23056233_Sujita Sherpa.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:105979916

Submission Date

Jul 26, 2025, 10:03 PM GMT+5:45

Download Date

Jul 26, 2025, 10:06 PM GMT+5:45

File Name

23056233_Sujita Sherpa.docx

File Size

64.8 KB

100 Pages

10,169 Words

53,742 Characters

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

99 Not Cited or Quoted 13%
Matches with neither in-text citation nor quotation marks

13 Missing Quotations 1%
Matches that are still very similar to source material

1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

7%	Internet sources
3%	Publications
13%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

99 Not Cited or Quoted 13%
Matches with neither in-text citation nor quotation marks

13 Missing Quotations 1%
Matches that are still very similar to source material

1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

7%	Internet sources
3%	Publications
13%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	**www.coursehero.com**	**2%**
2	Submitted works	**University of West London on 2024-04-17**	**1%**
3	Internet	**www.megavideomovies.com**	**<1%**
4	Internet	**cyberw1ng.medium.com**	**<1%**
5	Internet	**epdf.pub**	**<1%**
6	Submitted works	**Bahrain Polytechnic on 2023-12-19**	**<1%**
7	Internet	**booksite.elsevier.com**	**<1%**
8	Submitted works	**University of Derby on 2016-02-18**	**<1%**
9	Submitted works	**Arab Open University on 2016-12-15**	**<1%**
10	Internet	**routeicr.com**	**<1%**

11	Submitted works	University of West London on 2024-04-16	<1%
12	Submitted works	University of West London on 2025-04-19	<1%
13	Submitted works	Charles Darwin University on 2024-10-19	<1%
14	Submitted works	University of West London on 2023-04-28	<1%
15	Submitted works	Bahrain Polytechnic on 2023-12-15	<1%
16	Submitted works	Manipal University Jaipur Online on 2025-06-28	<1%
17	Internet	www.examcollection.com	<1%
18	Submitted works	London Metropolitan University on 2023-04-19	<1%
19	Submitted works	University of West London on 2025-04-18	<1%
20	Internet	www.researchgate.net	<1%
21	Submitted works	American Public University System on 2024-02-26	<1%
22	Submitted works	University of West London on 2024-04-17	<1%
23	Submitted works	University of West London on 2025-04-19	<1%
24	Internet	www.teanglann.ie	<1%

25	Submitted works	University of West London on 2023-05-03	<1%
26	Submitted works	University of West London on 2024-04-19	<1%
27	Submitted works	University of Maryland, Global Campus on 2023-04-04	<1%
28	Internet	csalpha.ist.unomaha.edu	<1%
29	Internet	mathbook.pugetsound.edu	<1%
30	Submitted works	Bahrain Polytechnic on 2023-12-19	<1%
31	Submitted works	Birkbeck College on 2020-11-08	<1%
32	Publication	Nik Zulkarnaen Khidzir, Shekh Abdullah-Al-Musa Ahmed. "Guardians of Data - A C...	<1%
33	Submitted works	University of West London on 2023-05-08	<1%
34	Submitted works	University of West London on 2025-04-05	<1%
35	Internet	securityboulevard.com	<1%
36	Submitted works	Bahrain Polytechnic on 2023-12-19	<1%
37	Submitted works	Coventry University on 2008-09-11	<1%
38	Submitted works	Manipal University Jaipur Online on 2025-06-26	<1%

39	Submitted works	Arab Open University on 2024-11-07	<1%
40	Submitted works	Arab Open University on 2024-11-07	<1%
41	Submitted works	Purdue University on 2023-10-04	<1%
42	Submitted works	University of West London on 2024-04-23	<1%
43	Submitted works	Pentecost University College on 2020-09-09	<1%
44	Submitted works	Westford School of Management on 2025-05-23	<1%
45	Submitted works	Queen Mary and Westfield College on 2018-12-04	<1%
46	Submitted works	K12 Incorporated on 2024-05-30	<1%
47	Submitted works	Manipal University Jaipur Online on 2025-05-30	<1%
48	Submitted works	Manipal University Jaipur Online on 2025-06-21	<1%
49	Submitted works	St. Leonard's College on 2017-06-17	<1%
50	Submitted works	The University of the West of Scotland on 2023-04-19	<1%
51	Submitted works	University of Derby on 2016-02-17	<1%
52	Submitted works	University of Essex on 2024-09-18	<1%

53 Submitted works	
University of West London on 2025-04-05	<1%
54 Internet	
alumni-portal.sasin.edu	<1%
55 Submitted works	
Visvesvaraya Technological University on 2014-11-08	<1%
56 Submitted works	
Queensland University of Technology on 2019-03-26	<1%
57 Submitted works	
University of West London on 2025-04-07	<1%

1. Introduction

Security and cryptography are interconnected concepts that are vital for safeguarding our digital environment. They are important for preserving information in the current digital era. This paper aims to explore various encryption types, including symmetric and asymmetric encryption, as well as their historical evolution. Each method has its benefits and drawbacks, which are essential for maintaining the confidentiality, integrity, and accessibility of information. (Chanana & Divya, 2019)

This paper will examine how cryptography converts data into a format that is unreadable except by those with proper authorization. The discussion will highlight how cryptography changes data into an unreadable state, available solely to those with the right permissions, serving as the foundation of contemporary cybersecurity. Understanding these principles is crucial for anyone interested in securing digital information. Additionally, it will highlight the evolution of encryption techniques and their growing significance in protecting information in today’s interconnected world.

1.1 What is security?

The widespread use of the Internet has greatly complicated computer security. In the past, computer systems were typically used by a small group of individuals, often within the same organization. Programs were usually created in-house or by a limited number of vendors, and data was transferred manually via tapes or disks. However, today, with nearly half a billion people worldwide connected to the Internet, new security challenges have emerged. (Thomas & Ciza, 2020)

As technology advances, so do cyber threats. Malware, viruses, and trojans are becoming more sophisticated, making cyberattacks more aggressive, covert, and potentially devastating. Cybercriminals are increasingly using hard-to-detect techniques to breach systems and steal sensitive information. Techniques like man-in-the-middle attacks, phishing, ransomware, denial-of-service attacks, SQL injection, zero-day exploits, and cross-site scripting are examples of the evolving threats that make digital systems more vulnerable. (Thomas & Ciza, 2020)

To protect against these threats, computer systems must be secured. Security involves safeguarding systems, networks, and data from unauthorized access, theft, damage, and other risks. It includes a range of practices and technologies designed to ensure the confidentiality, integrity, and availability of information. This can be achieved through measures such as firewalls, encryption, authentication protocols, regular system updates, and cryptography. (Stosic & Lazar, 2013) Cryptography plays a critical role in securing communication and protecting data from various types of cyberattacks (Chanana & Divya, 2019).

1.2 Core Principles of Security

Cybersecurity is now a key part of keeping organizations successful and secure in the modern digital landscape. In response to the increasing risks and challenges posed by cyber threats, organizations are implementing robust security strategies that integrate systems, tools, and regulations. These efforts help protect their IT systems from new dangers. By focusing on these steps, organizations can build strong defences, safeguard important data, and maintain trust in a world that is more connected than ever. (Abdalbasit, 2019).

The core of security is built on three essential components known as the CIA Triad:

1.2.1 Confidentiality

Confidentiality refers to protecting sensitive information/data from being accessed by unauthorized individuals. This means that only those who have the authority can view certain data whether it's personal customer details, financial records, or whatever data it may be. To achieve confidentiality, techniques such as encryption, access controls, and authentication like passwords and multi-factor authentication, are used to safeguard the data. For instance, only authorized employees should be able to access a company's financial database ensuring that sensitive information like salaries or profit margins is not exposed to the public or competitors. (Osara & Osazuwa, 2023).

1.2.2 Integrity

Maintaining integrity ensures that data remains precise, consistent, and reliable. It protects against unauthorized changes, whether intentional or accidental, that could compromise the quality or reliability of the information. Preserving integrity means that when someone accesses data, they can be assured that it has not been altered or manipulated. For example, a financial report should remain unchanged after receiving approval, guaranteeing that the figures presented to stakeholders are both accurate and trustworthy. (Osara & Osazuwa, 2023).

