryption peer processes. Overall, the working framework for integrating CloudSim and iFogSim as simulators on the Eclipse integrated development environment, querying on an Intel(R) Core-i3 processor running at 2.27 GHz and 4 GB of RAM on Windows 10 is still complete. For the experimental evaluation, MATLAB 2020a was used for this research work. CloudSim is one of the most well-known and effective simulators for cloud-based applications. Four case studies are analyzed in this study and discussed in the following subsections.

4.1. Case study 1

Let us consider mapping the plain text to the points on the circle. Let us consider the equation $x^2 + y^2 = R_c^{\,2}$, i.e., $x^2 + y^2 = 25$. Table 1 records the obtained data points from the circle mapping for case study 1.

In this illustration, the points are obtained from the first circular cross-section of the coil. Similarly, multiple points can be obtained by considering the subsequent circular cross-sections 2, 3,$\ldots N$ of the coil.

- The public key $PU_c = \pi \times R_c^{\,2}\, mod\, P = \frac{22}{7} \times 25\, mod\, 17 = 13$

Table 1
Points obtained from circle mapping to plain text (case study 1)

Plain text	A	B	C	D	E	F
Points on the circle	(0,5)	(0,–5)	(5,0)	(–5,0)	(3,4)	(–3,4)
Plain text	G	H	I	J	K	L
Points on the circle	(–3,–4)	(3,–4)	(4,3)	(–4,3)	(–4,–3)	(4,–3)

- Encryption of the plain text points using Equations (3) and (4) and the obtained cipher text points are recorded in Table 2.

- Determine the private key as follows:

$PU_c \times PR_c = 1 mod(P-1) = 13 * PR_c = 1 mod(16)$, hence $PR_c = 5$. Table 3 records the obtained decrypted Cipher test using Equations (6) and (7).

4.2. Case study 2

Let us consider mapping the plain text to the points on the circle. Let us consider the equation $x^2 + y^2 = R_c^{\,2} mod\, P$, i.e., $x^2 + y^2 = 25\, mod\, 17$. Table 4 records the points obtained from circle mapping to plain text for case study 2.

Table 2
Cipher text points obtained from Equations (3) and (4) (case study 1)

Plain text	A	B	C	D	E	F
Points on the circle	(0,5)	(0,−5)	(5,0)	(−5,0)	(3,4)	(−3,4)
Cipher points	(0,3)	(0,−3)	(3,0)	(−3,0)	(12,4)	(−12,4)
Plain text	G	H	I	J	K	L
Points on the circle	(−3,−4)	(3,−4)	(4,3)	(−4,3)	(−4,−3)	(4,−3)
Cipher points	(−12,−4)	(12,−4)	(4,12)	(−4,12)	(−4,−12)	(12,−4)

Table 3
Decrypting the cipher text obtained using the Equations (6) and (7)

Plain text	Points on the circle	Cipher text	Decrypted cipher text
A	(0,5)	(0,3)	(0,5)
B	(0,−5)	(0,−3)	(0,−5)
C	(5,0)	(3,0)	(5,0)
D	(−5,0)	(−3,0)	(−5,0)
E	(3,4)	(12,4)	(3,4)
F	(−3,4)	(−12,4)	(−3,4)
G	(−3,−4)	(−12,−4)	(−3,−4)
H	(3,−4)	(12,−4)	(3,−4)
I	(4,3)	(4,12)	(4,3)
J	(−4,3)	(−4,12)	(−4,3)
K	(−4,−3)	(−4,−12)	(−4,−3)
L	(4,−3)	(12,−4)	(4,−3)

In this illustration, the points are obtained from the first circular cross-section of the coil. Similarly, multiple points can be obtained by considering the subsequent circular cross-sections 2, 3,…N of the coil [16]. The public key $PU_c = \pi \times {R_c}^2 \; mod \, P = \frac{22}{7} * 25 \; mod$ $17 = 13$. The encryption of the plain text points using Equations (3) and (4) and the obtained cipher text points are recorded in Table 5.

Determine the private key $PU_c \times PR_c = 1mod(P-1) = 13 \times PRc = 1mod(16)$, hence $PR_c = 5$. Table 6 records the obtained decrypted cipher test using Equations (6) and (7) for case study 2.

