LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5004NI Security in Computing

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2023 -24 Autumn

Student Name: Angana Bhattarai

London Met ID: 22067110

College ID: NP01NT4A220074

Assignment Due Date: January 15, 2024

Assignment Submission Date: January 14, 2024

Word Count (Where Required): 7124

I confirm that I understand my coursework needs to be submitted online via My Second Teacher under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

The aim of this report is to provide consumers a thorough grasp of cryptography and how it protects data and communication in the context of information technology. It discusses the many forms of encryption and the CIA triad, including concept, both symmetric and asymmetric. There is additional discussion on the application and constraints of a traditional cypher and the function of logic gates in modifying the chosen cypher. In addition to testing and evaluating the updated cipher, the significance of security in contemporary society and the range of countermeasures to cyber threats are underlined. The study also finds possible applications for the recently developed encryption algorithm and conducts a comprehensive analysis of its advantages and disadvantages. The algorithm is compared to other encryption techniques and its flaws and potential enhancements are discussed. There are considerations made for implementing the algorithm in real-world scenarios.

Table of Contents

Table of Figures

1. Introduction

In computing, security is the safeguarding of data and systems against unapproved access, use, disclosure, disruption, alteration, or destruction. It is the process of making sure that data and systems are available, secure, and intact. The three components of the CIA triad information security framework are availability, integrity, and secrecy. Using algorithms, protocols, and cryptographic keys to secure data and communication is known as cryptography.

In the modern world, cryptography is crucial. The necessity to safeguard sensitive data against attacks and unauthorized access has grown due to the increased reliance on electronic communication and the storage of such data on computers and other digital devices. A wide range of applications, including online banking, e-commerce, and the transfer of sensitive data, depend on the confidentiality, integrity, and availability of data, all of which are made possible by cryptography. It is also essential for defending against cyberattacks on vital infrastructure and safeguarding national security. Cryptography is therefore a vital tool for preserving people's and organizations' security and privacy in the current environment.

1.1. Aims and Objectives

The aims and objectives of this report are listed below:

- To understand the function of the CIA triad in information security
- Understanding security in relation to cryptography and information technology.
- To comprehend the nature of cryptography and how data and communication are secured using it.
- To become aware about the many forms of encryption, such as symmetric and asymmetric encryption.
- To understand about the logic gates.
- To learn about the uses and limitations of a traditional cipher.

- To learn about the role of logic gates and how they may be used to modify the selected cipher.
- To test the modified cipher.
- To evaluate the modified cipher.
- To recognize the significance of security in present-day society and the scope of countermeasures available for cyber-attacks.

2. Background

2.1. What is security?

Security in Information Technology is the safeguarding of data and systems against unwanted access, use, disclosure, disruption, alteration, or destruction. It is the process of making sure that data and systems are available, secure, and intact. Security in the context of cryptography refers to the precautions used to shield data from assaults or unauthorized access. It is the procedure that makes sure the data is safe from any efforts to compromise it and that only authorized parties may access or alter it. A variety of cryptographic approaches, including authentication and encryption, can be used to accomplish this. These techniques work to make it more difficult for an attacker to access or alter the data. In the current digital era, where sensitive data is increasingly being sent and kept electronically, information security is extremely crucial.

2.2. What is C.I.A triad?

Confidentiality, integrity and availability. These are the three components of the CIA triad, an information security model designed to protect sensitive information from data breaches. ‘Confidentiality’ is the first element of the CIA triad, which means keeping sensitive information private and secure. The second element of the CIA triad is ‘Integrity’. This refers to the completeness and accuracy of data, as well as the organisation’s ability to protect it from corruption. The third element of the CIA triad is ‘Availability’. This refers to an organisation’s ability to access information when needed. (Irwin, 2023) Photo Source: (IBM, 2023)

Figure 1: The C.I.A triad

2.3. Why is C.I.A triad important?

The CIA triad constitutes the core basis for the development of security systems and policies for institutions. As such, the CIA triad plays a critical part in maintaining your data safe and protected against growing cyber threats. When a security incident, such as information swiping or a security breach, occurs, it is deemed that an organization has been unsuccessful in properly enforcing one or more of these regulations. The CIA triad is crucial to information security since it enriches security posture, enables organizations to stay obedient to complex regulations, and guarantees business continuity. (Prakash, 2023)

3. Cryptography

3.1. What is cryptography?

Cryptography is the art of keeping information secure by transforming it into form that unintended recipients cannot understand. In cryptography, an original human readable message, referred to as plaintext, is changed by means of an algorithm, or series of mathematical operations, into something that to an uninformed observer would look like gibberish; this gibberish is called cipher-text. Cryptographic systems require some method for the intended recipient to be able to make use of the encrypted message usually, though not always, by transforming the cipher-text back into plaintext. (Frulinger, 2022)
In addition to encryption and decryption, cryptography is used for digital signature, a method that enables the sender of a communication to demonstrate that the message was sent by them and wasn't modified in transit. The use of digital signature guarantees the integrity and authenticity of transactions and electronic messages. In the digital age, cryptography is essential for protecting sensitive data and ensuring communication security.

3.2. Key terminologies of Cryptography

Encryption: The process of encoding data or a message such that only the owner of the decryption key may read or access it is known as encryption.

Decryption: Decryption is the process of implementing the right decryption key to decode data or messages that have been encrypted.

Cryptographic Key: To encrypt and decode data or messages, a long string of random numbers and letters is called a cryptographic key. Asymmetric keys, which utilize two keys a (public key and a private key) for encryption and decryption, and symmetric keys, (which use one key for both) are two examples of the various kinds of cryptographic keys.