1.2.3 Availability

48 Ensuring availability means that information and systems are reachable to authorized users whenever necessary. This encompasses maintaining operational systems, reducing downtime, and swiftly recovering from interruptions like hardware malfunctions, cyber threats, or natural catastrophes. For instance, a company's website should stay functional even during high-traffic periods or cyberattacks, enabling customers to use services without any disruption. (Osara & Osazuwa, 2023).

The combination of these three principles forms the basis of a secure information environment. When an organization neglects confidentiality, integrity, and availability, its data could be vulnerable to attacks, misuse, or loss, leading to financial damage, legal issues, and a decline in trust.

1.3 Cryptography

Cryptography involves the creation and use of coded algorithms to safeguard information, ensuring that only those with the right permissions and access can utilize it. It secures communication by transforming information, known as plaintext, into a format that is unreadable to anyone lacking the means to decode it. (Chanana & Divya, 2019).

In cryptography, data in its original and legible state is called "plaintext." To shield this data from unauthorized access, it undergoes a process called "encryption," which converts the plaintext into an unreadable format referred to as "ciphertext." This conversion depends on a set of guidelines known as "encryption algorithms." These algorithms require an encryption key, which is a unique piece of information provided along with the initial message to facilitate the transformation into the scrambled ciphertext format. When retrieval of the original information is needed, a method called "decryption" is used. This method necessitates the application of a specific decryption algorithm alongside the correct key to revert the ciphertext into readable plaintext. This process allows the user to interpret the data as it was originally intended to be understood. The entire procedure safeguards sensitive information during both storage and transmission, ensuring that only those with the authorized keys and tools can access it. (Abdalbasit, 2019) In the modern digital landscape, cryptography is crucial in our everyday lives, ensuring that our private data, such as credit card numbers, e-commerce transactions, and even text messages, stays secure and confidential (Chanana & Divya, 2019).

The word "encryption" has its roots in the Greek word kryptos, meaning hidden. Historically, people have sought to conceal sensitive information from those not authorized to see it. In ancient eras, individuals would substitute parts of information with symbols, numbers, or images. Different groups have utilized cryptography for various reasons. The Assyrians aimed to protect their trade secrets concerning pottery crafting. The Chinese intended to keep their methods of silk production secret. The Germans focused on shielding their military intelligence. (F & F, 1977) With the evolution of computers and the internet, numerous companies, businesses, and industries have had to secure their sensitive data against intruders. This paper will explore how different encryption techniques have evolved from ancient times to the current era.

1.3.1 History of Cryptography

Cryptography, an art as well as a scientific field focused on safeguarding information, has been utilized for millennia to secure communication. Nevertheless, it was around a hundred years ago that cryptography started to be examined as a formal area of scientific study. (Abdalbasit, 2019)

46 The roots of cryptography can be traced back to ancient societies, where individuals devised different techniques to shield their messages from unauthorized individuals. The earliest known examples of efforts to secure information were found in a carving
21 dating around 1900 BC, situated in the tomb of the noble Khnumhotep II in Egypt. This

carving, known as the "Cryptography Inscription," described a method for obscuring the meaning of hieroglyphic texts by utilizing symbols to represent each letter. These symbols were then rearranged in a specific way to make the writing difficult to decipher. (Thwate inc, 2013).

The main intention behind the "Cryptography Inscription" was not to hide the message but to enhance its visual presentation, making it look more formal and dignified. Even though the symbols in the inscription were shuffled, they remained understandable to those who were knowledgeable about the substitution method used. This suggests that the inscription was crafted for a particular audience already familiar with the technique, rather than intending to keep the message hidden from all possible viewers. (Qadir & agarwal, 2024).

Around 100 BC, Julius Caesar employed a simple encryption technique known as the Caesar cipher to send confidential messages to the leaders of his army. This cipher is a classic form of substitution cipher, where each letter in the original message is

18 replaced with another letter that has been advanced by a set number of positions in the alphabet. In the case of the Caesar cipher, this shift is set to three letters. (Thwate inc, 2013).

18 For example, the character 'A' changes to 'D,' while the character 'X' cycles back to the start of the alphabet and becomes 'A.' Although this technique was quite reliable in its simplicity of decryption highlights a significant flaw of substitution ciphers, as they are vulnerable to scrutiny through letter frequency analysis, which uncovers how often each letter appears in the text and aids in breaking the code. (Thwate inc, 2013).

49 In the 16th century, the cryptographer Blaise de Vigenère created a more advanced type of encryption called the Vigenère cipher, which employed a confidential encryption key to improve security. This ground-breaking technique involved continuously repeating the selected key throughout the message, leading to a more intricate combination of letters through modular arithmetic. Although the Vigenère cipher was eventually broken, it represented a significant step forward compared to the simpler Caesar cipher by introducing the fundamental concept of using keys for encryption. In contrast to earlier techniques that depended only on the strength of the encryption algorithm, the security of the Vigenère cipher primarily relied on keeping the key confidential, emphasizing the necessity to protect the key from unauthorized users. This shift in approach significantly changed the landscape of cryptography at that time. Vigenère ciphers make use of a table referred to as the Vigenère Square. (Qadir & agarwal, 2024).

For example, if the key "KEY" is used to encrypt "HELLO WORLD", the letters in the original text are aligned with the key, which is repeated as needed to match the length of the plaintext. The encryption process involves shifting each letter of the plaintext by the number of positions specified by the corresponding letter of the key, using the Vigenère square. Each plaintext letter is replaced by the letter found at the intersection of the plaintext row and key column.

Plaintext

HELLO WORLD

Key

KEYKEYKEYKE

Encrypted Message

RIJVS UYVJN

Table 1. Example of Vigenère Cipher

By the 19th century, technological advancements led to electromechanical encryption devices. Hebern developed the rotor machine, which used a rotating disc to encrypt messages. Each key press resulted in a unique substitution, and the disc rotated with each character, changing the encryption for subsequent letters. However, this system was still vulnerable to frequency analysis. (National Museum Of American History, 2020).

41 A significant advancement occurred with the creation of the Enigma machine, developed by Arthur Scherbius towards the end of World War I and extensively utilized by Germany during World War II. The machine employed multiple rotors that rotated at varying speeds, generating highly complex ciphers. The encryption relied on the initial rotor settings, which served as the key. While the Enigma machine was more sophisticated than previous systems, its code was eventually broken by Polish and British cryptographers, who developed methods to deduce the daily keys. Until World War II, cryptography was primarily used for military purposes to protect classified information. After the war, it began gaining attention in the commercial sector, as businesses sought ways to secure their data. (Qadir & agarwal, 2024).

In the 1970s, IBM identified a growing need for robust encryption techniques to safeguard sensitive data from unauthorized access. To address this, the company created an innovative cipher known as Lucifer, aimed at delivering a high degree of security. This project was led by the well-known cryptographer Horst Feistel, who used advanced techniques to create a cipher that met the increasing need for secure encryption. As the cipher attracted interest, the National Bureau of Standards (currently known as NIST) evaluated its performance and acknowledged its capability to securely encrypt information. The agency formally adopted the cipher, rebranding it as the Data Encryption Standard (DES), which became the official method for encrypting non-classified data. (Konheim, 2018). DES quickly became widely used across various industries, offering a foundational layer of security for everything from financial transactions to personal data. (Sorkin, 1984)

However, as technology advanced, particularly in the realm of computing power, the weaknesses of DES became apparent. The cipher’s relatively short key length, which was originally considered secure, rendered it vulnerable to increasingly efficient brute-force attacks. Attackers could methodically try every possible key until they found the correct one, and as computers grew faster, this type of attack became more feasible. Over time, these vulnerabilities led to a decline in the trust and usage of DES, ultimately prompting the development of stronger encryption standards to replace it. (Konheim, 2018) In 1997, NIST requested proposals for a new encryption standard. After reviewing 50 submissions, it selected Rijndael in 2000 and named it AES (Advanced Encryption Standard) which remains a widely used standard for symmetric encryption today (Dworkin, 2023).

Recent developments in quantum computing have sparked worries regarding the security of existing cryptographic techniques. One potential remedy is quantum cryptography. In this method, "quantum" designates the tiniest measurable unit, a photon, or light particle. As photons move, they oscillate, and the orientation can convey encrypted information. If an individual tries to intercept the communication, the orientation shifts, making any eavesdropping immediately noticeable. This guarantees secure data transmission. (Schneider & Smalley, 2024).In 2016, NIST launched an international initiative aimed at creating quantum-resistant algorithms. By 2020, the organization revealed four finalists, which represented a significant advancement in protecting data from possible quantum threats. (NIST.gov, June 5, 2022)

1.4 Types of Algorithms

32 There are two common types of encryptions in use today:

Symmetric Algorithm,

Asymmetric Algorithm.

