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

NEW DATA HIDING METHOD BASED ON DNA AND VIGENERE AUTOKEY

[1]P Gayatri, [2]Maddimsetti Sri Chaitanya
[1]Assistant Professor, [2]MCA 2nd year
[2]Master of Computer Applications,
[2]Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

ABSTRACT
Security play's major role at the time of data transmission. To provide better security to data. a distinct type of hiding approach is adopted in this proposed system that is DNA sequences based on cryptographic method called Auto Key cipher[1]. The proposed mechanism hides the secret message via converting it along with the key to DNA sequence and then applying the Auto Key cipher using a special table created with DNA-XOR operator to increase the security of the proposed mechanism. So, it can meet the requirements such as high embedding[2] capacity and reasonable level of security.

INDEX TERMS: steganography, cryptography, DNA-based steganography, authentication, concealment strategy, Vigenere Autokey, DNA sequences, DNA-XOR operator, embedding capacity, visual imperceptibility, security

I.INTRODUCTION

Data concealing is one of the methods that has been created recently in tandem with technological advancements. Data hiding is a method for concealing information that is only known to the sender and recipient. So, only someone who possesses the stego key can access the secret data. Data can be concealed in seemingly benign multimedia files like images, audio signals, videos, etc. as long as the size of the cover media is not exceeded during transmission. The receiver must know the stego key and use the embedding technique[3] in reverse order at their end in order to access the concealed data (Leier, Richter, Banzhaf, & Rauhe, 2000; Tur, Lin, Lee, & Tao, 2012). for uses involving intellectual property. Most secret data transmission research in recent years has concentrated on data masking strategies to fend off harmful attempts and provide a secure transfer. Whereas images are the most popular form of data concealing, it is challenging to conceal a secret message within an image without altering the original image noticeably (Shiu et al. 2010).

II.SYSTEM ANALYSIS

The proposed project introduces a novel data hiding method that combines DNA sequences and the Vigenere Autokey cryptography technique. The system analysis reveals that this approach offers a unique and robust concealment strategy for sensitive data. By leveraging the inherent characteristics of DNA sequences, which exhibit minimal differentiation between genuine and fabricated sequences, the method ensures a high level of security. The process involves converting the secret message and the key into DNA sequences, which are then used to implement the Autokey cipher with the aid of a custom table that employs the DNA-XOR[4] operator for enhanced security. The system demonstrates several desirable attributes of a good steganographic system, including a high embedding capacity, ensuring that a significant amount of data can be concealed, good visual imperceptibility, enabling the hidden information to remain undetectable, and an acceptable level of security, safeguarding the concealed data from unauthorized access or usage. Visual invisibility, which allows the concealed data to stay unseen, and a respectable degree of security, which protects the concealed data from unauthorised access[25] or use.

II.A)EXISTING SYSTEM

The current method of data concealment often makes use of audio, video, and image files that have been steganographically altered. These techniques entail hiding information inside the file itself and use subtle media alterations to do so. Even though these methods are common, scientists have discovered that DNA has the ability to be a channel for secret data storage or authentication. However, DNA-based steganography systems[5] now in use have their limitations, necessitating the development of fresh strategies that make use of the special properties of DNA sequences. The standard encryption methods found in cryptography are frequently unable to be utilised successfully by the present methods. A more sophisticated method that combines DNA sequences with powerful cryptography algorithms[6], such as the Vigenere Autokey cypher to increase the steganographic process's capacity and security.

II.B)PROPOSED SYSTEM

It is simple to identify a DNA sequence's unique quality from genuine DNA sequences. In other words, the distinction between a genuine DNA sequence and a false one is almost non-existent [. In order to create an injective mapping, this work has taken advantage of this characteristic in addition to the complimentary rule[7] that has been stated by (Shiu et al. 2010). In other words, a complement, designated as C(x), is given to each letter x. For instance, we may use the complimentary rule that reads as follows: Where C(A) = T, C(C) = A, C(G) = C, and C(T) = G, we have (AT)(CA)(GC)(TG).There are six complementary legal rules: (AT)(TC)(CG)(GA), (AT)(TG)(GC)(CA), (AC)(CT)(TG)(GA), (AC)(CG)(GT)(TA), and (AG)(GT)(TC)(CA).The next stage allows for the demonstration of the suggested plan.

III.SPECIFICATION

III.A)HARDWARE REQUIREMENTS (Minimum Requirement)

1.Processor : Intel i3 & Above
2.RAM : 4 GB
3.Hard disk : 500 GB
4.Keyboard : Standard 102 Keys
5.Mouse : Standard

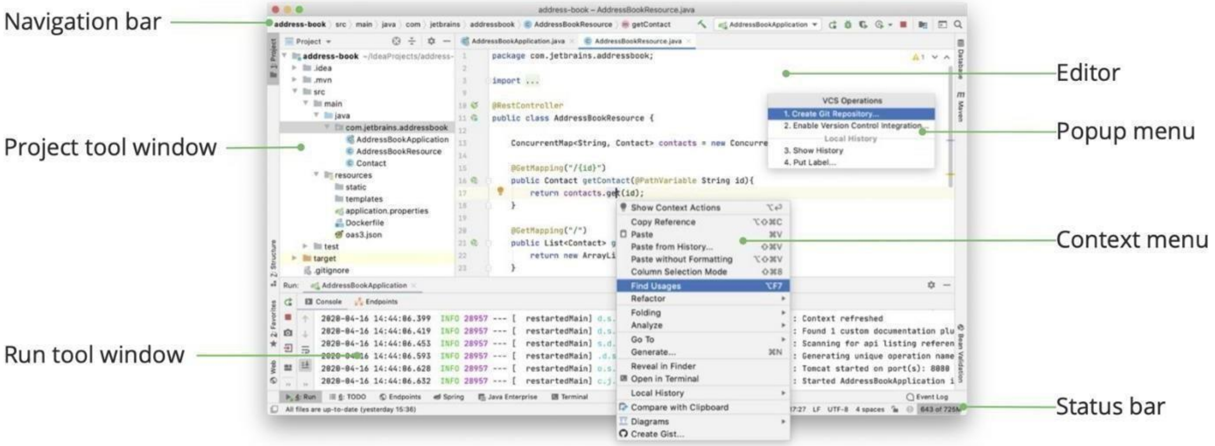
III.B)SOFTWARE REQUIREMENTS

1.Domain: Python
2.Version: Python IDLE (3.11.2)
3.Code Editors: PyCharm
4.Frameworks and Dependencies: tkinter,PIL,ImageTK
5.Operating System: Windows 10