Cipher: An method or mathematical function used to encode and decode data or communications is called a cipher. There are various kinds of ciphers, such as public-key ciphers that employ two keys (a public key and a private key) for encryption and decryption, and symmetric-key ciphers that employ a single key for both operations.

Hash Function: A hash function is a mathematical operation that takes an input, such a message, and outputs a unique, fixed-size value for it, which is referred to as a hash or hash value. Because any modifications made to the input will result in a different hash value, hash functions are used to ensure the integrity of electronic messages and transactions.

Digital Signature: Digital signature is an electronic signature that is used to verify the sender's identity and make sure the message hasn't been modified while in transit. Cryptographic methods are used by digital signatures to accomplish these objectives.

ASCII table: The ASCII (American Standard Code for Information Interchange) table is a standardized set of characters consisting of 128 characters (7-bit code). These consist of control characters without an equivalent printable representation in addition to the 95 printable characters. Because ASCII is compatible with the majority of contemporary computer systems and consists of simple characters that can be easily conveyed via a number of communication routes, it is widely used for data transmission and storage. Other character encodings like Latin-1 (ISO 8859-1) and Unicode are based on ASCII. The characters are mapped to their equivalent decimal, octal, and hexadecimal values using the ASCII table (ASCII, 2024).

Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char
0	00	000	0000000	NUL (null character)	32	20	040	0100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	01	001	0000001	SOH (start of header)	33	21	041	0100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	02	002	0000010	STX (start of text)	34	22	042	0100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	03	003	0000011	ETX (end of text)	35	23	043	0100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	04	004	0000100	EOT (end of transmission)	36	24	044	0100100	$	68	44	104	1000100	D	100	64	144	1100100	d
5	05	005	0000101	ENQ (enquiry)	37	25	045	0100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	06	006	0000110	ACK (acknowledge)	38	26	046	0100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	07	007	0000111	BEL (bell (ring))	39	27	047	0100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	08	010	0001000	BS (backspace)	40	28	050	0101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	09	011	0001001	HT (horizontal tab)	41	29	051	0101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	012	0001010	LF (line feed)	42	2A	052	0101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	013	0001011	VT (vertical tab)	43	2B	053	0101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	014	0001100	FF (form feed)	44	2C	054	0101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	015	0001101	CR (carriage return)	45	2D	055	0101101	-	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	016	0001110	SO (shift out)	46	2E	056	0101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	017	0001111	SI (shift in)	47	2F	057	0101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	020	0010000	DLE (data link escape)	48	30	060	0110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	021	0010001	DC1 (device control 1)	49	31	061	0110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	022	0010010	DC2 (device control 2)	50	32	062	0110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	023	0010011	DC3 (device control 3)	51	33	063	0110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	024	0010100	DC4 (device control 4)	52	34	064	0110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	025	0010101	NAK (negative acknowledge)	53	35	065	0110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	026	0010110	SYN (synchronize)	54	36	066	0110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	027	0010111	ETB (end transmission block)	55	37	067	0110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	030	0011000	CAN (cancel)	56	38	070	0111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	031	0011001	EM (end of medium)	57	39	071	0111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	032	0011010	SUB (substitute)	58	3A	072	0111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	033	0011011	ESC (escape)	59	3B	073	0111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	034	0011100	FS (file separator)	60	3C	074	0111100	<	92	5C	134	1011100	\	124	7C	174	1111100	|
29	1D	035	0011101	GS (group separator)	61	3D	075	0111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	036	0011110	RS (record separator)	62	3E	076	0111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	037	0011111	US (unit separator)	63	3F	077	0111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL

Figure 2: ASCII table

Source: (Table, 2024)

3.3. History of Cryptography

Figure 3: Hieroglyph − The Oldest Cryptographic Technique

Cryptography has a long and fascinating history, dating back to ancient civilizations. Some of the earliest known examples of cryptography include the use of simple ciphers, such as the Caesar cipher, which was used by the Roman emperor Julius Caesar to encode his military messages.

In the Middle Ages, the use of ciphers and codes became more widespread, and various methods were developed to encode and decode messages, such as the Vigenère cipher and the Enigma machine, which was used by the Germans during World War II to encode their military communications.

During World War II, the Allies were able to break the Enigma code, which was a major factor in their victory. This led to the development of more advanced cryptographic techniques and the creation of the first electronic computers, which were used to break codes and perform other cryptographic tasks.

In the digital age, the development of computers and the internet has led to the development of more advanced cryptographic algorithms and protocols, such as the RSA algorithm and the AES (Advanced Encryption Standard) algorithm. These algorithms are widely used to secure communication and protect sensitive information in the digital world.

Overall, cryptography's history is intriguing, and it is still developing as new technology and security risks are discovered (Sidhpurwala, 2023). Photo Source: (Tutorialspoint, 2024)

3.4. What are the types of encryption?

Depending on the kind of key used, encryption techniques can be broadly divided into two groups: symmetric encryption and asymmetric encryption.

