JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DUAL TEXT DATA ENCRYPTION AND DECRYPTION USING RIJNDAEL AND DNA CRYPTOGRAPHY

Meghana Harish[1], Dr. Prasanna B T[2]

Department of Computer Science & Engineering, JSS STU, Mysore, India

Department of Computer Science & Engineering, JSS STU, Mysore, India

Abstract: **In today's digital world, computer security is crucial for everything from email transmission to financial transactions. Security is one of the most important aspects of a cloud computing system due to the fact that users' sensitive and important information is stored there. Numerous new encryption methods are developed every day, and a great deal of research is being done to find a trustworthy cryptographic algorithm. In addition to using sophisticated logic and mathematics to create secure techniques of data encryption and decryption, or getting the original data back. With the use of cryptography, a message may be sent between a sender and a recipient while remaining impenetrable to anybody listening in. We need a strong algorithm, a strong key, and a strong idea for the encryption and decryption process in order to achieve this. In this work we have used the two level of encryption and decryption process for storing the data on the multiple cloud and for downloading the data from multiple cloud. First level of encryption is done by using Rijndael Algorithm which produces the base 64 cipher text. Second level of encryption is done by using DNA sequence which take the input of Rijndael Algorithm output. Base64 cipher text is splitted and on the splitted data DNA sequence is applied which produces the cipher text. Cipher text is stored on the multiple cloud service provider. The proposed work adds two layers of security and it is difficult for an attacker to get the original data.**

Keyword: Multi-Cloud, Rijndael Algorithm, DNA Cryptography, Encryption, Decryption.

I. INTRODUCTION

To achieve coherence and economies of scale, cloud computing, like a utility provided through a network, depends on resource sharing. The core of cloud computing is shared services and converged infrastructure. Although cloud computing has many benefits, it also presents a number of security risks and difficulties. Utilizing a ‘‘single cloud” provider is becoming less common as a result of risks such service interruption, data theft, and information leakage. Multiple clouds, sometimes referred to as ‘‘multi-clouds”, “inter-clouds”, or “clouds of clouds”, may be used to actualize this. To increase data accessibility and security, two level of encryption is made on user information, In the first level of encryption on user data Rijndael Algorithm is applied it produces the cipher text, in the second level of encryption cipher text will be divided into smaller chunks utilizing data hiding concepts borrowed from biological DNA sequences. The DNA-encrypted data pieces will then be dispersed across the several Cloud Service Providers (CSP) and also Cross Site Scripting attack is prevented. As a result, this study proposes a viable solution to security and privacy issues with cloud technology.

II. LITERATURE SURVEY

Anwar et al, offer an XOR operation-based approach that encrypts plain text using the symmetric key. Matrix computations and DNA hybridization are utilized to shorten the duration of this method [1]. Siahaan et al, has offered a standard encoding technique. This technique uses Base64 to convert an 8-bit character to a 6-bit one [2]. Karandeep Kaur, proposed a layered approach combining DNA and RSA encryption algorithms. The reference DNA strand from a genetic database serves as the secret key for the DNA encryption [3]. Kalsi et al, proposed the use of deep learning with DNA cryptography to conceal the ciphertext. They have also suggested a technique for producing keys through natural selection [4]. Saha and Haque, proposed encoding to DNA bases was done using a dynamic mapping. To increase its security, they have also implemented procedures like Roll in encoding and data and key arrangement [5]. Ashish Kumar Kaundal et al, this paper gives a brief overview of DNA cryptology along with a new algorithm based on the fusion of symmetric-key cryptography, DNA nucleotides, and XOR operation is proposed [6]. Y. Singh et al, which proposes an SCMCS (secured cost-effective multi-cloud storage) model for cloud computing. According to their findings, customers may make better decisions using this proposed model since it takes into account their available budgets [7]. W. Liu, Cloud computing concepts and characteristics are explored in this article, which introduces a variety of cloud computing platforms and investigates the cloud computing security challenge and its solution.[8]. D. Sureshraj et al, attempts to offer a technique for implementing data hiding in DNA sequences. Data hiding in the cloud is implemented using DNA sequence characteristics. The approach described here is based on complementary pair theory and binary coding [9].

III. PROPOSED WORK

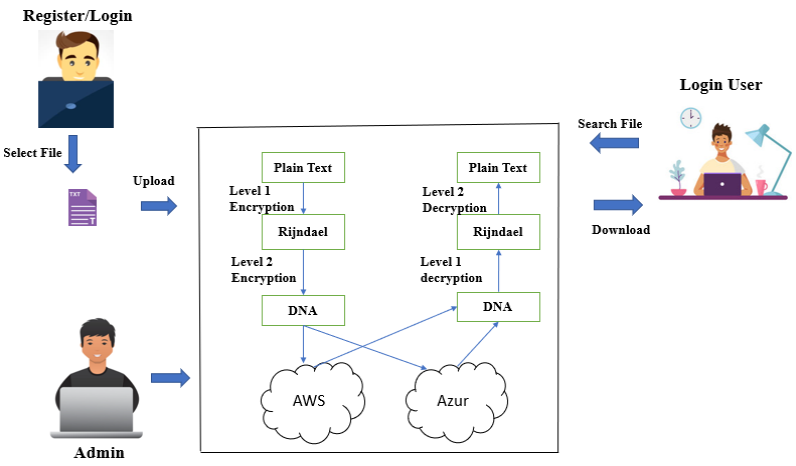

Fig 1. System architecture

The fig 1 is the system architecture is divided in to four models Admin, Client, Rijndael, DNA. Where Admin will login by giving valid name and password and creates a multiple cloud service provider**.** If client needs to store the data in the multiple cloud service provider, first he needs to register by giving user details. Once he registers, he can login and store the data in cloud or he can download the stored data from the cloud. The data which client want to store in the cloud will be encrypted twice. Two Level of encryption is done. First level of encryption is done by using Rijndael Algorithm. Plain text will be passed to Rijndael Algorithm. By using user defined keys, the plain text will be converted to base64. In the second encryption is done by using DNA cryptography. The output of the first encryption is splitted. And the splitted cipher text is passed as the input for DNA cryptograph. After applying DNA sequence on the splitted data the second encrypted data is produced. When Client want to download the data from the multiple cloud decryption process is done, inverse of the above two stages are performed. In the decoding process, the original lossless file is obtained.

