DOI: https://doi.org/10.52756/ijerr.2023.v32.028 *Int. J. Exp. Res. Rev.*, Vol. 32: 323-339 (2023)

Original Article | *Peer Reviewed* | *Open Access*

International Journal of Experimental Research and Review (IJERR)

ISSN: 2455-4855 (Online)
www.iaph.in
ISSN 2455-4855

A Novel Cryptographic Technique for Cloud Environment Based on Feedback DNA

Nadia Sharfuddin[1*], Faisal Anwer[2] and Salman Ali[3]

[1]Department of Computer Science, Aligarh Muslim University, Aligarh-202002, India; [2]Department of Computer Science, Aligarh Muslim University, Aligarh-202002, India; [3]Department of Computer Science, Aligarh Muslim University, Aligarh-202002, India.

E-mail/Orcid Id:

NS, nadiasharfu@gmail.com, https://orcid.org/0009-0007-4510-4398; *FA,* faisalanwer.cs@myamu.ac.in, https://orcid.org/0000-0001-7198-704X; *SA,* salmanali.amu@gmail.com, https://orcid.org/0000-0001-6495-1591

Article History:

Received: 3rd Jul., 2023
Accepted: 21st Aug., 2023
Published: 30th Aug., 2023

Keywords:
Cloud Environment, DNA-based Key generation, DNA sequence, Data Security, Cryptography.

Abstract: In the era of cloud computing, ensuring data security is crucial for users and organizations. This paper presents an innovative approach called F-DNAES, which combines DNA cryptography and the Advanced Encryption Standard (AES) algorithm, to enhance data security in cloud-based environments. The proposed methodology focuses on securing data during its transmission and storage in the cloud. By leveraging the unique properties of DNA, such as its storage capacity and stability, along with the robustness of AES encryption, F-DNAES provides a highly secure and efficient solution. The proposed method also ensures the authentication of the Data owners and the users. The F-DNAES model effectively mitigates potential security threats such as phishing attacks, collision attacks, and password guessing attacks. It also offers superior performance in terms of key generation, encryption, and decryption processes. The proposed approach can be applied to various cloud-based scenarios, including IoT infrastructure, web-based applications, and network security. F-DNAES contributes to strengthening data privacy and security, providing users with a trusted and reliable solution in cloud environments.

Introduction

The rapid advancement of information and communication technology has transformed the world into a global village, enabling seamless access and sharing of information across geographical boundaries. However, this interconnectedness has also introduced significant challenges, particularly in terms of data security and privacy. Cloud computing has emerged as a powerful solution to address the increasing demands for storage, processing power, and accessibility (Sakr et al., 2011). It offers a flexible and scalable infrastructure that allows users to store and access their data remotely. However, the inherent nature of cloud computing brings forth unique security risks and concerns (Noor et al., 2013). Ensuring cloud security becomes paramount in safeguarding the confidentiality, integrity, and availability of data stored and processed in cloud environments. With numerous potential threats such as data breaches, unauthorized access, and data loss, robust security measures must be implemented to protect sensitive information (Varadharajan and Tupakula, 2014). Cryptography, a fundamental concept in information security, plays a vital role in providing the necessary security services (Stallings, 2005).

Cryptography employs mathematical algorithms to encrypt and decrypt data, rendering it unreadable to unauthorized individuals. By using encryption techniques, data can be securely transmitted and stored, ensuring that only authorized parties can access and interpret the information. The encryption key is crucial for decrypting the data, adding an extra layer of security (Rama Devi and Bhuvaneswari, 2022).

In recent years, DNA-based encryption has emerged as a novel approach in the field of cryptography. Leveraging the unique properties of DNA, such as its capacity to store and process vast amounts of information, DNA

***Corresponding Author:** nadiasharfu@gmail.com

cryptography holds promising potential for secure data transmission and storage (Pavithran et al., 2021). Binary data is converted into a DNA sequence, which is then stored and transmitted. Decryption can only be achieved through a unique key generated by a computer program.

The proposed approach combines the power of DNA-based encryption, specifically the Feedback DNA technique, with the widely adopted Advanced Encryption Standard (AES) algorithm. This hybrid approach aims to optimize the computational time required for key generation, encryption, and decryption (Patnala and Kiran Kumar, 2019). The master key is generated from an elliptic curve, ensuring robust security. The plaintext is then encrypted using AES and represented in the form of DNA.

By integrating cloud security with DNA-based encryption and AES, the proposed approach provides a robust and efficient solution for safeguarding sensitive data in cloud environments (Rosado et al., 2012). This research contributes to the advancement of secure data storage and transmission, addressing the growing concerns surrounding cloud security.

The remaining part of the paper is structured as follows: The literature review for this paper is presented in Section 2. The background of the proposed model and proposed scheme are described in Sections 3 and 4. Section 5 contains the complete System Setup, Section 6 discusses the Security Analysis of the proposed Model, the comparison and the effectiveness of the suggested methodology is in Section 7, and the conclusion is in Section 8.

Review of Literature

The approach Varsha Kolate proposed in the paper (Kolate and Joshi, 2021) offers multilevel security in addition to DNA-based AES encryption, in which the input is DNA-based and takes the shape of the four nucleotides A, T, G, and C before being processed by AES and a DNA-based key. In this study, the message is converted to ASCII format first, with the bits taken in pairs like 00, 01, 11, etc., and then DNA nucleotides A, T, G, and C are applied. The AES algorithm is then used over the DNA-based input, with the master key of AES being DNA-based, and the generated subkeys are also based on this master key. This algorithm's primary goal is to transmit and receive business information safely.

Al Husainy et al. (2021) proposed a DNA-based cryptographic system in the paper Click or tap here to enter text.. The author has recommended that pairs of bits have the following values: A-00, T-01, C-10, and G-11, with each bit corresponding to A, T, G, or C. The key size used in the cryptographic system is inversely proportional to the data block size. To ensure a desired level of security, the proposed cryptographic system involves features like the randomness of the key and its length. Consequently, IoT devices that utilize high-speed cryptographic systems are necessary to encrypt shared data quickly. The author of this paper introduced an adaptable, lightweight IOT device encryption method. To work around the memory and computation limitations of IOT devices, the system employs dynamic data sizing and reliable logical operations for data encryption and decryption. Compared to AES, the suggested encryption scheme has demonstrated superior results regarding encryption time. The suggested encryption approach incorporates DNA sequences to generate high levels of randomness, making it difficult for attackers to decrypt the keys. The proposed encryption technique effectively enhances data security by creating complexity and scattering data. Additionally, the suggested encryption system passed the avalanche test with a score of above 50%, indicating that it is resistant to attacks utilizing statistical analysis. As a result, the suggested system is a potential method to be applied in most IOT devices with various capabilities.

In the paper (Tiwari & Kim, 2018), both encryption and decryption are performed using the mapped characters. Characters are given non-repeating DNA sequences that have been sorted. The author advised selecting the DNA sequence to be utilized first, then sequence padding, division, sorting, and character mapping, and last, converting the DNA message to an integer for encryption. From the perspective of security performance, ECC is used. Each sensor's DNA sequences are selected randomly from a collection of DNA sequences.

The author of the paper (Institute of Electrical and Electronics Engineers. Bangladesh Section & IEEE Communications Society, n.d.-a) developed a unique encryption technique utilizing subsequent DNA cryptography and a delayed chaotic neural network. The binary sequence produced by a chaotic neural network is required for the XOR operation with a message block. When DNA cryptography is included, the proposed technique excels in security. Once DNA cryptography is included, communication between the sender and the receiver becomes secure. To increase the safety of the suggested model, the additional DNA cryptographic approach performs a castoff of the Cipher text obtained from the first-level encryption.

The author of the paper (Kadhim and Ali, 2019) proposes a new method for improving the security of the advanced encryption standard (AES) by combining 3D chaos theory and DNA operations. The authors believe that increasing the encryption process's complexity can improve AES's security and reliability. The proposed

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

Table 1. Proposed vs. existing models.

Method	Authentication Mechanisms	Encryption Algorithm	Key Length	Security
Proposed Method (with AES)	XOR operations, F-DNAES-based encoding	F-DNAES	Alphanumeric DNA-based 256-bit master key	Innovative and potentially robust security measures, Resistance to Traditional Attacks, Threat Mitigation, Potential Quantum Resistance
AES with DNA (Kolate & Joshi, 2021)	DNA-based input, ASCII conversion	DNA-based AES Encryption	128-bit DNA-based key	Safe transmission and reception of business information
Lightweight cryptographic systems (Husainy et al., 2021)	DNA encryption (A-00, T-01, C-10, G-11)	DNA encryption	24-bit key size	Vulnerability assessment, Intrusion Prevention, Security Awareness
DNA Mapping for ECC (Tiwari & Kim, 2018)	ECC-based encryption and decryption	Elliptic Curve Cryptography	Variable (Determined by the ECC algorithm used)	Resilience to timing and SPA attacks
DNA Cryptography with Chaotic Neural Network (Institute of Electrical and Electronics Engineers. Bangladesh Section & IEEE Communications Society, n.d.-a)	DNA cryptography, Chaotic Neural network	XOR operation with chaotic neural network output and message block	Variable (Permutation-based key generation.)	Enhanced network security
3D Chaos-DNA AES (Kadhim & Ali, 2019)	AES-based logistic map and DNA	Advanced Encryption Standard (AES)	128-bit round key	Improved security and resistance to attacks

method involves using a 3D chaotic map to generate a chaotic sequence, which is then combined with DNA operations such as complementation, addition, and multiplication. The resulting sequence is used to alter the plaintext before AES encryption. According to the authors, this method improves security and resistance to various attacks, including differential and linear attacks. The proposed method is evaluated and compared to traditional AES and other existing methods using various statistical tests. The results demonstrate that the proposed method outperforms others in terms of security and complexity.