Figure 4: Types of encryption

Source: (Thakkar, 2020)

Symmetric encryption:

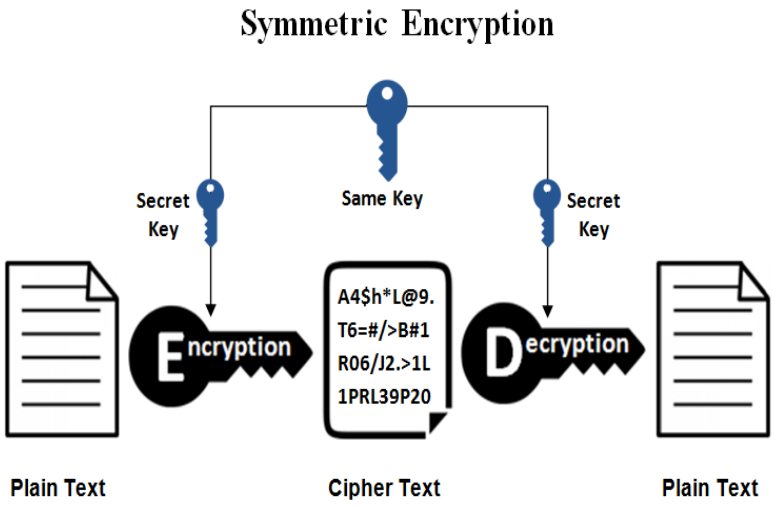

Figure 5: Symmetric encryption

Source: (SSL, 2023)

Symmetric encryption, also dubbed single key encryption, is the type of encryption where a single key can be used to encrypt and decrypt information. In this form of encryption, the receiver uses an agreed shared secret key to decrypt the encrypted data. Symmetric encryption is the oldest form of encryption and is still relevant in organizations that value the speed of information transmission over security authentication (Okeke, 2022). Data encryption known as symmetric encryption encrypts and decrypts data using a single, secret cryptographic key. Plaintext (pre-encryption or post-decryption state) and cipher-text (post-encryption or pre-decryption state) can both be encrypted and decrypted with this key. One of the most used encryption methods, symmetric encryption is used to protect sensitive data in the banking, healthcare, aerospace, and defense sectors. Data is encrypted and decrypted using either a stream cipher or a block cipher. A block cipher uses a predefined key length, such as 128, 192, or 256 bits, to convert entire blocks of plaintext into cipher-text, whereas a stream cipher turns plaintext one byte at a time. Both the sender and recipient must have the secret key in order to encrypt and decrypt the data being transferred.

Asymmetric encryption:

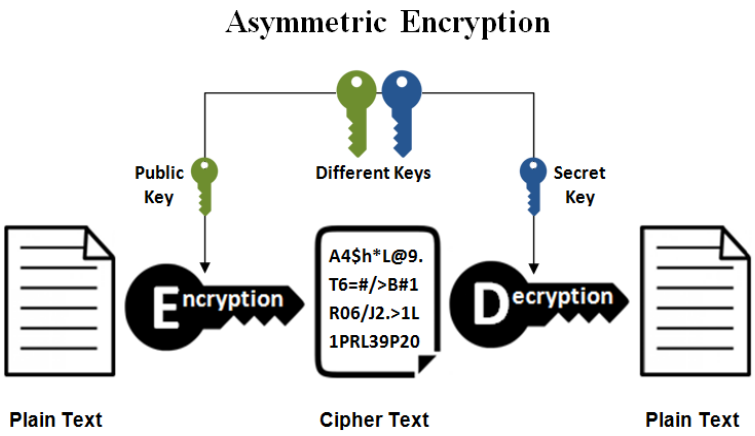

Figure 6: Asymmetric encryption

Source: (SSL, 2023)

Asymmetric encryption, on the other hand, encrypts and decrypts data using a pair of cryptographic keys: a public key and a private key. Data is encrypted using the public key and decrypted with the private key. Asymmetric or public key cryptography is the form of encryption that involves using public and private keys for encryption and decryption. With asymmetric encryption, there is no worry about what a malicious individual can do to your encrypted data as long as you have the secret key for the decryption. Some popular asymmetric key encryption includes DSA, RSA, PKCS and EIGamal (Okeke, 2022). Asymmetric encryption is a fundamental part of several cybersecurity protocols, including PGP (Pretty Good Privacy) and SSL/TLS (Secure Sockets Layer/Transport Layer Security), and is frequently used for secure communication over the internet. It is regarded to be more secure than symmetric encryption since it does not need the exchange of a secret key. Alternatively, the data can be encrypted by the sender using the recipient's public key, then decrypted by the recipient using their private key.

The key differences between symmetric and asymmetric encryption are speed and security preferences. Generally speaking, symmetric encryption is faster and simpler but is often viewed as less secure than asymmetric encryption. But as we've discussed, encryption really boils down to two things: key size and the security of the media storing encryption keys. (Danial, 2021)

4. Background of the selected cryptographic algorithm

4.1. Introduction to ROT-13 cipher

ROT13 is one of the oldest encryption techniques, dating back to ancient Rome, where it was used for military communications. The method was later revived in the 20th century and is now commonly used in computer applications and online communication platforms. ROT13, also known as Caesar Cipher, is a substitution cipher that replaces each letter of the alphabet with the letter 13 positions ahead of it (Nagaraj, 2023).

To encode a message using the ROT13 cipher, you can simply shift each letter in the message by 13 places. For example, the message "Hello, world!" becomes "Uryyb, jbeyq!" when encoded using the ROT13 cipher. To decode a message that has been encoded using the ROT13 cipher, you can simply shift each letter in the message by 13 places in the opposite direction. For example, the message "Uryyb, jbeyq!" becomes "Hello, world!" when decoded using the ROT13 cipher.