Admin: The Admin has the following Modules

- Login: Admin will login by giving valid name and password and creates a multiple cloud service provider**.**
- Validation: The client is validated. If he/she is approved then the admin will assign multiple cloud service provider to the client.

Client: The client has the following Modules

- Register: client get register by giving user details.
- Login: client will login by giving valid username and password.
- Purchase cloud service provider: The client purchases the new cloud service provider or if data is expired of existing client's cloud service provider, the client will send a new request for admin to provide a new cloud service provider.
- Upload: upload the data to cloud.
- Download: download the data from cloud.

Rijndael Managed:

- Plain text will be passed as input data.
- Plain text will be encoded into sequence of bytes.
- Plain text will get encrypted by using user defined keys.
- Once encrypted to binary later data will be converted to base 64.

DNA: The DNA has the following Modules

- Splitting of data: Data is split based on domain.
- DNA Encryption: The data should be encrypted. The encrypted data should be sent to the cloud service provider.
- DNA decryption: The data that has been uploaded on the service providers should be combined. Then the combined data should be decrypted before downloading the data**.**

Flow Chart Diagram for Admin and Client

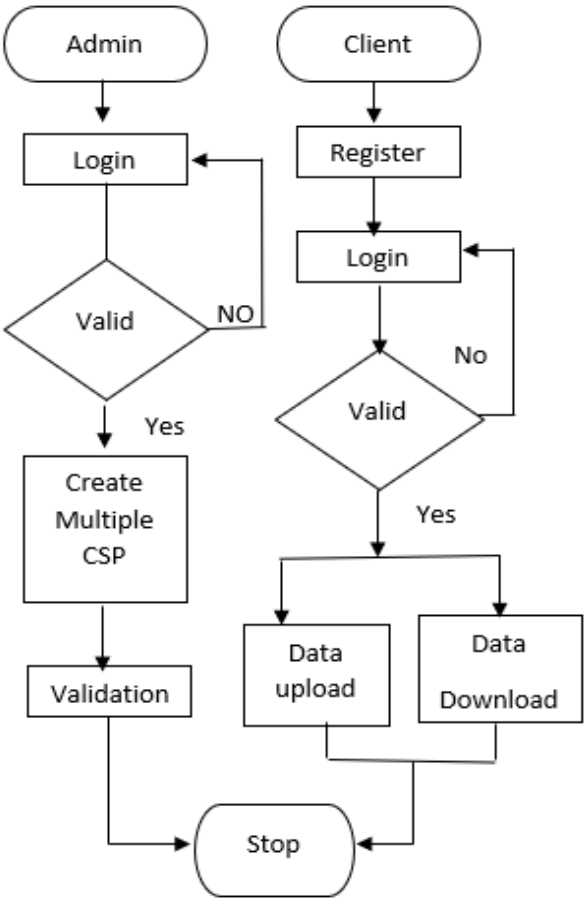

Fig 2: Flowchart Diagram for Admin and Client

In the fig 2 admin will login by giving the login credential and he will create multiple cloud service provider. Client will register by giving user details in order to store the data in the cloud or to download the data from cloud. Once client get registered, he can login by giving the credential. After the login, client can store the data in the cloud or he can download the data from cloud.

Flow Chart Diagram for Encryption

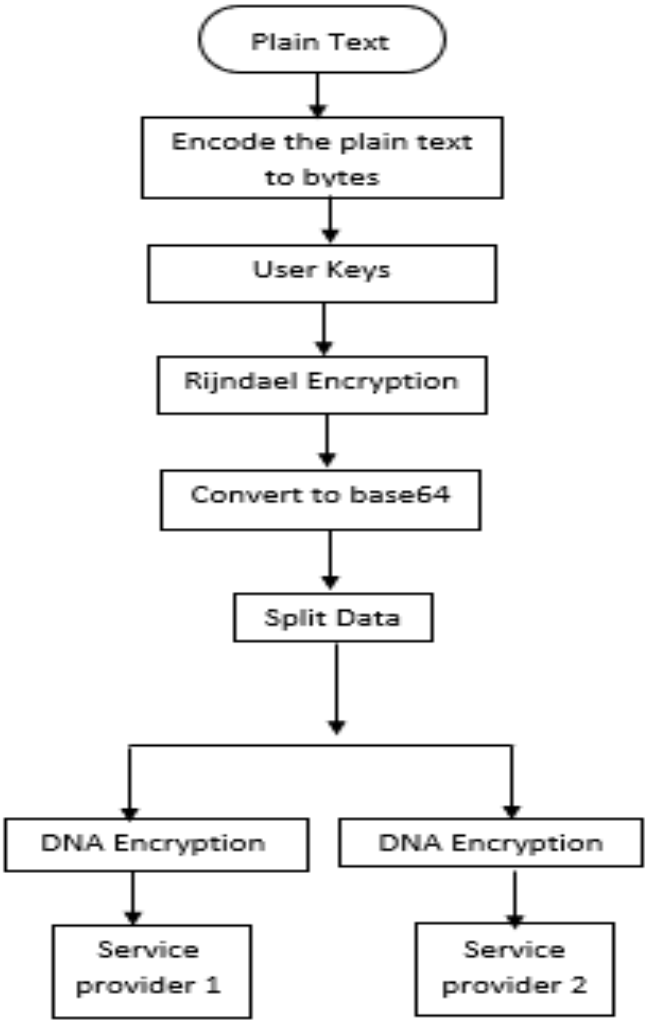

Fig 3: Flowchart Diagram for Encryption

In the fig 3 process of encryption, user data (plain text) will be converted to bytes. User defined keys will be passed to Rijndael Encryption method which will produce the encrypted data. Encrypted data will be converted to base64. Encrypted base64 will be splitted as a small junk. On the splitted data DNA encryption is applied. Then the encrypted data will be stored on the multiple cloud service provider.