1.4.1 Symmetric Algorithm

16 Symmetric encryption is a technique for safeguarding data. It utilizes the same key for
both encrypting and decrypting information. The sender and receiver need to have a
shared key so that the receiver can interpret the encrypted message. In symmetric
encryption, the initial data is transformed into a jumbled format known as ciphertext.
43 This renders it unreadable to anyone who does not possess the secret key, allowing
the receiver to use the same key to revert the ciphertext to its original, readable state.
The shared key may consist of a password, a code, or a sequence of random
characters generated by a secure system. This encryption method is efficient and
particularly effective for securing large volumes of data. It is frequently employed for
files, databases, and online communications. However, the effectiveness of symmetric
encryption is heavily reliant on the safeguarding of the key. (Chandra, et al., 2014)

Figure 4. Explanation of the whole process of symmetric encryption (Mohammad Ubaidullah Bokhari, August,2016).

17 Symmetric encryption uses the same key for both encryption and decryption. Algorithms like AES, DES, and Blowfish process data in fixed-size blocks with specific key lengths. The plaintext is the original data, which is converted into ciphertext using a secret key. Decryption then reverses the process to recover the original plaintext. (Mohammad Ubaidullah Bokhari, August,2016).

S.N

20 Algorithm Name

Key size

Uniqueness in regards to the technique

Vulnerable to attack

1

Cesar Cipher

23

Employs a straightforward substitution of letters based on a fixed shift value.

Brute Force Attack

2

Playfair Cipher

25

Encodes pairs of letters using a 5x5 matrix constructed with a key and the remaining alphabet.

Prone to frequency analysis and brute force attacks.

3

Vernan Cipher

126,192,256 Bits

Utilizes XOR operations between the plaintext and key bits, with variable key sizes.

Exposed to known plaintext attacks.

4

AES

32-34 Bits

Implements 10, 12, or 14 rounds involving substitution and permutation operations.

Vulnerable to side-channel and known plaintext attacks.

5

20 Blowfish

32-448 168 bits

Features a key-independent S-box with 16 rounds, supporting flexible key lengths.

Susceptible to weak key issues and second-order differential attacks.

6

TDES

112 bytes or 168 bits

Uses a Feistel network structure with three unique keys and 148 rounds of processing.

Can be attacked via chosen or known plaintext, though mainly theoretical.

7

DES

56 bits

Operates with a Feistel structure, incorporating 16 rounds, left shifts, and 32-bit substitutions.

Targeted by brute force, linear cryptanalysis, and differential attacks.

Table 2. Symmetric encryption types examples (Mohammad Ubaidullah Bokhari, August,2016)

1.4.1.1 Advantages of Symmetric Algorithm

A symmetric cryptosystem operates at a faster speed,

27 Encrypted information can be transmitted over a link even if interception is risky. Since no key is transmitted along with the data, the likelihood of the data being decrypted is effectively zero,

27 It employs password authentication to verify the identity of the recipient,

A message can only be decrypted by a system that possesses the secret key.

1.4.1.2 Disadvantages of Symmetric Algorithm

It faces a key distribution challenge, as the secret key needs to be shared with the receiving system before sending the actual message, given that all electronic communication can be susceptible to interception, the most secure method for exchanging keys is in person,

They cannot provide digital signatures that are irrefutable,

To ensure effective cryptography, it is essential to frequently change the key, preferably for every communication session.

1.4.2 Asymmetric Encryption Algorithm

37 Asymmetric encryption referred to as public key encryption, operates differently
17 compared to symmetric encryption. This form of encryption employs two keys: a public
44 key for encrypting information and a private key for the decryption process. The public
key is accessible to everyone, while the private key remains confidential. This

technique is commonly utilized to secure communication over the Internet. Asymmetric encryption relies on the RSA algorithm, which is a reliable option. It utilizes large key pairs, typically 1024 or 2048 bits, to provide robust security. Although it is slower than symmetric encryption, it ensures message confidentiality and authenticates the identity of the sender. (Khan, et al., October, 2018)

Figure 5. Asymmetric encryption algorithm process (Infosec Insights, April 25, 2020).

S.N

Algorithm name

Key size

Uniqueness in technique

Vulnerable to Attack

1

RSA

1024 bits, Commonly

2048 bits

Relies on the difficulty of factoring large integers. Uses two large, prime numbers and an auxiliary value for the public key generation.

Vulnerable to

quantum

attacks (future risk)

2

DH (Diffie Hellman)

-

Allows secure exchange of a secret key over an insecure medium using discrete algorithms in a finite field

Man-in-the-middle attacks, algorithm-based attacks

3

ECC

160 bits (as secure as RSA with 1024 bits)

Employs elliptic curve principles to create cryptographic keys that are quicker, more compact, and more efficient. Perfect for devices with constrained resources.

Vulnerable to side-channel attacks

4

DSA

Dynamic depends on data size

Generates digital signatures for authenticity using asymmetric Cryptography. Suitable for detecting forgery or tampering.

Susceptible to weak random number generators.

Table 3. Asymmetric Algorithm encryption Examples (Annie Badman, 8 August, 2024).

1.4.2.1 Advantages of the Asymmetric Algorithm

Only the private key must be kept confidential, making management simpler. A pair of private and public keys can stay the same for extended durations, depending on their application.

Public-key systems frequently offer effective digital signature options and necessitate fewer keys in extensive networks compared to symmetric-key systems.

1.4.2.2 Disadvantages of the Asymmetric Algorithm

Asymmetric encryption techniques are notably slower than those using symmetric keys. The key sizes are considerably larger in comparison to symmetric encryption, necessitating more storage and processing power.

Public-key signatures are larger than those used for data origin authentication in symmetric-key methods, resulting in greater resource consumption.

2. Background Studies of the Selected Algorithm

For this Coursework, the algorithm that I'll be working on is the Playfair Cipher, alongside Binary operations and XOR for further encryption.

2.1 What is the Playfair Cipher?

The Playfair cipher is a traditional method of encryption that was created in the late 1800s by Charles Wheatstone. Its purpose was to enhance basic substitution ciphers by processing pairs of letters, known as digraphs, rather than individual characters (Gustavus & , 2013).

The main concept of the Playfair cipher involves utilizing a 5x5 grid, with the arrangement of letters dictated by a key. Each two-letter combination from the plaintext is encoded by referencing the matrix and then following distinct guidelines to generate the ciphertext. (Gupta & Das, January, 2020). Initially, the Playfair cipher's key is usually a keyword or phrase (such as “keyword”) from which duplicate letters are 23 eliminated. This keyword is then utilized to populate a 5x5 grid. The letters from the keyword are placed into the grid sequentially, row by row. If any letter in the keyword appears more than once, it is substituted with 'X'. After populating the grid with the 30 keyword, the remaining spaces are filled with the rest of the alphabet in order, omitting 34 'J' since it is typically combined with 'I'. (Wang, 2021). For Example, if the keyword is “SUJITA”, the Matrix might look like this:

S

U

J/I

3 T

A

B

C

D

E

F

G

H

K

L

M

N

O

P

Q

R

V

W

X

Y

Z

Table 4. 5x5 matrix example

The Play Fair algorithm utilizes a 5X5 letter matrix that is formed using a specific key. The encryption of plaintext follows these guidelines (Wang, 2021).

Rule 01: If there are repeating letters in the plaintext that are part of the same pair, they are separated by an additional letter, often "X".

Rule 02: Letters from plaintext that are located in the same row are substituted with letters to their right.

Rule 03: Letters from plaintext that share the same column are replaced by the letters that are positioned directly below.

Rule 04: If both letters in a plaintext pair are present, they are exchanged for the letters found in their respective row and column of the other plaintext letter.