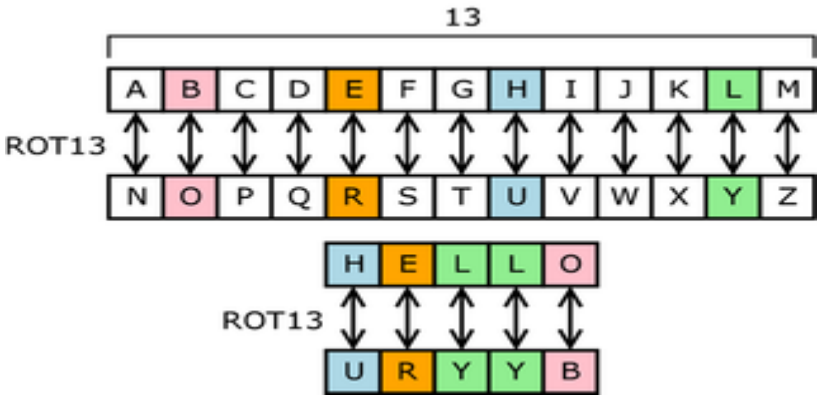

Figure 7: ROT-13 Cipher

Source: (Academic, 2024)

Despite the fact that ROT13 has a fair number of benefits and drawbacks, its simplicity of use and ability to be strengthened against potential weaknesses make it a strong cipher that the intended user can readily interpret. For these reasons, it was chosen as the base for this report.

4.2. Example of encryption and decryption in ROT-13 cipher

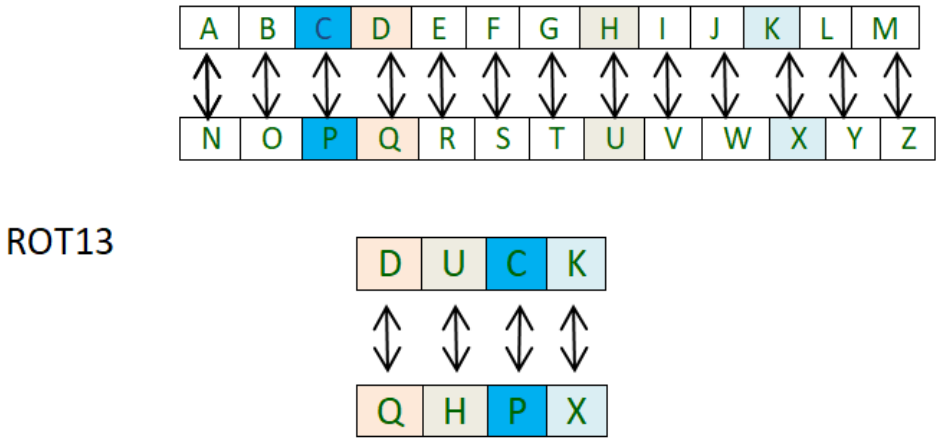

Figure 8: Encryption & Decryption of ROT-13

Source: (GeeksForGeeks, 2024)

Plain text: "duck"

To encrypt the message using the ROT13 cipher, we shift each letter in the message by 13 counts.

Encrypted text: "qhpx"

To decrypt the message, we shift each letter in the encrypted message by 13 places in the opposite direction.

Decrypted text: "duck"

4.3. Pros and Cons of ROT-13

The pros of ROT-13 are:

- Given that the encryption and decryption processes are identical; it is simply decrypted.
- It can be applied as an easy approach to protect text and reduce its readability without really trying to protect the data.
- Using it is quite simple because all you have to do is swap out each letter for a different one.

The cons of ROT-13 are:

- Because the encryption is easily cracked, it offers very little security.
- When transmitting sensitive data or in other situations where serious safety is necessary, it is not recommended.
- Since it's a commonly used cipher, it's not a smart idea to conceal information from others who could be aware of it.

Despite the fact that ROT13 has a fair number of benefits and drawbacks, its simplicity of use and ability to be strengthened against potential weaknesses make it a strong cipher that the intended user can freely interpret. For these reasons, it was chosen as the foundation for this report.

5. Development

5.1. Logic Gates

Computers use logic gates to transform the 1s and 0s from input wires. A logic gate accepts inputs and then outputs a result based on their state (Academy, 2024). Logic gates are electronic devices that perform operations on one or more inputs and produce an output based on a set of rules. There are several types of logic gates, including AND, OR, NOT, NAND, NOR, XOR and XNOR. These gates are the basic building blocks of digital circuits and can be used to perform a wide range of functions.

AND gates have one output and two or more inputs. Only when every input is "true" (high or 1) is the output considered "true." OR gates have one output and two or more inputs. If any of the inputs are "true," then the output is "true". The input and output of a NOT gate are one. In other words, if the input is "true," the output is "false," and vice versa. The output is the opposite of the input. The AND, OR, and NOT gates are modified into NAND, NOR, XOR and XNOR gates, which use combinations of these gates to accomplish various tasks.

For this modification, XNOR gate is being used. The XNOR gate is the complement of the XOR gate. It is a hybrid gate. Simply, it is the combination of the XOR gate and NOT gate. The output level of the XNOR gate is high only when both of its inputs are the same, either 0 or 1. The symbol of the XNOR gate is the same as XOR, only complement sign is added. Sometimes, the XNOR gate is also called the Equivalence gate (Javatpoint, 2024)**.**

The XNOR operation takes two input values, which can be either 0 (false) or 1 (true), and compares them. The result of the XNOR operation is true if both inputs are the same, and false if the inputs are different. In other words, the XNOR outputs a true value when both inputs match (both true or both false) and outputs a false value when the inputs do not match (one is true and the other is false) (Deep, 2024).

The XNOR logical operation is a fundamental concept in the field of digital logic and Boolean algebra. Its ability to compare inputs and determine equality makes it an indispensable tool in the design and analysis of digital systems. Understanding XNOR and its applications is crucial for anyone involved in the development of digital electronics, computer algorithms, and various applications that rely on logical operations (Deep, 2024).