Flow Chart for Decryption Process

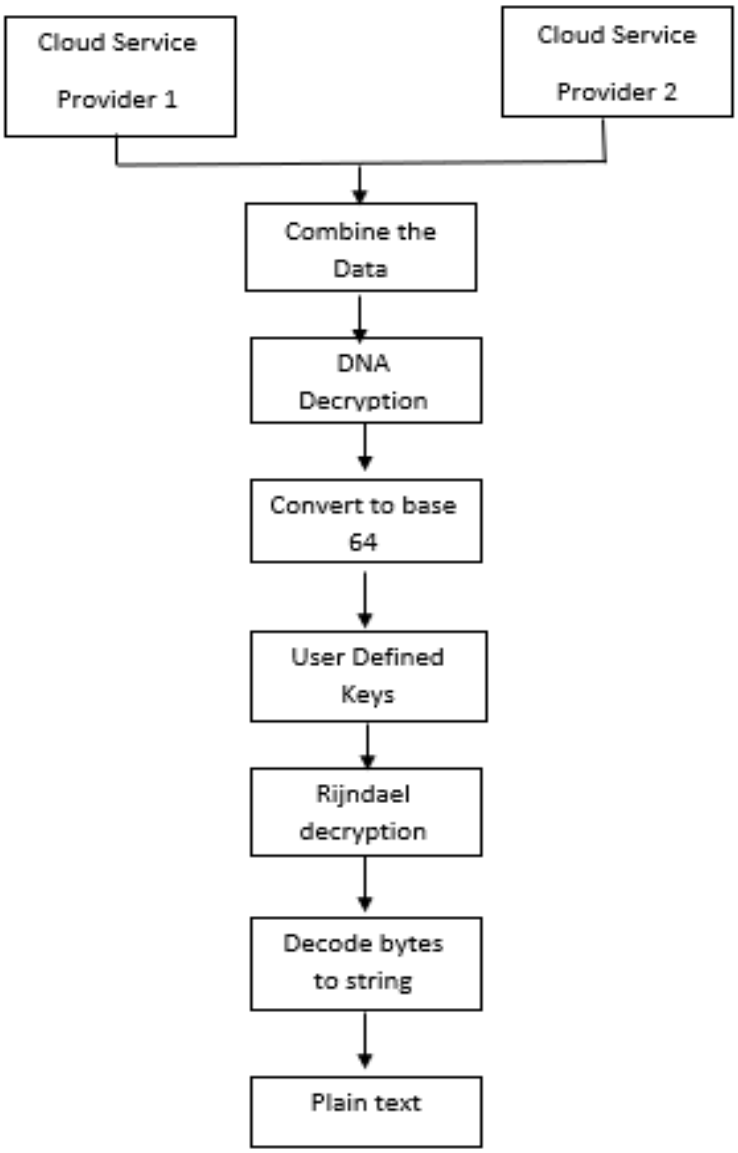

Fig 4: Flow Chart for Decryption Process

In the fig 4 process decryption, Encrypted data stored on the multiple cloud service provider is combined and the combined data will be passed to DNA cryptography for the decryption process. The cipher text obtained from the DNA sequence will be converted to base64. By using user defined keys, the base64 data will be decrypted by the Rijndael Algorithm. The output obtained be Rijndael algorithm will be decoded from bytes to string. And finally, the user stored data is obtained.

The system goes through 3 different applications, they are:

1. Rijndael Managed
2. DNA Based Cryptography
3. Cross Site Scripting

1. Rijndael Managed

Rijndael is used for data encryption when using the AES method (pronounced rain-dahl). Because of its higher level of security, the National Institute of Standards and Technology (NIST) chose it to replace the outdated and dangerous Data Encryption Standard (DES).

Token cypher Being an iterated block cypher, Rijndael indicates that a block of data is encoded and decoded again, round after round. Depending on the block size, encryption keys might be 128, 192, or 256 bits long.

Working of Rijndael

Using Rijndael, a sequence of matrix transformations or rounds is used to encrypt data. Each key or block size has a different number of rounds:

- 128 bits = 9 rounds
- 192 bits = 11 rounds
- 256 bits = 13 rounds

XOR and byte-by-byte substitution are all part of the Rijndael algorithm. The steps are as follows:

- It creates 10 128-bit keys from the 128-bit key, which are kept in 4x4 tables.
- The plaintext is broken into 128-bit tables, each of which is partitioned into 4x4 columns.
- A configurable number of rounds is performed on each piece of 128-bit plaintext, as described above. After the 10th round, the code is generated.

Each round consists of four steps:

- **Byte Sub:** The S-box contains a substitute for each byte in the block.
- **Shift Row:** The first 16 bytes of a block of bytes are arranged in a rectangle and can be moved around the block based on the size of the block.
- **Mix Column:** Each column is multiplied by the matrix in this example of matrix multiplication. Polynomials, not numbers, are used to multiply bytes. XORing the binary 9-bit string 100011011 with the result wipes out any extra bits in the result. In cyclic redundancy checks, a similar technique is used.
- **Add Round Key:** The current round's subkey is XORed here.

The security of Rijndael is considerably increased when it is done several times with different round keys.

Rijndael Managed Class in .NET: Key lengths of 128, 192, or 256 bits are supported by this formula, with 256 bits being the default. This formula supports block sizes of 128, 192, or 256 bits in the.NET Framework, with 128 bits being the default.

Algorithm for Rijndael Encryption

1. Plain Text.
2. Convert the plain text to bytes.
3. User defined Keys.
4. Passing the keys and plain text data bytes to encryptor.
5. Create Memory stream which will be used to hold encrypted data.
6. Create Cryptographic stream always write mode for encryption.
7. Start encrypting.
8. Finish encrypting.
9. Close the streams.
10. Convert encrypted data into a base 64 encoded String.

Algorithm for Rijndael Decryption

1. Encrypted Text.
2. Convert the encrypted text to base64.
3. Passing the keys and base64 encrypted text to decryptor.
4. Create Memory stream
5. Create Cryptographic stream.
6. Start decrypting.
7. Stop decrypting.
8. Close the stream.
9. Convert decrypted data into a string.
10. Plain Text.

2. DNA Based Cryptography

Secure data transmission across open networks, such as the Internet, is made possible by the Data concealing technique. Complementary and binary coding principles will be used to encrypt the data in DNA. In order to hide the created cipher-text, the DNA reference sequence will be inserted. Adenine (A), Cytosine (C), Guanine (G), and Thymine (T) are the four significant influences on the success that make up the DNA double helix structure in the real world (T). Generate a DNA sequence based on any combination of these four nucleotides The basic synthesis of these nucleotides in a true natural environment is described in the following constant universal laws.

Rules for Watson-Crick Base Pairing

The Pyrimidine Thymine is paired with the Purine Adenine (A) (T).

The Purine Guanine is paired with the Pyrimidine cytosine (C) (G).

