

GENEPROTECT: SECURE STORAGE AND ENCRYPTION OF GENETIC DATA

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology degree in Biotechnology
Interdisciplinary project

By

AARTHI RAGHU (Reg. No. 41230001)

HASIKA R (Reg. No. 41230022)



**DEPARTMENT OF BIOTECHNOLOGY
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY-2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF BIOTECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **Aarthi Raghu** (Reg. No. 41230001) and **Hasika R** (Reg. No. 41230022), who carried out the Interdisciplinary Project entitled "**GENEPROTECT: SECURE STORAGE AND ENCRYPTION OF GENETIC DATA**" under our supervision from January 2024 to April 2024.

Internal Guide

Dr. SAQIB HASAN, M.Sc., Ph.D.

Head of the Department

DR. V. RAMESH KUMAR, M. PHIL., PH.D.,

Submitted for Viva voce Examination held on 06.05.2024

Internal Examiner

External Examiner

DECLARATION

We, **Aarthi Raghu (Reg. No. 41230001)** and **Hasika R (Reg. No. 41230022)** hereby declare that the Project Report entitled “**GENE PROTECT: SECURE STORAGE AND ENCRYPTION OF GENETIC DATA**” done by us under the guidance of **Dr.SAQIB HASAN,M.Sc.,Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Biotechnology.

A handwritten signature in black ink, appearing to read 'Aarthi Raghu', with a long, sweeping horizontal stroke extending to the right.

DATE: 06.05.2024

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

AKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my sincere thanks to **Dr. V. Ramesh Kumar, M.Phil., Ph.D., Head of the Department, Department of Biotechnology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Dr.Saqib Hasan,M.Sc.,Ph.D., Assistant Professor, Department of Biotechnology** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I also express my thanks to all **Teaching and Non-teaching staff members** of the Department of Biotechnology who were helpful in many ways for the completion of the project work.

ABSTRACT

In the present era of crime scene investigations, DNA and RNA samples collected from crime scenes are extremely essential in cracking the case and bringing the criminal to justice. However, the advancement of genetic technologies, bioinformatic analysis, and storage has raised major concerns about the proper safeguarding of privacy and security of these genetic data. In response to this need of the hour, GeneProtect is a highly advanced double-encrypted cloud storage-based database that is meticulously built and programmed to ensure the security and privacy of collected DNA and RNA samples. It is a powerful web application tool that provides a complete defense against access that may be unauthorized and potential breaches of sensitive genetic data by hackers. It achieves this by employing a robust system of double encryption following advanced encryption standard (AES) which ensures that every piece of genetic information is shielded with layers of cryptographic protection by assigning multiple values for a single character, making it virtually impervious to intruders and malicious phishers. This advanced encryption algorithm not only enhances the protection of the stored samples against potential theft of data but also reduces the chance of false accusations by helping provide solid genetic evidence during the time of trial. GeneProtect incorporates data anonymization which is essential for further safeguarding of the identities of individuals associated with the collected genetic samples. GeneProtect therefore showcases the potential of DNA and RNA encrypted technology to revolutionize data storage, offering a secure and efficient solution to safeguard sensitive information like forensic data against unauthorized access, tampering, and data breaches.

Keywords: Advanced Encryption Standard (AES), Crime Scene Investigation, Double Encryption, DNA, GeneProtect, RNA

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF FIGURES	iii
	LIST OF ABBREVIATIONS	iv
1	INTRODUCTION	1
2	REVIEW OF LITERATURE	6
	2.1 History of DNA Database Formation in Forensic Science	6
	2.2 Drawbacks in Current Forensic DNA Databases	7
	2.3 Security Vulnerabilities in Existing Databases	8
	2.4 AES in DNA Databases for Forensic Data	9
	2.5 Enhancing Security and Confidentiality	10
3	AIM AND SCOPE	11
	3.1 Aim	
	3.2 Objective	
	3.3 Scope	
4	MATERIALS AND METHODOLOGY	12
	4.1 Hardware Requirements	12
	4.2 Software Requirements	12
	4.3 DNA Encryption	14
	4.4 DNA Encoding	17
	4.5 SQL Schema for Genetic Data Storage	22
	4.6 Retrieval of Data from Database	23
5	RESULTS AND DISCUSSION	26
	5.1 Web Application User Interface	26
	5.2 Back-End Working of Code	29
6	SUMMARY AND CONCLUSION	32
	REFERENCES	33

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Short Tandem Repeat (STR) Typing	2
1.2	Cyberbiosecurity Aspects	3
1.3	Workflow of DNA Analysis and Data Interpretation	4
1.4	Double Encryption Mechanism	4
2.1	CODIS Database Features	6
2.4	AES Encryption and Decryption Mechanism	9
4.3.1	Flowchart of AES Encryption	14
4.3.2	AES Parameters	15
4.3.3	AES Data Structures	16
4.4	DNA Encoding	17
4.6	Encryption and Decryption of Data	25
5.1.1	Account Creation	27
5.1.2	Database Details	27
5.1.3	DNA Data Storage	28
5.1.4	Assigning Encryption Key	28
5.2.1	Source Code	29
5.2.2	File Encryption Key Size Selection	29
5.2.3	Successful Encryption of File	30
5.2.4	File Decryption Parameters	30
5.2.5	Successful Decryption of File	31

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
CODIS	Combined DNA Index System
DNA	Deoxyribonucleic Acid
HSM	Hardware Security Module
NGS	Next-Generation Sequencing
NIST	US National Institute of Standards and Technology
RNA	Ribonucleic Acid
SSD	RAID-Configured Solid-State Drive
SSL	Secure Socket Layer
STR	Short Tandem Repeat
TLS	Transport Layer Security

CHAPTER 1

INTRODUCTION

Forensic DNA testing stands as a seminal advancement in modern criminal justice, fundamentally reshaping investigative procedures since its advent in the mid-1980s. Its far-reaching impact transcends mere convictions, playing pivotal roles in both incrimination and exoneration. By furnishing critical evidence leading to the conviction of perpetrators and contributing to the exoneration of the wrongfully accused, forensic DNA testing serves as an essential arbiter of truth in legal proceedings. (Butler, *et al.*, 2014). The continual evolution of forensic laboratories underscores a steadfast commitment to advancing scientific methodologies and technologies. Through stringent validation protocols and the adoption of state-of-the-art techniques, laboratories endeavour to augment the sensitivity and informativeness of DNA analysis. Automation has emerged as a cornerstone in addressing the mounting demands for throughput, efficaciously streamlining workflows and ensuring expeditious and precise sample processing. (Arshad, *et al.*, 2021)

Short tandem repeat (STR) typing, the cornerstone of forensic DNA analysis, enables the probabilistic individualization of DNA profiles, allowing for the statistical assessment of matches (fig 1.1). While the probability of individualization is generally achievable, even in cases involving identical twins, the integration of additional genetic markers further enhances the discriminatory power of DNA analysis, facilitating the resolution of complex forensic scenarios. (Butler, *et al.*, 2014)

The digitization of DNA technology heralds a new era of efficiency and collaboration, empowering scientists with advanced software tools for genomic analysis. This digital transformation revolutionizes traditional laboratory practices, facilitating seamless data sharing, analysis, and collaboration across interdisciplinary boundaries. However, amidst the myriad benefits of digitization, ensuring the security and privacy of genetic information emerges as a paramount concern. The digitization of DNA data introduces novel challenges in safeguarding sensitive genetic information from unauthorized access, tampering, and exploitation. As the custodians of genetic data, stakeholders must navigate the complex landscape of cybersecurity and privacy regulations to uphold the integrity and confidentiality of genomic information. (Wang, *et al.*, 2017)

Forensic STR Analysis

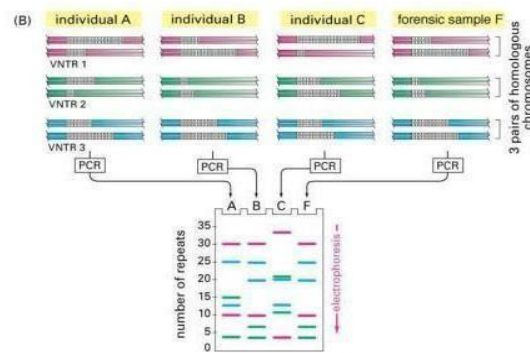


Fig.1.1: Short Tandem Repeat (STR) Typing

Three pairs of homologous chromosomes are taken from each suspect and the number of tandem repeats for those individuals is analyzed and matched with that of the forensic sample obtained from the scene