2.1.1 Analyzing the Playfair Cipher

Example:

Key: DIXANT

Plaintext: KHADKA

D

I/J

X

A

N

T

24 B

C

E

F

G

H

K

L

M

O

P

Q

R

S

U

V

W

Y

Z

Table 5. 5x5 matrix table

Solution,

Here,

To cipher,

Plaintext= KHADKA

Now,

Split the Plaintext

Plaintext: KHADKA → Split into pairs: KH, AD, KA

KH AD KA

Here,

Encrypt Each Pair Using the Rules

For KH, we'll use Rule 3 as both K and H are in the same column. Move each letter down one position in the column:

K → L

H → K

Result: LK

For AD,

11 Rectangle Rule: A and D form the corners of a rectangle. Replace each with the letter on the same row but at the opposite corner:

A → N

D → I

Result: NI

For KA,

12 Rule 3 (Same Column): Both K and A are in the same column. Move each letter down one position in the column:

K → L

A → X

Result: LX

Hence, the cipher text for KHADKA = LKNILX

Lastly,

Decryption Process,

Ciphertext: LKNILX → Split into pairs: LK, NI, LX

Decrypt Each Pair Using the Rules,

For LK,

12 Rule 3 (Same Column): Both L and K are in the same column. Move each letter up one position in the column:

L → K

K → H

Result: KH

Again,

For NI,

11 Rectangle Rule: N and I form the corners of a rectangle. Replace each with the letter on the same row but at the opposite corner:

N → A

I → D

Result: AD

Again,

For LX,

12 Rule 3 (Same Column): Both L and X are in the same column. Move each letter up one position in the column:

L → K

X → A

Result: KA

Lastly,

Combining the Deciphered result,

KH AD KA = KHADKA

14 2.1.2 History of Playfair Cipher

14 The Playfair Cipher is a significant early encryption technique that emerged in the mid-19th century. Created by Sir Charles Wheatstone in 1854, it was later named in honor of his associate Baron Playfair. This type of polygraphic substitution cipher is among the first methods used for message encryption. The Playfair Cipher was designed to meet the growing demand for more secure encryption methods during an era when initial cryptographic strategies were becoming vital. Sir Charles Wheatstone, a British

researcher recognized for his contributions to physics, acoustics, and cryptography, worked alongside Baron Playfair, a Scottish diplomat, to develop this technique. 53 (Wang, 2021) The cipher formulates a 5x5 letter grid based on a keyword and fills it 36 with all the letters of the alphabet, excluding "J" (or occasionally "Q"). The plaintext is separated into pairs of letters, and each pair goes through specific encryption rules: If 2 the letters appear in the same row, they are substituted with the letters directly to their right, wrapping around if necessary (Nagaraj & Kathikeyan, March 21, 2023).

26 If two letters are positioned in the same row, substitute them with the letters that are directly to their right (looping back to the beginning if necessary). If they are in the same column, change them to the letters that are directly beneath (also wrapping 36 around). When the letters are neither in the same row nor column, exchange them with the corresponding rows and columns (Reddy & Malla, April,2017). This cipher was 4 initially introduced in a paper by Wheatstone in 1854, titled “The Application of the Principle of the Symmetrical Cipher System to the Interpretation of Topographical and Geographical Maps,” which was presented to the Royal Society in London. It became widely used during the Boer War and continued to be utilized in both World War I and 4 World War II, primarily by military forces for secure communications. Over time, it was replaced by more advanced encryption techniques as cryptography evolved. (Nagaraj & Kathikeyan, March 21, 2023).

The Playfair Cipher remains a valuable instructional resource that illustrates early 4 encryption techniques and has influenced more intricate polygraphy substitution ciphers such as the Four-Square and Trifid Ciphers. These innovations provided the foundation for contemporary block ciphers like AES and DES, which are essential components of today's encryption standard. (Gupta & Das, January, 2020). Despite its widespread use during World War I, the Playfair cipher is now considered insecure; however, its encryption principles have been carried forward. Researchers are now blending modern mathematical concepts and encryption algorithms with the traditional Playfair cipher to develop new encryption methods. (Wang, 2021).

2.1.3 Pros of Using Playfair Cipher

The Playfair Cipher provides enhanced encryption through the use of Polygram substitution. This method encrypts letter pairs simultaneously, increasing the complexity and making it more challenging for attackers to decipher without the proper key. Its straightforward design, featuring a 5x5 letter matrix, allows for easy comprehension and implementation. The encryption of pairs of letters introduces an additional security measure, increasing its resistance to frequency analysis. (Nagaraj & Kathikeyan, March 21, 2023).

2.1.4 Cons of Using PlayFair Cipher

The Playfair Cipher has various drawbacks. It is limited to encrypting only alphabetic characters, which means it cannot process numbers, symbols, or other non-alphabetic characters, restricting its application. The 5x5 grid results in a small key space, leaving it susceptible to brute-force attacks. Furthermore, the letter frequency patterns present in the ciphertext can be studied, potentially exposing the original message if there is sufficient data. This absence of perfect secrecy and its vulnerability to known plaintext attacks render it less effective for secure communication in contemporary situations. (Sileshi & Dagim, March 18, 2024)

13

3. Development of New Cryptographic Algorithm

To create a new cryptographic algorithm, the Playfair cipher was first used as the foundational technique. The plaintext was encoded using this classical cipher, after which binary operations were applied to convert the encoded text into binary digits. An XOR operation was then performed on these binary values, introducing an additional layer of encryption and enhancing the overall security of the process.

After the fundamentals and historical context of the Playfair cipher were explored, the focus was shifted to binary operations and their integration with the Playfair cipher. Through this combination, a new algorithm—named XOR-Enhanced Playfair Cipher

(XEFR) was developed to provide stronger encryption compared to the traditional approach.

3.1 What is the Binary System?

The binary numeral system, which consists solely of 0 and 1, is foundational to contemporary computing as it corresponds directly to how computers process data through electrical signals that represent "on" and "off" states (Lande, 2014).

Binary is vital in the field of cryptography, as it facilitates secure encoding, encryption, and decryption of information by simplifying complex algorithms into a dependable format. Its application in electronic components, such as transistors, drives all computing technologies. Additionally, binary efficiently represents digital information like text, images, and sound, rendering it essential in both common technology and sophisticated systems such as artificial intelligence. Pioneers in mathematics, including Gottfried Leibniz and George Boole, significantly contributed to the development of binary, which has evolved into the universal language of computing. Its inherent simplicity, reliability, and versatility make it crucial for a range of applications, from secure communications to cutting-edge technological advancements. (Lande, 2014).

3.2 XOR Operation

XOR, short for "Exclusive OR," is a logical function that compares two binary inputs and outputs 1 if the inputs differ or 0 if they are the same. This operation plays a crucial role in computing and cryptography because of its simplicity, efficiency, and unique properties. One key feature of XOR is its reversibility. When applied twice with the same value, it restores the original data, making it especially useful for encryption and decryption tasks. XOR can also flip individual bits, combine data, and detect errors. (Training, 2024).

For example, applying XOR to the binary numbers A=1010A = 1010A=1010 and B=1100B = 1100B=1100 yields 011001100110, with each bit processed independently. Its ability to securely mix and unmix information along with its straightforward design makes XOR indispensable in digital circuits, cryptographic algorithms, and error-checking systems. (Training, 2024).

X

Y

X^Y

0

0

0

0

1

1

1

0

1

1

1

0

Table 6. XOR table

3.3 How did we use binary and XOR Operation with Playfair cipher?

The combination of the Playfair Cipher, binary conversion, and XOR operation creates a multi-layered encryption approach that merges classical and modern cryptographic techniques. The process begins with encoding plaintext using the Playfair Cipher, which manipulates letter pairs based on a 5x5 grid. The resulting ciphertext is then converted into binary format, enabling compatibility with digital systems. XOR operation is applied using a secret binary key, adding a robust layer of security by making the ciphertext dependent on the key. Optionally, the binary output can be converted to hexadecimal for compactness and ease of handling. Decryption reverses these steps, recovering the original plaintext. The Playfair Cipher introduces non-linear complexity, while binary and XOR enhance resistance against brute force and statistical attacks. The addition of hexadecimal conversion makes the ciphertext compact and transmission-friendly.

3.3.1 Encrypting and Decrypting Steps

For Encryption,

Create a Playfair Grid

Use a keyword to construct a 5x5 grid of letters, excluding duplicates and omitting "J." Fill the remaining spaces with other letters in alphabetical order.