- **Base pairing rule:** Changes will be made to these universal rules in order to increase complexity and make it more difficult for an attacker to acquire access. A is synthesized to T in biology, but we may suppose A to C, T, or G or anything else and so on. Thus, possible number of such base pairing rules become 4×3×2×1 = 24. The attacker's odds of making a precise guess are thus 1 in 24. Nucleotide Synthesis in the Real World.
- **Binary coding rule or Complementary pairing rule:** Using this method, you can generate a DNA sequence from binary data. Consider T = 00, A = 01, G = 10, and C = 11. Additionally, each time this is run it will be different such that C can be 00 in the next run or it might be G or A or T. This means that there are four fundamental nucleotides, and the ultimate number of potential rules is 4x3x2x1=24 possible rules. As a result, the chance of having it correctly are 1 in 24.
- **Generate DNA reference sequence:** Using the EBI online database, which has over 163 million distinct DNA sequences, is one method of directly selecting a DNA reference sequence. Although we will construct a DNA sequence by randomly shifting four nucleotides, we will do it in order to attain more security. As a result, the shift-key range of one to sixteen will be generated at random by the system. The technique can generate 185 million unique DNA reference sequences, which is more than the EBI database's more than 163 million.

DNA Sub Part 2 Key	DNA Sub Part 2 Value
T	C
G	T
C	A
A	G

Fig 5: Base Pairing Role

Binary Digits	DNA Base
00	A
01	T
10	C
11	G

Fig 6: Binary Coding Rule

Character	ASCII Value (Decimal value)	Binary Value	DNA Sequence of Character
1	49	00110001	AGAT
a	97	01100000	TCAA
A	65	01000001	TAAT

Fig 7: DNA Sequence for the Characters

DNA Sequence Key	DNA Sequence Value
AA	1
AT	2
CC	3
CG	4
CT	5
GA	6
CA	7
AC	8
TT	9
GT	10
TC	11
AG	12
GG	13
TA	14
GC	15
TG	16

Fig 8: DNA reference sequence

Algorithm for DNA Encryption

1. M is the Cipher text obtained by the Rijndael Algorithm.
2. M will be splitted as a small junk M1 and M2.
3. M1 will be converted to ASCII.
4. M1 ASCII will be converted to Binary Number.
5. Apply Binary coding rule. (Fig 6)
6. Output of rule execution is M1' = DNA sequence (Binary data converted to DNA nucleotides).
7. Apply base pairing rule. (Fig 5)
8. Get M1''= new form of M1'.
9. Find index of Nucleotides in DNA reference sequence. (Fig 8)
10. Get M1''' = Cipher text will be stored on the cloud

Same method is applied on M2 data.

Algorithm for DNA Decryption

1. M1''' and M2''' are the cipher text stored on the multiple cloud and combine the M1''' and M2''' = M'''.
2. M'''= Cipher text.
3. Find Index of Nucleotides in DNA reference Sequence. (Fig 8)
4. M'' = Previous Form of M1'.
5. Apply base pairing Rules in reverse way (Fig 5)
6. Get M'= DNA Sequence.
7. Convert M' to binary using binary coding rule. (Fig 6)
8. Get M is a binary number.
9. Convert the Binary number to ASCII code.
10. ASCII code will be converted to character (Cipher text)
11. Cipher text will be passed to Rijndael Algorithm.

3. Cross Site Scripting

Malicious scripts can be inserted via the cross-site scripting technique onto otherwise reliable websites (XSS). An XSS attack happens when a hacker uses an online application to deliver malicious code to another end user, generally in the form of a browser-side script. These kinds of attacks are possible because many online apps include user input without validating or encrypting the data.

An attacker can use XSS to send a malicious script to a careless user. The script will be executed by the end user's browser even if it shouldn't be trusted. Since it thinks the script is coming from a trustworthy source, the malicious script can access any cookies, session tokens, or other sensitive data kept by the browser and utilized with that site. These programmes have the power to totally alter the content of HTML pages.

Cross-Site Scripting (XSS) attacks occur when:

- Data entry into a Web application typically occurs from an unreliable source, such as a web request.
- A web user is provided with the information as dynamic content that hasn't been scanned for viruses.

Prevent Cross-Site Scripting (XSS) in ASP.NET Core: A security weakness called cross-site scripting (XSS) enables an attacker to introduce client-side programmes, frequently JavaScript, into online pages. The scripts of the attacker will run when subsequent users visit the vulnerable sites, giving the attacker the ability to steal cookies and session tokens, alter the contents of the website via DOM manipulation, or reroute the browser to another page. Generally speaking, XSS flaws emerge when a user enters data into an application and the programmed delivers that data to a website without validating, encoding, or escaping it.

Anti-Cross Site Scripting Library: With its Security Runtime Engine, AntiXSS protects your current apps from cross-site scripting assaults while also protecting your historical applications. AntiXSS includes elements that have been drastically and innovatively rethought, providing you with a fresh, more effective weapon against the commonly used cross-site scripting (XSS) assault. AntiXSS provides you with:

- Increased performance. AntiXSS has been fully redesigned with performance in mind, while still providing the essential XSS security that you have come to rely on for your apps.
- Safeguard Globalization. The internet is a global marketplace, and cross-site scripting is a worldwide problem. An attack may be coded in any language, and Anti-XSS now protects against XSS assaults written in hundreds of them.

- Adherence to standards. AntiXSS is written in accordance with current web standards. You may secure your web application without altering its user interface.

Correct output encoding and input validation will resolve the XSS issue. Use AntiXSS Library's extensive encoding features for output encoding. AntiXSS works by inspecting all of the characters in the input and encoding those that are not on the whitelist.

Antixssencoder HtmlEncode: Encodes the provided string used as text in HTML markup, with the option of using HTML 4.0 named entities. HtmlEncode (String, TextWriter) encodes the supplied string used as text in HTML markup and outputs it using the text writer specified.

Syntax

public static string HtmlEncode (string input, bool useNamedEntities);

AntiXssEncoder.HtmlEncode(TxtUserName.Text,true);

When a malicious attacker passes the script code in the user input field an error message is popped up so further the attacker can’t pass through the application and attacker will be blocked.

IV. RESULT

Plain Text

Cryptography is the science of protecting information by transforming it into a secure format. This process, called encryption, has been used for centuries to prevent handwritten messages from being read by unintended recipients. Today, cryptography is used to protect digital data. It is a division of computer science that focuses on transforming data into formats that cannot be recognized by unauthorized users.