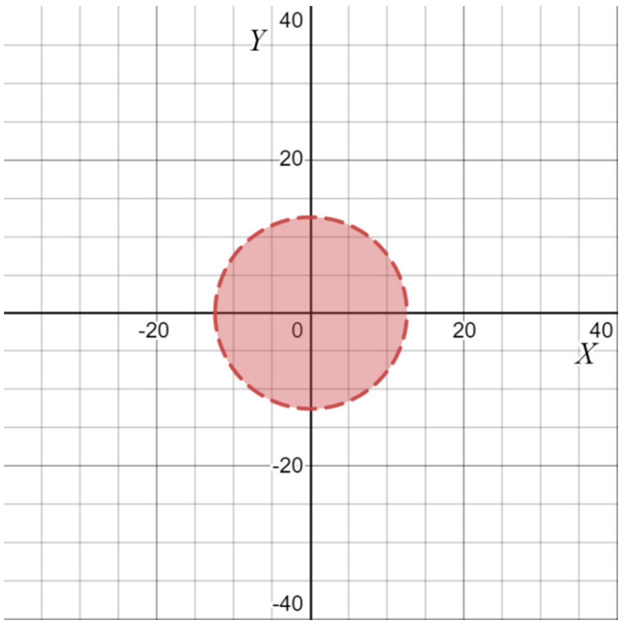
Another illustration of the proposed algorithm is discussed as follows. Consider the value of $R_c = 13$. The possible (x, y) coordinates that satisfy the equation $x^2 + y^2 = 169$ are as follows. The location of (x, y) coordinates along four quadrants is shown in Figure 3.

$$x = \pm 12, y = \pm 5$$

$$x = \pm 5, y = \pm 12$$

$$x = \pm 13, y = 0$$

Figure 3
(x, y) points of the circle along the 4-quadrants



$$x = 0, y = \pm 13$$

According to the mathematical relation between R_c and prime number P, the following relation is satisfied.

$${R_c}^2 + 2 \times R_c = 1modP$$

$$13^2 + 2 \times 13 = 1modP$$

$$169 + 26 = 1modP$$

$$195 = 1modP$$

The above equation is satisfied for $P = 97$, i.e., $195mod97 = 1$.

Table 4
Points obtained from circle mapping to plain text (case study 2)

Plain text	A	B	C	D	E	F	G	H	I
Points on the circle	(0,5)	(0,12)	(2,2)	(2,15)	(3,4)	(3,13)	(4,3)	(4,14)	(5,0)
Plain text	K	L	M	N	O	P	Q	R	S
Points on the circle	(12,0)	(0,−5)	(0,−12)	(−12,0)	(−2,2)	(−2,−2)	(2,−2)	(−2,15)	(−2,−15)
Plain text	T	U	V	W	X	Y	Z	0	1
Points on the circle	(2,−15)	(−3,4)	(−3,−4)	(3,−4)	(−3,13)	(−3,−13)	(3,−13)	(−4,3)	(−4,−3)
Plain text	2	3	4	5					
Points on the circle	(4,−3)	(−3,−4)	(−4,−14)	(4,−14)					

Table 5
Cipher text points obtained from Equations (3) and (4) (case study 2)

Plain text	Points on the circle	Cipher text
A	(0,5)	(0,3)
B	(0,12)	(0,14)
C	(2,2)	(15,15)
D	(2,15)	(15,2)
E	(3,4)	(12,4)
F	(3,13)	(12,13)
G	(4,3)	(4,12)
H	(4,14)	(4,5)
I	(5,0)	(3,0)
J	(12,0)	(14,0)
K	(0,–5)	(0,–3)
L	(0,–12)	(0,–14)
M	(–5,0)	(–3,0)
N	(–12,0)	(–14,0)
O	(–2,2)	(–15,15)
P	(–2,–2)	(–15,–15)
Q	(2,–2)	(15,–15)
R	(–2,15)	(–15,2)
S	(–2,–15)	(–15,–2)
T	(2,–15)	(15,–2)
U	(–3,4)	(–12,4)
V	(–3,–4)	(–12,–4)
W	(3,–4)	(12,–4)
X	(–3,13)	(–12,13)
Y	(–3,–13)	(–12,–13)
Z	(3,–13)	(–12,–13)
0	(–4,3)	(–4,12)
1	(–4,–3)	(–4,–12)
2	(4,–3)	(4,–12)
3	(–4,14)	(–4,5)
4	(–4,–14)	(–4,–5)
5	(4,–14)	(4,–5)

Table 6
Decrypting the cipher text obtained using the Equations (6) and (7) for case study 2

Plain text	Points on the circle	Cipher text	Decrypted plain text
A	(0,5)	(0,3)	(0,5)
B	(0,12)	(0,14)	(0,12)
C	(2,2)	(15,15)	(2,2)
D	(2,15)	(15,2)	(2,15)
E	(3,4)	(12,4)	(3,4)
F	(3,13)	(12,13)	(3,13)
G	(4,3)	(4,12)	(4,3)
H	(4,14)	(4,5)	(4,14)
I	(5,0)	(3,0)	(5,0)
J	(12,0)	(14,0)	(12,0)
K	(0,–5)	(0,–3)	(0,–5)
L	(0,–12)	(0,–14)	(0,–12)
M	(–5,0)	(–3,0)	(–5,0)
N	(–12,0)	(–14,0)	(–12,0)
O	(–2,2)	(–15,15)	(–2,2)
P	(–2,–2)	(–15,–15)	(–2,–2)
Q	(2,–2)	(15,–15)	(2,–2)
R	(–2,15)	(–15,2)	(–2,15)
S	(–2,–15)	(–15,–2)	(–2,–15)
T	(2,–15)	(15,–2)	(2,–15)
U	(–3,4)	(–12,4)	(–3,4)
V	(–3,–4)	(–12,–4)	(–3,–4)
W	(3,–4)	(12,–4)	(3,–4)
X	(–3,13)	(–12,13)	(–3,13)
Y	(–3,–13)	(–12,–13)	(–3,–13)
Z	(3,–13)	(–12,–13)	(3,–13)
1	(–4,3)	(–4,12)	(–4,3)
2	(–4,–3)	(–4,–12)	(–4,–3)
3	(4,–3)	(4,–12)	(4,–3)
4	(–4,14)	(–4,5)	(–4,14)
5	(–4,–14)	(–4,–5)	(–4,–14)
6	(4,–14)	(4,–5)	(4,–14)