Split the Plaintext

19 Divide the plaintext into pairs of letters. If a pair has repeated letters, insert a filler letter (e.g., "X"). Add an extra filler if the length is odd.

Encrypt Using Playfair Rules

Apply the Playfair Cipher rules to encrypt each pair:

Convert to Binary

Letter

ASCII Code

Binary

A

065

01000001

B

066

01000010

C

067

01000011

D

068

01000100

E

069

01000101

F

070

01000110

G

071

01000111

H

072

01001000

I

073

01001001

J

074

01001010

K

075

01001011

L

076

01001100

M

077

01001101

N

078

01001110

O

1 079

01001111

P

080

01010000

Q

081

01010001

R

082

01010010

S

083

01010011

T

084

01010100

U

085

01010101

V

086

01010110

W

087

01010111

X

088

01011000

Y

089

01011001

Z

090

01011010

Translate each letter of the ciphertext into its binary representation of ASCII values from the ASCII value table,

Combine Binary Values

Concatenate the binary values of the ciphertext letters into a single binary string.

Apply XOR Operation

Choose a binary key of the same length as the binary ciphertext.

Perform XOR between the binary ciphertext and the key.

X

Y

X^Y

5 0

0

0

0

1

1

1

0

1

1

1

0

Step 7: Perform Binary Addition to the Key and XOR result

For Decryption Steps,

Step 1: Perform Binary subtraction with the cipher text and the same key

Step 2: Reverse XOR Operation

Use the same binary key to XOR with the binary ciphertext to retrieve the original binary string.

Step 3: Binary to Playfair Cipher Text

Translate the binary values back into their corresponding letters.

Step 4: Playfair Cipher Decryption

Using the same Playfair grid, decrypt each pair by reversing the rules

Step 5: Recover the Plaintext

Step 6: Combine the decrypted letter pairs to reconstruct the original plaintext.

Together, these steps create a secure and versatile encryption mechanism suitable for modern digital systems.

3.3.1.1 Example Use

Keyword: INSTRUMENT

Plaintext: IKSONG

Step 1: Use PlayFair Cipher

Here,

We create a 5x5 matrix grid using the Keyword,

I

N

S

T

R

U

M

E

A

B

C

D

F

G

H

K

L

O

P

Q

V

W

X

Y

Z

Table 7. 5x5 matrix grid for Example

Solution,

PLAINTEXT= IKSONG

Now,

Pairing the plaintext,

IK SO NG

IK= VU (same column which is rule no 3),

SO= XE (same column which is rule no 3),

NG = DT (Rectangle Rule)

Hence, Cipher text = VUXEDT

Step 2: Converting the cipher text into Binary

Here, we use the Binary value of the letters to convert them into Binary,

Letter

ASCII Code

Binary

A

065

01000001

B

066

01000010

C

067

01000011

D

068

01000100

E

069

01000101

F

070

01000110

G

071

01000111

H

072

01001000

I

073

01001001

J

074

01001010

K

075

01001011

L

076

01001100

M

077

01001101

N

078

01001110

O

079

01001111

P

080

01010000

Q

081

01010001

R

082

01010010

S

083

01010011

T

084

01010100

U

085

01010101

V

086

01010110

W

087

01010111

X

088

01011000

Y

089

01011001

Z

090

01011010

Table 8. Binary Representation of Letters

Now,

VUXEDT,

V= 01010110

U=01010101

X=01011000

E=01000101

D=01000100

T=01010100

Encrypted Text =01010110 01010101 01011000 01000101 01000100 01010100

Step 3. Use XOR Operation

Encrypted Text: 01010110 01010101 01011000 01000101 01000100 01010100

Now.

Let's assume the key is given as:

Key:10101100 11001001 10010110 00101011 11110001 01101110

XOR Calculation for each block,

For Block1,

Cipher Text=01010110

Key=10101100

Cipher Text

0

1

0

1

0

1

1

0

Key

1

0

1

0

1

1

0

0

XOR

1

1

1

1

1

0

1

0

Table 9. Block 1 XOR Operation

Result = 11111010

Now,

For Block 2,

Cipher Text = 01010101

Key = 11001001

Cipher Text

0

1

0

1

0

1

0

1

Key

1

1

0

0

1

0

0

1

XOR

1

0

0

1

1

1

0

0

Table 10. Block 2 XOR Operation

Result = 10011100

Again,

For Block 3,

Cipher Text = 01011000

Key = 10010110

Cipher Text

0

1

0

1

1

0

0

0

Key

1

0

0

1

0

1

1

0

XOR

1

1

0

0

1

1

1

0

Table 11. Block 3 XOR Operation

Result = 11001110

Again,

For Block 4,

Cipher text = 01000101

Key = 00101011

Cipher Text

0

1

0

0

0

1

0

1

Key

0

0

1

0

1

0

1

1

XOR

0

1

1

0

1

1

1

0

Table 12. Block 4 XOR Operations

Result = 01101110

Again,

For Block 5,

Cipher Text = 01000100

Key = 11110001

9 Cipher Text

0

1

0

0

0

1

0

0

Key

1

1

1

1

0

0

0

1

XOR

1

0

1

0

0

1

0

1

Table 13. Block 5 XOR Operation

Result = 10100101

Lastly,

For Block 6,

Cipher Text =01010100

Key = 01101110

9

Cipher Text

0

1

0

1

0

1

0

0

Key

0

1

1

0

1

1

1

0

XOR

0

0

1

1

1

0

1

0

Table 14. Block 6 XOR Operation

Result: 00111010

Now,

Combining the XOR Results,

XOR encrypted Result: 11111010 10011100 11001110 01101110 10100101 00111010

Step 4: Performing Binary Addition with Key

Key: 10101100 11001001 10010110 00101011 11110001 01101110

XOR Result: 11111010 10011100 11001110 01101110 10100101 00111010

Now,

After Binary Addition,

Encrypted string:10100110 01100101 01100100 10011001 10010110 10101000

Hence, the cipher text is 10100110 01100101 01100100 10011001 10010110 10101000

3.3.2 Flowcharts

3.3.2.1 Encryption

Decryption

4. Testing

Let’s test our newly developed algorithm with a few test cases to evaluate its performance and accuracy.

4.1 Test Case 1

Encrypting,

Keyword: RETINOL

Plaintext: SERUM

Step 1: Create the Playfair Cipher Grid

We begin by creating the Playfair cipher grid using the keyword RETINOL,

Now,

Creating a 5x5 matrix,

R

E

T

I/J

N

O

L

A

B

C

D

F

G

H

K

M

P

Q

S

U

V

W

X

Y

Z

Table 15. 5x5 Matrix for test case 1

Now,

We Pair the Plaintext SERUM,

Since the plaintext has an odd number of letters, we add an 'X' at the end to make it even we get,

SE RU MX

Now, we Apply the Playfair Cipher rules to Each Pair

SE → PI

RU → NM

MX → QV

Hence, the cipher text is PINMQV.

Step 2: Converting the Ciphertext into Binary (ASCII Values)

Cipher Text = PINMQV

Now,

Using the ASCII table to convert each letter into their Binary Values,

P → 0101000

I → 01001001

N → 01001110

M → 01001101

Q → 01010001

V → 01010110

Hence the Binary String value is 01010000 01001001 01001110 01001101 01010001 01010110.

Step 3: Perform XOR Operation

Encrypted Text: 01010000 01001001 01001110 01001101 01010001 01010110

Now.