Background of the proposed scheme

This proposed approach focuses on two key elements:

System Mode

Cloud Service Provider

The cloud service provider is the central authority responsible for managing the cloud environment and offering various services. It operates multiple servers with sufficient power and storage capacity to support infrastructure and accommodate user needs.

Data Owner

Data owners store confidential or regular data on the cloud server's database. They rely on the cloud service provider to handle their data securely.

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

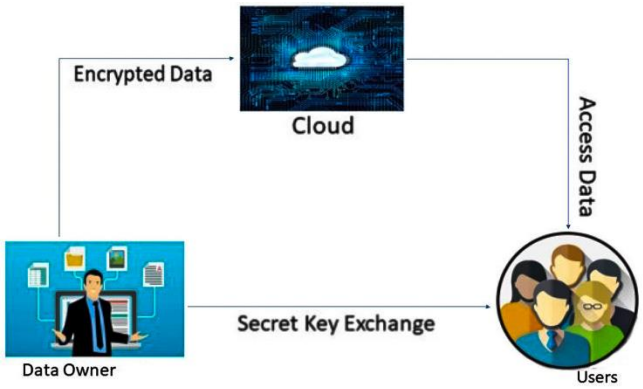


Figure 1. System Model of F-DNAES

User

Users represent authorized entities or individuals aiming to access data content or utilize services from the cloud server. The system model of F-DNAES is depicted in Figure 1.

Design Objectives

The proposed scheme is guided by specific system requirements and design objectives, which are as follows:

Precise access control

The primary objective is to ensure that only authorized users can access data stored on the cloud server. Unauthorized access is strictly prohibited. To achieve this, the Data Owner must assign appropriate access rights to their data, allowing only authorized users to access and view them. The proposed scheme focuses on implementing robust access control mechanisms to enhance data security and protect the privacy of the stored data.

Enhanced Security

Security is a critical concern in cloud environments due to the presence of potential hackers and attackers. Protecting users' sensitive data is of utmost importance. The cloud service provider (CSP) is responsible for implementing robust security measures to safeguard users' confidential files. The proposed scheme prioritizes the development of efficient and scalable data storage solutions that ensure strong security for users' sensitive information.

Minimized System Overhead

In existing schemes, data owners (DOs) are required to be continuously online throughout the communication process, resulting in increased system overhead. The F-DNAES approach aims to reduce this overhead by optimizing system operations and streamlining the data communication phase. Efforts are focused on minimizing unnecessary resource consumption and enhancing overall system efficiency (Namasudra et al., 2021).

Proposed scheme

The proposed scheme employs a DNA-based secret key to encrypt users' confidential data or messages, thereby enhancing security. The key generation process involves multiple stages, further strengthening the security measures. The scheme is structured around five core phases: system setup, registration, login, DNA-based storage, and data access. The complete workflow of F-DNAES is depicted in Figure 2 illustrating the step-by-step progression of the scheme.

System Setup

In the system setup phase, the three key entities come into play: the Cloud Service Provider (CSP), the Data Owner (DO), and the User. The first step is the registration process, where users create their profiles and gain access to their accounts. With their unique credentials, users can log in and utilize the system's functionalities.

Once logged in, users have the option to securely send their data to the cloud. The data undergoes encryption using the advanced F-DNAES encryption technique, ensuring its confidentiality and protection. The encrypted data is then stored in the cloud, ready to be retrieved when requested.

When a user desires to access their data, they contact the Data Owner and make the request. The Data Owner retrieves the requested data from the cloud, decrypts it using the appropriate keys, and securely delivers it back to the user. This seamless process ensures the confidentiality and integrity of the user's data throughout the entire data transfer cycle.

Overall, this innovative system offers a secure and efficient way for users to store and retrieve their data in the cloud, providing peace of mind and ensuring the privacy of their valuable information.

Registration

To register on the cloud server, users are required to submit a registration request to the service provider. During the registration process, the CSP collects essential personal and confidential information from the users, including their name, date of birth, age, and address, to create a unique user profile. To maintain security, a secure communication channel, such as a secure socket layer, is established to facilitate the exchange of the key pair, registration, and confirmation. A similar registration procedure is followed for Data Owners (DOs) on the

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

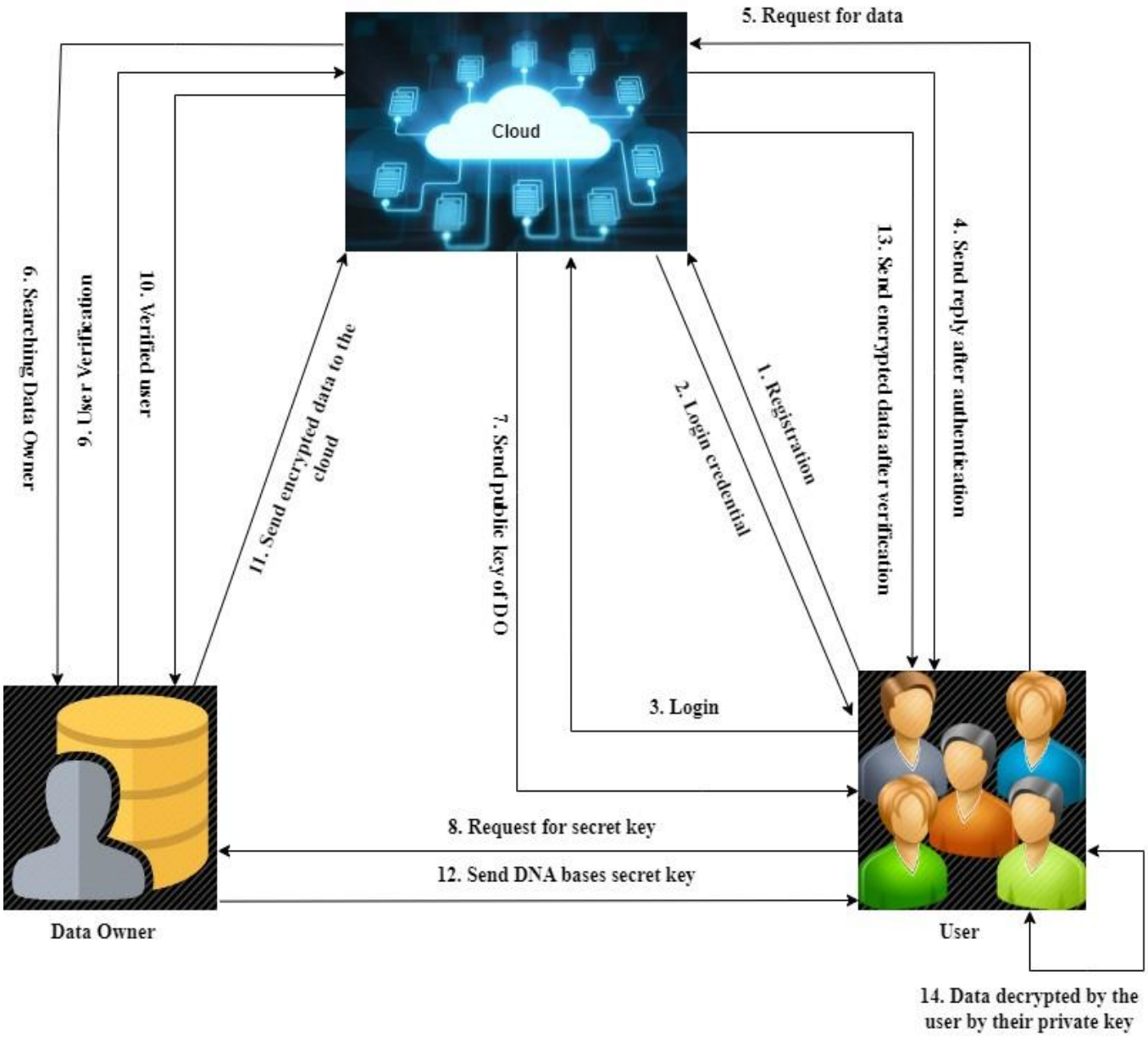


Figure 2. Workflow of the proposed F-DNAES

cloud server to maintain a consistent security approach (Muttik and Barton, 2009).