The public key $PU_c = \left(\frac{22}{7}\right) \times R_c^2 \, mod \, P$. By substituting values of R_c and P, we obtain

$$PU_c = \left(\frac{22}{7}\right) * (13)^2 \, mod \, 97$$

$$PU_c = (22) * 14 * (13)^2 mod 97$$

i.e., $(7)^{-1} mod 97 = 14$, and $PU_c = 60$. Since (x, y) coordinates from all the quadrants are mapped to the plain text, and to retain the sign of the plain text, the public key and private key obtained should satisfy the condition. PU_c mod $2 = 1$ and PR_c mod $2 = 1$, hence the value of $PU_c = 61$.

Encryption – Let us consider the plain text as (−5,−12).

$$Cx = (X)^{PU_c} modP$$

$$Cx = (-5)^{61} mod \, 97$$

$$Cx = -68$$

$$Cy = (Y)^{PU_c} mod \, P$$

$$Cy = (-12)^{61} mod \, 97$$

$$Cy = -70$$

The cipher text is obtained as (−68,−70).

Decryption – The relationship between PU_c and PR_c is as follows:

$$PUc * PRc = 1 mod(P - 1)$$

$$61 * PRc = 1 mod 96$$

Therefore, $PR_c = 85$.

$$X = (Cx)^{PRc} modP$$

$$X = (-68)^{85} mod 97$$

$$X = (-5)$$

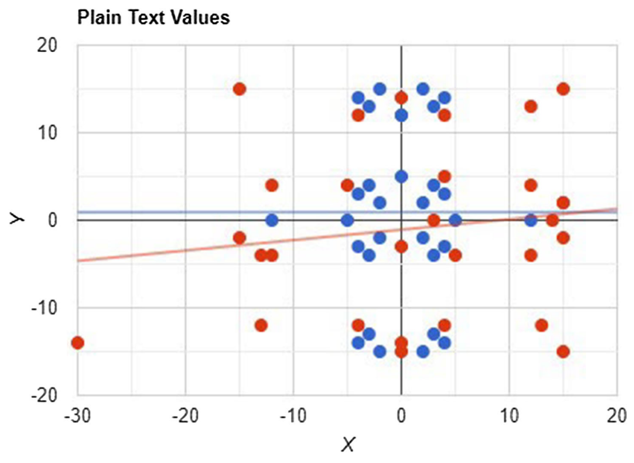
$$Y = (Cy)^{PRc} modP$$

$$Y = (-70)^{85} mod 97$$

$$Y = (-12)$$

Therefore, the plain text (−5,−12) is obtained after decryption. Figure 4 shows the mapping of plain text and cipher text points in coordinates.

Figure 4
Mapping of plain text and cipher text points



4.3. Case study 3

Let us consider Encryption with Galois Field Modular Arithmetic [17]. Consider the $Gf(p^m)$ Table 7 shown below where $p(x) = 1 + x + x^3$.

- Encryption

Let us consider the plain text as (x, y).

$$Cx = (X)^{PUc} modGF(p^m) \tag{8}$$

$$Cx = (Y)^{PUc} modGF(p^m) \tag{9}$$

The cipher text is obtained as $(Cx\ , Cy)$.

- Decryption

Table 7
Polynomial representation of GF(2^4)

Power representation	Polynomial representation	4-tuple representation
0	0	0000
1	1	1000
α	α	0100
α^2	α^2	0010
α^3	α^3	0001
α^4	$1 + \alpha$	1100
α^5	$\alpha + \alpha^2$	0110
α^6	$\alpha^3 + \alpha^2$	0011
α^7	$1 + \alpha^3 + \alpha$	1101
α^8	$1 + \alpha^2$	1010
α^9	$\alpha^3 + \alpha$	0101
α^{10}	$1 + \alpha^1 + \alpha^2$	1110
α^{11}	$\alpha + \alpha^2 + \alpha^3$	0111
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	1111
α^{13}	$1 + \alpha^2 \ + \alpha^3$	1011
α^{14}	$1 + \alpha^3$	1001