The Combined DNA Index System (CODIS), established by the federal government, integrates national, state, and local DNA databases, enabling law enforcement agencies to compare DNA evidence from crime scenes with profiles from convicted offenders. This interconnected system aids in identifying serial offenders and linking DNA evidence across different crime scenes. (National Institute of Standards and Technology, *et. al.*, 2001) While genomic data holds immense potential for applications such as paternity testing, disease screening, and ancestry tracing, safeguarding privacy is paramount. Cyberbiosecurity has emerged as a field dedicated to addressing vulnerabilities in genomic data storage and transmission, protecting individuals and organizations from threats such as DNA theft and genetic warfare. (Sun, *et al.*, 2018)

Efforts to secure genomic data must balance privacy with utility. (fig 1.2) Encryption techniques such as AES (Advanced Encryption Standard) play a crucial role in protecting sensitive genetic information while allowing for its practical use in medical and genomic research. Innovative approaches, like combining Arithmetic coding with AES, offer both security and compression, enabling efficient data transmission and storage. (Abdullah, *et al.*, 2017). Asymmetric key encryption, utilizing genetic algorithms, presents a promising avenue for enhancing security in data transmission. Its use of key pairs, with one public and one private key, overcomes challenges associated with symmetric key encryption, ensuring secure communication channels. (Naik, *et al.*, 2012)

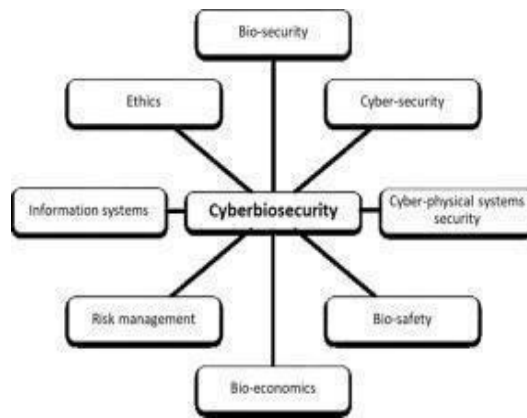


Fig:1.2: Cyberbiosecurity Aspects

Summary of cyberbiosecurity aspects across the spectrum of the biology. Each sector of the chart describes specific areas that span the bioinformatics, biological information systems, risk management and biotechnology

The advent of next-generation sequencing has generated vast amounts of data, prompting laboratories to turn to cloud services for storage and management. However, concerns regarding data curation and security underscore the importance of robust protocols for safeguarding genetic databases. (Carter, *et al.*, 2017) Each DNA sample undergoes a rigorous sequencing process (fig 1.3) before being stored in a double-encrypted format, safeguarding its integrity and ensuring proper storage in the encrypted database. (Lederma, *et al.*, 2019)

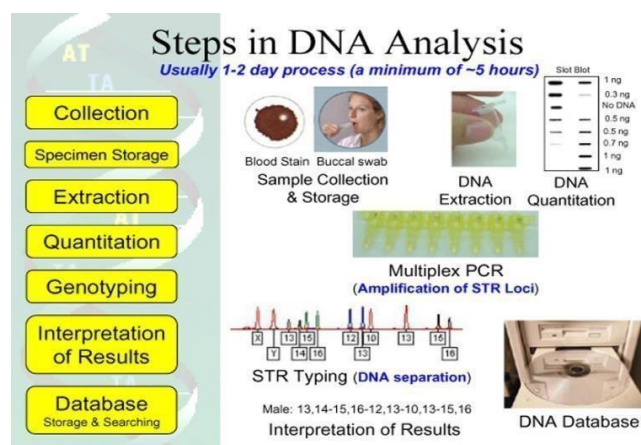


Fig:1.3: Workflow of DNA Analysis and Data Interpretation

The steps in DNA analysis include sample collection and storage, extraction and quantitation of DNA, genotyping to generate an individual pattern of short tandem repeat (STR) loci, and interpretation and storage of the results.

At the core of Geneprotect's security architecture lies its exclusive subscription-based system, which implements stringent access controls to ensure that only authorized personnel can access genetic data stored within the platform. By prioritizing user privacy and confidentiality, it maintains the trust and integrity of forensic investigations, fostering a collaborative and transparent environment for all stakeholders. Central to the security infrastructure is the renowned AES (Advanced Encryption Standard) technology, which provides robust encryption capabilities to safeguard genetic information against interception or compromise. Employing multiple layers of encryption (fig 1.4) and sophisticated key generation techniques, it ensures that sensitive genetic data remains protected at all times. (Patel, et al., 2014)

■ Double Encryption

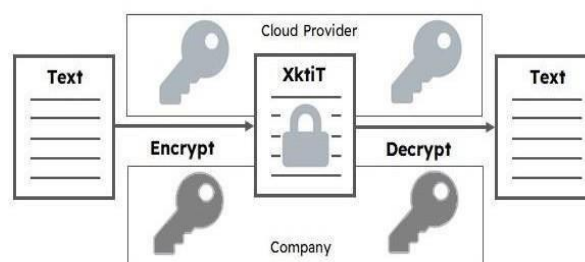


Fig:1.4: Double Encryption Mechanism

(Double encryption is where two or more independent layers of encryption are enabled to protect against compromises of any one layer of encryption. Using two layers of encryption mitigates threats that come with encrypting data.)

In addition to its formidable encryption capabilities, it upholds rigorous data management practices, adhering to industry best practices and regulatory standards. By adopting a proactive approach to data security, it mitigates the risk of data breaches and maintains the confidentiality of genetic information, instilling confidence and peace of mind among users. Beyond its role in forensic investigations, the technology holds promise for diverse fields, including medical research, personalized medicine, and ancestry tracing. By enabling secure access to genetic information, Geneprotect ensures the confidentiality and protection of sensitive data from the hands of unauthorized people.

CHAPTER 2

REVIEW OF LITERATURE

2.1 HISTORY OF DNA DATABASE FORMATION IN FORENSIC SCIENCE

A major revolutionary change occurred in forensic investigations in the late 1980s and early 1990s with the establishment of databases for storage of DNA data. Ground-breaking efforts were made in Europe, majorly exemplified by the National DNA Database of United Kingdom initiated in 1995, paved the way for a paradigm shift. Unlike traditional fingerprint identification methods, DNA analysis offered unmatched accuracy and the astonishing ability to crack seemingly unrelated crimes through unidentified DNA evidence found at the scene. (Gill, *et al.*, 1994)

In the same era, simultaneously, in 1994, the United States of America established its primary unit, the FBI's Combined DNA Index System (CODIS). CODIS enabled efficient comparison of DNA profiles across jurisdictions, leading to the identification of serial killers, exoneration of the wrongly convicted, and resolution of cold cases that lacked traditional fingerprint or eyewitness evidence.(fig 2.1) (Butler, *et al.*, 2003)

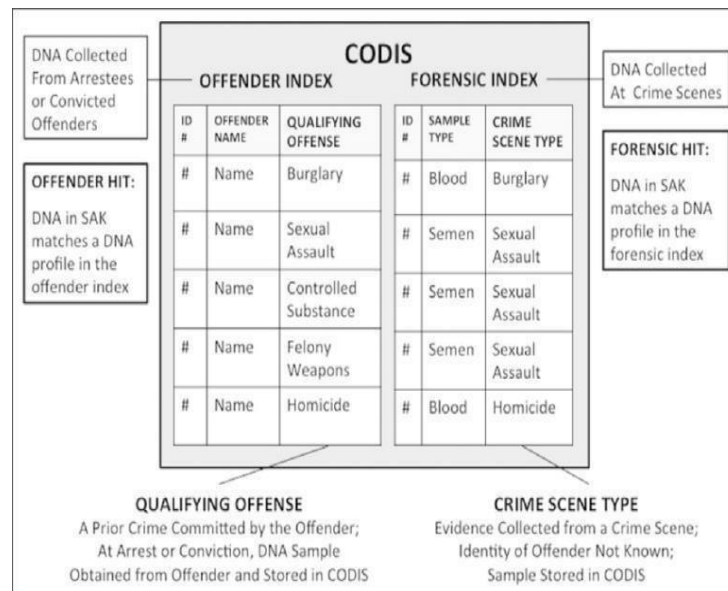


Fig:2.1: CODIS Database Features

A heuristic representation of how CODIS is structured and the types of information stored within this database

2.2 DRAWBACKS IN CURRENT FORENSIC DNA DATABASES

While DNA databases have revolutionized crime scene investigations, their efficacy depends on overcoming several critical issues. Data integrity remains paramount. Errors introduced during sample collection, handling, analysis, or storage, though statistically rare, can have unimaginable consequences, leading to inaccurate results and wrongful convictions. Mitigating this risk requires a multi-pronged approach. Meticulous protocols with standardized operating procedures for every stage of the process are essential. Equally important is a highly trained workforce with expertise in proper sample handling and rigorous adherence to quality control measures throughout the entire DNA analysis workflow. (Butler, *et al.*, 2003)

Database comprehensiveness is another challenge that can hinder investigations. Incomplete databases may exclude profiles from crimes deemed minor offenses. This creates a blind spot for repeat offenders who haven't been previously convicted of serious crimes. Furthermore, unsolved cold cases often lack corresponding DNA profiles in the database. This absence hinders the ability to link these cases to future crimes or identify potential suspects, potentially leaving victims and their families without closure. (Budowle, *et al.*, 2009) Resource constraints pose a significant logistical challenge. Law enforcement agencies and forensic laboratories often grapple with high caseloads, leading to backlogs of DNA samples awaiting analysis.