User Login

Only authorized users or customers who have completed the registration phase are permitted to log into the system or server. After the login process, users can send a request to the service provider to access their Data. The CSP responds by providing the DO's public key in encrypted form, which is required to obtain the certificate or access rights and the secret key or password from the respective DO. The user can only request data from the corresponding DO once they have obtained the DO's public key.

Upon confirming the user's authenticity, the CSP facilitates the delivery of the requested data's certificate and secret key in encrypted form, ensuring secure and authenticated access for the authorized user (Fernandes et al., 2014).

In this Methodology, the encryption and decryption processes use DNA-based keys. The key created for this process is generated randomly, and the encryption and decryption processes are performed using AES by utilizing a shared secret key. AES is a block cipher and symmetric key algorithm. The proposed methodology is shown in Figure 3 for data encryption and decryption in the cloud environment.

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

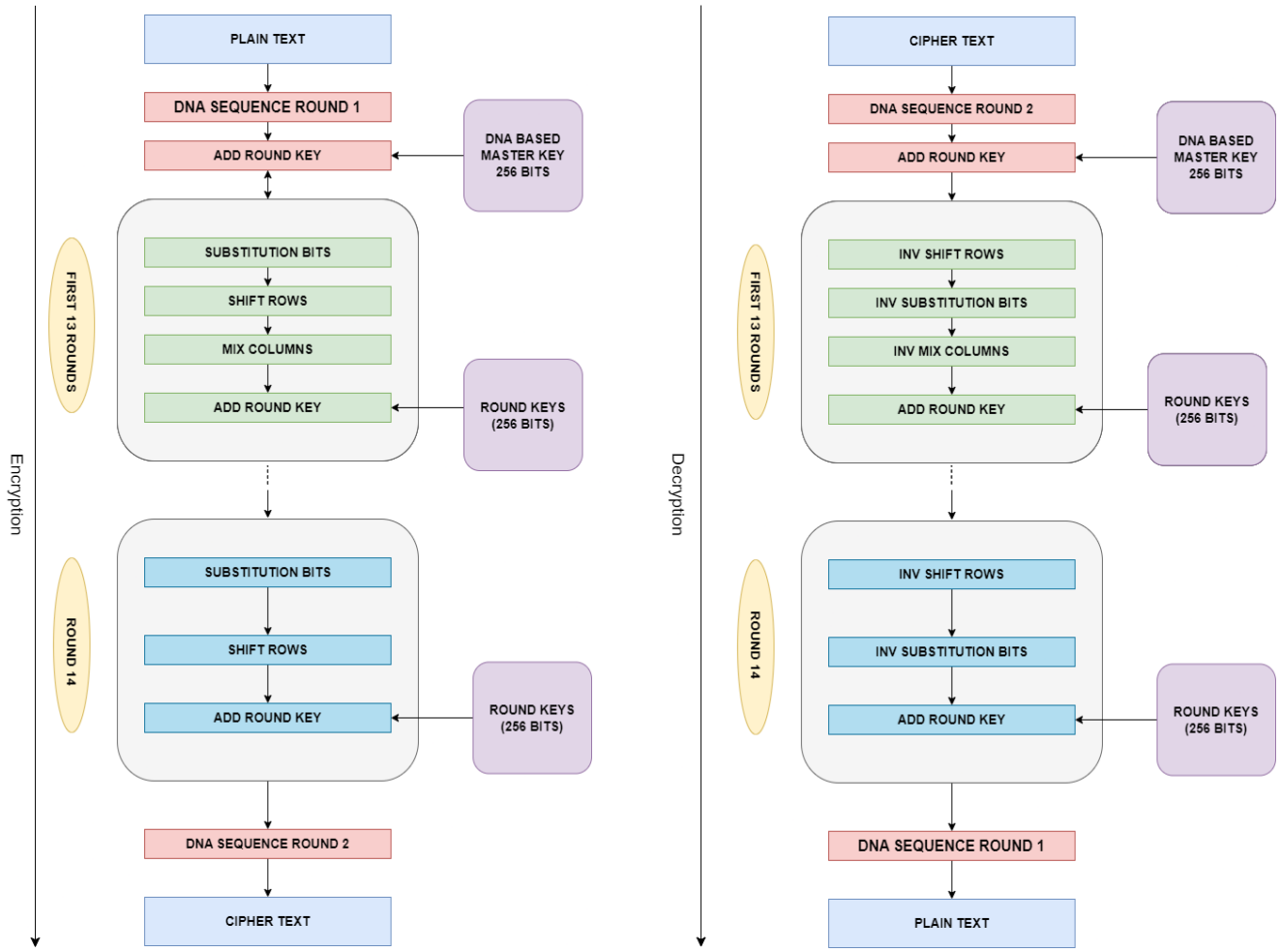


Figure 3. Structure of Feedback DNA along with AES Algorithm.

Nucleotides, such as A, T, G, and C, encode DNA sequences. Proteins are created by combining the two opposing binary strands of the DNA. In DNA-based encryption, only two letters, C and T, are utilized, and the letter A is used as feedback or to maintain track of the terms used during the XOR operation rather than working on all the nucleotides (Young et al., 2007). This process involves dual DNA encoding, where the plaintext is first converted to DNA, then the DNA is converted back to plaintext, and finally, the cipher text is converted to DNA. This results in the final ciphered text produced by these operations (Safaa et al., 2023).

Working of F-DNA (Feedback DNA)

Step 1: Choosing plaintext as in raw data or a file.

Step 2: Computing the binary of the plaintext using Figure 4.

Step 3: After calculating binary data, take an 8-bit at one time, and make four pairs of 2-bits. Now, perform the iteration over each octet in pairs of 2 bits.

For example, suppose the binary data is 00 01 10 11. If two bits are 00 or 10, then perform XOR with 01. Else, perform XOR with 10 By the above logic.

Note: As seen in Table 2, we got 01 = C (1 in decimal) and 11 = T (3 in decimal) twice. This will raise an issue at the time of decryption as the program will fail to recognize what binary was used at the time of the XOR operation. Due to this reason, we will append "A" we get 01 after performing the XOR operation with 10 and also in the case where we get 11 after performing XOR 10.

Step 4: With the same discussed logic, we will encrypt the plaintext. After encrypting the plaintext, we can further encode the cipher text in DNA (using the same logic as given in step 3).

Description with examples as follows:

Example:

Table 2 presents the XOR operations conducted on the binary data as demonstrated in the provided example:

1. Binary digit 00011011

2. Taking two digits together 00, 01, 10, 11

3.00 or 10, then perform the XOR operation with 01. Else perform the XOR operation with 10.

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char
0	0000 0000	00	[NUL]	32	0010 0000	20	space	64	0100 0000	40	@	96	0110 0000	60	`
1	0000 0001	01	[SOH]	33	0010 0001	21	!	65	0100 0001	41	A	97	0110 0001	61	a
2	0000 0010	02	[STX]	34	0010 0010	22	"	66	0100 0010	42	B	98	0110 0010	62	b
3	0000 0011	03	[ETX]	35	0010 0011	23	#	67	0100 0011	43	C	99	0110 0011	63	c
4	0000 0100	04	[EOT]	36	0010 0100	24	$	68	0100 0100	44	D	100	0110 0100	64	d
5	0000 0101	05	[ENQ]	37	0010 0101	25	%	69	0100 0101	45	E	101	0110 0101	65	e
6	0000 0110	06	[ACK]	38	0010 0110	26	&	70	0100 0110	46	F	102	0110 0110	66	f
7	0000 0111	07	[BEL]	39	0010 0111	27	'	71	0100 0111	47	G	103	0110 0111	67	g
8	0000 1000	08	[BS]	40	0010 1000	28	(	72	0100 1000	48	H	104	0110 1000	68	h
9	0000 1001	09	[TAB]	41	0010 1001	29	)	73	0100 1001	49	I	105	0110 1001	69	i
10	0000 1010	0A	[LF]	42	0010 1010	2A	*	74	0100 1010	4A	J	106	0110 1010	6A	j
11	0000 1011	0B	[VT]	43	0010 1011	2B	+	75	0100 1011	4B	K	107	0110 1011	6B	k
12	0000 1100	0C	[FF]	44	0010 1100	2C	,	76	0100 1100	4C	L	108	0110 1100	6C	l
13	0000 1101	0D	[CR]	45	0010 1101	2D	-	77	0100 1101	4D	M	109	0110 1101	6D	m
14	0000 1110	0E	[SO]	46	0010 1110	2E	.	78	0100 1110	4E	N	110	0110 1110	6E	n
15	0000 1111	0F	[SI]	47	0010 1111	2F	/	79	0100 1111	4F	O	111	0110 1111	6F	o
16	0001 0000	10	[DLE]	48	0011 0000	30	0	80	0101 0000	50	P	112	0111 0000	70	p
17	0001 0001	11	[DC1]	49	0011 0001	31	1	81	0101 0001	51	Q	113	0111 0001	71	q
18	0001 0010	12	[DC2]	50	0011 0010	32	2	82	0101 0010	52	R	114	0111 0010	72	r
19	0001 0011	13	[DC3]	51	0011 0011	33	3	83	0101 0011	53	S	115	0111 0011	73	s
20	0001 0100	14	[DC4]	52	0011 0100	34	4	84	0101 0100	54	T	116	0111 0100	74	t
21	0001 0101	15	[NAK]	53	0011 0101	35	5	85	0101 0101	55	U	117	0111 0101	75	u
22	0001 0110	16	[SYN]	54	0011 0110	36	6	86	0101 0110	56	V	118	0111 0110	76	v
23	0001 0111	17	[ETB]	55	0011 0111	37	7	87	0101 0111	57	W	119	0111 0111	77	w
24	0001 1000	18	[CAN]	56	0011 1000	38	8	88	0101 1000	58	X	120	0111 1000	78	x
25	0001 1001	19	[EM]	57	0011 1001	39	9	89	0101 1001	59	Y	121	0111 1001	79	y
26	0001 1010	1A	[SUB]	58	0011 1010	3A	:	90	0101 1010	5A	Z	122	0111 1010	7A	z
27	0001 1011	1B	[ESC]	59	0011 1011	3B	;	91	0101 1011	5B	[	123	0111 1011	7B	{
28	0001 1100	1C	[FS]	60	0011 1100	3C	<	92	0101 1100	5C	\	124	0111 1100	7C	|
29	0001 1101	1D	[GS]	61	0011 1101	3D	=	93	0101 1101	5D	]	125	0111 1101	7D	}
30	0001 1110	1E	[RS]	62	0011 1110	3E	>	94	0101 1110	5E	^	126	0111 1110	7E	~
31	0001 1111	1F	[US]	63	0011 1111	3F	?	95	0101 1111	5F	_	127	0111 1111	7F	[DEL]