Rijndael Encryption (Level 1 Encryption)

NI4Y54f2C2XqZbWmT88sw0r4zgHbkD61aahZ6X+Lx8DQB8tAdLIJcao4rozi/k2BUPZnzdl1aiCscQ5ELsjbQ8I
v15YVKrfYchixHfwcNwlKvToIUjf2PuWW7hGl2q7uD6kxcRgUDiddiIX23LTzKPkRNwNhjd84/kd72BT2DEt
Ve6eLpykwGg9ud3WrvHrcLe4hfnUXy5qJdO8wl7Lwnx0NHD2om6lx6fHHoow6xP8xZ13Sm6ZeZQFjea4O/cJ3
5lSjKoRnt2HB3Y2DEFhScgmrvoiKwxnTpA7sHKFhNWeN4+meegqqx0ouPpy4flRiEfNMjsLTniniVGuh26026
BWnqwTg3DlZDpOgpBqCBzdKAx4L5TmGagfWHXvbfNUlYPI97cxlbcXyMFoQ9+3eKxbnRxk2SyUdqsAdE
GzggEf5pTvmWaAwUhzs9u9YTEdMd3nVr6d3NBb5dyIBIE/WvQ9vXhfb1+v/vjA8cVExJPwK+7CKyFTqE8rjz
GTy3zbzo47ZtDuzTPSy6Mgdrab13GZ1FodtENaFT0B1ymFRdKg=

DNA Encryption (Level 2 Encryption)

414481043810310477106410106312515317635711341012101251055101356104517541276724410710157157
157123110731262416512101244315461012544157441648417107153141045631451786972106463331331714517
747161015715784105107103151034341651071763151012485710151033837425677387107127851241277557 1
041455716425734314483371771063135335351057124571610651510553441077251271036753344787474784 8
312106101041634514231372364145541471271741012104697274105106463410644435437731077341651358 7
255457510853741010355657412567104167310471277714333125810351541745161012557161054165571451 2
101341441244106314711107716512107774124123143145510751231310125123110151010310711107317331 3
154771737151045166971041101010371631071423143671454106412461010381064443477123107107 5