The relationship between PUc and PRc is as follows:

$$PUc * PRc = 1mod\ GF\big(p^{m)}\big) - 1$$

$$X = (Cx)^{PRc} modGF(p^m) \tag{10}$$

$$Y = (Cy)^{PRc} modGF(p^m) \tag{11}$$

Encryption:
Let us consider the plain text as (0,5)

$$Cx = (X)^{PUc} modGF(p^m)$$

$$Cx = (0)^2 mod\ GF(2^4)$$

$$Cx = 0$$

$$Cy = (Y)^{PUc} modGF(p^m)$$

$$Cy = (5)^2 mod\ GF(2^4)$$

$$Cy = 2$$

The cipher text is obtained as (0,2)
Decryption:
The relationship between PUc and PRc is

$$PUc * PRc = 1mod\ GF\big(p^{m)}\big) - 1$$

$$2{*}PRc = 1mod15$$

Therefore, $PRc = 8$.

$$X = (Cx)^{PRc} modP$$

$$X = (0)^8 mod\ GF(2^4)$$

$$X = (0)$$

$$Y = (Cy)^{PRc} mod\ GF(2^4)$$

$$Y = (2)^8 mod\ GF(2^4)$$

$$Y = (5)$$

So, we have obtained plain text as (0,5).

Encryption with Galois Field Modular Arithmetic for higher order

Here is another example with a higher order of Galois Field arithmetic [18]. Consider the $Gf(p^m)$ table shown below where $p(x) = 1 + x + x^5$. Table 8 shows the polynomial representation of $Gf(2^5)$.

Encryption:

Let us consider the plain text as (0,5).

$$Cx = (X)^{PUc} modGF(p^m)$$

$$Cx = (0)^2 mod\ GF(2^5)$$

$$Cx = 0$$

$$Cy = (Y)^{PUc} mod\ GF(2^5)$$

$$Cy = (5)^2\ mod\ GF(2^5)$$

$$Cy = 17$$

The cipher text is obtained as (0,17).

Table 8
Polynomial representation of $Gf(2^5)$

Galois field $Gf(2^5)$elements	Binary representation	Decimal representation
0	0 0 0 0 0	0
α	1 0 0 0 0	1
$α^1$	0 1 0 0 0	2
$α^2$	0 0 1 0 0	4
$α^3$	0 0 0 1 0	8
$α^4$	0 0 0 0 1	16
$α^5$	1 0 1 0 0	5
$α^6$	0 1 0 1 0	10
$α^7$	0 0 1 0 1	20
$α^8$	1 0 1 1 0	13
$α^9$	0 1 0 1 1	26
$α^{10}$	1 0 0 0 1	17
$α^{11}$	1 1 1 0 0	7
$α^{12}$	0 1 1 1 0	14
$α^{13}$	0 0 1 1 1	28
$α^{14}$	1 0 1 1 1	29
$α^{15}$	1 1 1 1 1	31
$α^{16}$	1 1 0 1 1	27
$α^{17}$	1 1 0 0 1	19
$α^{18}$	1 1 0 0 0	3
$α^{19}$	0 1 1 0 0	6
$α^{20}$	0 0 1 1 0	12
$α^{21}$	0 0 0 1 1	24
$α^{22}$	1 0 1 0 1	21
$α^{23}$	1 1 1 1 0	15
$α^{24}$	0 1 1 1 1	30
$α^{25}$	1 0 0 1 1	25
$α^{26}$	1 1 1 0 1	23
$α^{27}$	1 1 0 1 0	11
$α^{28}$	0 1 1 0 1	22
$α^{29}$	1 0 0 1 0	9
$α^{30}$	0 1 0 0 1	18

Decryption
The relationship between PUc and PRc is as follows:

$$PUc * PRc = 1mod\ GF\big(p^{m)}\big) - 1$$

$$2 * PRc = 1mod31$$

Therefore, $PRc = 16$.

$$X = (Cx)^{PRc}mod\ GF(p^m)$$

$$X = (0)^8\ mod\ GF(2^5)$$

$$X = (0)$$

$$Y = (Cy)^{PRc}mod\ GF(2^5)$$

$$Y = (17)^8mod\ GF(2^5)$$

$$Y = (5)$$

So, we have obtained plain text as (0,5).

4.4. Case study 4

Let us consider GF (33). The elements of GF (33) are constructed using the primitive polynomial $p(x) = x^3 + 2x^2 + 1$. Let α be the root of the polynomial $p(x)$ [19].

$$p(x = \alpha) = \alpha^3 + 2\alpha^2 + 1 = 0$$

$$\alpha^3 = -2\alpha^2 - 1$$

Hence, $\alpha^3 = \alpha^2 + 2$. The polynomial representation is recorded in Table 9.

Encryption:
Let us consider the plain text as $(0, 5)$ or $(0, \alpha^{11})$, $PUC = \alpha$.

$$Cx = (X)^{PUc}modGF(p^m)$$

$$Cx = (0)^{\alpha}mod\ GF(3^3)$$

$$Cx = 0$$

$$Cy = (Y)^{PUc}mod\ GF(3^3\ )$$

$$Cy = (\alpha)^{11\ \alpha}mod\ GF(3^3)$$

$$Cy = \alpha^7$$

The cipher text is obtained as (0, α^7) or (0, 10).