These delays can significantly impede investigations and prosecutions, potentially hindering the timely administration of justice. (National Institute of Justice, *et al.*, 2009) Investing in adequate resources is crucial. Addressing these challenges is critical to ensure the continued effectiveness and reliability of DNA databases as a cornerstone of modern forensic investigations. By prioritizing data integrity, fostering database comprehensiveness, and overcoming resource constraints, DNA databases can continue to be a powerful tool for solving crimes, exonerating the wrongly convicted, and identifying unknown remains.

2.3 SECURITY VULNERABILITIES IN EXISTING DATABASES

The inherent sensitivity of DNA data stored in forensic databases raises a multitude of privacy concerns. Individuals have a legitimate right to fear unauthorized access or misuse of their genetic information. Hacking attempts or security vulnerabilities could expose this sensitive data, leading to ethical and legal concerns about discrimination or exploitation based on genetic information. (National Institute of Justice, *et al.*, 2009) A major concern lies in the potential misuse of DNA data for discriminatory purposes. This could encompass scenarios such as denying employment opportunities, increased insurance premiums, or even social ostracization based on genetic predispositions. Robust regulations and strong oversight mechanisms are crucial to prevent such misuse and ensure the responsible handling of sensitive genetic data. (National Institute of Justice, *et al.*, 2009)

DNA databases can be a powerful investigative tool through familial searching, a technique that compares an unknown suspect's DNA profile to profiles of relatives within the database. While valuable for investigations, it raises privacy concerns for family members who may have no connection to the crime. Clear guidelines and legal frameworks are necessary to govern the use of familial searching, ensuring a balance between investigative needs and individual privacy rights. (Gostin, *et al.*, 2001) Determining the appropriate timeframe for retaining DNA profiles in databases is a complex issue. Balancing the potential for future investigations with individual privacy concerns necessitates careful consideration.

Arguments exist for indefinite retention, particularly for unsolved cases. However, others advocate for specific retention periods based on the severity of the crime or the individual's criminal history. Striking a balance between these viewpoints requires ongoing discourse and the development of clear retention policies that prioritize both public safety and individual privacy rights. (International Society for Forensic Genetics, *et al.*, 2019) By acknowledging these privacy concerns and implementing robust safeguards, we can ensure that DNA databases remain a valuable tool for forensic investigations while protecting the privacy and security of individuals' genetic information.

2.4 AES IN DNA DATABASES FOR FORENSIC DATA

While not yet widely implemented, Advanced Encryption Standard (AES) offers a promising approach to securing DNA data within forensic databases. Developed by the US National Institute of Standards and Technology (NIST), AES is a robust encryption standard that scrambles genetic information using a secret key of different lengths each consisting of different rounds. (fig 2.4) (National Human Genome Research Institute, *et al.*, 2020) This renders the data unreadable to anyone without the decryption key, providing an additional layer of security against unauthorized access or misuse.

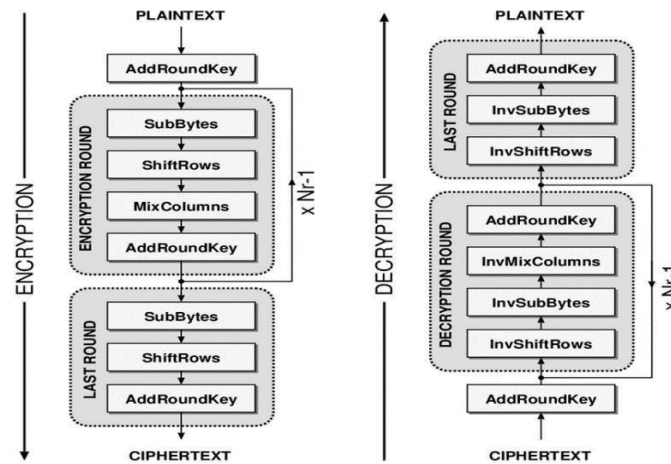


Fig:2.4: AES Encryption and Decryption Mechanism

A block diagram of AES encryption and decryption

However, implementing AES in forensic databases presents challenges. Searching encrypted databases for matches requires specialized techniques. Secure computation methods offer a solution, allowing investigators to compare DNA profiles and identify matches without decrypting the data itself. This mitigates privacy risks associated with decryption while maintaining investigative capabilities. (National Institute of Standards and Technology, *et al.*, 2001)

Researchers are actively exploring these secure computation methods to enhance the functionality of encrypted DNA databases. Additionally, robust access control measures remain essential to limit access to DNA databases only to authorized personnel. Implementing these measures in conjunction with AES encryption can significantly strengthen the security of sensitive genetic data.

2.5 ENHANCING SECURITY AND CONFIDENTIALITY IN FORENSIC DNA DATABASES

The future of forensic DNA databases hinges on advancements that bolster security, efficiency, and address privacy concerns. One promising area lies in bioinformatics and secure computation. Bioinformatics, the application of computer science and information technology to biological data, can be leveraged to develop sophisticated algorithms for DNA database analysis and searching. This could lead to faster and more efficient identification of matches, even in complex cases. (Wang, *et al.*, 2017) Additionally, secure computation techniques, previously mentioned, allow for searching encrypted databases without decryption, further enhancing security. Blockchain technology offers another avenue for improvement. This decentralized and distributed ledger system ensures data integrity and security. Implementing blockchain technology in DNA databases could yield several benefits. Its distributed nature makes it tamper-proof, as any alteration to a record would be immediately detectable across the entire network.

Additionally, blockchain can provide a verifiable audit trail for all data access and modifications, enhancing transparency and accountability. (Frere, *et al.*, 2015) Traditional DNA profiling methods analyse a limited number of short tandem repeat (STR) loci. However, Next-Generation Sequencing (NGS) technologies can sequence entire genomes or much larger portions of DNA, providing a more comprehensive genetic profile. While NGS holds immense potential for forensic investigations, ethical considerations and legal frameworks need to be established to address privacy concerns associated with storing and using whole genome sequences. (Alyahya, *et al.*, 2018) Forensic DNA databases like GeneProtect can continue to be a powerful tool for solving crimes, exonerating the wrongly convicted, and identifying unknown remains, while ensuring the highest level of security and respecting individual privacy rights.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

To build and develop an encrypted database using AES encryption and advanced encoding techniques to store and safeguard genetic information obtained from crime scene investigation.

3.2 OBJECTIVE

- To ensure that the stored data remains confidential and tamper-proof by encoding digital data into DNA and RNA sequences and employing advanced encryption algorithms such as AES (Advanced Encryption Standard) and random encryption.
- To provide categorical storage in the web application and to optimize database responsiveness through the utilization of advanced indexing and search algorithms, facilitating rapid retrieval of genetic information crucial for criminal investigations.
- To provide a diverse range of encryption keys assigned to different security requirements and access levels.

3.3 SCOPE

- To integrate blockchain technology for immutable and transparent record-keeping of database transactions, ensuring auditability and accountability while enhancing trust in forensic data integrity.
- To fortify data security measures by incorporating biometric authentication methods alongside encryption protocols, thereby bolstering access control and mitigating unauthorized access risks.
- To leverage machine learning and artificial intelligence algorithms for automated anomaly detection and pattern recognition within genetic datasets.
- To expand the application of the database to storage of newly identified chemical structures, research findings from research units and confidential health records.
- To implement comprehensive data backup and disaster recovery mechanisms to ensure the resilience and availability of genetic data in the event of system failures or unexpected incidents.

CHAPTER 4

MATERIALS AND METHODOLOGY

4.1 HARDWARE REQUIREMENTS

1. High-Performance Computing System:

Ensures computational efficiency for robust encryption/decryption and database operations.

2. RAID-Configured Solid-State Drive (SSDs) or High-Capacity Hard Drive:

Provides resilient and secure storage infrastructure with redundancy for encrypted genetic data.