Figure 4. ASCII Conversion chart.

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

Table 2. General XOR operation

00	01	10	11
XOR	XOR	XOR	XOR
01	10	01	10
01	11	11	01

2-bit binary

00 01 10 11

1st 2nd 3rd 4th

00 ⊕ 01 = 01

01 ⊕ 10 = 11

10 ⊕ 01 = 11

11 ⊕ 10 = 01

We can see there can be either 01 or 11 as DNA sequence output.

Now 01=C and 11=T

(ii) and (iii) both give 11 and (i) and (iv) give 01

If 1st and 3rd is the case, take it as usual. If 2nd and 4th is the case, then take it as unusual. (i.e., Append 'A' before) If DNA 'A' is before T, the decoded character will be 11(case ii). If 'A' is before C, the decoded character will be 01(case iv).

Otherwise, for cases (i) and (iii), perform XOR with 01.

Example1:

Encrypt

E = 01000101, n = 01101110, c = 01100011, r = 01110010, y = 01111001, p = 01110000, t = 01110100

let's take E=01000101, pairing it then:

Table 3. XOR operation for E

01	00	01	01
XOR	XOR	XOR	XOR
10	01	10	10
11	01	11	11

We can see in Table 3 that the output is 11, 01, 11, 11

01 ⊕ 10 = 11

00 ⊕ 01 = 01

01 ⊕ 10 = 11

01 ⊕ 10 = 11

01 with 10 is an unusual case, then append 'A' and then XOR 10= 11(T)

Cipher text = AT

00 XOR 01= 01(C) it is usual case so, Cipher text= ATC

01 XOR 10 is an unusual case then Append 'A' before and then 01 XOR 10 =11(T)=AT

Cipher text = ATCAT

Similarly, the 4th

Cipher text for E= ATCATAT

n = 01101110

Table 4. XOR operation for n

01	10	11	10
XOR	XOR	XOR	XOR
10	01	10	01
11	11	01	11

Based on Table 4,

01 ⊕ 10 is an unusual case, append 'A' and then 01 XOR 10 = 11(T) = AT

Cipher text= AT

10 ⊕ 01 is the usual case, then 10 XOR 01 = 11=T

Cipher text= ATT

11 ⊕ 10 = 01(C) unusual case, Append 'A' before C

Cipher text = ATTAC

10 ⊕ 01= 11= T, this is the usual case

Final cipher text for n= ATTACT

c =01100011

01 ⊕ 10 = 11 unusual case = AT,

Cipher text = AT

10 ⊕ 01 =11(T), usual case = T, Cipher text = ATT

00 ⊕ 01= 01(C), usual case, Cipher text = ATTC

11 ⊕10= 11(T)= AT, unusual case, Cipher text = ATTCAT

r = 01110010

Table 5. XOR operation for c.

01	10	00	11
XOR	XOR	XOR	XOR
10	01	01	10
11	11	01	11

Table 6. XOR operation for r.

01	11	00	10
XOR	XOR	XOR	XOR
10	10	01	01
11	01	01	11

01 ⊕ 10 =11(T), unusual case = AT, cipher text = AT

11 ⊕ 10 = 01(C), unusual case = AC, cipher text = ATAC

00 ⊕ 01 = 01, usual case=C, cipher text = ATACC

10 ⊕ 01 = 11, usual case=T, cipher text = ATACCT

y = 01111001

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

Table 7. XOR operation y

01	11	10	01
XOR	XOR	XOR	XOR
10	10	01	10
11	01	11	11

01 ⊕10 =11(T), unusual case=AT, cipher text=AT
11 ⊕ 10 = 01(C), unusual case= AC, cipher text= ATAC
10 ⊕ 01 = 11, usual case=T, cipher text= ATACT
01 ⊕ 10 = 11, unusual case=AT, cipher text= ATACTAT
p = 01110000

Table 8. XOR operation for p

01	11	00	00
XOR	XOR	XOR	XOR
10	10	01	01
11	01	01	01

01 ⊕ 10 =11(T), unusual case=AT, cipher text=AT
11 ⊕10 = 01(C), unusual case= AC, cipher text= ATAC
00 ⊕ 01 = 01, usual case=C, cipher text= ATACC
00 ⊕ 01 = 01, usual case=C, cipher text= ATACCC
t = 01110100

Table 9. XOR operation for t

01	11	01	00
XOR	XOR	XOR	XOR
10	10	10	01
11	01	11	01

01 ⊕ 10 =11(T), unusual case=AT, cipher text=AT
11 ⊕ 10 = 01(C), unusual case= AC, cipher text= ATAC
01 ⊕ 10 = 11(T), unusual case=AT, cipher text= ATACAT
00 ⊕ 01 = 01, usual case=C, cipher text= ATACATC

Encrypt:

01000101 01101110 01100011 01110010 01111001 01110000 01110100

Cipher text for Encrypt= ATCATATATTACTAT-TCATATACCTATACTATATACCCATACATC

Decryption:

At the time of decryption, if a letter such as “T” and “C” are prefixed by “A,” it means that it was acquired after the XOR operation between 10 and 11, respectively.

Working of Cloud-Based F-DNAES

DNA-based AES, along with a feedback factor, uses the DNA bases A, T, C, and G, out of which only two bases, T and C, are used, while 'A' is used for feedback. This algorithm uses a DNA-based master key. The plaintext is first encrypted by using the F-DNA method and then it passes to AES, the cipher text generated by the AES is further encrypted using the DNA feedback method, resulting in the final cipher text.

The DNA-based AES and the feedback factor use two DNA bases, T and C, and 'A' is used as feedback. Therefore, three DNA nucleotides are used in the whole process. A 256-bit key is used for encoding.

Algorithm

Key generation

Step 1. The user requests a secret key and certificate from the Data Owner (DO).
Step 2. DO verifies the user's authorization and proceeds if valid; otherwise, the request is declined.
Step 3. The user provides a password as requested by the DO.
Step 4. Password and necessary user attributes are collected.
Step 5. A random key is selected from an elliptic curve.
Step 6. The key is converted into binary format.
Step 7. A binary key is transformed into an alphanumeric representation by grouping 5 bits at a time.
Step 8. The alphanumeric key is further converted into a DNA-based master key.

Encryption

Step 1: Initiate the process by sending a request from the user to the Data Owner (DO) to acquire the secret key and certificate.
Step 2: Verify the user's authorization by the DO. If the user is valid, proceed to the next step. Otherwise, reject the request.
Step 3: Prompt the user to provide a password.
Step 4: Gather the password and necessary user attributes.
Step 5. Take input from the user as plaintext to encrypt.
Step 6. Convert the plaintext into binary.
Step 7. Encrypt the binary data into a DNA sequence.
Step 8. Save the DNA-encoded data into a file.
Step 9. Encrypt the file using AES.
Step 10. Send and store the file to the CSP.

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

Decryption

Step 1. Decode the attributes and password using a reverse decimal encoding scheme.

Step 2. Validate the password by comparing it with the stored credentials.

Step 3. Grant access to the requested data if the user is authorized.

Step 4. If access is denied, promptly inform the user about the unauthorized attempt.