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Figure 9: XNOR logic gate

Source: (Dahl, 2022)

5.2. Modifications of the cipher

5.2.1 Methodology Implied

Choose a random word that will function as your key for both encryption and decryption. Any word, such as "gold," "rose," "simple," and so on, may be the key. Shift each character 13 counts right to each letter to address a new value from the plain-text. Next, compare the word's letter count with the key as indicated below. To apply XNOR on the word and the key, the word lengths must match precisely. In order to address this, if the key contains more characters than the word, we remove the excess characters by removing characters until the initial positions of the keys include no more

characters. As an alternative, add the missing characters as "x" after the last character until there are the same amount of characters if there are less characters in the key than in the word. There is no need for adjustment if the word's letter count matches the letter count of the key. Then, with the equal length of plain-text and key, convert the value of each character into binary form using ASCII table. That value converted into characters in the encrypted text.

Again, with the binary value of key and encrypted text, we apply XNOR gate to obtain a new binary value. We then convert the new binary value obtained from the encrypted text and key using ASCII table. Then we shift the value 13 counts each again to get the decrypted text.

5.2.2. Logical representation of the modification

Given,

Plain-text (P) = “xxx”

Key (K) = “xxx”

If:

Number of characters in Key = Number of characteristics in Plain-text

Key = Plain-text

Else:

Number of characteristics in Key > Number of characteristics in Plain-text

Eradicate the extra characters from the first position of Key.

(Number of characters in Key – Number of characteristics in Plain-text)

For instance,

Plain-text (P) = cake

Key (K) = cup

Then,

Key = Remove (P- K) characters from first position

Remove 1 letter from last position

= cake

Else if:

Number of characters in Key < Number of characters in Plain-text

Implement the necessary characters to equal the key length

(Number of characters in Plain-text – Number of characters in Key)

(Add the character 'x' to fill up the length and use the binary value of character 'x' itself)

For instance,

Plain-text = cake

Key = cup

Then,

Key = Add (P – K) characters 'x' to last position of Key

Add 1 letter 'x' to last position of key

= cupx

In flowchart,

Let,

Length of Plain-text be PL

Length of Key be KL

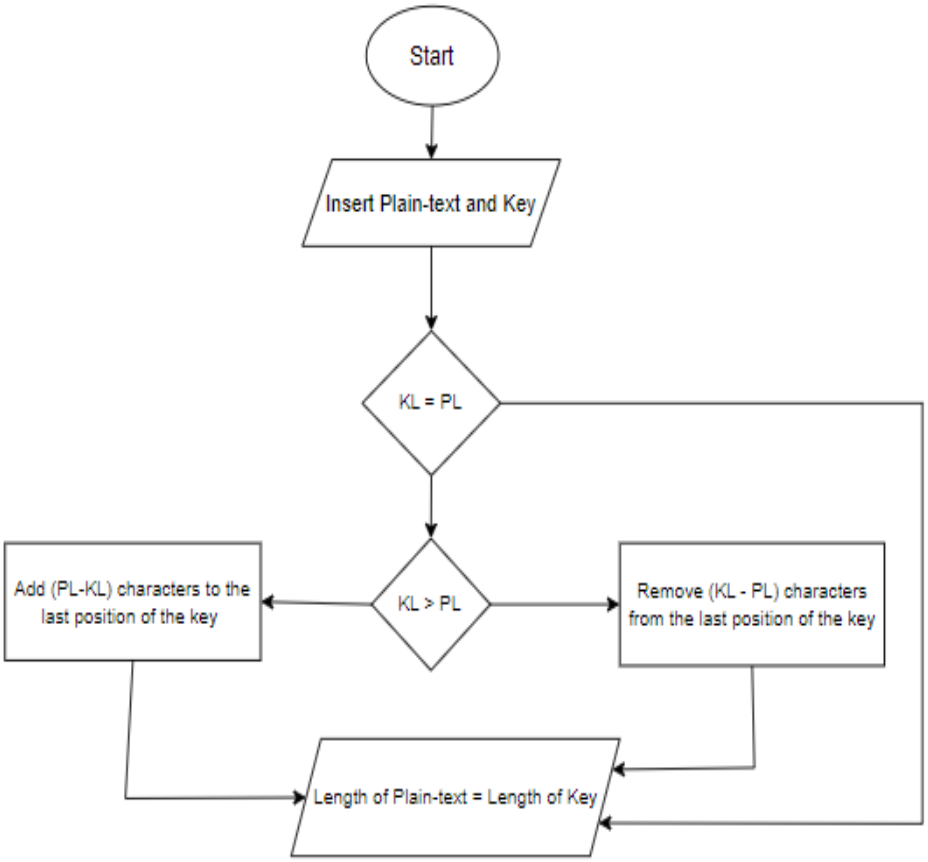

Figure 10: Logical modification of ROT-13 in flowchart

- Convert the value of plain text and key to ASCII code.
- Apply XNOR gate to the values of 13th place after plain text and key.
- Convert the binary code back to its character form to obtain the encrypted message.
- Again to decrypt the message, apply XNOR gate to the binary values obtained from the ASCII table and shift 13 counts for each character.

This new algorithm may be called ROT 2.0 cipher.

5.3. Why was this modification necessary?

The ROT-13 algorithm was particularly vulnerable to exploitation because of its many weaknesses, making it unsecure for sending private messages. There is great potential for this cipher if a few things were adjusted. Thus, this is the goal.