3. Hardware Security Module (HSM):

Enhances data security through dedicated hardware for safeguarding encryption keys and cryptographic processes.

4. Networking Equipment:

Facilitates secure data transfer and communication between system components, ensuring reliable connectivity.

5. High-Resolution Monitor:

Supports seamless management and analysis of genetic data with optimal visual clarity.

6. Reliable Power Source and Backup System:

Ensures continuous operation and data integrity by mitigating risks associated with power disruptions.

4.2 SOFTWARE REQUIREMENTS

1. Windows and Linux Operating System:

Supports versatile server environments for seamless deployment and management of the GeneProtect system.

2. Web Server Software:

Enables hosting of the GeneProtect website, ensuring secure access to encrypted genetic data.

3. Python Scripting Language:

Empowers robust backend development and server-side scripting functionalities.

4. MySQL Database Management System:

Provides a secure and scalable platform for storing and managing genetic information.

5. OpenSSL, Crypto++, or PyCryptodome Encryption Library:

Implements industry-standard encryption algorithms to ensure double encryption of genetic data, enhancing security.

6. Intrusion Detection Software:

Monitors and detects unauthorized access attempts or security breaches within the GeneProtect database infrastructure.

7. Secure Socket Layer (SSL) or Transport Layer Security (TLS):

Encrypts data transmission channels to guarantee confidentiality and integrity during communication between the GeneProtect server and client applications.

4.3 DNA ENCRYPTION

GeneProtect utilizes symmetric encryption Advanced Encryption Standard (AES) algorithm to encrypt digital data. It utilizes symmetric encryption (fig 4.3.1) in which the same key is used for both encryption and decryption, ensuring efficient and secure data transmission.

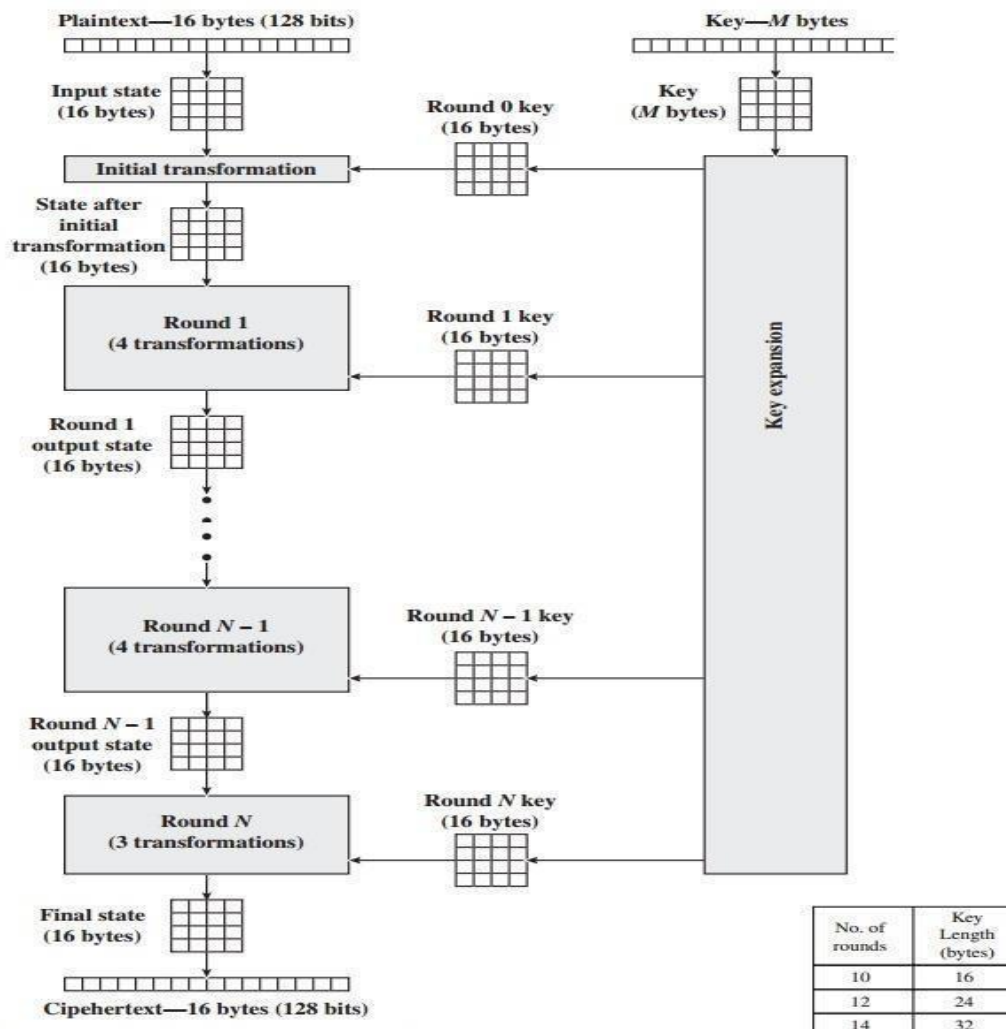


Fig:4.3.1: Flowchart of AES Encryption

The AES algorithm consists of sequences of 128 bits for each input and output. The cipher key is a sequence of 128, 192 or 256 bits. The number of rounds of encryption varies for each key size. Initially, the 128-bit key is expanded into eleven so-called round keys, each of them 128 bits in size. To ensure high security of the encryption each round contains transformation by utilizing the corresponding key of cipher.

1. Key Generation: Generate a cryptographically secure random encryption key of suitable length, typically 128, 192, or 256 bits, adhering to established cryptographic standards (fig 4.3.2).

Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

Fig:4.3.2: AES Parameters

The encryption key assigned for the input text may be of varying sizes of 128,192 or 256 bits. The number of rounds of encryption varies based on the encryption key assigned. These parameters are used for the generation of the key and initialization of the encryption process

- 2. Padding:** Apply padding to the plaintext data if its length is not a multiple of the block size (128 bits) to ensure uniformity and compatibility with the AES algorithm.
- 3. Initial Round Key Addition:** Combine the initial round key, derived from the encryption key, with the plaintext data to initialize the encryption process.
- 4. Encryption Rounds:** Execute a predetermined number of encryption rounds, determined by the key size (10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys), each comprising distinct transformations.
- 5. SubBytes Transformation:** Substitute each byte of the data matrix with a corresponding value from the AES S-box, a predefined lookup table, enhancing confusion and diffusion properties.
- 6. ShiftRows Transformation:** Perform cyclic left shifts on each row of the data matrix to introduce diffusion and achieve resistance against linear attacks.
- 7. MixColumns Transformation:** Apply a matrix multiplication operation to mix the columns of the data matrix, facilitating further diffusion and nonlinear interactions between bytes.

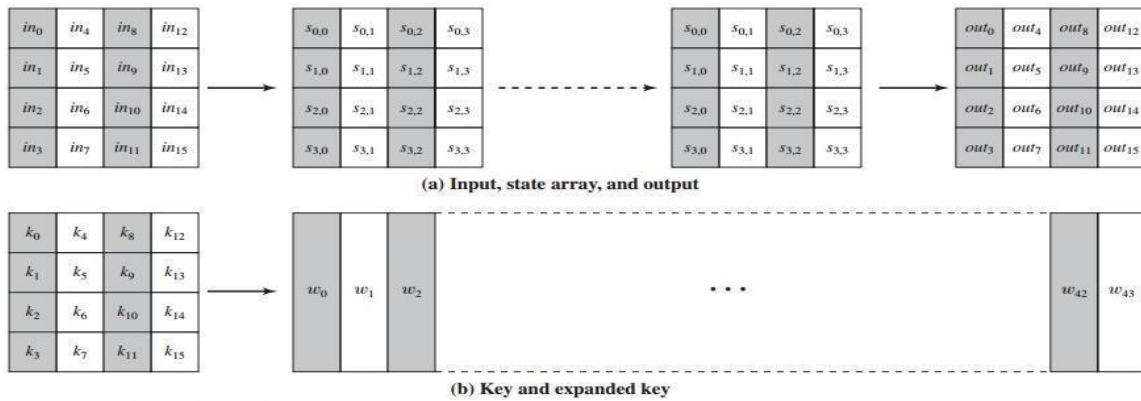


Fig :4.3.3: AES Data Structures

The cipher is views as alternate XOR operations followed by scrambling of the block to form datasets as shown

8. Add RoundKey Transformation: Combine the data matrix with a portion of the expanded encryption key through bitwise XOR operations (fig 4.3.3), introducing additional randomness and ensuring each round's uniqueness.

9. Final Round (for 128-bit keys): Conduct an additional encryption round to finalize the transformation process, reinforcing security and achieving a higher level of cryptographic strength.