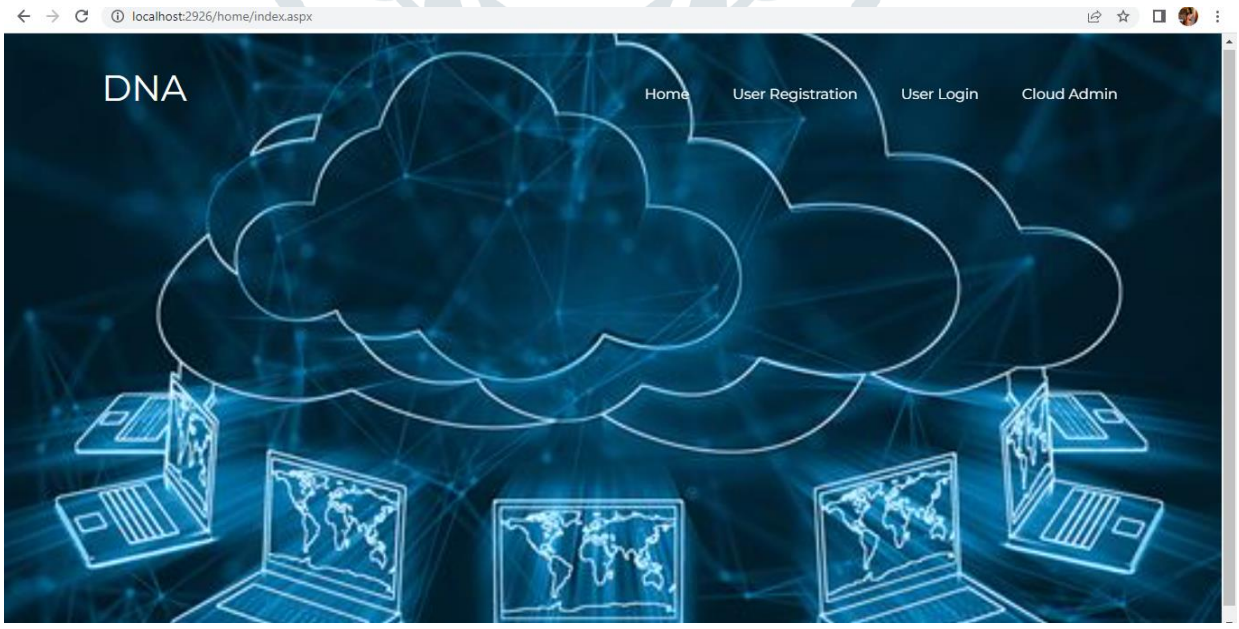

Fig 9 :Home Page

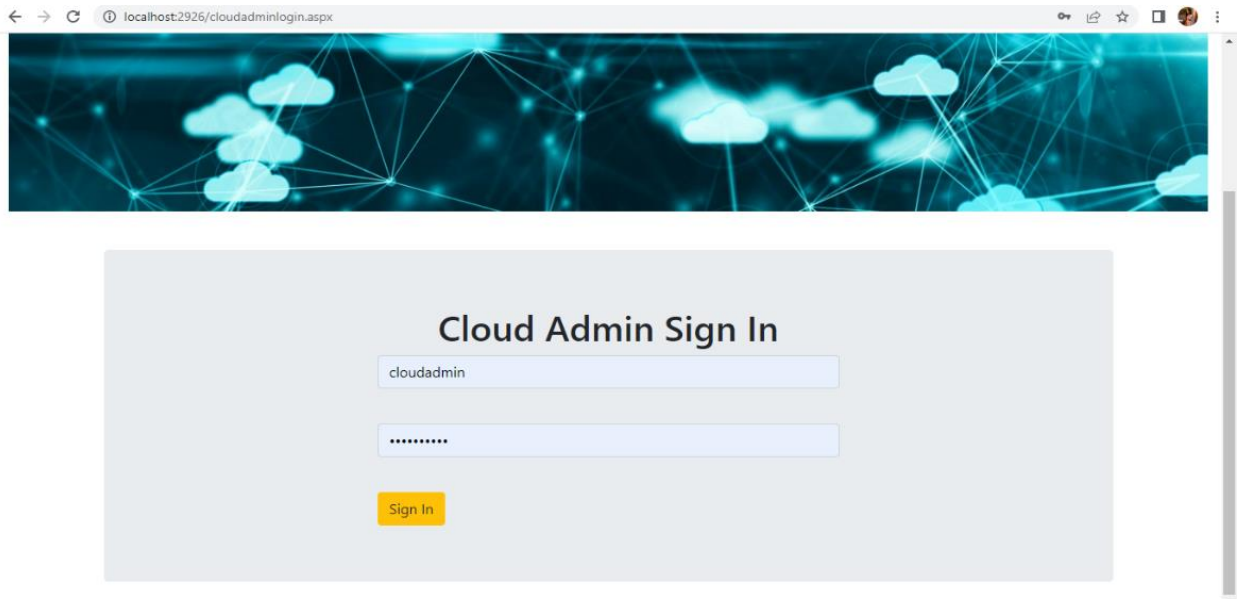

Fig 10 : Cloud Admin Sign in page

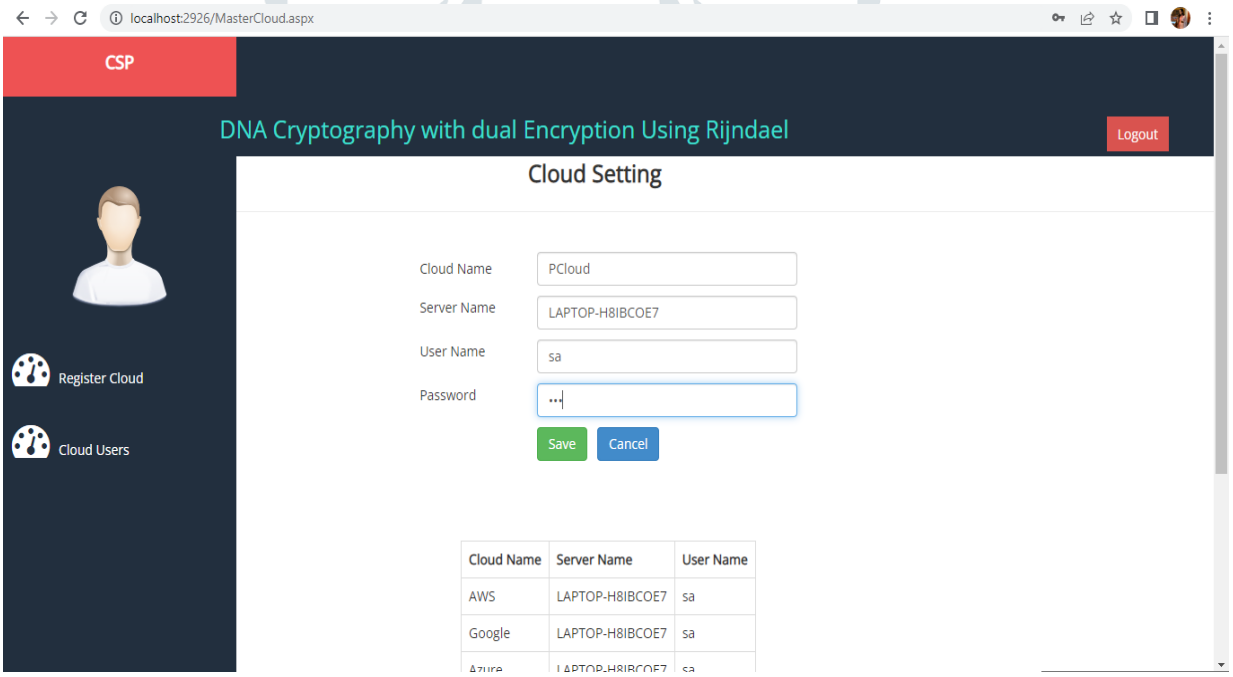

Fig 11 :Admin Cloud Creation Page

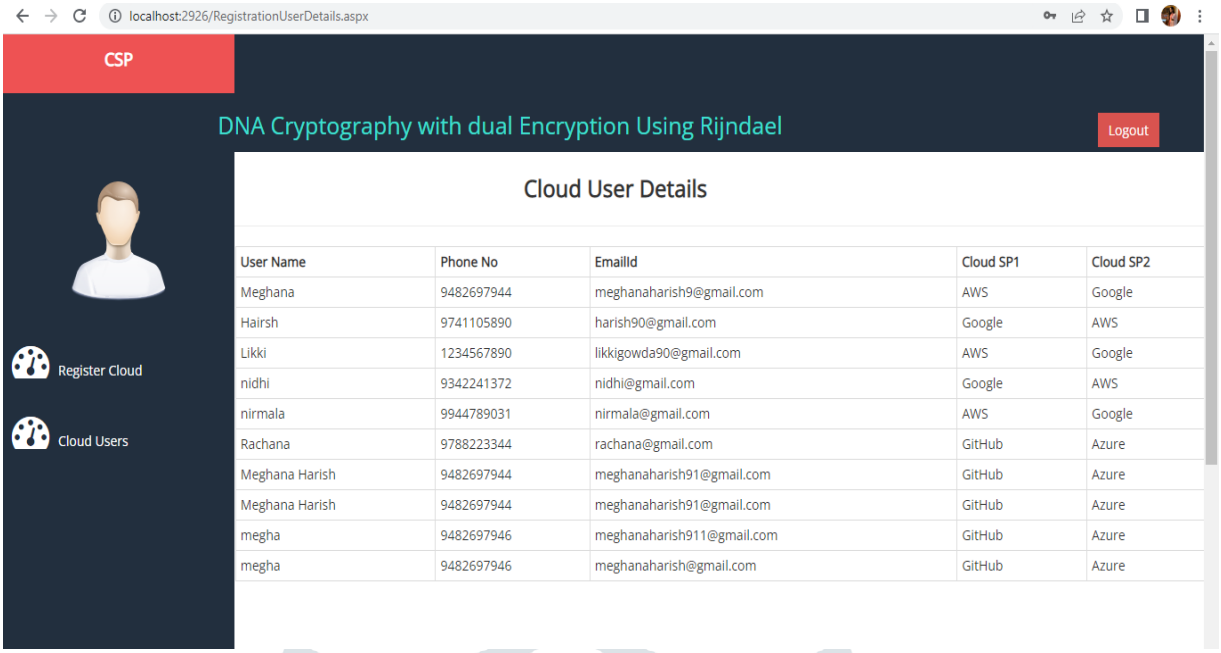

Fig 12: Admin View the Cloud Users Page

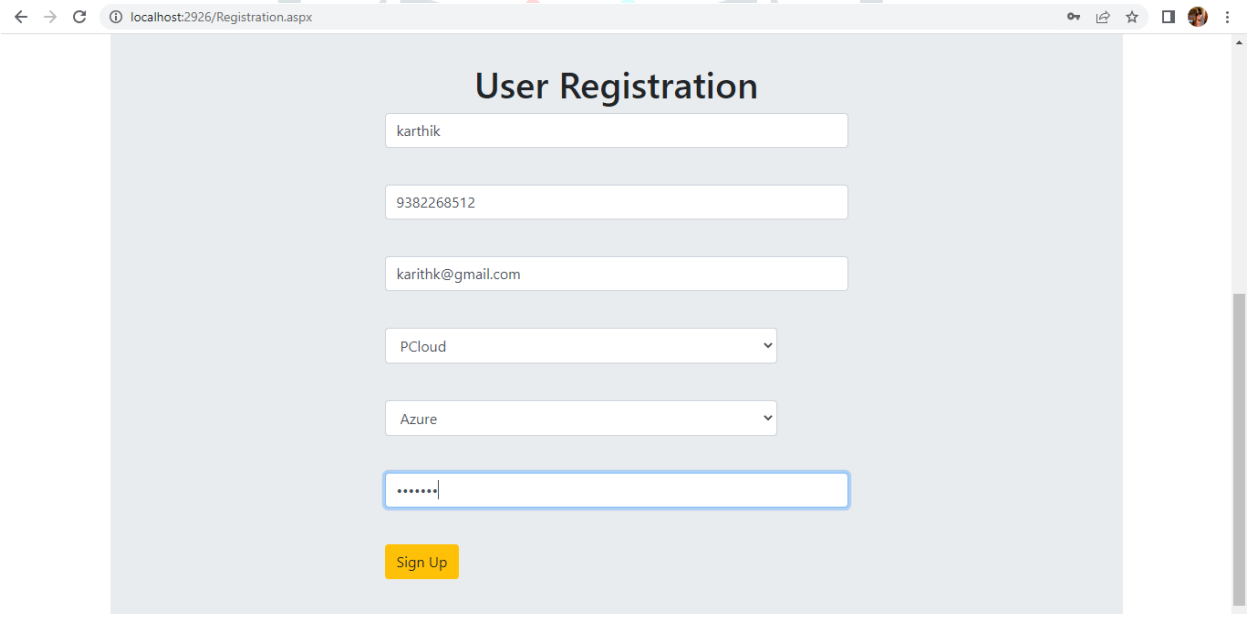

Fig 13: User Registration Page

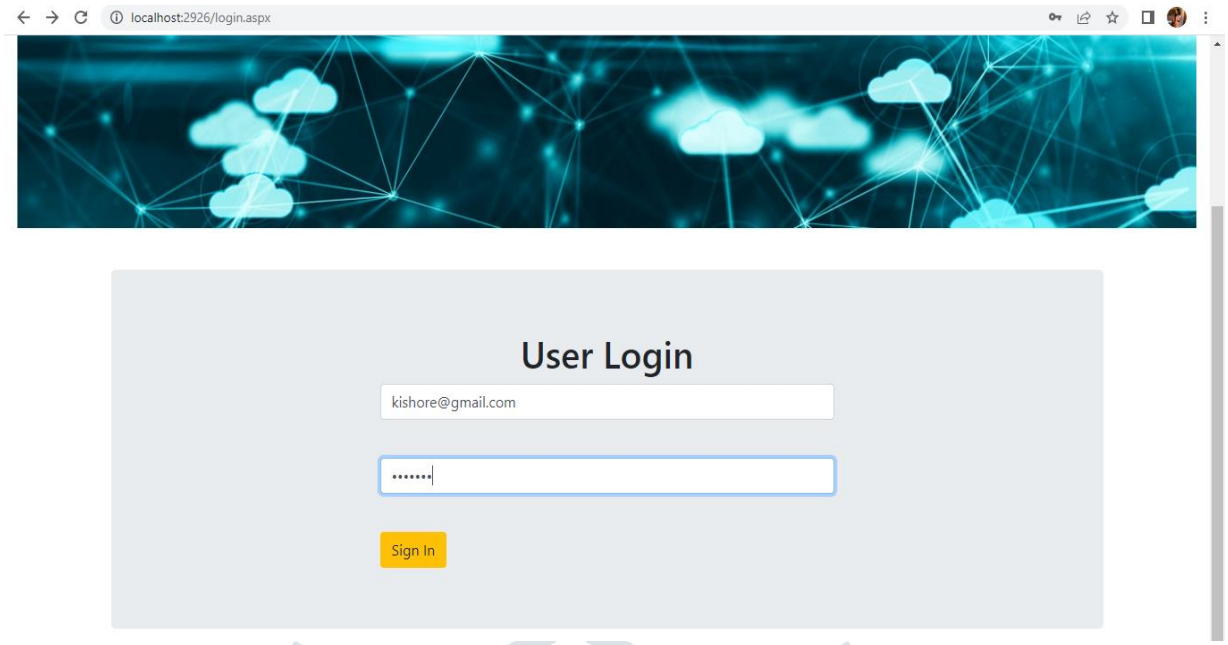

Fig 14: User Login Page

Fig 15: User Uploading the Confidential Data Page

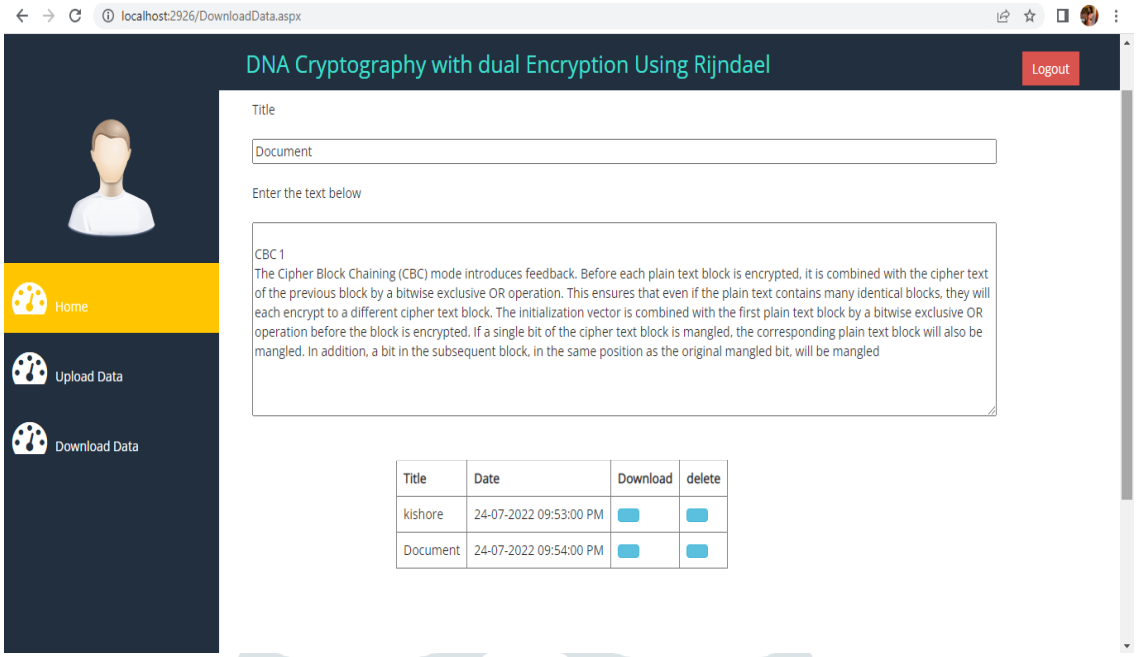

Fig 16: User Downloading the Confidential Data Page

Number of Words	Encryption Time(ms)
50	0.06
100	0.09
500	0.14
1000	0.38

Fig 17: Number of Words and Encryption Time(ms)

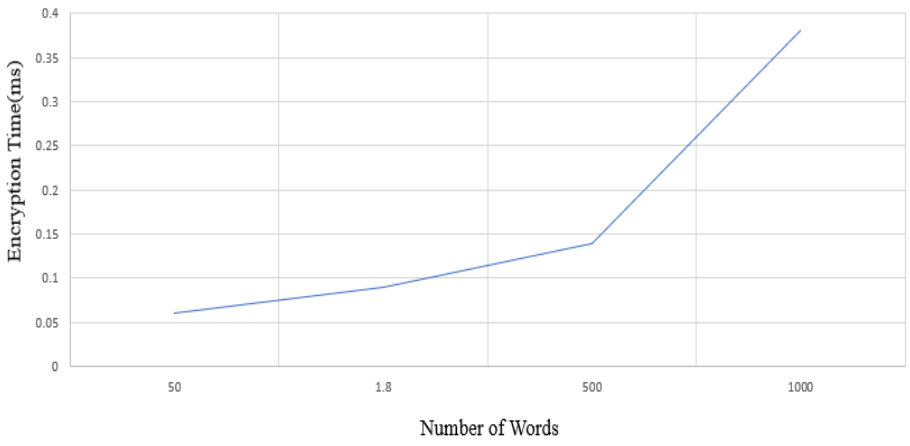

Fig 18: Line Plot

The time required to encrypt the data after Rijndael and DNA Sequence is compared with time required to complete encryption process and store data in cloud as number of words increase the time required for encryption also increase.

V. CONCLUSION AND FUTURE WORK

The proposed work employs the Rijndael algorithm and DNA Cryptography at two levels of encryption and decryption to store data in the cloud and download data from multiple cloud service providers. This work's key contribution is to raise the information security level to a high degree. When