The original ROT-13 cipher's limited security and use cases were intended to be addressed by the ROT 2.0 cipher, which was created by modifying the ROT-13 cipher. Compared to the original ROT-13 cypher, the ROR 2.0 cipher offers greater encryption by applying the XNOR gate to the binary forms of the plain text and key. This strengthens the encryption method's security by making it harder for an attacker to decrypt the cipher text. The ROT 2.0 cipher also gives the customer more control and freedom by letting them select their own key, which further enhances the encryption process. The method of key length correction additionally helps in removing the potential for errors brought about by mismatched key and plain text lengths.

5.4. Encryption Algorithm for ROT 2.0

Step 1: Start

Step 2: Address a plain-text and a key.

Step 3: Shift the position of plain-text by 13 characters.

Step 4: Adjust the length of the key as per the condition to equal the length of both plain-text and key.

Step 5: Using ASCII table, convert the values of the equal length, shifted value of plain-text and key to its binary form.

Step 6: Apply the XNOR gate to the shifted value of plain-text and the key to receive a new binary output.

Step 7: This output in binary form is then converted back to the characters using ASCII table to get out encrypted text.

Step 8: End

In flowchart:

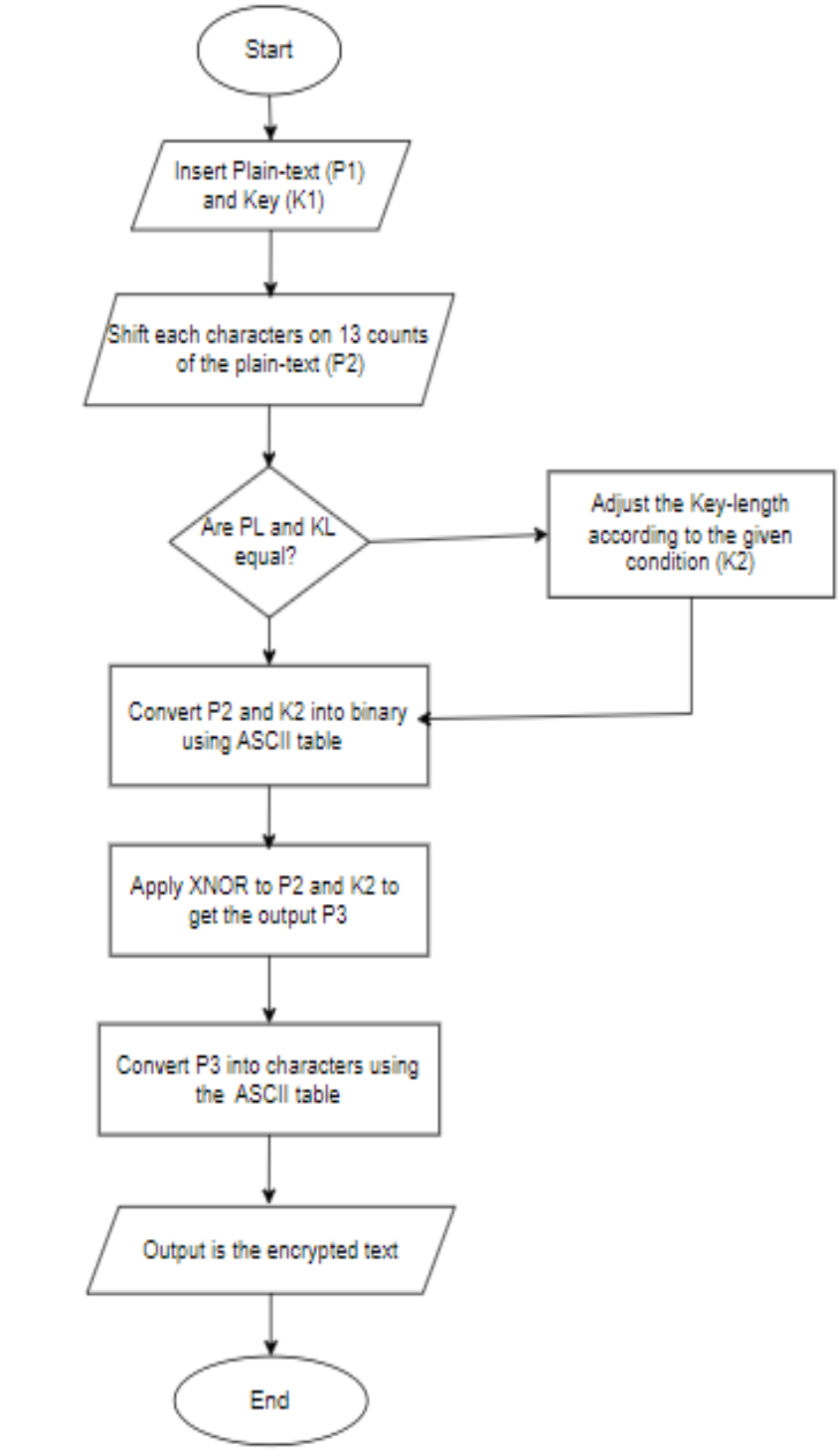

Figure 11: Encryption for ROT 2.0 in flowchart

5.5. Decryption Algorithm for ROT 2.0

Step 1: Start

Step 2: Use the encrypted text and the key used in encryption process.

Step 3: Adjust the length of the key as per the condition to equal the length of both encrypted plain-text and key.

Step 4: Using ASCII table, convert the values of the equal length, encrypted value of plain-text and key to its binary form.

Step 5: Apply the XNOR gate to the key and encrypted plain-text to receive a new binary output.