IV.CODE EDITORS

IV.A)PyCharm

PyCharm is an integrated development environment[8] (IDE) used In computer programming, specifically for the Python language. It is developed by the Czech company Jet Brains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda



```
package com.jetbrains.addressbook;

import ...

@RestController
public class AddressBookResource {

    ConcurrentMap<String, Contact> contacts = new Concurre

    @GetMapping("/{id}")
    public Contact getContact(@PathVariable String id){
        return contacts.get(id);
    }

    @GetMapping("/")
    public List<Contact> g
        return new ArrayLi
    }
```

```
2020-04-16 14:44:06.399  INFO 28957 --- [  restartedMain] d.s.   : Context refreshed
2020-04-16 14:44:06.419  INFO 28957 --- [  restartedMain] d.s.   : Found 1 custom documentation plu
2020-04-16 14:44:06.453  INFO 28957 --- [  restartedMain] s.d.   : Scanning for api listing referen
2020-04-16 14:44:06.593  INFO 28957 --- [  restartedMain] .d.s   : Generating unique operation name
2020-04-16 14:44:06.628  INFO 28957 --- [  restartedMain] o.s.   : Tomcat started on port(s): 8080
2020-04-16 14:44:06.632  INFO 28957 --- [  restartedMain] c.j.   : Started AddressBookApplication i
```

Figure1: PyCharm screen

•Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
•Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
•Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
•Integrated Python debugger
•Integrated unit testing, with line-by-line code coverage
•Google App Engine Python development
•Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
•Support for scientific tools like matplotlib, numpy and scipy [professional edition only]

PyCharm provide an API so that developers can write their own plugins to extend PyCharm features. Several plugins from other JetBrains IDE also work with PyCharm. There are more than 1000 plugins which are compatible with PyCharm

IV.B)LIBRARY USED

IV.B.1)TKINTER: Tkinter is a Python standard library for creating desktop application graphical user interfaces[(GUIs). It makes it simple to create desktop applications. Our primary GUI toolkit will be Tk, which is Python's default GUI framework. Tkinter, a Python interface, will be used to access Tk (short for Tk interface). Running python -m Tkinter from the command line should display a simple Tk interface, indicating that Tkinter is properly installed on your system and also indicating what version of Tcl/Tk is installed, allowing you to browse the documentation for that version. Tkinter works with various Tcl/Tk versions, both with and without thread support. Tcl/Tk 8.6 threaded is included in the official Python binary distribution[9]. More information about supported versions can be found in the _tkinter module's source code. Tkinter is not a light wrapper, but it does contribute some logic to make the experience more pythonic. The documentation will focus on these additions and updates, with aspects that remain unchanged being referred to as the standard Tcl/Tk documentation.

Features:
1.Displaying Text and Images.
2.Displaying Clickable Buttons.
3.Getting User Input.
4.Getting Multiline User Input.
5.Assigning Widgets to Frames.
6.Adjusting Frame Appearance.

IV.B.2)Tkinter Message box: The message box module in Python is used to show message boxes in programs that use them. A variety of functions are used, depending on the application's requirements, to display the pertinent messages. There are seven functions in the Tkinter Message box[10] there are listed below
a. showinfo(): When we need to provide the user with some pertinent information, we use the showinfo() message box
b. showwarning(): The warning is shown to the user using this technique.
c. showerror(): This technique is used to show the user the error notice.
d. askquestion(): This approach is used to pose questions to users that they can respond to with a simple yes or no.
e. askokcancle(): This technique is employed to verify the user's participation in an application activity.
f. askyesno(): Using this technique, a question concerning action is posed
g. askretrycancel(): Using this technique, the user is asked whether they would like to perform a specific task again.

V.RELATED WORK

An Autokey Cipher (also known as the autoclave cipher) is a cipher that incorporates the message (the plaintext) into the key. The key is generated from the message in some automated fashion, sometimes by selecting certain letters from the text or, more commonly, by adding a short primer key to the front of the message. There are teo forms of autokey cipher: Key-autokey and text-autokey ciphers. A Key-autokey cipher uses previous members of the keystream to determine the next element in the keystream. A text-autokey uses the previous message text to determine the next element in the keystream. In modern Crypotography, Self-Synchonising[11] Stream Ciphers are Autokey Ciphers.
This cipher was invented in 1586 by Blaise de Vigenère with a reciprocal table of ten alphabets. Vigenère's version used an agreed-upon letter of the alphabet as a primer, making the key by writing down that letter and then the rest of the message.
More popular autokeys use a tabula recta, a square with 26 copies of the alphabet, the first line starting with 'A', the next line starting with 'B' etc. Instead of a single letter, a short agreed upon keyword is used, and the key is generated by writing down the primer and then the rest of the message, as in Vigenère's version. To encrypt a plaintext, the row with the first letter of the message and the column with the first letter of the key are located. The letter in which the row and the column cross is the ciphertext[12] letter