Table 9
Polynomial representation of case study 4

Elements	Polynomial representation	Ternary and decimal representation
0	0	$(000)_3 = (0)$
1	1	$(100)_3 = (1)$
α	α	$(010)_3 = (3)$
α^2	α^2	$(001)_3 = (9)$
α^3	$2 + \alpha^2$	$(201)_3 = (11)$
α^4	$2 + 2\alpha + \alpha^2$	$(221)_3 = (17)$
α^5	$2 + 2\alpha$	$(220)_3 = (8)$
α^6	$2\alpha + 2\alpha^2$	$(022)_3$=(24)
α^7	$1 + \alpha^2$	$(101)_3 = (10)$
α^8	$2 + \alpha + \alpha^2$	$(211)_3 = (14)$
α^9	$2 + 2\alpha + 2\alpha^2$	$(222)_3$=(26)
α^{10}	$1 + 2\alpha + \alpha^2$	$(121)_3 = (16)$
α^{11}	$2 + \alpha$	$(210)_3$=(5)
α^{12}	$2\alpha + \alpha^2$	$(021)_3 = (15)$
α^{13}	2	$(200)_3 = (2)$
α^{14}	2α	$(020)_3 = (6)$
α^{15}	$2\alpha^2$	$(002)_3 = (18)$
α^{16}	$1 + 2\alpha^2$	$(102)_3 = (19)$
α^{17}	$1 + \alpha + 2\alpha^2$	$(112)_3 = (22)$
α^{18}	$1 + \alpha$	$(110)_3 = (4)$
α^{19}	$\alpha + \alpha^2$	$(011)_3 = (12)$
α^{20}	$2 + 2\alpha^2$	$(202)_3 = (20)$
α^{21}	$1 + 2\alpha + 2\alpha^2$	$(122)_3 = (25)$
α^{22}	$1 + \alpha + \alpha^2$	$(111)_3 = (13)$
α^{23}	$2 + \alpha + 2\alpha^2$	$(212)_3 = (23)$
α^{24}	$1 + 2\alpha$	$(120)_3 = (7)$
α^{25}	$\alpha + 2\alpha^2$	$(012)_3 = (21)$
α^{26}	1	$(100)_3 = (1)$

Table 10
Comparative time analysis of the proposed method with ECC

Time	ECC	Proposed method
Time consumption for encryption and decryption	0.059 s and 0.7346 s	0.0214 s and 0.0429 s
Key generation time	0.043 s	0.0298 s
Overall processing time	0.1770 s	0.1236 s
Throughput	1627 bps	2330 bps

Decryption:
The relationship between PUc and PRc is

$$PUc * PRc = 1 mod\ GF\left(p^{m)}\right) - 1$$

$$\alpha^2 * PRc = 1 mod\ GF(3^3)$$

Therefore, $PRc = \alpha^2$ or 9

$$X = (Cx)^{PRc} modGF(p^m)$$

$$X = \left(0\right)^{\alpha^2}\ modGF\left(p^m\right)$$

$$X = (0)$$

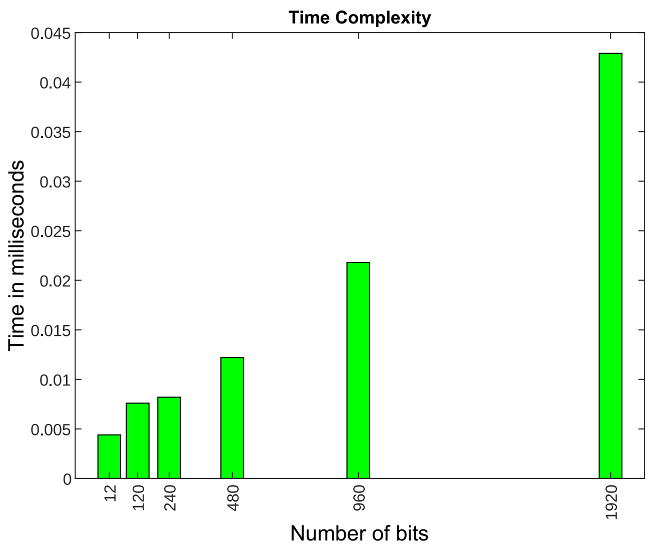
$$Y = (Cy)^{PRc} modGF(p^m)$$

$$Y = (\alpha^7)^{\alpha^2} mod\ GF(3^3)$$

$$Y = \alpha^{11}$$

Therefore, the plain text is obtained as $(0, \alpha^{11})$. Figure 5 shows the time complexity analysis obtained as $O(N)$.

Figure 5
Time complexity analysis



Any new cryptographic algorithm has to be proven in relation to security and randomness since these are the performance metrics that determine the algorithm's effectiveness. In this proposed algorithm, performance is assessed using the NIST statistical suite. The security of any algorithm depends on the computational feasibility of encryption decryption keys. In Brute Force Attack, the hacker has some information regarding the plain text and the complete information about the cipher text to determine the key. By obtaining the key using the Brute Force Attack, the hacker gets all the information the two parties communicate. This algorithm recovering the data is impossible since all the data are concealed within a cylindrical coil of radius R.