Let's assume the key is given as:

Key: 11001001 10010010 10101010 01010101 11110000 01101101

XOR Calculation for each block,

Block

Cipher Text

Key

XOR result

1

01010000

11001001

10011001

2

01001001

10010010

11011011

3

01001110

10101010

11100100

4

01001101

01010101

00011000

5

01010001

11110000

10100001

6

01010110

01101101

00111011

Table 16. XOR result of Encryption Test Case 1

XOR results = 10011001 11011011 11100100 00011000 10100001 00111011

Step 4: Performing Binary Addition with Key

Key:11001001 10010010 10101010 01010101 11110000 01101101

XOR Result:10011001 11011011 11100100 00011000 10100001 00111011

Now,

After Binary Addition,

Encrypted text = 01100010 01101110 10011111 01101110 10000001 01001001

Hence, encrypted text is 01100010 01101110 10011111 01101110 10000001 01001001

DECRYPTION

Now,

Let’s move on to Decrypting,

Cipher text = 01100010 01101110 10011111 01101110 10000001 01001001

Step 1: Binary Subtraction is done to the ciphertext and Key

Cipher text = 01100010 01101110 10011111 01101110 10000001 01001001

Key: 11001001 10010010 10101010 01010101 11110000 01101101

After the Binary Subtraction,

Encrypted text = 10011001 11011100 11110101 00011001 10010001 11011100

Step 2: Reverse XOR Operation

Using the same key we used earlier for encryption,

Key = 11001001 10010010 10101010 01010101 11110000 01101101

Block

Encrypted text

Key

XOR

1

10011001

11001001

01010000

2

11011011

10010010

01001001

3

11100100

10101010

01001110

4

00011000

01010101

01001101

5

10100001

11110000

01010001

6

00111011

01101101

01010110

Table 17. XOR result of Decryption of Test 1

Hence, the reverse XOR result is: 01010000 01001001 01001110 01001101 01010001 01010110

Step 3: Convert Binary to letters

Now, we convert the binary values back to the letters using the ASCII table

Cipher Text = 01010000 01001001 01001110 01001101 01010001 01010110

0101000 → P

01001001 → I

01001110 → N

01001101 → M

01010001 → Q

01010110 → V

So, The Binary Values map to PINMQV

Step 3: Playfair cipher Decryption

Now let’s look at the 5x5 grid Again,

R

E

T

I/J

N

O

L

A

B

C

D

F

G

H

K

M

P

Q

S

U

V

W

X

Y

Z

Table 18. 5x5 grid (Test case 1)

Now,

Decrypting each pair from PINMQV

23 Now, let's decrypt each pair of the ciphertext PI NM QV

1st Pair: PI

P is located at (4th row, 2nd column)

I is located at (1st row, 4th column)

2 - These two letters are not in the same row or column; they form a rectangle on the Playfair grid. So, following the rectangle rule for decryption:

8 P (4th row, 2nd column) swaps with I (1st row, 4th column) by swapping their columns.

P moves to the position (4th row, 4th column) → S

"I" moves to the position (1st row, 2nd column) → E.

Hence, the decrypted pair is SE

2nd Pair: NM

51 N is located at (1st row, 5th column)

M is located at (4th row, 1st column)

2 These two letters are not in the same row or column but they form a rectangle on the Playfair grid.

So, the decrypted pair is RU

3rd Pair: QV

Q is located at (4th row, 3rd column)

2 V is located at (5th row, 1st column)

These two letters are not in the same row or column once again they form a rectangle on the Playfair grid. So, by following the rectangle rule for decryption:

8 Q (4th row, 3rd column) swaps with V (5th row, 1st column) by swapping their columns. Q moves to the position (4th row, 1st column) → M

V moves to the position (5th row, 3rd column) → X

Hence, the decrypted pair is MX

Lastly,

Combining the decrypted pairs we get, SE RU MX.

Now,

Remove the filler letter X which was added during encryption to make the plaintext length even.

After Combining the deciphered text, we get SERUM which was the original Plaintext.

Hence, this test was successful.

4.2 Test Case 2

Encrypting,

Keyword: DREAM

Plaintext: BRIGHT

Step 1: Create the Playfair Cipher Grid

We begin by creating the Playfair cipher grid using the keyword DREAM,

Now,

Creating a 5x5 matrix using the keyword, and eliminating duplicate letters. Then fill the grid with the remaining letters of the alphabet. For this example, we treat I/J as the same letter.

D

R

E

A

M

B

C

F

G

H

I/J

K

L

N

O

P

Q

S

T

U

V

W

X

Y

Z

Table 19. 5x5 matrix for Test case 2

Now,

We Pair the Plaintext BRIGHT,

Split into pairs

BR IG HT

6 Now, we Apply the Playfair Cipher rules to Each Pair

BR → CD

IG → NB

HT → GU

So, the final cipher text is CD NB GU

Step 2: Convert Ciphertext into Binary (ASCII Values)

Cipher Text= CDNBGU

Now, let’s convert the ciphertext into binary using the ASCII values:

C (ASCII 67) → 01000011

D (ASCII 68) → 01000100

N (ASCII 78) → 01001110

B (ASCII 66) → 01000010

G (ASCII 71) → 01000111

U (ASCII 85) → 01010101

Thus, the binary ciphertext is:

01000011 01000100 01001110 01000010 01000111 01010101

Step 3: Perform XOR Operation

Cipher text:01000011 01000100 01001110 01000010 01000111 01010101

Let's assume the key is in binary:

11001011 10101100 01110101 11001010 10101110 01110111

Now, perform the XOR between the ciphertext and the key:

Block

Cipher Text

Key

XOR result

1

01000011

11001011

10001000

2

01000100

10101100

11110000

3

01001110

01110101

00111011

4

01000010

11001010

10001000

5

01000111

10101110

11110001

6

01010101

01110111

00010010

Table 20. XOR operation for encryption for Test Case 2

The XOR result is: 10001000 11101000 00111011 10001000 11101001 00100010

Step:4 Performing Binary Addition with key

Key: 11001011 10101100 01110101 11001010 10101110 01110111

XOR Result: 10001000 11101000 00111011 10001000 11101001 00100010

Now,

After Binary Addition,

Cipher text: 01010011 10010100 10110000 01010010 10010111 10011001

Hence, the cipher text is 01010011 10010100 10110000 01010010 10010111 10011001

DECRYPTION

Step 1: Binary Subtraction is done to the ciphertext and the Key

Here,

Cipher text: 01010011 10010100 10110000 01010010 10010111 10011001

Key: 11001011 10101100 01110101 11001010 10101110 01110111

Now, performing binary subtraction, we get, Decipher text=10001000 11101000 00111011 10001000 11101001 00100010

Step 2: Reverse XOR Operation

Decipher Text: 10001000 11101000 00111011 10001000 11101001 00100010

Using the same key (in binary) as before:

Key:11001011 10101100 01110101 11001010 10101110 01110111

Now perform the XOR again to get the original binary:

Block

Cipher Text

Key

XOR result

1

10001000

11001011

01000011

2

1101000

10101100

01000100

3

00111011

01110101

01001110

4

10001000

11001010

01000010

5

11101001

10101110

01000111

6

00100010

01110111

01010101

Table 21. Reverse XOR operation for Decryption Test case 2

The binary result is = 01000011 01000100 01001110 01000010 01000111 01010101

Step 3: Convert Binary to Letters (ASCII values)

Now, convert the binary values back into letters:

01000011 → C

01000100 → D

01001110 → N

01000010 → B

01000111 → G

01010101 → U

Hence, the result is CDNBGU

Step 4: Playfair cipher Decryption

Now let’s look at the 5x5 grid Again,

D

R

E

A

M

B

C

F

G

H

I/J

K

L

N

O

P

Q

S

T

U

V

W

X

Y

Z

Table 22. 5x5 matrix for Decryption of Test Case 2

Decrypting each pair from CDNBGU

Now, Pairing the Cipher text

CD NB GU

Now,

1st Pair: CD

2 C is located at (1st row, 3rd column).

D is located at (1st row, 4th column).

These letters are in the same row (same row rule). To decrypt, we shift them to the left:

Result: BR

2nd Pair: NB

N is located at (4th row, 2nd column).

8 B is located at (1st row, 2nd column).

These letters are in the same column (same column rule). To decrypt, we shift them up:

Result: IG

3rd Pair: GU

8 G is located at (2nd row, 1st column).

U is located at (4th row, 5th column).

2 These letters are not in the same row or column (they form a rectangle). To decrypt

them, we swap their columns:

Result: HT

Combining the Results

CD → BR

NB → IG

GU → HT

The final decrypted plaintext is BRIGHT which is the original Plaintext. Hence, this test was successful.