VI.STRUCTURE OF PART OF A DNA

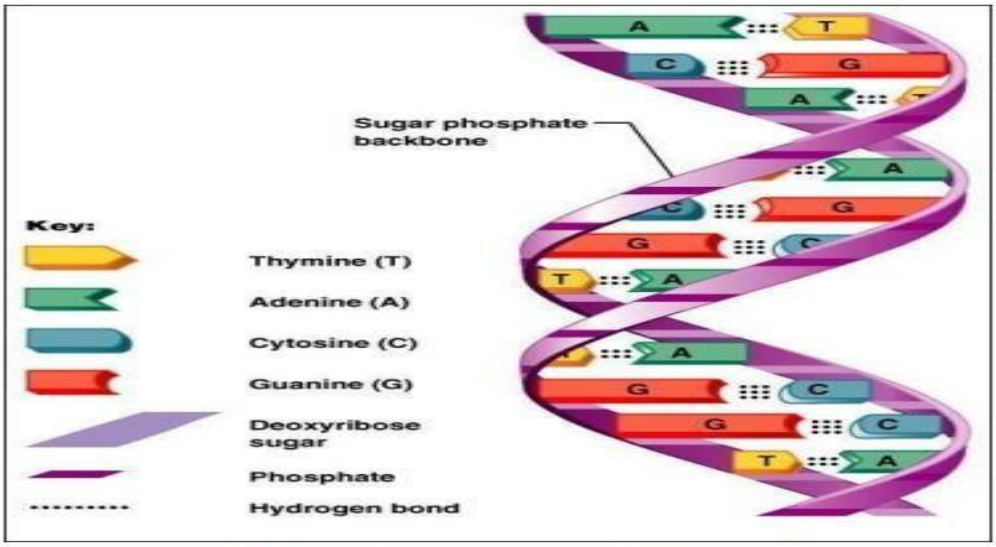


Fig. 1. The structure of part of a DNA.

VI.STATE CHART DIAGRAM

VI.A)STATE CHART DIAGRAM FOR SENDER

State diagrams are used to describe the behaviour of a system. State diagram describe all the possible state of an object as events occur. Each diagram usually represents objects of a single class and track the different state of its objects through the system. Not all classes will require a state diagram and state diagram are not useful for describing the collaboration of all objects in a use case. State diagram have very few elements. This is the state of the object when it is created. After the initial state the object begins changing states. State chart diagrams are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

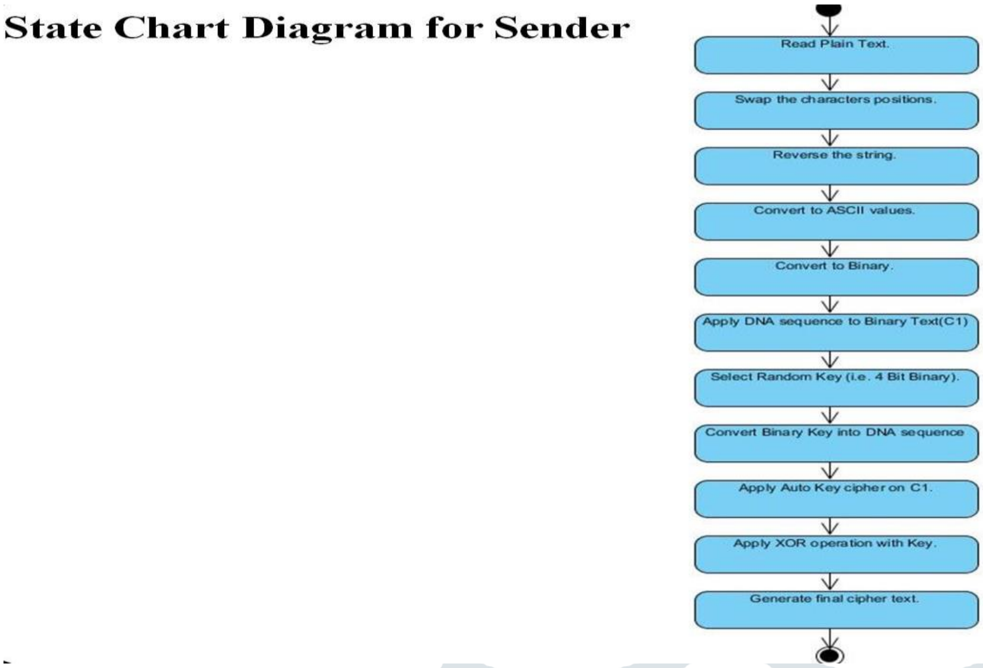

State Chart Diagram for Sender
Read Plain Text.
Swap the characters positions.
Reverse the string.
Convert to ASCII values.
Convert to Binary.
Apply DNA sequence to Binary Text(C1)
Select Random Key (i.e. 4 Bit Binary).
Convert Binary Key into DNA sequence
Apply Auto Key cipher on C1.
Apply XOR operation with Key.
Generate final cipher text.

Figure 3.State-Chart Diagram for sender

Description:

The sender gives the plain text as input. Swap characters and reverse the string and convert characters into ASCII Values after that Convert ASCII Values into Binary values apply DNA Sequence[13] on Binary Values and select some Random Kry apply autokey cipher, After that we Apply XOR operation on those two values View cipher text apply XOR With Key and Apply inverse Autokey cipher Generate Apply DNA Sequence Convert Binary to ASCII and ASCII to String generate plain text.