the Rijndael algorithm is used to apply the first level of encryption to plain text, base 64 encrypted data is created. In the second level of encryption, the encrypted data will first be divided, and DNA cryptography will be used on the divided data. This establishes the proposed work's cryptographical security and adds two additional levels of protection. The original data is hard for an attacker to obtain. It is difficult to successfully decrypt information without knowledge of the Rijndael Algorithm using user-defined passwords if the DNA sequence is known to any of the attackers. Additionally, the Cross-Site Scripting (XSS) attack, which allows an attacker to embed client-side scripts into web pages, is stopped. Therefore, customers won't experience data theft, data leakage, or the possibility of a hostile insider attack leading to the loss of their private information. The application of this technique to picture, video, audio, and soon is its future potential.

REFERENCES

[1] T. Anwar, A. Kumar, S. Paul, "DNA cryptography based on symmetric key exchange," International Journal of Engineering and Technology (IJET'15), June 2015, vol. 7, no. 3, pp. 938-950.

[2] Isnar Sumartono, Andysah Putera Utama Siahaan, Arpan, "Base64 character encoding and decoding modeling", International Journal of Recent Trends in Engineering & Research (IJRTER), December-2016.

[3] Karandeep Kaur, "A Double Layer Encryption Algorithm based on DNA and RSA for Security on Cloud", International Research Journal of Engineering and Technology (IRJET), March-2016.

[4] Kalsi, Shruti, HarleenKaur, and Victor Chang, "DNA Cryptography and Deep Learning using Genetic Algorithm with NW", Journal of Medical Systems, December-2017.

[5] Haque, Rejwana, and RoniSaha, "A novel Rolling based DNA Cryptography", Journal of Bioinformatics and Genomics, April-2014, Vol.1, No.3, pp.1-6.

[6] Ashish Kumar Kaundal and A. K. Verma, "Extending Feistel structure to DNA Cryptography" Journal of Discrete Mathematical Sciences and Cryptography, April-2014, Vol.18, pp. 349-362.

[7] Y. Singh, F. Kandah, and W. Zhang, "A Secured Cost-effective Multi-Cloud Storage in Cloud Computing", IEEE Workshop on Computer Communications and Cloud Computing, April–2011, pp. 619 – 624.

[8] W. Liu, "Research on Cloud Computing Security Problems and Strategy", IEEE conference on Consumer Electronics, Communications and Networks, April-2012, pp. 1216 – 1219.

[9] D. Sureshraj, and V. Bhaskaran, "Automatic DNA Sequence Generation for Secured Cost-effective Multi-Cloud Storage", IEEE conference on Mobile Application Modeling and Cloud Computing, December–2012, pp. 1 – 6.