Since the hacker is unaware of mapping the data from the cylindrical coil, he cannot recover the original information [20].

Randomness is the other important criterion for evaluating any cipher considered a true random number generator. The algorithm has been tested in all the 15 core tests using NIST to evaluate the randomness. According to the NIST statistical suite, the P value depends on the randomness of the cipher. The Value of the P should be greater than 0.01 for all 15 core tests of NIST. The inputs to the NIST suite to test our algorithm original and the cipher obtained and revealed the value of P greater than 0.1229 for all the 15 core tests. Hence, it can be concluded that there is sufficient randomness in the cipher. The comparative time analysis of the proposed method is shown in Table 10. The results proved that this method consumes less time and is more secure than the conventional elliptic curve method [21,22].

In Figure 6, the x–y coordinate values can be positive, negative, or a real number. The plain text is mapped on the cylindrically helical coil, as shown in Figure. (The blue circles). They appear in the shape of a cylindrical coil if all the blue dots are joined together.

Upon computation, the cipher text obtained can also be mapped to the cylindrical helical coil, as shown in Figure 4 (The red stars). They appear in the shape of a cylindrical coil if all the red stars are joined together. The randomness test signifies no statistical relation between the plain text and cipher text [23,24]. 1920 values of data are considered to determine the time complexity. Data can be positive, negative, and a real number. Encryption time: 0.02145 s and the decryption time: 0.0429 s.

The strength of the algorithm is as follows: It can be directly linked to the data type of the message and the key. There are "N" turns in the cylindrical coil, which is of non-uniform size. The number of turns of the cylindrical coil "N" is decided only between the transacting parties. Each turn of the cylindrical coil has many integers. These integer values are mapped to the plain text. As the coil progresses from turn to turn, the coil grows dimensionally [25]. More integer points are obtained for mapping the plain text. Similarly, upon computation of the cipher text, it was experimentally observed that they lie on any of the turns of the cylindrical helical coil. Hence, the results of the run test exhibit the randomness between the plain text and cipher text points.

Figure 7 shows the mapping of plain text and cipher text points of ECC. Here, the x–y coordinate values are positive [26]. The plain text is mapped on the cubic polynomial over the prime Galois field, as shown in Figure. These points do not take any particular geometric structure. Upon computation, the cipher text obtained is mapped on the cubic polynomial over the prime Galois field, as shown in Figure. These points do not take any particular geometric structure. The randomness test signifies no statistical relation between the plain text and cipher text [27]. 1920 values of data are considered to determine the time complexity. Data are positive [28]. Encryption time: 0.059 s and the decryption time: 0.7346 s.

Figure 6
Mapping of plain text and cipher text points of the proposed method

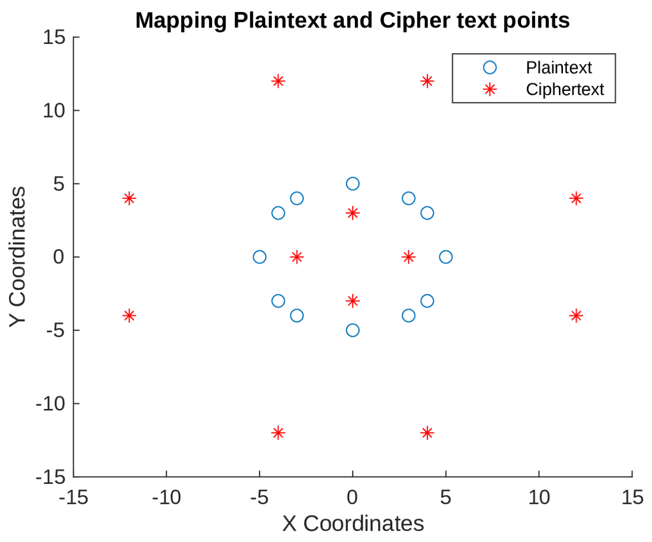
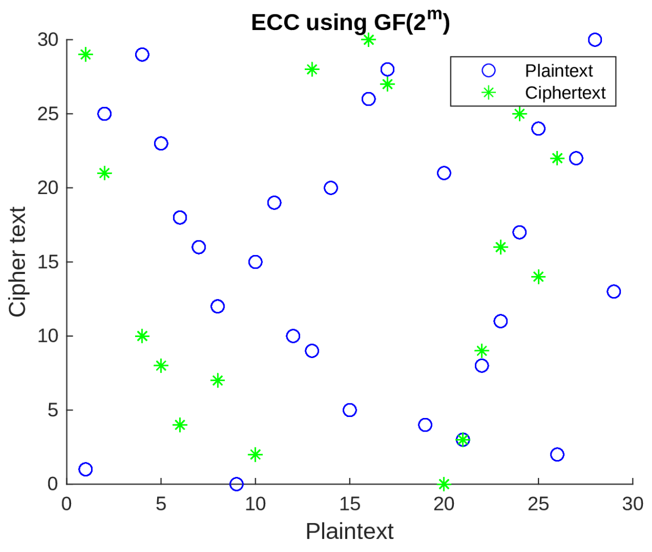