Step 5. If access is granted, retrieve the encrypted data and decryption key from the trusted Cloud Service Provider (CSP).

Step 6. Deliver the decrypted data to the user, ensuring its confidentiality and integrity.

Step 7. Finalize the decryption process, securely closing the connection with the CSP.

By following these carefully designed steps, users can confidently and securely decrypt their data stored in the cloud, thanks to the reliable assistance of the Cloud Service Provider (CSP).

Implementation

In the proposed methodology, the implementation begins with the setup of the cloud environment by the CSP, providing a secure and scalable infrastructure for data storage and processing. Users register with the CSP, providing necessary information for authentication and profile creation. Upon successful registration, users authenticate themselves to gain access to the cloud services. When a user wants to store data in the cloud, it undergoes encryption using the AES algorithm and DNA sequences. The user generates a secret key and certificate request, which are verified by the DO to ensure authorization. User attributes and passwords are securely collected and transformed into binary values, which are then used to generate DNA sequences representing the encrypted data. These encrypted DNA sequences are stored securely in the cloud infrastructure managed by the CSP. When a user requests access to their data, the encrypted DNA sequences are retrieved and decrypted using the user's secret key and the AES algorithm (Daemen & Rijmen, 1999). The integration of encryption techniques, DNA sequences, and AES algorithm ensures the secure storage and retrieval of data in the cloud, providing enhanced confidentiality and security for users' sensitive information.

The implementation of the proposed method is done in Python by using the library “Pycryptodome.” The system is equipped with an Intel Core™ i5-6300U processor running at a clock speed of 2.40GHz (or 2496 MHz). It features 2 cores and 4 logical processors, providing efficient multitasking capabilities and performance.

Security Analysis

The proposed F-DNAES model represents a state-of-the-art cloud security framework that incorporates advanced measures to protect data confidentiality, integrity, and availability. In this model, the collaborative roles of Cloud Service Providers (CSPs), Data Owners (DOs), and authorized users play a crucial role in ensuring a secure cloud environment. To maintain data confidentiality, the F-DNAES model employs robust encryption techniques (Zhang et al., 2014). Authorized users are granted access to encrypted data through a secure authentication process facilitated by the CSP (Di Pietro & Lombardi, 2018). By utilizing unique keys, unauthorized users are effectively prevented from deciphering sensitive information, ensuring that data remains confidential and secure throughout its lifecycle.

Data integrity is a top priority in the F-DNAES model. Through the implementation of cryptographic hash functions, the system generates checksums for each stored file, serving as digital fingerprints. This enables the detection of any tampering or unauthorized modifications to the data, ensuring its integrity is preserved at all times. The availability of data is guaranteed by the resilient infrastructure provided by the CSP. By implementing redundant storage mechanisms, efficient backup strategies, and load-balancing techniques, the model ensures uninterrupted access to data for authorized users. This mitigates the risk of data loss and minimizes the chances of service disruptions, resulting in a seamless user experience. The F-DNAES model addresses various security threats, including collision attacks, phishing attacks, and masquerade attacks. The generation of strong cryptographic keys, stringent user authentication protocols, and secure communication channels between the CSP, DOs, and users are key components of this model. These measures effectively safeguard against unauthorized access attempts, identity theft, and fraudulent activities (Imam et al., 2022; Kaufman, 2010).

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

In short, the F-DNAES model offers a comprehensive and cutting-edge cloud security solution. By leveraging DNA-based encryption, the AES algorithm, and the collaborative efforts of CSPs, DOs, and users, it ensures the highest levels of data protection. This instills confidence in cloud data storage and transmission, empowering users to leverage the full potential of cloud computing with peace of mind and trust in the security of their valuable data.

Our evaluations confirm its robustness against various attacks such as collision Attacks, Password Guessing Attack, Phishing Attack, and Masquerade Attack ensuring the secure protection of data from unauthorized access (Kandukuri et al., 2009; Namasudra et al., 2021).

Collision Attack

The F-DNAES approach efficiently addresses collision attacks by employing a distinctive method for generating secret keys based on user attributes and chosen passwords. User attributes are treated as confidential and distinct, while user-selected passwords enhance the overall security measures. The DO and CSP strictly adhere to privacy policies and ensure secure data handling, establishing the proposed scheme as F-DNAES highly resistant to collision attacks. This robustness guarantees the preservation of data integrity and confidentiality throughout the system.

Password Guessing Attack

The F-DNAES methodology, implemented within the cloud security framework, effectively mitigates password-guessing attacks. The Cloud Service Provider (CSP) and Data Owner (DO) collaborate to securely store user passwords using encryption techniques. Strong password requirements and account lockout policies further enhance security, preventing repetitive or brute-force guessing attempts. These measures ensure the confidentiality and integrity of user accounts and data, creating a secure cloud computing environment.

Phishing Attack

In the context of phishing attacks, unauthorized individuals attempt to obtain sensitive information, such as user IDs, voter ID numbers, and passwords, from legitimate users. These attackers then exploit this acquired information to gain unauthorized access to cloud services. However, in the F-DNAES approach, the Cloud Service Provider (CSP) follows a secure registration process where user details are collected and a public-private key pair is generated using a secure socket layer. Similarly, the CSP employs the same approach for registering Data Owners (DOs). When a user requests access to data, the CSP provides the respective DO's public key in encrypted form. The DO then confirms the user's authenticity and shares the DNA-based key. Importantly, both the DO and CSP maintain strict confidentiality, ensuring that sensitive information remains protected even from authorized users. As a result, the F-DNAES scheme effectively thwarts phishing attacks by preventing unauthorized individuals from accessing user information.

Masquerade Attack

In a masquerade attack, unauthorized users aim to gain system access by employing false identities or deceptive information. However, the proposed scheme requires all users to undergo a registration process and login with legitimate credentials to the cloud server, ensuring that only authenticated users can access it. Even if a hacker manages to log in and request file access, the Cloud Service Provider (CSP) provides the Data Owner's (DO) public key for obtaining the secret key and certificate. Before generating and sending the DNA-based key and certificate, the DO verifies the user's authenticity through the CSP. If a hacker tries to access Data files with a fake identity, the CSP does not verify their authenticity, resulting in the DO rejecting the request for the secret key and certificate. This robust security measure effectively prevents unauthorized access to the cloud server using fake identities, making the F-DNAES scheme highly resistant to masquerade attacks.

Results and Discussion

Compared to the model from the paper (Bahig & Nassr, 2019), the model in this article is faster, more secure, more efficient, and less time-consuming. When the input size is enormous, this model uses significantly less computing power than others. Many varied input sizes are used to observe the encryption and decryption procedures. The proposed model produces small-size cipher text compared to the DNAES model's size. The main difference is that the DNAES model uses codons for each bit, whereas each character has eight bits and eight different codons, each of which has three characters. For a single byte of encryption, 8(bits)*3(characters of codons) = 24 characters. There will be little difference in time complexity, but using escaped codons unnecessarily uses computational power.

For Example:

The encoded format of ‘video conference’ in the paper (Bahig and Nassr, 2019) is:

“GGA-GTC-GGC-ACA-GGC-GTC-TGG-GGC-CAA-GGC-TCA-GGA-GAT-AGA-GGC-GTC-GGA-ACA-GGA-GTC-CTA-GGA-ACC-GGC-GTA-TGG-GGC-TAG-GGA-TGT-GGA-GAC-AGG-GGC-GTA-GGC-ACC-GGC-GTC-TGG-GGA-CAG-GGC-TCA-GGA-GAC-AGC-AGC-GCA-AAG-AGC-CCA-GCC-CTC-

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

CAA-CCA-CCA-GCC-CTC-CTA-GTC-TAA-TGT-GGC-AAG-TGG-CCA-AGG-TGA-GGC-CTA-TGC-TCC-CCC-GGC-GAC-ATC-TGT-CCA-CAA-CTC-TGA-GCC-GGA-ACA-GGC-CTC-GCA-ACC-GGA-CCC-CTA-GTA-AAG-ACC-CTA-ATG-ACA-CGC-TGG-TCA-TGA-GGA-AGG-GGC-GCC-GAC-GAT-CAA-GGC-GAC-CTA-CCC-ACC-GAC-CCA-GCC-CCC-GGC-AAA-TGG-TCC-GGC-AAC-TGT-AGT-CTA-AGC-GGA-GTC-GGC-ACA-GGC-GTA-TGG-GGC-CAA-GG A-TCC-GGC- GAC"

In addition, the Model proposed in this paper has the encoded form for 'video conference' as:
"ATACATTATTTATATTATCATTATATATTACACC TCCATTCACATTACACATTACTATTATTATTATAT ATACCTATTATATATTACTATTCACATTATAT".

Table 10 presents a comparison of the performance between the proposed F-DNAES scheme and the existing DNAES scheme. The table displays the number of blocks and the time of encryption for each scheme.