10. Output: Obtain the resulting state matrix after the final encryption round, representing the ciphertext, which conceals the original plaintext data through robust encryption.

11. Store or Transmit: Securely store or transmit the ciphertext over trusted channels, safeguarding the confidentiality and integrity of sensitive information against unauthorized access or tampering.

AES ENCRYPTION CODE

```
import argparse
import os
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.primitives import padding
from cryptography.hazmat.backends import default_backend

def encrypt_file(file_path, key, key_size):
    if key_size == 128:

        iv = os.urandom(16)

        cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
        elif key_size == 192:
            iv = os.urandom(16)

            cipher = Cipher(algorithms.AES(key[:24]), modes.CBC(iv),
                            backend=default_backend())
        elif key_size == 256:

            iv = os.urandom(16)

            cipher = Cipher(algorithms.AES(key[:32]), modes.CBC(iv),
                            backend=default_backend())
        else:

            raise ValueError("Invalid key size. Key size must be 128, 192, or 256 bits.")
        encryptor = cipher.encryptor()
        with open(file_path, 'rb') as file:

            plaintext = file.read()

            # Pad the plaintext before encryption

            padder = padding.PKCS7(algorithms.AES.block_size).padder()
            padded_plaintext = padder.update(plaintext) + padder.finalize()
            ciphertext = encryptor.update(padded_plaintext) + encryptor.finalize()
            output_file = f'encrypted_file_{key_size}_bits.txt'
```

```

with open(output_file, 'wb') as encrypted_file:
    encrypted_file.write(ciphertext)

print(f"File encrypted with {key_size}-bit key. Encrypted file saved as {output_file}")
def decrypt_file(file_path, key, key_size):
    if key_size == 128:

32

    iv = os.urandom(16)

    cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
    elif key_size == 192:
        iv = os.urandom(16)

    cipher = Cipher(algorithms.AES(key[:24]), modes.CBC(iv),
        backend=default_backend())
    elif key_size == 256:

    iv = os.urandom(16)

    cipher = Cipher(algorithms.AES(key[:32]), modes.CBC(iv),
        backend=default_backend())
    else:

    raise ValueError("Invalid key size. Key size must be 128, 192, or 256 bits.")
    decryptor = cipher.decryptor()
    with open(file_path, 'rb') as file:

    ciphertext = file.read()

    plaintext = decryptor.update(ciphertext) + decryptor.finalize()
    output_file = f'decrypted_file.txt'
    with open(output_file, 'wb') as decrypted_file:
        decrypted_file.write(plaintext)
    print(f"File decrypted with {key_size}-bit key. Decrypted file saved as {output_file}")
def main():
    parser = argparse.ArgumentParser(description='Encrypt/Decrypt a file using AES

```

```

algorithm.')
parser.add_argument('mode', choices=['encrypt', 'decrypt'], help='Mode: encrypt or
decrypt')

parser.add_argument('file_path', help='Path to the file')

parser.add_argument('key_size', type=int, choices=[128, 192, 256], help='Key size
(128, 192,
or 256 bits)')

parser.add_argument('--key', required=True, help='AES key')
args = parser.parse_args()
# Validate the length of the key based on the key size
if len(args.key.encode('ascii')) != args.key_size // 8:
    raise ValueError(f'Invalid key length for {args.key_size}-bit key.')
key_bytes = args.key.encode('ascii')
if args.key_size == 192 and len(key_bytes) < 24:
    key_bytes = key_bytes.ljust(24, b'\x00')
elif args.key_size == 256 and len(key_bytes) < 32:
    key_bytes = key_bytes.ljust(32, b'\x00')
if args.mode == 'encrypt':

    encrypt_file(args.file_path, key_bytes, args.key_size)
else:
    decrypt_file(args.file_path, key_bytes, args.key_size)
33
if __name__ == "__main__":
    main()

```

4.4 DNA ENCODING

1. DNA/RNA Sequence Generation: Mapping digital data to nucleotide sequences using a bijective mapping scheme (fig 4.4), ensuring accurate representation of genetic information.

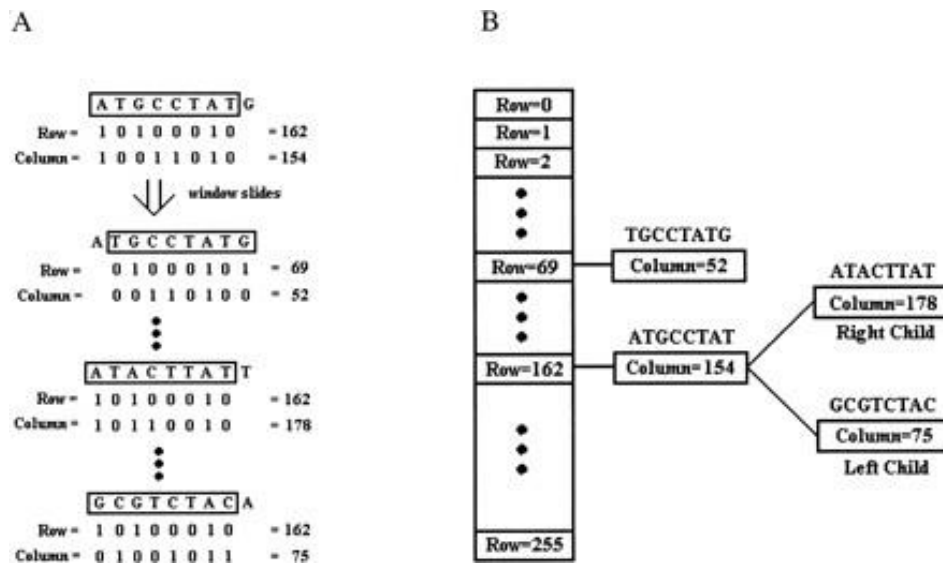


Fig:4.4 DNA Encoding

Systematic assigning of binary code to each nucleotide based on complementarity using bijective mapping scheme

2. Error Correction: To mitigate errors introduced during data encoding and decoding, GeneProtect integrates error correction techniques such as redundancy and error detection codes. These techniques enhance data reliability and integrity, minimizing the risk of data loss or corruption.

3. Reverse Mapping: During data retrieval, GeneProtect reverses the encoding process to decode DNA/RNA sequences back into digital format. This reverse mapping involves decoding nucleotide sequences based on the predefined encoding rules, ensuring accurate reconstruction of digital data.