Figure 7
Mapping of plain text and cipher text points of ECC



5. Conclusion

Using maximum length random sequence generation, this study suggests a new cylindrical coil-based cryptographic scheme and a data mapping technique on a circular cross-section over a finite field. This suggested technique is used to secure data in cloud systems. The proposed scheme is tested on prime fields of different bit lengths and Galois field, and the experimental results show very good resilience to cryptanalytic attacks like a random walk and enhanced computing performance compared to conventional approaches. To ensure data confidentiality and information integrity of users' data in the cloud computing environment, an effective security framework is developed that provides a way by which communication is protected and unauthorized access is limited. Cloud users can safely manage data integrity and privacy thanks to the proposed security architecture. It also makes it possible to use the network, store data in the cloud, and protect personal information without depending on the viability of the cloud provider. Using the AES algorithm establishes a strong basis for protecting data kept in the cloud and limiting access to individuals who have properly authenticated and validated their identities. In the real world, delays can happen under various conditions, and our framework does not compensate for all of them. Finite field arithmetic applied to ternary data can extend the suggested work. To increase security, scientists can design and implement finite field multipliers over polynomial and optimum normal bases for pipeline and parallel architectures.

Acknowledgements

We would like to thank the management of Dayananda Sagar College of Engineering for the support rendered.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Smitha Sasi: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Visualization. **Srividya Bharadwaj Venkata Subbu:** Conceptualization, Methodology, Validation, Investigation, Data curation, Writing – original draft, Visualization. **Premkumar Manoharan:** Software, Validation, Formal analysis, Resources, Writing – review & editing, Visualization, Supervision, Project administration. **Anju Vijayakumar Kulkarni:** Formal analysis, Resources, Writing – review & editing, Supervision. **Laith Abualigah:** Formal analysis, Resources, Writing – review & editing.

References

[1] Bs, V., Arvindhan, M., & Kalimuthu, S. (2023). The crucial function that clouds access security brokers play in ensuring the safety of cloud computing. In *2023 16th International Conference on Security of Information and Networks*, 1–5. https://doi.org/10.1109/SIN60469.2023.10475014

[2] Kumar, N., Upreti, K., & Mohan, D. (2022). Blockchain adoption for provenance and traceability in the retail food supply chain: A consumer perspective. *International Journal of E-Business Research*, *18*(2), 1–17. http://doi.org/10.4018/IJEBR.294110

[3] Mahmood, G. S., Huang, D. J., & Jaleel, B. A. (2019). Achieving an effective, confidentiality and integrity of data in cloud

computing. *International Journal of Network Security*, *21*(2), 326–332. http://doi.org/10.6633/IJNS.201903 21(2).17

[4] Saeed, O., & Shaikh, R. A. (2018). A user-based trust model for cloud computing environment. *International Journal of Advanced Computer Science and Applications*, *9*(3), 337–346. https://doi.org/10.14569/IJACSA.2018.090347

[5] Shukla, P. K., Aljaedi, A., Pareek, P. K., Alharbi, A. R., & Jamal, S. S. (2022). AES based white box cryptography in digital signature verification. *Sensors*, *22*(23), 9444. https://doi.org/10.3390/s22239444

[6] de Los Reyes, E. M., Sison, A. M., & Medina, R. (2019). Modified AES cipher round and key schedule. *Indonesian Journal of Electrical Engineering and Informatics*, *7*(1), 29–36. http://dx.doi.org/10.52549/ijeei.v7i1.652

[7] Pradeep, K. V., Vijayakumar, V., & Subramaniyaswamy, V. (2019). An efficient framework for sharing a file in a secure manner using asymmetric key distribution management in cloud environment. *Journal of Computer Networks and Communications*, *2019*, 9852472. https://doi.org/10.1155/2019/9852472

[8] Sivakumar, P., NandhaKumar, M., Jayaraj, R., & Kumaran, A. S. (2019). Securing data and reducing the time traffic using AES encryption with dual cloud. In 2019 *IEEE International Conference on System, Computation, Automation and Networking*, 1–5. https://doi.org/10.1109/ICSCAN.2019.8878749

[9] Khan, M. R., Upreti, K., Alam, M. I., Khan, H., Siddiqui, S. T., Haque, M., & Parashar, J. (2023). Analysis of Elliptic Curve cryptography & RSA. *Journal of ICT Standardization*, *11*(4), 355–378. https://doi.org/10.13052/jicts2245-800X.1142

[10] Chowdary, G. N. M., Nylu, Y., Deepthi, B., Prasad, K. V., & Kannaiah, S. K. (2023). Elliptic Curve cryptography for network security. In *2023 International Conference on Inventive Computation Technologies,* 1500–1503. https://doi.org/10.1109/ICICT57646.2023.10134492