Table 10. The average execution time for F-DNAES in seconds

Data	Number of Blocks	Time of Encryption(s)
F-DNAES	1024	2.24
DNAES	1000	2.37

In the proposed approach, we took the average of 5 executions of 5 different datasets of Different Algorithms to check the validity of the suggested method. The time for encryption, decryption, and key generation along with the different key sizes of F-DNAES shown in Figure 5, Figure 6, and Figure 7, and depicted in Table 11, and the comparison with Different Algorithms like RSA, DES, AES, ECC, and F-DNAES is shown in Figure 7, Figure 8 and Figure 9:

The F-DNAES scheme demonstrated superior performance compared to RSA, DES, AES, and ECC algorithms in terms of key generation, encryption, and decryption given in Figure 7, Figure 8, and Figure 9. It offers efficient and secure data storage and transmission in cloud environments. The scheme leverages the Cloud Service Provider (CSP) to ensure secure communication and user authentication, protecting sensitive information from phishing and masquerade attacks (Imam et al., 2021). With its robust security measures and compatibility with IoT, F-DNAES is a promising solution for cloud-based data management.

Advantages of the Proposed Methodology

Strengthened Cloud Security:

The proposed methodology significantly enhances cloud security by implementing robust measures to protect data stored and transmitted within the cloud environment. The Cloud Service Provider (CSP) plays a pivotal role in ensuring secure user authentication and establishing trusted communication channels, ensuring data confidentiality, integrity, and availability.

Granular Access Control:

The methodology offers fine-grained access control, granting access privileges only to authorized users authenticated by the CSP. The Data Owner (DO) maintains control over data access rights, allowing specific permissions to be assigned to different users based on their roles and responsibilities.

Efficient Key Generation:

Leveraging DNA-based encryption, the methodology enables efficient and streamlined key generation processes. This ensures secure encryption and decryption operations while optimizing computational resources and minimizing processing time.

Data Confidentiality:

Leveraging DNA-based encryption techniques transforms sensitive data into encrypted DNA sequences, ensuring robust confidentiality. Only the DO possesses the unique key required for decryption, adding an extra layer of security to protect the data from unauthorized access.

Limitations of the Proposed Methodology

Computational Complexity:

The adoption of DNA-based encryption and decryption processes may introduce computational complexity, particularly when dealing with large-scale datasets. Advanced algorithms and hardware acceleration techniques are necessary to address this challenge effectively.

Key Management:

Effective management of encryption keys is crucial to maintain the security of the proposed methodology. Secure storage and distribution of keys among the CSP, DO, and users must be established to prevent unauthorized access and potential key compromise.

Potential Errors in DNA Sequencing:

Although DNA sequencing is highly accurate, errors can still occur, which may impact the integrity and reliability of the encrypted data. Implementing error-correction mechanisms and conducting thorough validation processes are essential to mitigate this risk.

Adoption Challenges:

The successful implementation of DNA-based encryption in cloud environments may present various challenges, including infrastructure requirements, compatibility issues, and integration complexities. Proper planning and strategic measures should be taken to overcome these challenges and ensure a smooth transition.

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

Table 11. The Average time of 5 Dataset executed through F-DNAES.

Data size (bytes)	Time of key generation (sec)	Time of Encryption (sec)	Time of Decryption (sec)
128(Dataset 1)	0.001	0.03	16.37
256(Dataset 2)	0.002	0.09	14.64
512(Dataset 3)	0.001	0.08	17.85
960Dataset 4)	0.001	0.17	14.45
1024(Dataset 5)	0.001	0.17	13.89

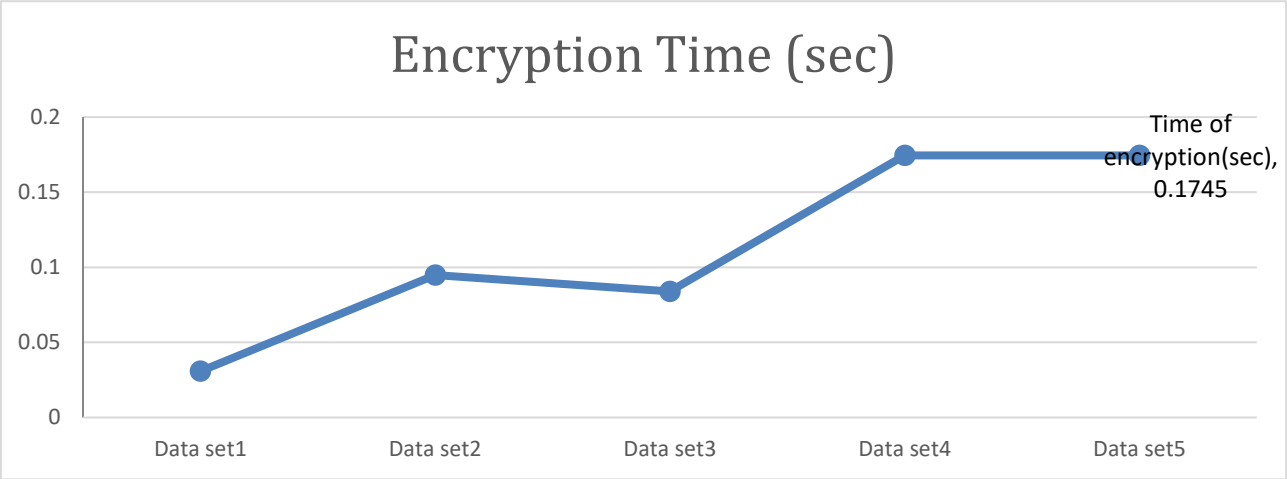


Figure 5. Encryption Time Comparison with Different Data Sizes.

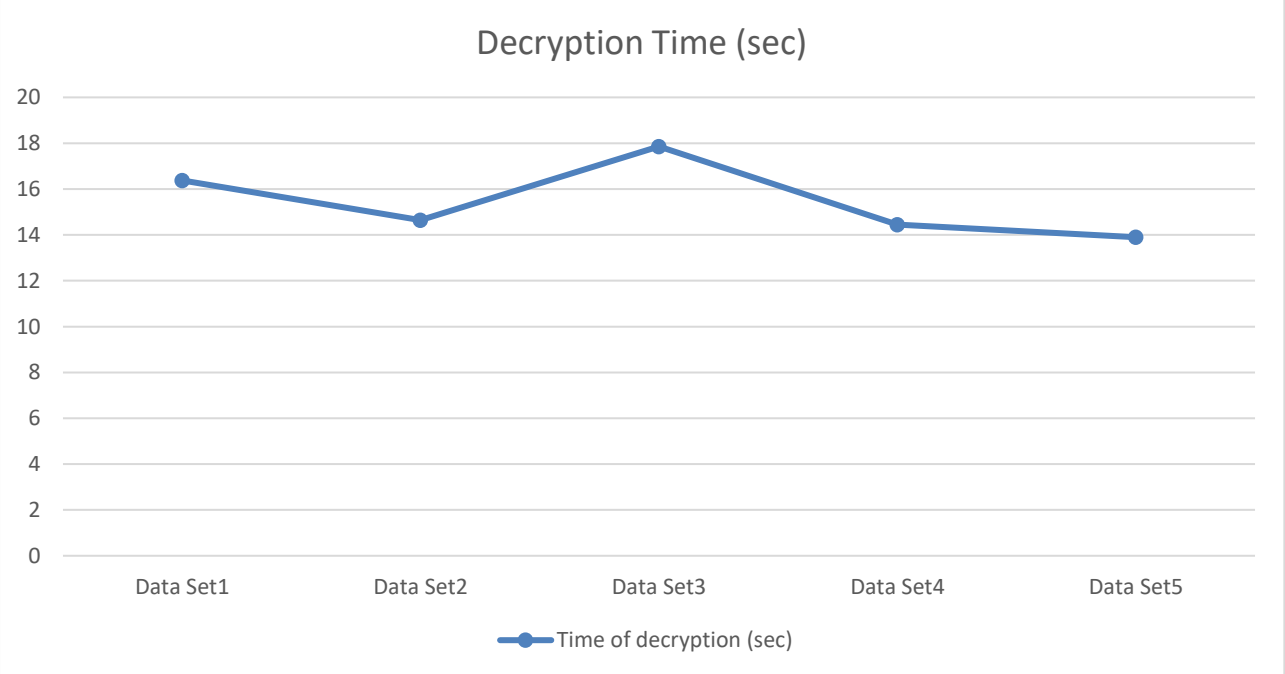


Figure 6. Decryption Time Comparison of F-DNAES with Different Data Sizes.