VI.B)STATE CHART DIAGRAM FOR RECEIVER

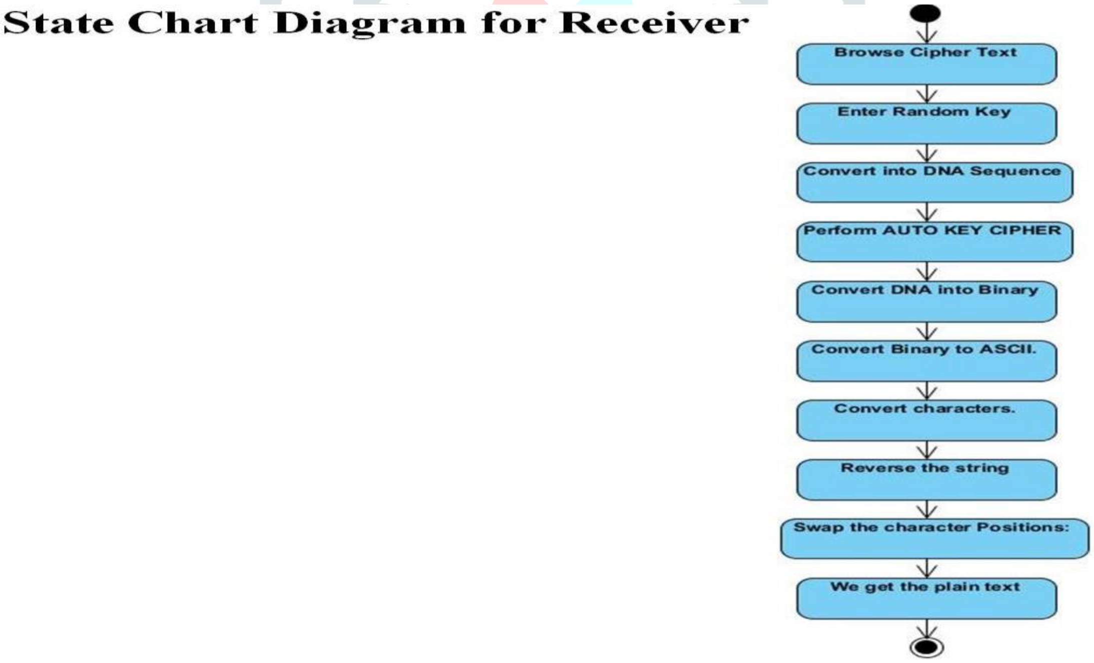


Figure4 State-Chart Diagram for receiver

Description:

First we Generate plain text, and the Convert Binary to ASCII and ASCII to String Generate Apply DNA Sequence Apply DNA Sequence apply inverse Auto cipher and apply XOR With Key View cipher text Apply XOR Operation on those teo values Select Random Key Apply Auto Key Cipher Apply DNA Sequence on Binary Values Convert ASCII[14] Values into Binary Values and convert characters into ASCII values and swap characters into reverse the string and Enter the Plain Text.

VII.ENCRYPTION AND DECRYPTION

Data that can be read and understood by anyone without any special knowledge about it is called plaintext or clear text. The method of disguising the plaintext in such a way as to hide the information is called encryption[15]. Encrypting plaintext results in unreadable gibberish called cipher text. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting cipher text to its original plaintext is called decryption[16]

VII.A)ENCRYPTION ALGORITHM

Step 1: Read Plain Text.

Step2: Swap the characters positions.

Step3: Reverse the string.

Step4: Convert the character values into ASCII values.
Step5: Then covert ASCII values to Binary.
Step6: Apply DNA sequence to Binary Text (C1).
Step7: Select Random Key (i.e., 4 Bit Binary).
Step7: Convert Binary Key value into DNA sequence.
Step8: Apply Auto Key cipher on C1.
Step9: Apply XOR operation[17] with Key value to C1.
Step10: Generate final cipher text.

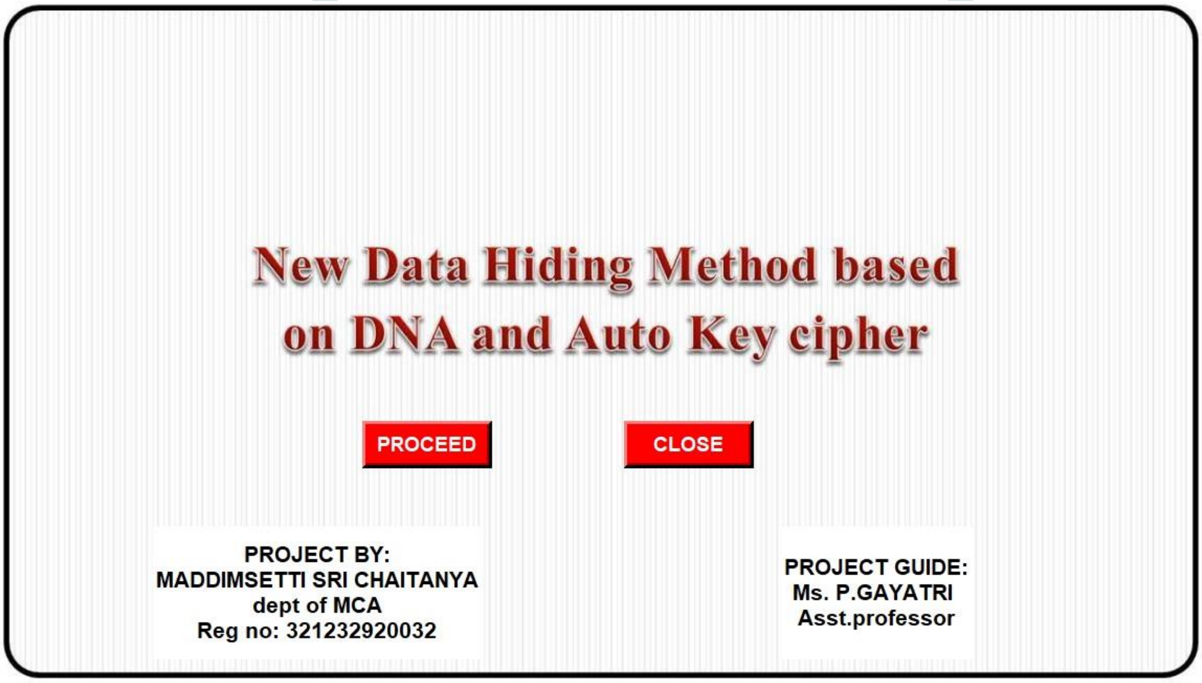
VII.B)DECRYPTION ALGORITHM

Step 1: Take a string as input from the sender.
Step 2: By considering a Random Key and convert into DNA Sequence
Step 3: Perform AUTO KEY CIPHER on Cipher Text1
Step 4: By using DNA Nucleotides convert the DNA sequence into Binary values.
Step 5: Convert Binary to ASCII.
Step 6: Convert each ASCII value into its characters.
Step7: Reverse the string
Step 8: Swap the character Positions:
Step 9: We get the plain text

VIII.OUTPUT SCREENS

VIII.A)HOME SCREEN

This is the home screen of the application from where the procedure will begin.