Step 6: This output in binary form is then converted back to the characters using ASCII table to get the decrypted text.

Step 7: End

In flowchart:

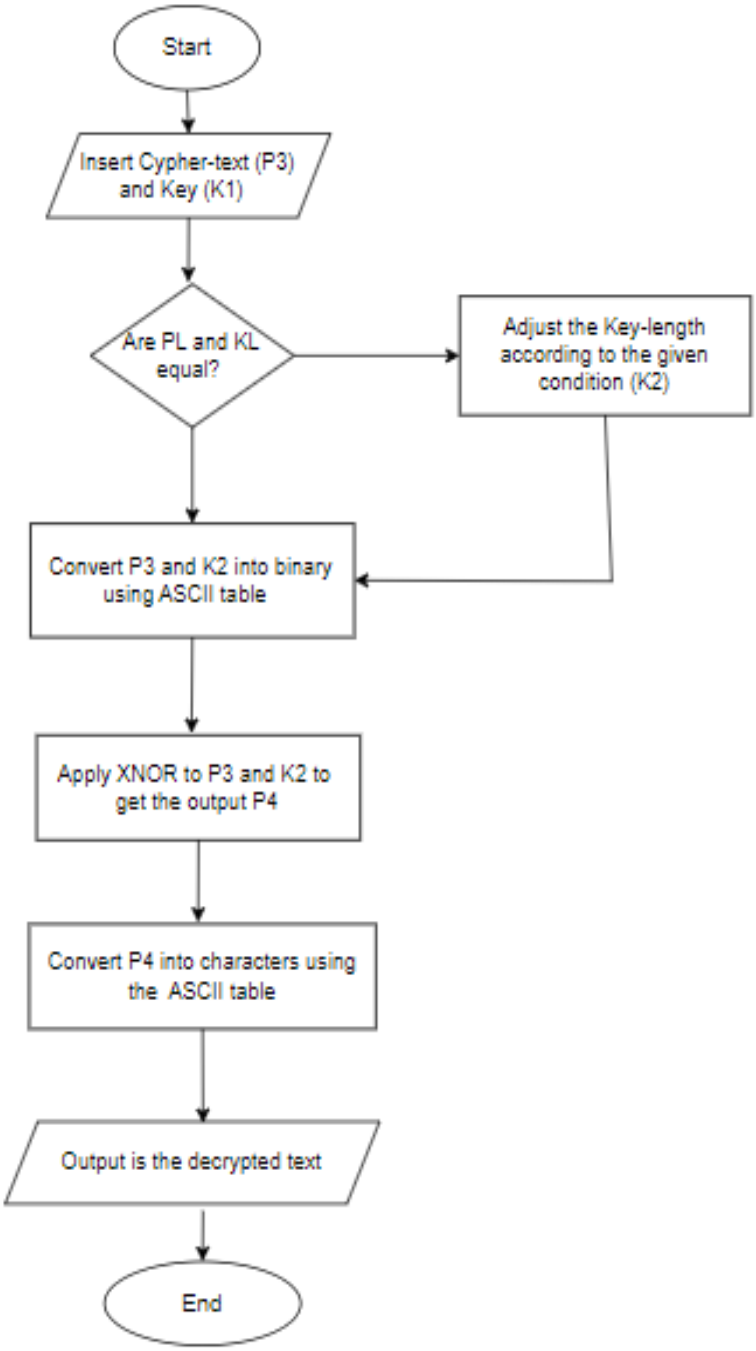

Figure 12: Decryption for ROT 2.0 in flowchart

6. Testing

6.1. Test 1

Given,

Plain-text (P1) = cupcake

Key (K1) = spoon

Encryption Process:

Shifting the value of plain-text by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

Now,

Shifted plain-text (P2) = phcpnxr

Then,

Key (K1) = spoon

Analysing the length of both key and plain-text

Number of characters in Plain-text > Number of characters in Key

Therefore,

Add two extra characters:

Key (K2) = spoonxx

Converting P2 and K2 to binary using ASCII table,

P2 (binary): p = 1110000

h = 1101000

c = 1100011

p = 1110000

n = 1101110

x = 1111000

r = 1110010

K2 (binary): s = 1110011

p = 1110000

o = 1100111

o = 1100111

n = 1101110

x = 1111000

x = 1111000

Applying XNOR gate to P2 and K2

P2 XNOR K2

1110000	XNOR	1110011	=	1111100
1101000	XNOR	1110000	=	1100111
1100011	XNOR	1101111	=	1110011
1110000	XNOR	1101111	=	1100000
1101110	XNOR	1101110	=	1111111
1111000	XNOR	1111000	=	1111111
1110010	XNOR	1111000	=	1110101

Let the value be addressed as P3

P3 (binary) = 1111100 1100111 1110011 1100000 1111111 1111111 1110101

Converting this binary value in characters using ASCII table.

P3 = |gs`DELDELu

Decryption Process:

Cypher-text (P3) = |gs`DELDELu

P3 (binary): | = 1111100

g = 1100111

s = 1110011

` = 1100000

DEL = 1111111

DEL = 1111111

u = 1110101

Key (K1) = spoon

Analysing the length of key and cypher-text.

P3 > K1

So,

We add the necessary characters to equal the length of K1 and P3

i.e. Key (K2) = spoonxx

Converting K2 into binary using ASCII table.