4.3 Test Case 3

Keyword: TEDDY

Plaintext: BEAR

Step 1: Create the Playfair Cipher Grid

Using the keyword TEDDY, we create the 5x5 Playfair cipher grid as,

T

E

D

Y

A

B

C

F

G

H

I/J

K

L

M

N

O

P

Q

R

S

U

V

W

X

Z

Table 23. 5x5 matrix for Test 3

Step 2: Pair the Plaintext

Plaintext: BEAR

Split the plaintext into pairs:

BE AR

Step 3: Encrypt Each Pair

BE → CT

AR → YS

Final ciphertext after Playfair encryption AC DS

Step 4: Convert Ciphertext to Binary

Ciphertext: ACDS

Using the ASCII table, convert each letter to binary:

A (ASCII 65) → 01000001

C (ASCII 67) → 01000011

D (ASCII 68) → 01000100

S (ASCII 83) → 01010011

Now,

Step 5: Concatenate the binary values:

01000001 01000011 01000100 01010011

Step 6: Perform XOR Operation

Cipher Text: 01000001 01000011 01000100 01010011

Key = 11001010 10101100 01111011 10010110

Now,

Perform XOR between the ciphertext binary and the key binary:

Block	Cipher Text	Key	XOR Result
1	01000001	11001010	10001011
2	01000011	10101100	11101111
3	01000100	01111011	00111111
4	01010011	10010110	11000101

Table 24. XOR operation for Test Case 3

XOR result: 10001011 11101111 00111111 11000101

Step 4: Performing Binary Addition with Key

Here,

Key:11001010 10101100 01111011 10010110

XOR Result:10001011 11101111 00111111 11000101

Now,

After Binary Addition,

We get,

Cipher text: 01010101 10011011 10111010 01011011

Hence, the ciphertext is 01010101 10011011 10111010 01011011

Decryption

Step 1: Binary Subtraction is done to the ciphertext and the Key

Here,

Cipher text: 01010101 10011011 10111010 01011011

Key: 11001010 10101100 01111011 10010110

We get,

Decipher text: 10001011 11101111 00111111 11000101

Now, to further decipher

Step 2: Reverse XOR Operation

Using the same key from the encryption,

Key = 11001010 10101100 01111011 10010110

Block

Cipher Text

Key

XOR result

1

10001011

11001010

01000001

2

11101111

10101100

01000011

3

00111111

01111011

01000100

4

11000101

10010110

01010011

Table 25. Reverse XOR for test case 3

Reverse XOR result = 01000001 01000011 01000100 01010011

Step 3: Convert Binary Back to Letters

Reverse XOR result = 01000001 01000011 01000100 01010011

Here,

Convert binary values back to ASCII letters:

01000001 → A

01000011 → C

01000100 → D

01010011 → S

Reconstructed ciphertext after XOR decryption: ACDS

Step 4: Decrypt Playfair Cipher

Here,

Let’s look into the grid once again,

T

E

D

7 Y

A

B

C

F

G

H

I/J

K

L

M

N

O

P

Q

R

S

U

V

W

X

Z

Table 26. 5x5 grid (Test Case 3)

Here,

Reconstructed ciphertext after XOR decryption = ACDS

Now,

Pairing up the cipher text we get,

1st Pair: AC

A is at (1st row, 1st column).

C is at (2nd row, 2nd column).

Decrypted pair = BE

2nd Pair: DS

D is at (4th row, 5th column).

S is at (1st row, 4th column).

Decrypted pair = AR

Lastly,

Pairing up the Decrypted text we get,

AC=BE and DS=AR

Therefore, the Final deciphered text is BEAR which matches the original plaintext.

Hence, this test was successful.

4.4 Test Case 4

Keyword = BLUE

Plaintext = SWEATER

Now,

Step 1: Create the Playfair Cipher Grid from the Keyword

Here, we use the keyword BLUE to fill in the Playfair cipher grid. First, we write down the letters of the keyword without repeating any letters. Then, we fill in the remaining letters of the alphabet, omitting J (since I/J are typically combined in the Playfair cipher grid).

B

L

U

E

A

C

D

F

G

H

I/J

K

M

N

O

P

Q

R

S

T

V

W

X

Y

Z

Table 27. 5x5 matrix for Test Case 4

Step 2: Prepare the Plaintext

Now, we prepare the plaintext SWEATER for encryption. Since Playfair cipher works in pairs of letters, we split SWEATER into pairs. If there’s an odd number of letters, we add a filler letter (usually X). So, we split SWEATER as follows:

SW and EA and TE and RX

Step 3: Encrypt Each Pair Using the Grid

SW → QY

EA → AB

TE → SA

RX → XU

Ciphered text = QY AB SA XU

Step 4: Convert Ciphertext to Binary

Ciphertext: QYABSAXU

To do this, we first need to assign each letter in the ciphertext a corresponding binary code. One common method is to use the ASCII binary representation for each letter.

Step 5: Convert each letter to its ASCII value:

Q: ASCII value = 81

Y: ASCII value = 89

A: ASCII value = 65

B: ASCII value = 66

S: ASCII value = 83

A: ASCII value = 65

X: ASCII value = 88

U: ASCII value = 85

Step 2: Convert each ASCII value to binary:

Q (ASCII 81) → 01010001

Y (ASCII 89) → 01011001

A (ASCII 65) → 01000001

56 B (ASCII 66) → 01000010

S (ASCII 83) → 01010011

A (ASCII 65) → 01000001

X (ASCII 88) → 01011000

U (ASCII 85) → 01010101

Step 5: Combine the Binary Strings

Now, let’s concatenate the binary values for each letter in the ciphertext "NBEORY":

Ciphertext "QYABSAXU" in binary is: 01010001 01011001 01000001 01000010 01010011 01000001 01011000 01010101

Step: 5 Use XOR Operations.

Here,

Reconstructed Cipher Text = 01010001 01011001 01000001 01000010 01010011 01000001 01011000 01010101

Now,

Key = 11001010 10101100 01111011 10010110 11001100 10111001 10010111 11101010

54 Now, we will perform the XOR operation between each block of the ciphertext and the

corresponding block of the key.

Block

Cipher Text

Key

XOR Result

1

01010001

11001010

10011011

2

01011001

10101100

11100101

3

01000001

01111011

00111010

4

01000010

10010110

11010100

5

01010011

11001100

10011111

6

01000001

10111001

11111000

7

01011000

10010111

11001111

8

01010101

11101010

10111111

Table 28. XOR Operation for Test Case 4

XOR Result = 10011011 11100101 00111010 11010100 10011111 11111000 11001111 10111111

Step 4: Performing Binary Addition with Key

Key:11001010 10101100 01111011 10010110 11001100 10111001 10010111 11101010

XOR result:10011011 11100101 00111010 11010100 10011111 11111000 11001111 10111111

Here,

Cipher text:01100101 10010001 10110101 01101010 01101011 10110001 01100110 10101001

Hence, the ciphertext is 01100101 10010001 10110101 01101010 01101011 10110001 01100110 10101001

Decryption

Step 1: Binary Subtraction is done to the ciphertext and Key

Here,

Cipher text: 01100101 10010001 10110101 01101010 01101011 10110001 01100110 10101001

Key(using the same key used in encryption): 11001010 10101100 01111011 10010110 11001100 10111001 10010111 11101010

We get,

Deciphered text: 10011011 11100101 00111010 11010100 10011111 11111000 11001111 10111111

Step 2: Reverse XOR

Now, for decryption, we reverse the XOR operation using the same key. Let's go through the XOR decryption process with the same key.

Reverse XOR operation:

Block

Cipher Text

Key

XOR

1

10011011

11001010

01010001

2

11100101

10101100

01011001

3

00111010

01111011

01000001

4

11010100

10010110

01000010

5

10011111

11001100

01010011

6

11111000

10111001

01000001

7	11001111	10010111	01011000
8	10111111	11101010	01010101

Table 29. Reverse XOR for Test Case 4

Reverse XOR result (binary): 01010001 01011001 01000001 01000010 01010011 01000001 01011000 01010101

Step 2: Convert the binary back to ASCII letters:

01010001 01011001 01000001 01000010 01010011 01000001 01011000 01010101

Now,

01010001 → Q

01011001 → Y

01000001 → A

01000010 → B

01010011 → S

01000001 → A

01011000 → X

01010101 → U

So, we get the reconstructed ciphertext: QYAB SA XU

Step 7: Decrypt Playfair Cipher

Finally, we decrypt the ciphertext using the Playfair cipher rules

QY → SW

AB → EA

SA → TE

XU → RX

Final Decrypted Plaintext: SW EA TE RX

So, the final decrypted plaintext is SWEATER, which matches the original plaintext.

Hence, it was successful.

4.5 Test Case 5

Let’s go through the Playfair cipher encryption and XOR operation for the keyword CERAMICS and the plaintext BOWL.