New Data Hiding Method based on DNA and Auto Key cipher

PROJECT BY:
MADDIMSETTI SRI CHAITANYA
dept of MCA
Reg no: 321232920032

PROJECT GUIDE:
Ms. P.GAYATRI
Asst.professor

Figure 5.Home Screen

VIII.B)ENCRYPTION

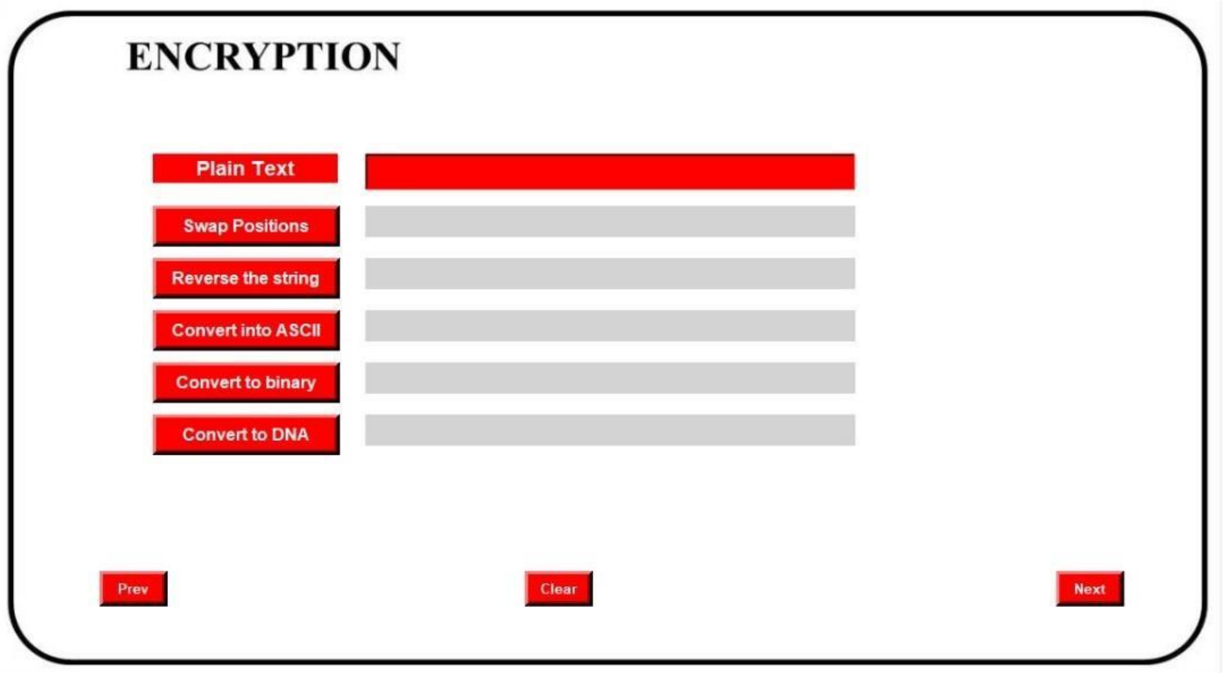
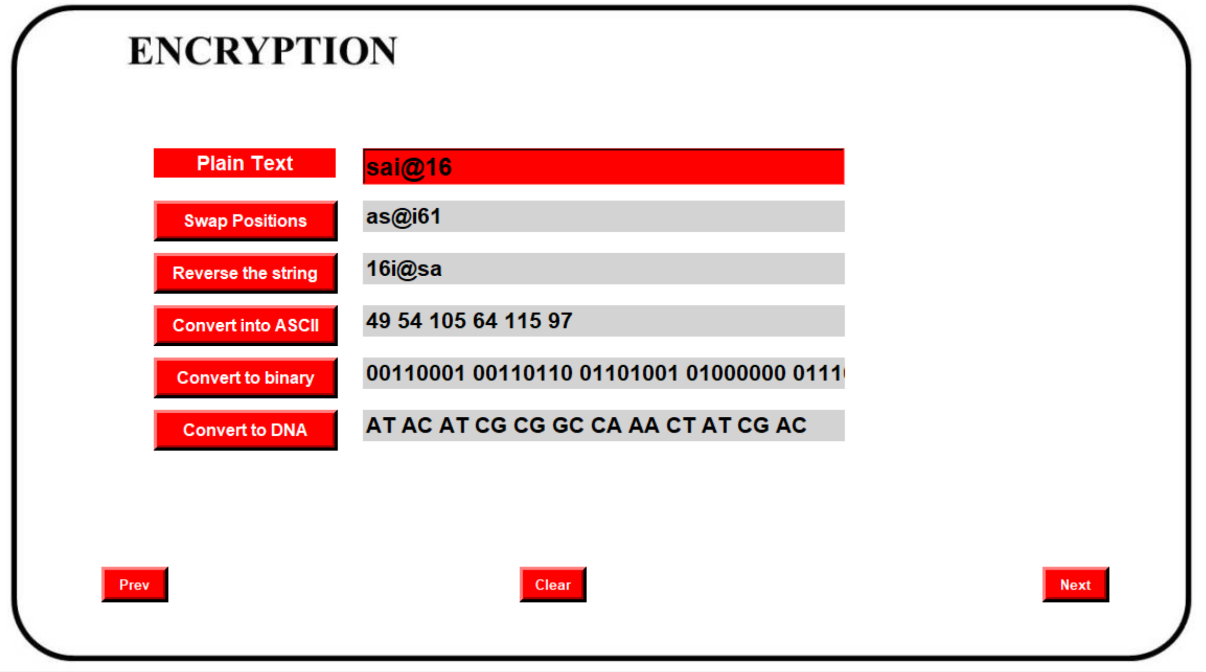


Figure 6. Encryption with plain text

This screen asks the sender to enter plain text of maximum length of 12 characters which has been applied circular left shift of the plain text. Later the text is split into 2 parts and reversed. These reversed bits are then merged to obtain a text which is converted to ASCII values followed by converting into binary values.