Workflow of Encryption and Decryption of Digital Data in the Database

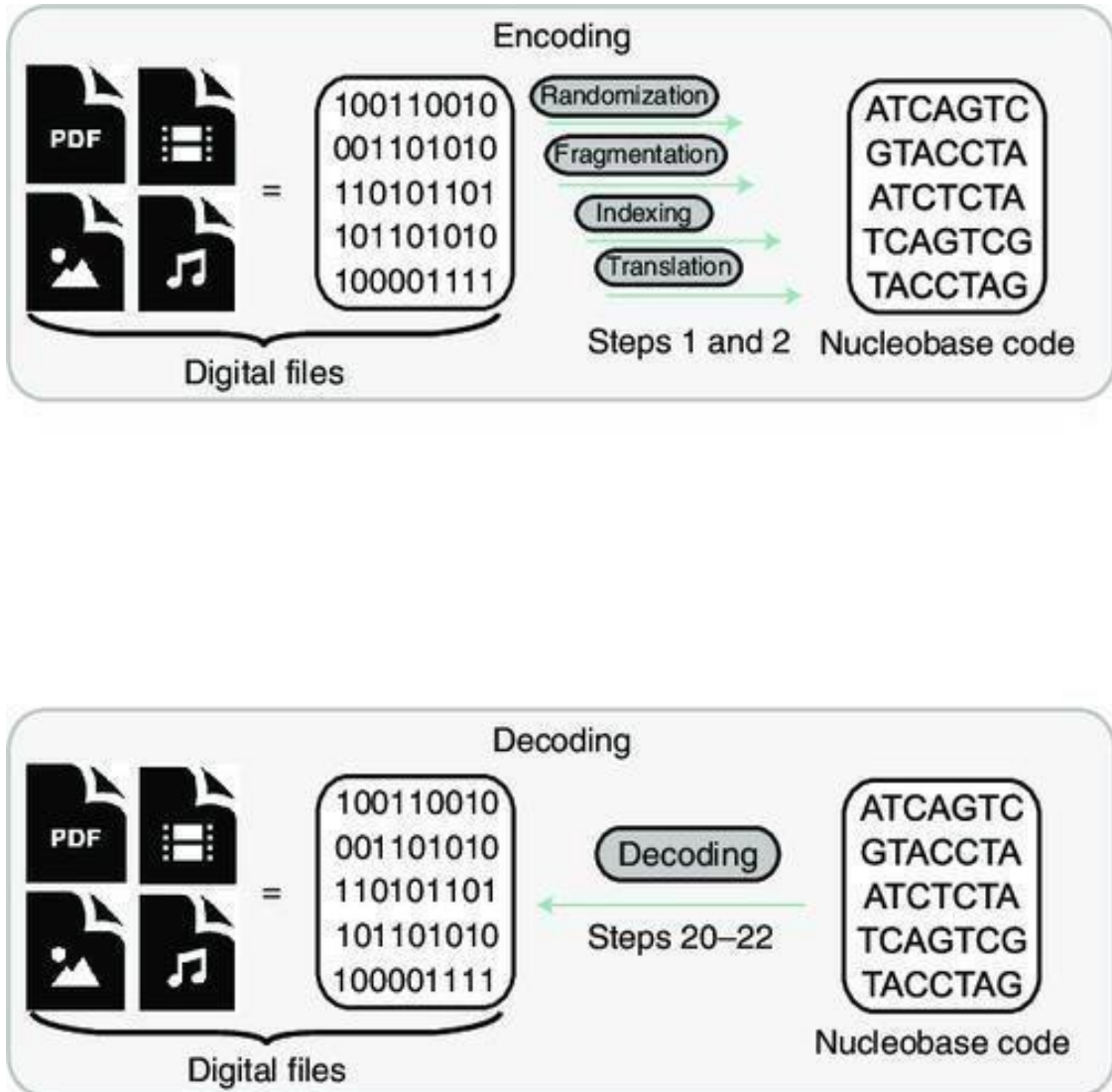


Fig:4.6: Encryption and Decryption of Data

Process overview of DNA data storage. The protocol includes the steps of mapping of bits to nucleotides, DNA writing, handling and storage of DNA, followed by reading and mapping of nucleotides back to bits. During encoding, bits are fragmented, randomized and appended with an index and redundancy. Then, bits are mapped to nucleotides

4.5 SQL SCHEMA FOR GENETIC DATA STORAGE

The SQL schema outlined below defines the database structure necessary for storing DNA/RNA sequences in the GeneProtect system. This schema ensures efficient storage, retrieval, and management of genetic data while maintaining data integrity and security. The **GeneticData** table includes essential columns such as **id** for unique identification, **dna_sequence** and **rna_sequence** for storing genetic sequences, **encrypted_data** for storing encrypted digital data, and **upload_date** for recording the timestamp of data uploads. This schema serves as the foundation for securely managing genetic information within the GeneProtect system, facilitating seamless integration and effective data management.

SOURCE CODE

```
CREATE TABLE GeneticData (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    dna_sequence TEXT NOT NULL,  
    rna_sequence TEXT NOT NULL,  
    encrypted_data TEXT NOT NULL,  
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Schema Description:

- **Table Name:** GeneticData
- **Columns:**
 - **id:** Unique identifier for each genetic data entry.
 - **dna_sequence:** Column to store DNA sequences.
 - **rna_sequence:** Column to store RNA sequences.
 - **encrypted_data:** Column to store encrypted digital data.
 - **upload_date:** Column to record the timestamp when the data was uploaded.

GUI CODE (COMMAND LINE TO GRAPHICAL USER INTERFACE)

```
from flask import Flask, render_template, request, jsonify
from aes_program import encrypt_file, decrypt_file
app = Flask(_name_template_folder="C:/Users/Aswin kumar/Downloads/pt2
project")
@app.route('/')
def index():
    return render_template('front_aes.html')
@app.route('/upload', methods=['POST'])
def upload_file():
    # Handle file upload and encryption
    file = request.files['file']
    key = request.form['key']

    key_size = int(request.form['key_size'])
    # Save the uploaded file temporarily
    file_path = r"C:/Users/Aswin kumar/Downloads/temp"
    file.save(file_path)
    # Encrypt the file
    encrypt_file(file_path, key, key_size)
    # Return success message or redirect to another page
    return jsonify({'message': 'File encrypted successfully'})
@app.route('/decrypt', methods=['POST'])
def decrypt_file():
    # Handle file decryption