DOI: https://doi.org/10.52756/ijerr.2023.v32.028

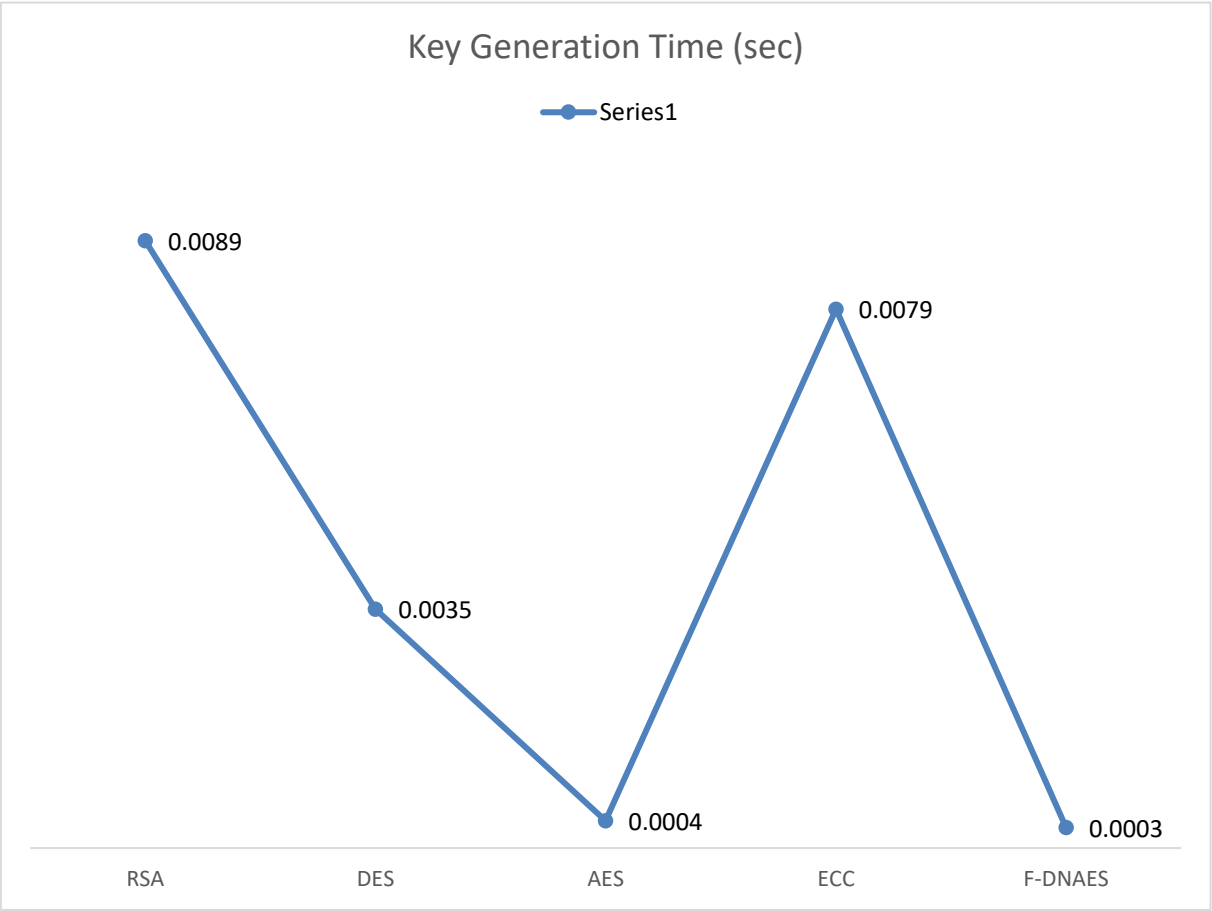


Figure 7. Key Generation Time Comparison of Different Algorithms

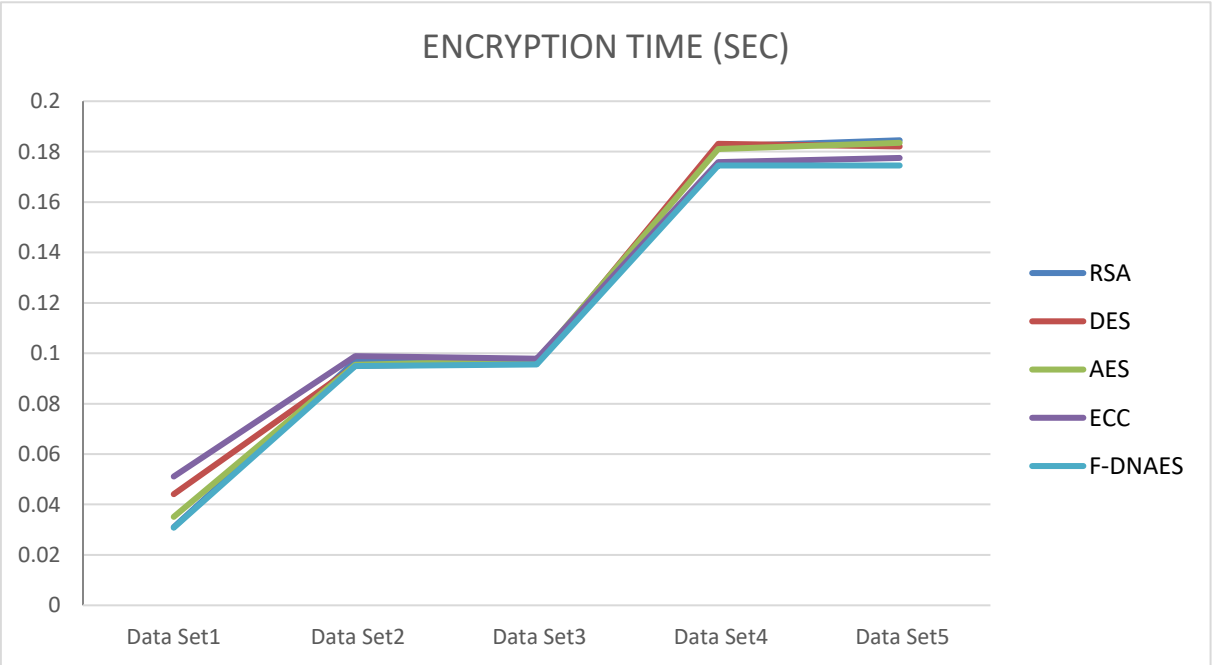
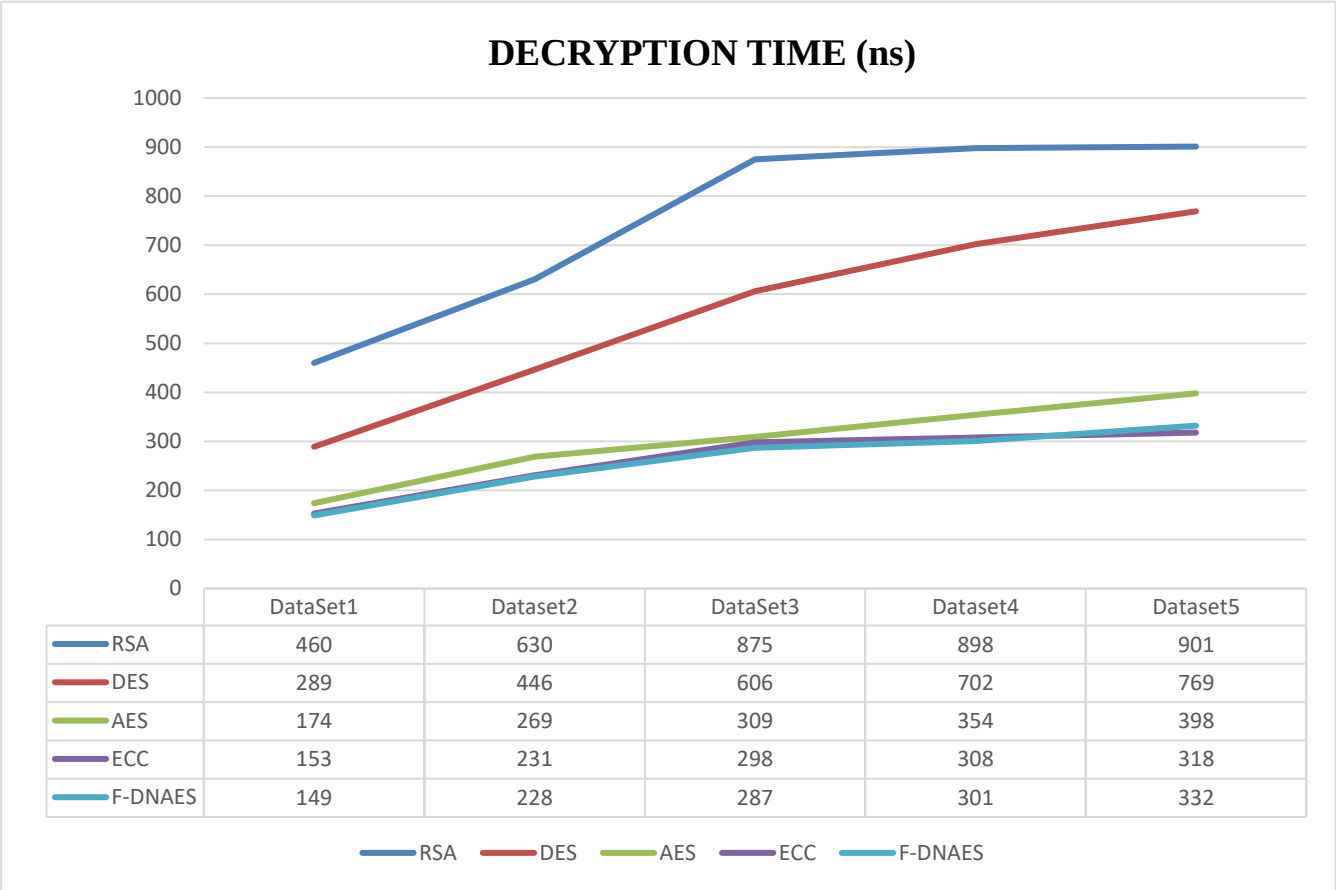


Figure 8. Encryption Time Comparison of Different Algorithms

DOI: https://doi.org/10.52756/ijerr.2023.v32.028



	DataSet1	Dataset2	DataSet3	Dataset4	Dataset5
RSA	460	630	875	898	901
DES	289	446	606	702	769
AES	174	269	309	354	398
ECC	153	231	298	308	318
F-DNAES	149	228	287	301	332

Figure 9. Decryption Time Comparison of Different Algorithms

Overall, the proposed methodology offers substantial advantages in terms of cloud security, granular access control, and efficient key generation. However, it is essential to address the computational complexity, key management, potential sequencing errors, and adoption challenges to fully leverage the benefits of this approach in real-world cloud environments.