Plain Text: sai@16

Swap Positions: as@i61

Reverse the string: 16i@sa

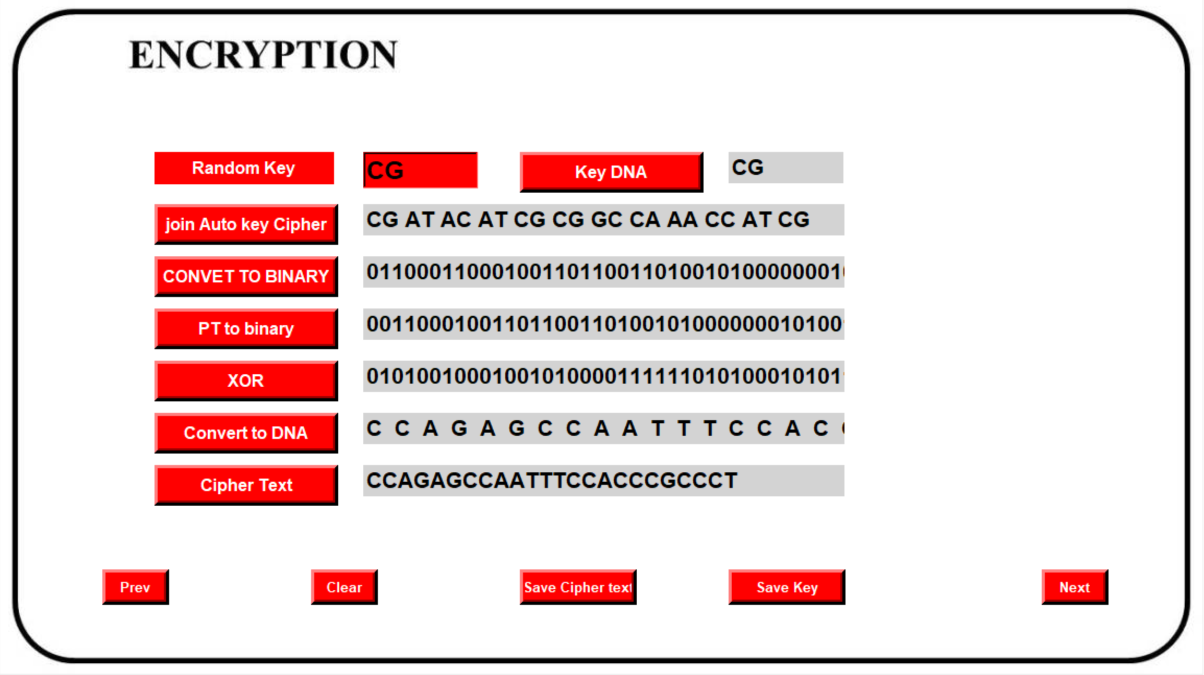
Convert into ASCII: 49 54 105 64 115 97

Convert to binary: 00110001 00110110 01101001 01000000 0111

Convert to DNA: AT AC AT CG CG GC CA AA CT AT CG AC

Figure 7. Encryption with Key

This screen asks for the key value which has to be saved. This key is then converted to binary values. These binary values are then XOR end with the plain texts' binary values[18] and being split into 2 parts namely Odd and Even parts.