    # Similar to file upload route but with decryption logic
    return jsonify({'message': 'File decrypted successfully'})
if __name__ == '__main__':
    app.run(debug=True)
```

4.6 RETRIEVAL OF DATA FROM THE DATABASE

1. Access Control:

Authenticate the user and ensure that they have appropriate permissions to access the encrypted data in the database.

2. Query Execution:

Execute a query or request to retrieve specific encrypted data from the database.

3. Data Retrieval:

Retrieve the encrypted data from the database based on the query parameters.

4. Decryption Key Feeding:

In AES encryption, the encryption key and decryption key are the same. Feed the decryption key to the database

5. Key Expansion:

Re-expand the encryption key to generate the round keys used for decryption. This is achieved through a process known as key expansion, which produces a series of round keys from the original encryption key.

6. Initial Round Key Addition:

Add the initial round key (the last round key used in encryption) to the ciphertext.

7. Decryption Rounds:

Perform a specified number of decryption rounds (10, 12, or 14) depending on the key size. Each round consists of InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey transformations.

8. InvSubBytes Transformation:

Substitute each byte of the state matrix with a corresponding byte from the inverse S-box (a predefined substitution table).

9. InvShiftRows Transformation:

Shift the rows of the state matrix cyclically to the right by different offsets, reversing the shift performed during encryption.

10. InvMixColumns Transformation:

Reverse the MixColumns operation by applying an inverse matrix multiplication operation to the state matrix.

11. AddRoundKey Transformation:

XOR the state matrix with a portion of the round key derived from the original encryption key using the key expansion process.

12. Final Round (for 128-bit keys):

Perform an additional round of InvSubBytes, InvShiftRows, and AddRoundKey transformations.

13. Output:

The resulting state matrix after the final decryption round represents the original plaintext.

14. Padding Removal:

Remove the padding from the decrypted plaintext to obtain the original data

CHAPTER 5

RESULTS AND DISCUSSION

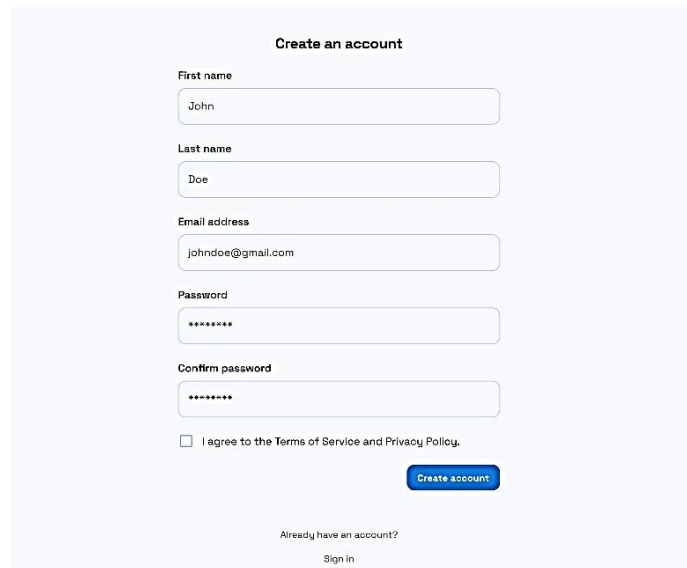
Through a comprehensive approach of integrating molecular biology and cybersecurity principles, GeneProtect has been utilized to address critical challenges in data security and encryption, ensuring secure and efficient molecular-level data storage solutions. The scalable architecture, built upon a client-server model, facilitates flexible and extensible data storage capabilities while ensuring robust security measures. The integration of advanced encryption algorithms, such as the Advanced Encryption Standard (AES), alongside bijective mapping schemes and error correction mechanisms, has enabled the secure encoding and decoding of digital information into DNA and RNA sequences.

5.1 WEB APPLICATION USER INTERFACE

The User Interface (UI) component provides a graphical interface for users to interact with the system. It facilitates actions such as uploading genetic data, retrieving data, and viewing data status. The web application consists of a sign-in page, authentication and verification, database storage, encryption, decryption and retrieval platform and customer-care helpline facility etc.

5.1.1 Registration or Login Page:

This page (fig 5.1.1) allows registration option for new user and login option for existing account holders. The necessary credentials are fed into the required sections in order to create an account or login to the existing account of the user.



Create an account

First name
John

Last name
Doe

Email address
johndoe@gmail.com

Password

Confirm password

☐ I agree to the Terms of Service and Privacy Policy.

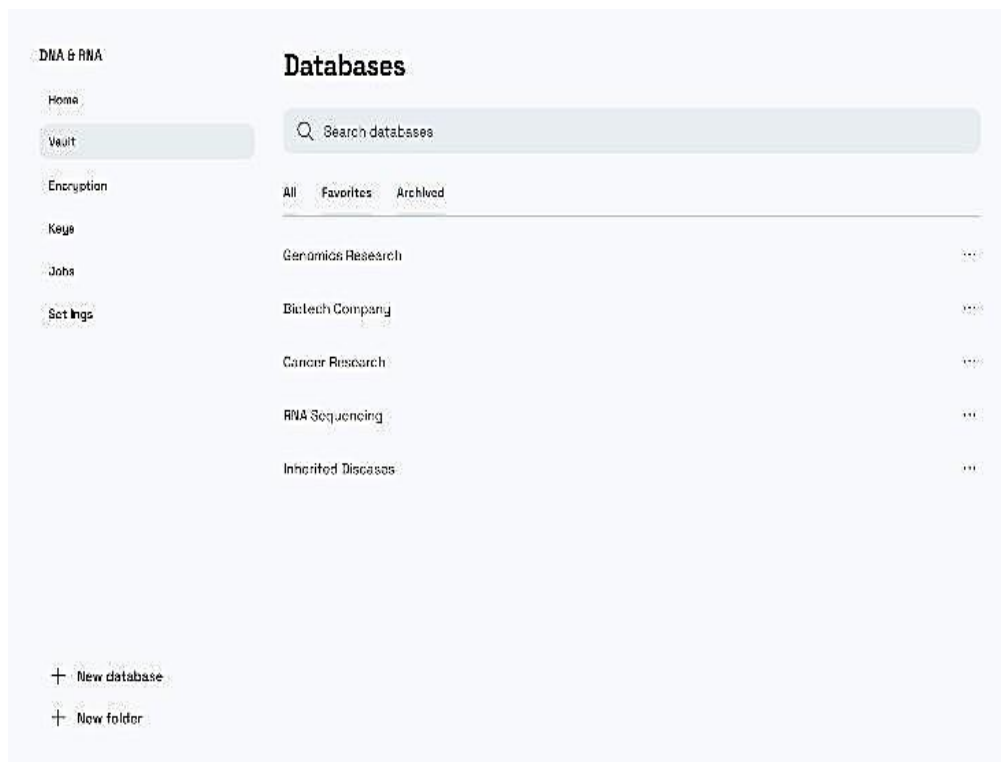
Create account

Already have an account?
Sign in

Fig 5.1.1: Account Creation

5.1.2 Database Access Page:

This page provides the database storage details (fig 5.1.2) and the data regarding the encrypted files in a secured environment.



DNA & RNA

Home

Vault

Encryption

Keys

Jobs

Settings

Databases

Search databases

All Favorites Archived

Genomics Research	...
Bioleth Company	...
Cancer Research	...
RNA Sequencing	...
Inherited Diseases	...

+ New database

+ New folder

Fig 5.1.2 Database Details

5.1.3 Digital Data Storage and Encryption:

The data is fed into the database with suitable name and description and encrypted with a key of required size based on the need of the user. This data can be chosen in the form of a text file with .txt file extension. A 16-character key is assigned by the user which is used for both encryption and decryption of the file. (fig 5.1.4).

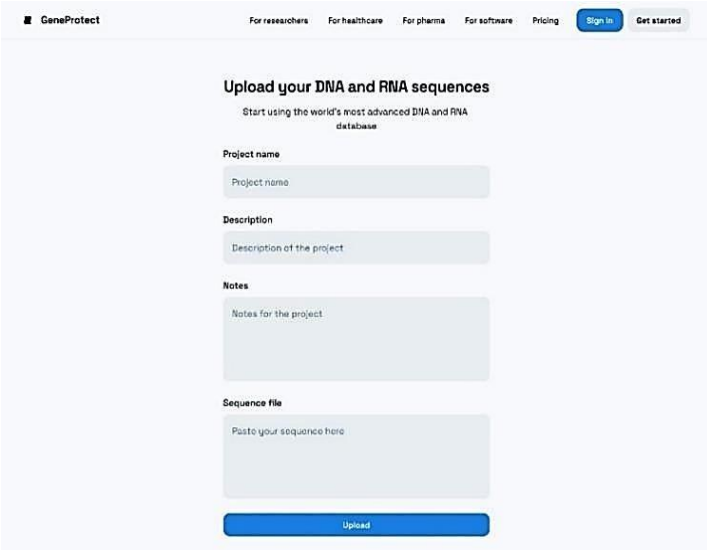
The screenshot shows the 'Upload your DNA and RNA sequences' page on the GeneProtect website. The page has a light blue header with the GeneProtect logo and navigation links: 'For researchers', 'For healthcare', 'For pharma', 'For software', 'Pricing', 'Sign in', and 'Get started'. The main content area is white and contains a form with the following sections: 'Project name' with a text input field containing 'Project name'; 'Description' with a text input field containing 'Description of the project'; 'Notes' with a larger text input field containing 'Notes for the project'; and 'Sequence file' with a text input field containing 'Paste your sequence here'. At the bottom of the form is a blue 'Upload' button.

Fig 5.1.3: DNA Data Storage

The encryption key must be preserved by the user for decrypting the encrypted genetic data file. The encryption key (fig 5.1.4) is generated and entered into the web page.

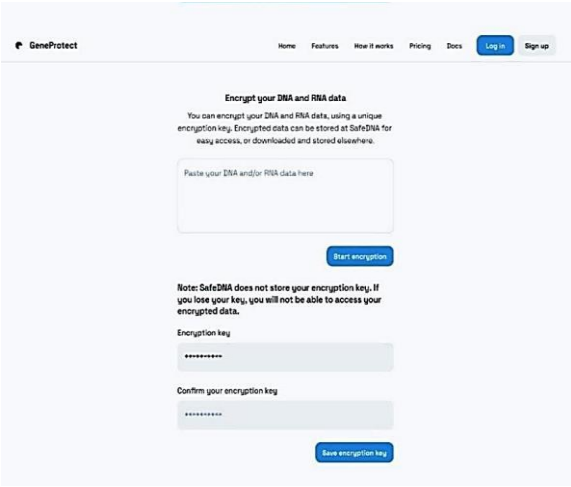
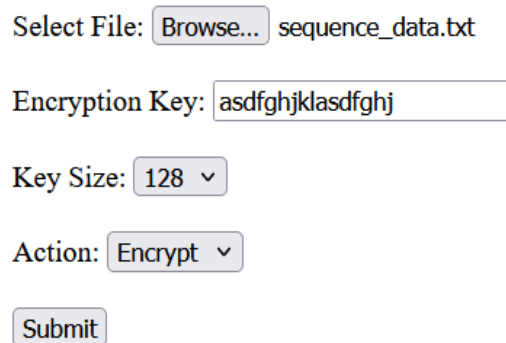
The screenshot shows the 'Encrypt your DNA and RNA data' page on the GeneProtect website. The page has a light blue header with the GeneProtect logo and navigation links: 'Home', 'Features', 'How it works', 'Pricing', 'Docs', 'Log in', and 'Sign up'. The main content area is white and contains a form with the following sections: 'Encrypt your DNA and RNA data' with a text input field containing 'Paste your DNA and/or RNA data here' and a blue 'Start encryption' button; a note stating 'Note: SafeDNA does not store your encryption key. If you lose your key, you will not be able to access your encrypted data.'; 'Encryption key' with a text input field containing '*****'; 'Confirm your encryption key' with a text input field containing '*****'; and a blue 'Save encryption key' button at the bottom.

Fig 5.1.4 Encryption Key Assigning

5.2.2 Opting Preferred Encryption Range:

The file encryption dialog box (fig 5.2.2) allows the user to select the file to be encrypted into the cloud storage or local storage. The user can choose the preferred encryption key size based on the level of security required and the purpose of encryption.

File Encryption



Select File: sequence_data.txt

Encryption Key:

Key Size: ▾

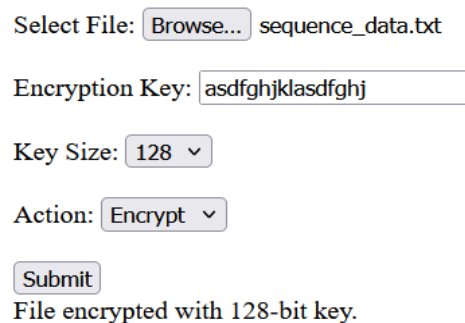
Action: ▾

Fig 5.2.2 File Encryption Key Size Selection

5.2.3 File Encryption:

The file undergoes n number of rounds of encryption and encoding based on the keysize selected by the user and a dialogue box appears showing the successful encryption of the file (fig 5.2.3) into the desired directory.

File Encryption



Select File: sequence_data.txt

Encryption Key:

Key Size: ▾

Action: ▾

File encrypted with 128-bit key.

Fig 5.2.3 Successful Encryption of File

5.2.4 Opting Required Decryption Range

When the user required the file to be decrypted, the dialogue box appears to choose the file from the directory. (fig 5.2.4) The same key is used to decrypt the file as AES encryption is a symmetric method of encryption. All of the necessary inputs are given.

File Encryption

Select File: encrypted_file_128_bits.txt

Encryption Key:

Key Size: ▾

Action: ▾

Fig 5.2.4: File Decryption Parameters

5.2.5 File Decryption:

The file is then successfully decrypted and saved in the file directory that was assigned for storage of the decrypted files to be used for further applications and analysis.