Given,

Keyword: CERAMICS

Plaintext: BOWL

Step 1: Create the Playfair Cipher Grid

We create the 5x5 grid using the keyword "CERAMICS" and fill in the remaining alphabet letters, excluding J (since I/J are combined in the Playfair cipher grid).

Keyword: CERAMICS

Write the letters of the keyword without repetition: C E R A M I/J S

C

E

R

A

M

I/J

S

B

D

F

G

H

K

L

N

O

P

Q

T

U

V

W

X

Y

Z

Table 30. 5x5 grid for Test Case 5

Step 2: Prepare the Plaintext

We prepare the plaintext BOWL. We split the plaintext into pairs since Playfair cipher works in pairs of letters. We add a filler letter (usually "X") if there's an odd number of letters.

Plaintext: BOWL

Pairs: BO, WL

Step 3: Encrypt Each Pair Using the Grid

BO → IQ

WL → YH

The ciphertext after Playfair encryption is: IQ YH

Step 4: Convert Ciphertext to Binary

Now, let's convert the ciphertext IQYH to binary using ASCII values.

6 Convert each letter to its ASCII value:

I: ASCII value = 73

Q: ASCII value = 81

Y: ASCII value = 89

H: ASCII value = 72

Convert ASCII values to binary:

I (ASCII 73) → 01001001

Q (ASCII 81) → 01010001

Y (ASCII 89) → 01011001

H (ASCII 72) → 01001000

Concatenate the binary strings:

The binary representation of IQYH is = 01001001 01010001 01011001 01001000,

Step 5. Perform XOR operation between ciphertext and key:

Cipher Text: 01001001 01010001 01011001 01001000

Key: 01000011 01000101 01010010 01000001

Now,

Block

Cipher Text

		Key	XOR
1	01001001	01000011	00001010
2	01010001	01000101	00010100
3	01011001	01010010	00001011
4	01001000	01000001	00001001

Table 31. XOR Operation for Test 5

XOR Result: 00001010 00010100 00001011 00001001

Step 4: Performing Binary Addition with Key

Here,

Key: 01000011 01000101 01010010 01000001

XOR result: 00001010 00010100 00001011 00001001

We get,

Cipher text: 01001101 01011001 01011101 01001010

Hence, the ciphertext is 01001101 01011001 01011101 01001010

Decryption

Step 1: Binary Subtraction is done to the ciphertext and Key

Here,

Cipher text: 01001101 01011001 01011101 01001010

Key: 01000011 01000101 01010010 01000001

We get,

Decipher text: 00001010 00010100 00001011 00001001

Now,

Further deciphering,

Step 2: Decryption (Reverse XOR)

For decryption, we reverse the XOR operation using the same key.

XOR Result: 00001010 00010100 00001011 00001001

Key (same as encryption): 01000011 01000101 01010010 01000001

Reverse XOR operation:

Block

XOR Result

Key

Reverse XOR Result

1

00001010

01000011

01001001

2

00010100

01000101

01010001

3

00001011

01010010

01011001

4

00001001

01000001

01001000

Table 32. Reverse XOR for Test Case 5

Reverse XOR Result: 01001001 01010001 01011001 01001000

Convert the binary back to ASCII:

01001001 → I

01010001 → Q

01011001 → Y

01001000 → H

So, the reconstructed ciphertext is: IQYH

Step 7: Decrypt Playfair Cipher

Finally, we decrypt the ciphertext IQYH using the Playfair cipher rules.

Pair the ciphertext:

IQ → BO

YH → WL

Final Decrypted Plaintext:

The final decrypted plaintext is BOWL, which matches the original plaintext. Hence, the test is successful.

5. Analysis

5.1 Introduction

The XOR-Enhanced Playfair Cipher Algorithm is a cryptographic technique combining the Playfair cipher with XOR-based binary transformations. It enhances traditional substitution ciphers by adding complexity through binary operations. While the approach is innovative, a critical evaluation reveals its strengths, weaknesses, and potential applications.

5.2 Strengths of the XOR-Enhanced Playfair Cipher Algorithm

Dual Layer Security:

Combining the Playfair cipher and XOR operations makes it harder to decrypt without access to both keys.

Ease of Implementation:

The algorithm uses simple and well-known methods, making it easy to program in different languages.

Flexibility:

Customizable keys for both Playfair and XOR layers allow adaptation to specific security needs.

Efficiency:

XOR operations are computationally light, making the algorithm suitable for low-resource environments.

Frequency Analysis Resistance:

The XOR transformation disrupts frequency patterns, reducing susceptibility to basic cryptanalysis.

5.3 Weaknesses of the XOR-Enhanced Playfair Cipher Algorithm

If the XOR key is exposed or poorly managed, the encryption can be reversed easily.

The Playfair cipher and XOR operations alone are not robust against modern cryptanalytic techniques.

Managing and securely sharing two keys (Playfair and XOR) increases logistical challenges.

The block-based nature of the Playfair cipher may hinder its application to large datasets or streaming data.

The added complexity may not provide proportional security benefits compared to modern encryption standards like AES.

5.4 Comparative Analysis

Against Classical Ciphers

The XOR-Enhanced Playfair Cipher builds on classical ciphers by introducing binary operations, offering enhanced complexity and resistance to basic cryptanalytic techniques like frequency analysis. However, it still shares inherent limitations of classical ciphers, such as reliance on shared key secrecy, making it less suitable for scenarios requiring high levels of security.

Against Modern Cryptographic Standards

Compared to modern algorithms like AES and RSA, the XOR-Enhanced Playfair Cipher lacks the robust mathematical foundations and proven security guarantees these standards provide. It does not meet the rigorous cryptographic requirements, such as resistance to advanced attacks (e.g., differential and linear cryptanalysis) or scalability for key management in large systems. While innovative, it is better suited for educational purposes or low-security applications rather than as a replacement for modern encryption standards.

5.5 Potential Use Cases

Ideal for educational purposes, demonstrating the integration of classical and modern cryptographic techniques clearly and practically.

Suitable for low-risk applications, such as securing local files or transmitting non-sensitive information.

Well-suited for resource-constrained environments, like IoT devices, where simplicity and low computational demands are prioritized over advanced security measures.

Hence, The XOR-Enhanced Playfair Cipher Algorithm is an innovative fusion of classical cryptographic methods and binary techniques, offering a simple yet effective framework for certain applications. Its flexibility makes it valuable for educational purposes, illustrating the integration of traditional and modern encryption concepts. While it lacks the robustness required for high-security environments, it is well-suited for niche applications, such as protecting non-sensitive data or securing low-risk communications. With further refinements and rigorous testing, it has the potential to become a practical tool in resource-constrained scenarios, providing a balance between efficiency and basic security needs.

6. Conclusion

This research took a deep dive into the evolution of cryptography, with a focus on enhancing the classic Playfair Cipher using cutting-edge techniques like binary systems and XOR operations. We've seen first-hand how the core principles of cryptography confidentiality, integrity, and availability are the backbone of securing sensitive information in today's fast-moving digital world. By revisiting the Playfair Cipher, we not only explored its historical importance but also its practical strengths and limitations in modern applications.

We introduced a powerful new version of the Playfair Cipher, boosted by XOR logic, proving that merging traditional cryptographic methods with modern innovations can create a much stronger security system. The addition of binary operations and XOR made the cipher far more robust, addressing the vulnerabilities that have held it back for decades. After rigorous testing, we confirmed that this enhanced algorithm is highly effective for encrypting and decrypting data. It is efficient, adaptable, and reliable in real-world scenarios, especially in resource-limited environments.

What this research truly highlights is the need for constant innovation in cryptography to outpace the ever-growing threats in the cybersecurity landscape. The new algorithm isn’t just an academic exercise but it opens up the possibility of lightweight, secure encryption systems that can be deployed in the real world on mobile devices, in IoT applications, or in environments where computational power is a luxury. More than that, it serves as a call to action: encryption systems need to evolve continuously to stay ahead of emerging cyber threats and ensure long-term security.

In the end, this work offers a renewed appreciation for how cryptography has evolved from its origins in classical methods to the sophisticated algorithms we rely on today. It also provides a roadmap for where future advancements might take us, offering powerful insights for the next wave of research and practical applications in the world of cybersecurity.