CG

CG

CG AT AC AT CG CG GC CA AA CC AT CG

01100011000100110110011010010100000001

0011000100110110011010010100000000 10100

0101001000100101000011111101010001010 1

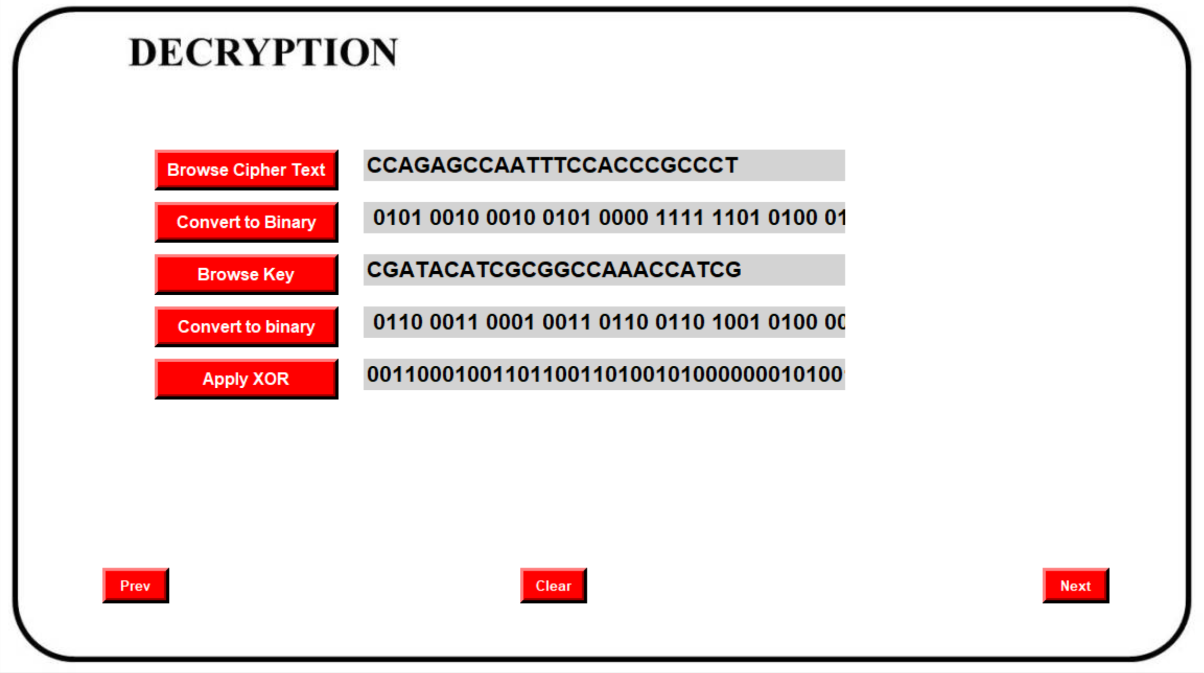
C C A G A G C C A A T T T C C A C

CCAGAGCCAATTTCCACCCGCCCT

Figure 8. Cipher Text

This screen contains the joining the odd and even parts of the XORed bit parts which are converted decimal values which are compared with ASCII table to obtain Cipher text

VIII.C)DECRYPTION



CCAGAGCCAATTTCCACCCGCCCT

0101 0010 0010 0101 0000 1111 1101 0100 01

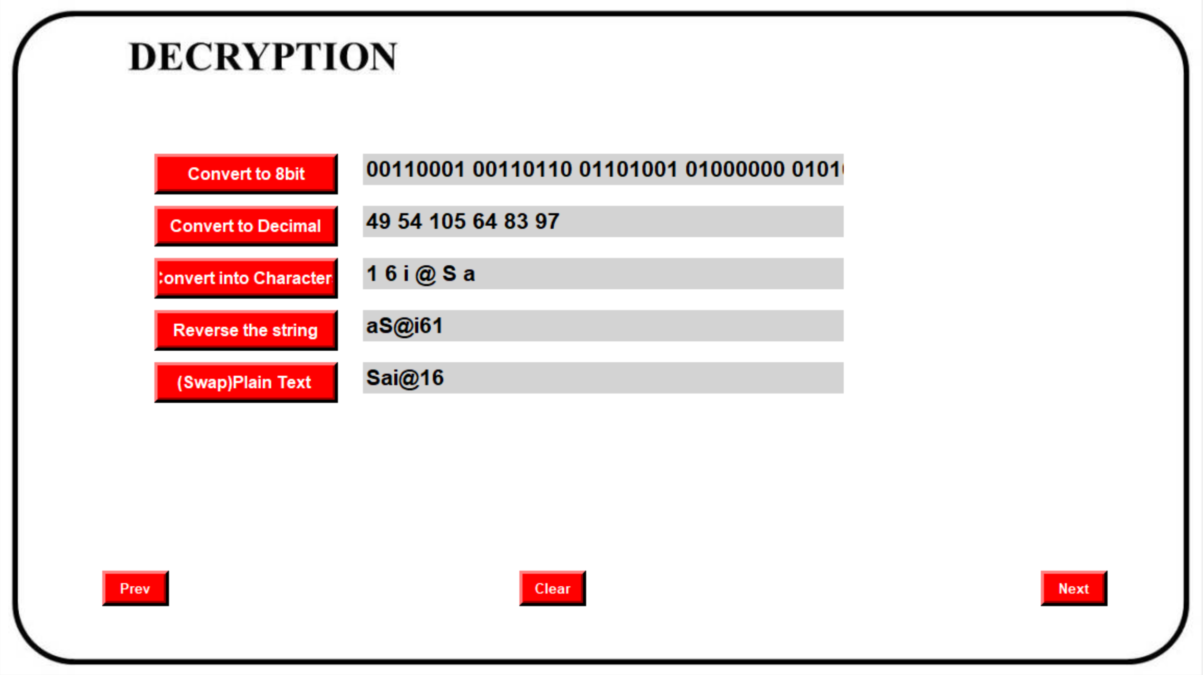
CGATACATCGCGGCCAAACCATCG

0110 0011 0001 0011 0110 0110 1001 0100 00

0011000100110110011010010100000000 10100

Figure 9.Decryption page1

This screen asks the receiver to browse the cipher text obtained from the sender. This text is converted to binary values of left and right parts. Now gather 4 bits from each block and convert to 8-bit blocks.



00110001 00110110 01101001 01000000 0101

49 54 105 64 83 97

1 6 i @ S a

aS@i61

Sai@16

Figure 10.Decryption page 2

Now browse for the key sent by the sender. Find binary values of the key and perform XOR for these values with the cipher text's binary values to obtain a single bitstream. Finally find ASCII values of the obtained binary values

IX.TESTING

It finds the differences between the object design model and its corresponding components. In this test each component is tested independent of the other thus allowing parallelism[19] in testing activity. Ex: individual units like selecting the plain text and the key are not match each other than the system will not generate the appropriate result/plaintext

Structural testing: It finds difference between the system design model and a subset of integrated subsystems[20].

Functional testing: It finds differences between the use case model and the system.

Performance testing: It finds difference between non-functional requirements and actual system performance.

Test Case #	Test Case	Description	Expected result	Obtained Result	Pass/Fail
1	Encryption Test	Enter the plain text for encryption process.	The plain text should be displayed on the screen.	Plain text is displayed	PASS
2	Encryption Test	Sender clicks on Encryption	Encryption process is done will get the Cipher text.	Got the ciphertext.	PASS
3	Key Generation	Random Key should be given as input	Plain/Cipher should be obtained	Plain/Cipher text obtained	PASS
4	Decryption Test	Receiver select the Encoded Bits	Retrieve Encoded bits from the destination.	Got the EncodedBits	PASS
5	Decryption Test	User clicks on decryption.	Decryption	Got the Cipher Text.	PASS

X.CONCLUSION

In this project a secure, reversible and applicable data hiding algorithm is proposed based on DNA technology. Where, the proposed method made use of the special properties of DNA Sequences for data hiding which is, there is almost no difference between a real DNA sequence and a fake one therefore hiding data in DNA sequences and producing new sequence of DNA will not be discovered by the intruder since, there are approximately 163 million publicly available DNA sequences and it is virtually impossible to guess this sequence. The proposed hiding algorithm provides a new way of embedding mechanism[21] which used Auto key cipher for hiding the secret message within a generated DNA sequence. The proposed algorithm utilized DNA-XOR operator for creating a table to be used by the mechanism of Auto key cipher. To the best of our knowledge, this mechanism has not been adopted by other works.