File Encryption

Select File: encrypted_file_128_bits.txt

Encryption Key:

Key Size: ▾

Action: ▾

File decrypted with 128-bit key.

Fig 5.2.5: Successful Decryption of File

When implementing AES encryption within GeneProtect to safeguard genetic data gathered from crime scenes, several pertinent considerations arise. Firstly, while AES is renowned for its security, the intricacies of key management present a challenge, demanding meticulous handling to avert mismanagement or compromise within its framework. Resource constraints, particularly in environments with limited computational resources or extensive data volumes, exacerbate these challenges.

By centralizing genetic information securely, GeneProtect enhances efficiency in forensic analysis, enabling seamless cross-referencing and bolstering accuracy in criminal investigations. This innovative platform represents a groundbreaking advancement in forensic science, revolutionizing genetic profiling for law enforcement globally in comparison with the existing databases.

CHAPTER 6

SUMMARY AND CONCLUSION

The GeneProtect DNA storage system is a sophisticated and comprehensive web application dedicated to ensuring the confidentiality and integrity of genetic data. Through the strategic encoding of digital data into DNA and RNA sequences and the deployment of robust encryption algorithms such as the Advanced Encryption Standard (AES), it stands as an essential tool against unauthorized access and tampering attempts. This fortification of genetic information underscores its important feature to upholding the highest standards of data security and integrity within forensic contexts. Furthermore, its meticulous implementation of categorical storage within the web application, coupled with the utilization of advanced indexing and search algorithms, exemplifies its dedication to optimizing database responsiveness.

This strategic approach not only facilitates the seamless organization and management of genetic data but also expedites the retrieval process vital for swift and effective criminal investigations. By prioritizing efficiency and efficacy, GeneProtect enhances the operational efficiency of forensic procedures, thereby streamlining investigative workflows and bolstering the overall efficacy of forensic DNA analysis. The provision of a diverse range of encryption keys created to distinct security requirements and access levels ensures to providing a multifaceted approach to data protection. This complicated encryption strategy ensures that sensitive genetic information is safeguarded with utmost precision, catering to the varied needs and security protocols inherent in forensic applications.

REFERENCES

Abdullah, A. M. (2017). Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data. *Journal of Science and Technology*, 1(1), 5-8.

<https://doi.org/10.1016/j.jbi.2021.103710>

Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science*, 266(5187), 1021-1024.

<https://doi.org/10.1126/science.7973651>

Arshad, S., Arshad, J., Khan, M. M., & Parkinson, S. (2021). Analysis of security and privacy challenges for DNA-genomics applications and databases. *Journal of Biomedical Informatics*, 116, 103710. <https://doi.org/10.1016/j.jbi.2021.103710>

Boneh, D., & Shoup, V. (2000). A graduate course in applied cryptography (Vol. 11). Springer Science & Business Media.

Bornholt, J., Lopez, R., Carmean, D. M., Ceze, L., Seelig, G., & Strauss, K. (2016). A DNA-based archival storage system. In *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems* (pp. 637-649).

Butler, J. M. (2014). The future of forensic DNA analysis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1674), 20140252.

<https://doi.org/10.1098/rstb.2014.0252>

Carter, A. B. (2017). Considerations for genomic data privacy and security when working in the cloud. *Human Genomics*, 11(1), 15. [https://doi.org/10.1186/s40246-](https://doi.org/10.1186/s40246-017-0104-8)

[017-0104-8](https://doi.org/10.1186/s40246-017-0104-8)

Ceze, L. (2018). Random access in large-scale DNA data storage. *Nature Biotechnology*, 36(3), 242-248. <https://doi.org/10.1038/nbt.4079>

Church, G. M., Gao, Y., & Kosuri, S. (2012). Next-generation digital information storage in DNA. *Science*, 337(6102), 1628.
<https://doi.org/10.1126/science.1226355>

Cox, A. J., Bauer, M. J., Goldman, N., & Giannoulatou, E. (2021). Ultra-high-throughput functional DNA-based data storage using synthetic gene networks. *Nature Communications*, 12(1), 1-11. <https://doi.org/10.1038/s41467-021-27732-5>

Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media.

Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644-654.
<https://doi.org/10.1109/TIT.1976.1055638>

Gao, Y., & Church, G. M. (2012). Improving molecular information storage by encoding DNA with more letters. *Journal of Molecular Evolution*, 74(4-5), 332-338.
<https://doi.org/10.1007/s00239-012-9520-y>

Gobi, M., & Sridevi, R. (2015). Multi-Biometric Authentication through Hybrid Cryptographic System. *International Journal of Computing Algorithm*, 4, 19-21.
<https://doi.org/10.20894/IJCOA.101.004.001.004>

Goldman, N., Bertone, P., Chen, S., Dessimoz, C., LeProust, E. M., Sipos, B., ... & Birney, E. (2013). Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature*, 494(7435), 77-80.
<https://doi.org/10.1038/nature11875>

Grass, R. N., Heckel, R., Puddu, M., Paunescu, D., & Stark, W. J. (2015). Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angewandte Chemie International Edition*, 54(8), 2552-2555.

<https://doi.org/10.1002/anie.201410773>

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177), 203-209. <https://doi.org/10.1090/S0025-5718-1987-0866109-5>

Lederma, L. (n.d.). Forensics. *Future Science*. <https://doi.org/10.2144/05391TN01>

Lipton, R. J. (1995). DNA solution of hard computational problems. *Science*, 268(5210), 542-545. <https://doi.org/10.1126/science.7725092>

Meiser, L., Antkowiak, P., Koch, J., Chen, W., Kohll, X., Stark, W., Heckel, R., & Grass, R. (2019). Reading and writing digital data in DNA. *Nature Protocols*, 15. <https://doi.org/10.1038/s41596-019-0244-5>

Naik, P. G. (2012). Asymmetric Key Encryption using Genetic Algorithm. <https://doi.org/10.1.1.1058.6984>

National Institute of Standards and Technology (NIST). (2001). Advanced Encryption Standard (AES). Federal Information Processing Standards Publication, 197. <https://doi.org/10.6028/NIST.FIPS.197>

Organick, L., Ang, S. D., Chen, Y. J., Lopez, R., Yekhanin, S., Makarychev, K., ... & Ceze, L. (2018). Random access in large-scale DNA data storage. *Biotechnology*, 36(3), 242-248. <https://doi.org/10.1038/nbt.4079>

Paar, C., & Pelzl, J. (2009). Understanding cryptography: A textbook for students and practitioners. Springer Science & Business Media.

Patel, S. M., Pandya, M. S., & Pathak, S. (2014). Enhancing AES algorithm with arithmetic coding. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1685-1691).

<https://doi.org/10.1109/ICACCI.2014.6968446>

Rogaway, P., & Shrimpton, T. (2006). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance.

Schneier, B. (2015). *Applied cryptography: Protocols, algorithms, and source code* in C. John Wiley & Sons.

Shendure, J., & Ji, H. (2008). Next-generation DNA sequencing. *Nature Biotechnology*, 26(10), 1135-1145. <https://doi.org/10.1038/nbt1486>

Sun, W., Zhang, N., Lou, W., & Hou, Y. T. (2018). When gene meets cloud: Enabling scalable and efficient range query on encrypted genomic data. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 4(3), 193-207. <https://doi.org/10.1109/TMBMC.2018.2849142>

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., ... & Hoffman, S. L. (2001). The sequence of the human genome. *Science*, 291(5507), 1304-1351. <https://doi.org/10.1126/science.1058040>

Wang, B., Song, W., Lou, W., & Hou, Y. T. (2017). Privacy-preserving pattern matching over encrypted genetic data in cloud computing. *IEEE Transactions on Cloud Computing*, 6(1), 179-192. <https://doi.org/10.1109/TCC.2017.2698284>

Watson, J. D., & Crick, F. H. (1953). Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171(4356), 737-738.