Conclusion

When it comes to data security, the integration of DNA cryptography with cloud security principles emerges as a powerful and reliable solution. By leveraging the vast storage capacity of DNA and the robustness of the AES symmetric key technique, the proposed model ensures the secure storage and seamless sharing of information within IoT and cloud-based infrastructures. Extensive testing and analysis have demonstrated promising results, showcasing improved encryption, decryption, and key generation times compared to existing methods such as RSA, DES, AES, and ECC. This innovative approach not only enhances the overall security posture but also addresses the growing concerns surrounding data confidentiality and integrity in the cloud. The collaborative effort between Cloud Service Providers (CSPs), Data Owners (DOs), and users plays a crucial role in implementing and maintaining stringent security protocols to safeguard sensitive data from unauthorized access or malicious attacks. By adopting this approach, organizations can establish a more secure and resilient data storage and transmission environment, ensuring the confidentiality, integrity, and availability of their critical information assets.

Conflict of Interest

The authors declare no conflict of interest.

References

Al-Husainy, M. A. F., Al-Shargabi, B., & Aljawarneh, S. (2021). Lightweight cryptography system for IoT devices using DNA. *Computers and Electrical Engineering*, *95*. https://doi.org/10.1016/j.compeleceng.2021.107418

Bahig, H. M., & Nassr, D. I. (2019). DNA-Based AES with Silent Mutations. *Arabian Journal for Science and Engineering*, *44*(4), 3389–3403. https://doi.org/10.1007/s13369-018-3520-8

Daemen, J., & Rijmen, V. (n.d.). *AES Proposal: Rijndael*.

Di Pietro, R., & Lombardi, F. (2018.). *Virtualization Technologies and Cloud Security: advantages, issues, and perspectives*.

Fernandes, D. A. B., Soares, L. F. B., Gomes, J. V., Freire, M. M., & Inácio, P. R. M. (2014). Security issues in cloud environments: A survey. *International*

Journal of Information Security, *13*(2), 113–170. https://doi.org/10.1007/s10207-013-0208-7

Imam, R., Anwer, F., & Nadeem, M. (2022). An Effective and enhanced RSA based Public Key Encryption Scheme (XRSA). *International Journal of Information Technology (Singapore)*, *14*(5), 2645–2656. https://doi.org/10.1007/s41870-022-00993-y

Imam, R., Areeb, Q. M., Alturki, A., & Anwer, F. (2021). Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status. In *IEEE Access* (Vol. 9, pp. 155949–155976). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2021.3129224

Institute of Electrical and Electronics Engineers. Bangladesh Section, & IEEE Communications Society. (2017). *ICCIT : 2017 20th International Conference of Computer and Information Technology : 22-24 December 2017*.

Institute of Electrical and Electronics Engineers. Bangladesh Section, & IEEE Communications Society. (2017). *ICCIT : 2017 20th International Conference of Computer and Information Technology : 22-24 December 2017*.

Kadhim, A., & Ali, R. S. (2019). Enhancement AES based on 3D chaos theory and DNA operations addition. *Karbala International Journal of Modern Science*, *5*(2). https://doi.org/10.33640/2405-609X.1137

Kandukuri, B. R., Ramakrishna, P. V., & Rakshit, A. (2009). Cloud security issues. *SCC 2009 - 2009 IEEE International Conference on Services Computing*, 517–520. https://doi.org/10.1109/SCC.2009.84

Kaufman, L. M. (2010). *Can Public-Cloud Security Meet Its Unique Challenges?* http://csrc.nist.gov/groups/

Kolate, V., & Joshi, R. B. (2021). An Information Security Using DNA Cryptography along with AES Algorithm. In *Turkish Journal of Computer and Mathematics Education, 12(1), 2021*.

Muttik, I., & Barton, C. (2009). Cloud security technologies. *Information Security Technical Report*, *14*(1), 1–6. https://doi.org/10.1016/j.istr.2009.03.001

Namasudra, S., Chakraborty, R., Majumder, A., & Moparthi, N. R. (2021). Securing Multimedia by Using DNA-Based Encryption in the Cloud Computing Environment. *ACM Transactions on Multimedia Computing, Communications and Applications*, *16*(3s). https://doi.org/10.1145/3392665

Noor, T. H., Sheng, Q. Z., Zeadally, S., & Yu, J. (2013). Trust management of services in cloud environments: Obstacles and solutions. *ACM Computing Surveys*, *46*(1). https://doi.org/10.1145/2522968.2522980

Patnala, B. D., & Kiran Kumar, R. (2019). A Novel Level-Based DNA Security Algorithm Using DNA Codons. Springer Verlag. In *SpringerBriefs in Applied Sciences and Technology*, pp. 1–13. https://doi.org/10.1007/978-981-13-0544-3_1

Pavithran, P., Mathew, S., Namasudra, S., & Lorenz, P. (2021). A novel cryptosystem based on DNA cryptography and randomly generated mealy machine. *Computers and Security*, *104*. https://doi.org/10.1016/j.cose.2020.102160

Rama Devi, K., & Bhuvaneswari, E. (2022). An Enhancement in Data Security Using Trellis Algorithm with DNA Sequences in Symmetric DNA Cryptography. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-022-10102-8

Rosado, D. G., Gómez, R., Mellado, D., & Fernández-Medina, E. (2012). Security Analysis in the Migration to Cloud Environments. *Future Internet*, *4*(2), 469–487. https://doi.org/10.3390/fi4020469

Safaa, H., Adill, S., & Yakoob, A. (2023). *Information Security Using DNA Sequences*. *30(4)*. https://doi.org/10.29196/jubpas.v30i4.4397

Sakr, S., Liu, A., Batista, D. M., & Alomari, M. (2011). A survey of large scale data management approaches in cloud environments. In *IEEE Communications Surveys and Tutorials*,*13*(3), 311–336. https://doi.org/10.1109/SURV.2011.032211.00087

Stallings, W. (2005). *Cryptography and Network Security Principles and Practices, Fourth Edition*.

Tiwari, H. D., & Kim, J. H. (2018). Novel Method for DNA-Based Elliptic Curve Cryptography for IoT Devices. *ETRI Journal*, *40*(3), 396–409. https://doi.org/10.4218/etrij.2017-0220

Varadharajan, V., & Tupakula, U. (2014). Security as a service model for cloud environment. *IEEE Transactions on Network and Service Management*, *11*(1), 60–75. https://doi.org/10.1109/TNSM.2014.041614.120394

Young, K. H., Weisenburger, D. D., Dave, B. J., Smith, L., Sanger, W., Iqbal, J., Campo, E., Delabie, J.,

Gascoyne, R. D., Ott, G., Rimsza, L., Konrad, Mü Ller-Hermelink, H., Jaffe, E. S., Rosenwald, A., Staudt, L. M., Chan, W. C., & Greiner, T. C. (2007). *Mutations in the DNA-binding codons of TP53, which are associated with decreased expression of TRAIL receptor-2, predict for poor survival in diffuse large B-cell lymphoma*. https://doi.org/10.1182/blood-2007-02

Zhang, Y., Xiao, D., Wen, W., & Wong, K. W. (2014). On the security of symmetric ciphers based on DNA coding. *Information Sciences*, *289*(1), 254–261. https://doi.org/10.1016/j.ins.2014.08.005

How to cite this Article:

Beldar Shamshad Bee, Rakesh Kumar Jat and Sufiyan Ahmad (2023). Extraction, isolation and chromatographic estimation of Aloe-Emodin and Physcion from *Cassia Fistula* root. *International Journal of Experimental Research and Review*, 32, 323-339.

DOI : https://doi.org/10.52756/ ijerr.2023.v32.028