X.A)SCOPE FOR FUTURE DEVELOPMENT

Future advancement in this discipline has a sizable and highly promising future. In order to provide even higher degrees of security and embedding capability, the suggested system can be improved further by investigating other DNA-based steganographic methods and encryption algorithms. The conversion of DNA sequences may be optimised, and encoding and decoding operations can be made more accurate and efficient. Additionally, investigating the integration of other cutting-edge technologies[24], such as artificial intelligence[22] and machine learning, can improve the system's capabilities in identifying and thwarting future assaults on the hidden data. Future research can look into other uses outside data storage and authentication, including DNA-based covert communication routes or secure data transmission[23]. Working with geneticists and biotechnologists can provide researchers important insights on using the DNA's special features might lead to new developments in steganography and cryptography.

XI.REFERENCES

[1] An Article Reference of Auto Key cipher
https://ieeexplore.ieee.org/abstract/document/8674346
[2] An Article Reference of high embedding
https://dl.acm.org/doi/abs/10.1145/3082031.3083240
[3] An Article Reference of embedding technique
https://www.sciencedirect.com/science/article/abs/pii/S0950705118301540
[4] An Article Reference of DNA-XOR
https://www.sciencedirect.com/science/article/abs/pii/S0141938215300238
[5]An Article Reference of steganography systems
https://asmp-eurasipjournals.springeropen.com/articles/10.1186/1687-4722-2012-25/
[6] An Article Reference of Cryptography algorithm
https://ieeexplore.ieee.org/abstract/document/9328432
[7] An Web Reference of complimentary rule
https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.200804709
[8] An Web Reference of PyCharm is an integrated development environment
https://ieeexplore.ieee.org/abstract/document/8941213
[9] An Web Reference of Python binary distribution
https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.12200
[10] A Book Reference Of Tkinter Message Box
https://books.google.co.in/books?hl=en&lr=&id=YDe2DwAAQBAJ&oi=fnd&pg=PP1&dq=Tkinter+message+box+%2B+book&ots=fRye6pPoWq&sig=4ZyHqOGypZKsjc08pQGeFe-vr_Y&redir_esc=
[11] A Book Reference Of the Self-Synchonising
https://books.google.co.in/books?hl=en&lr=&id=IkUPEAAAQBAJ&oi=fnd&pg=PP1&dq=SelfSynchronising+%2B+book&ots=Ol3UQXeSrM&sig
[12] A Book Reference Of ciphertext
https://www.proquest.com/openview/e2bb3efd70988995df18545099745e0d/1?pqorigsite=gscholar&cbl=20263
[13] A Book Reference Of DNA Sequence
https://link.springer.com/book/10.1007/978-0-387-78168-6
[14] A Book Reference Of ASCII
https://go.gale.com/ps/i.do?id=GALE%7CA346926256&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=14814374&p=AONE&sw=w&userGroupName=anon%7E4729dcd1&aty=open+web+entry
[15]AN Article Reference of Encryption
https://www.earticle.net/Article/A245530
[16] A Book Reference Of Decryption
https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=367c9ab706164690122849e15147e1975af43
[17] AN Article Reference of XOR Operation
https://ieeexplore.ieee.org/abstract/document/6024565
[18]An Web Reference of Binary values
https://link.springer.com/chapter/10.1007/978-3-642-37256-8_11
[19] AN Article Reference of parallelism
https://www.cambridge.org/core/journals/journal-of-functional-programming/article/algorithm-strategy-parallelism/24CE696A9E76AEA63C2D6132BE25FC09
[20] AN Article Reference of integrated subsystem
https://www.proquest.com/openview/0c50bf93ad92fbf38bfe99a0ecb81eea/1?pq-origsite=gscholar&cbl=18750
[21] A Book Reference Of embedding mechanism
https://books.google.co.in/books?hl=en&lr=&id=36iLEAAAQBAJ&oi=fnd&pg=PR5&dq=embedding+mechanism+%2B+book&ots=rh-SUt-03P&sig=DNRzIPA1_UqyerWTg1rfVlQV34w&redir_esc=
[22] A Book Reference Of Artificial Intelligence

https://dl.acm.org/doi/abs/10.5555/129914
[23] An Web Reference of data transmission
https://pubs.acs.org/doi/full/10.1021/ja403828z
[24] AN Article Reference of cutting-edge technologies
https://www.sciencedirect.com/science/article/abs/pii/S0747563221000832
[25] A Book Reference Of Unauthorized access
https://library.oapen.org/handle/20.500.12657/40085

BIBILIOGRAPHY

POTNURI GAYATRI . She received her M Tech in Computer Science & engineering from JNTU Kakinada in January 2015. She received her B Tech Degree from VITAM College of Engineering, JNTU Hyderabad in 2004 . she currently working as Assistant Professor, CSE Dept in SVPEC ,Andhra Pradesh, India . Her Research Interests include Sensor Networks.

Maddimsetti Sri Chaitanya studying her 2nd year, Master of Computer Applications in Sanketika Vidya Parishad Engineering College, affiliated to Andhra University, accredited by NAAC.With her interest in machine learning method and as a part of academic project,he used New data hiding method based on DNA and vigenere autokey . As a result of a desire to comprehend the flaws in conventional reporting and to preserve timely and high-quality report output in DNA and Vigenere autokey . A completely developed project along with code has been submitted for Andhra University as an Academic Project. In completion of her MCA.