[11] Chinnasamy, P., Padmavathi, S., Swathy, R., & Rakesh, S. (2021). Efficient data security using hybrid cryptography on cloud computing. In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020*, 537–547. https://doi.org/10.1007/978-981-15-7345-3_46

[12] Chinnasamy, P., & Deepalakshmi, P. (2022). HCAC-EHR: Hybrid cryptographic access control for secure EHR retrieval in healthcare cloud. *Journal of Ambient Intelligence and Humanized Computing*, *13*(2), 1001–1019. https://doi.org/10.1007/s12652-021-02942-2

[13] Rachmat, N., & Samsuryadi. (2019). Performance analysis of 256-bit AES encryption algorithm on Android smartphone. *Journal of Physics: Conference Series*, *1196*, 012049. https://doi.org/10.1088/1742-6596/1196/1/012049

[14] Sugumar, D. R., & Joycee, K. A. M. (2018). FEDSACE: A framework for enhanced user data security algorithms in cloud computing environment. *International Journal on Future Revolution in Computer Science & Communication Engineering*, *4*(3), 49–52.

[15] Mohammed, Z. A., Gheni, H. Q., Hussein, Z. J., & Al-Qurabat, A. K. M. (2024). Advancing cloud image security via AES algorithm enhancement techniques. *Engineering, Technology & Applied Science Research*, *14*(1), 12694–12701. https://doi.org/10.48084/etasr.6601

[16] Adelmeyer, M., Walterbusch, M., Biermanski, P., & Teuteberg, F. (2018). Trust transitivity and trust propagation in cloud computing ecosystems. In *Twenty-Sixth European Conference on Information Systems*, 1–17.

[17] Meng, F., Lin, R., Wang, Z., Zou, H., & Zhou, S. (2018). A multi-connection encryption algorithm applied in secure channel service system. *EAI Endorsed Transactions on Security and Safety*, *5*(15), e1. http://dx.doi.org/10.4108/eai.15-5-2018.155167

[18] Marwan, M., Kartit, A., & Ouahmane, H. (2018). A framework to secure medical image storage in cloud computing environment. *Journal of Electronic Commerce in Organizations*, *16*(1), 1–16. https://doi.org/10.4018/JECO.2018010101

[19] El Mrabet, N., Guillevic, A., & Ionica, S. (2011). Efficient multiplication in finite field extensions of degree 5. In *Progress in Cryptology–AFRICACRYPT 2011: 4th International Conference on Cryptology*, 188–205. https://doi.org/10.1007/978-3-642-21969-6_12

[20] Kythe, D. K., & Kythe, P. K. (2012). *Algebraic and stochastic coding theory*. USA: CRC Press.

[21] Lin, S., & Costello, D. J. (1983). *Error control coding: Fundamentals and applications*. USA: Prentice Hall.

[22] Zholubak, I., & Hlukhov, V. (2023). Verification of synthesized by the IP-core generator multipliers of extended galois fields GF (p^n) elements. In *2023 13th International Conference on Dependable Systems, Services and Technologies*, 1–4. https://doi.org/10.1109/DESSERT61349.2023.10416466

[23] Steffy, R. C. A., Pushpa, S. E. P., Mariammal, K., Senbagakuzhalvaimozhi, S., & Varalakshmi, P. (2022). Area optimized implementation of Galois field fourier transform. In *2022 International Conference on Sustainable Computing and Data Communication Systems*, 736–743. https://doi.org/10.1109/ICSCDS53736.2022.9761047

[24] Asaker, A. A., Elsharkawy, Z. F., Nassar, S., Ayad, N., Zahran, O., & Abd El-Samie, F. E. (2021). A novel Iris cryptosystem using elliptic curve cryptography. In *2021 9th International Japan-Africa Conference on Electronics, Communications, and Computations*, 155–158. https://doi.org/10.1109/JAC-ECC54461.2021.9691307

[25] VenkataGiri, J., & Murty, A. S. R. (2021). Elliptical curve cryptography design principles. In *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology*, 889–893, https://doi.org/10.1109/RTEICT52294.2021.9573662

[26] Ihor, D., & Viktor, S. (2023). Fast exponential method on Galois fields for cryptographic applications. In *2023 13th International Conference on Dependable Systems, Services and Technologies*, 1–4. https://doi.org/10.1109/DESSERT61349.2023.10416519

[27] Liang, H., Liu, H., Dang, F., Yan, L., & Li, D. (2021). Information system security protection based on SDN technology in cloud computing environment. In *2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications*, 432–435. https://doi.org/10.1109/AEECA52519.2021.9574276

[28] Hazzazi, M. M., Attuluri, S., Bassfar, Z., & Joshi, K. (2023). A novel Cipher-based data encryption with Galois field theory. *Sensors*, *23*(6), 3287. https://doi.org/10.3390/s23063287

How to Cite: Sasi, S., Subbu, S. B. V., Manoharan, P., Kulkarni, A. V., & Abualigah, L. (2024). Design and Implementation of Discrete Field Arithmetic-Based Cylindrical Coil-Driven Crypto Framework for Cloud Data. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE42022816