K2 (binary): s = 1110011

p = 1110000

o = 1100111

o = 1100111

n = 1101110

x = 1111000

x = 1111000

Applying XNOR gate to K2 and P3

K2 XNOR P3

1110011	XNOR	1111100	=	1110000
1110000	XNOR	1100111	=	1101000
1101111	XNOR	1110011	=	1100011
1101111	XNOR	1100000	=	1110000
1101110	XNOR	1111111	=	1101110
1111000	XNOR	1111111	=	1111000
1111000	XNOR	1110101	=	1110010

Let the value be addressed as P4.

Converting the binary value into characters using ASCII table.

P4 = phcpnxr

Shifting the value of P4 by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

The decrypted cipher-text is cupcake.

6.2. Test 2

Given,

Plain-text (P1) = dog

Key (K1) = pan

Encryption Process:

Shifting the value of plain-text by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

Now,

Shifted plain-text (P2) = qbt

Then,

Key (K1) = pan

Analysing the length of both key and plain-text

P1 = K1

Converting P1 and K1 to binary using ASCII table,

P1 (binary): q = 1110001

b = 1100010

t = 1110100

K1 (binary): p = 1110000

a = 1100001

n = 1101110

Applying XNOR gate to P1 and K1

P1 XNOR K1

1110001	XNOR	1110000	=	1111110
1100010	XNOR	1100001	=	1111100
1110100	XNOR	1101110	=	1100101

Let the value be addressed as P2

P2 (binary) = 1111110 1111100 1100101

Converting this binary value in characters using ASCII table.

P2 = ~|e

Decryption Process:

Cypher-text (P3) = ~|e

P2 (binary): ~ = 1111110

| = 1111110

e = 1100101

Key (K1) = pan

Analysing the length of key and cypher-text.

P2 = K1

Converting K1 into binary using ASCII table.

K2 (binary): p = 1110000

a = 1100001

n = 1101110

Applying XNOR gate to K1 and P1

K1 XNOR P2

1110000	XNOR	1111110	= 1110001
1100001	XNOR	1111100	= 1100010
1101110	XNOR	1100101	= 1110100

Let the value be addressed as P3.

P3 = 1110001 1100010 1110100

Converting the binary value into characters using ASCII table.

P3 = qbt

Shifting the value of P3 by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

The decrypted cipher-text is dog.

6.3. Test 3

Given,

Plain-text (P1) = peacock

Key (K1) = elephant

Encryption Process:

Shifting the value of plain-text by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

Now,

Shifted plain-text (P2) = crnpbpx

Then,

Key (K1) = elephant

Analysing the length of both key and plain-text

Number of characters in Plain-text < Number of characters in Key

Therefore,

Remove the extra character:

Key (K2) = elephant

Converting P2 and K2 to binary using ASCII table,

P2 (binary): c = 1100011

r = 1110010

n = 1101110

p = 1110000

b = 1100010

p = 1110000

x = 1111000

K2 (binary): e = 1100101

l = 1101100

e = 1100101

p = 1110000

h = 1101000

a = 1100001

n = 1101110

Applying XNOR gate to P2 and K2

P2 XNOR K2

1100011	XNOR	1100101	=	1111001
1110010	XNOR	1101100	=	1100001
1101110	XNOR	1100101	=	1110100
1110000	XNOR	1110000	=	1111111
1100010	XNOR	1101000	=	1110101
1110000	XNOR	1100001	=	1101110
1111000	XNOR	1101110	=	1101001

Let the value be addressed as P3

P3 (binary) = 1111001 1100001 1110100 1111111 1110101 1101110 1101001

Converting this binary value in characters using ASCII table.

P3 = yatDELuni

Decryption Process:

Cypher-text (P3) = yatDELuni

P3 (binary): y = 1111001

a = 1100001

t = 1110100

DEL = 1111111

u = 1110101

n = 1101110

i = 1101001

Key (K1) = elephant

Analysing the length of key and cypher-text.

P3 < K1

So,

We remove the extra characters to equal the length of K1 and P3

i.e. Key (K2) = elephant

Converting K2 into binary using ASCII table.

K2 (binary): e = 1100101

l = 1101100

e = 1100101

p = 1110000

h = 1101000

a = 1100001

n = 1101110

Applying XNOR gate to K2 and P3

K2 XNOR P3

1100101	XNOR	1111001	=	1100011
1101100	XNOR	1100001	=	1110010
1100101	XNOR	1110100	=	1101110
1110000	XNOR	1111111	=	1110000
1101000	XNOR	1110101	=	1100010
1100001	XNOR	1101110	=	1110000
1101110	XNOR	1101001	=	1111000

Let the value be addressed as P4.

Converting the binary value into characters using ASCII table.

P4 = 1100011 1110010 1101110 1110000 1100010 1110000 1111000

P4 = crnpbpx

Shifting the value of P4 by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

The decrypted cipher-text is peacock.

6.4. Test 4

Given,

Plain-text (P1) = hello world

Key (K1) = food eater

Encryption Process:

Shifting the value of plain-text by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

Now,

Shifted plain-text (P2) = uryyb jbeyq

Then,

Key (K1) = food eater

Analysing the length of both key and plain-text

Number of characters in Plain-text > Number of characters in Key

Therefore,

Add one extra character:

Key (K2) = foodx eater

Converting P2 and K2 to binary using ASCII table,

P2 (binary): u = 1110101

r = 1110010

y = 1111001

y = 1111001

b = 1100010

j = 1101010

b = 1100010

e = 1100101

y = 1111001

q = 1110001

K2 (binary): f = 1100110

o = 1101111

o = 1101111

d = 1100100

x = 1111000

e = 1100101

a = 1100001

t = 1110100

e = 1100101

r = 1110010

Applying XNOR gate to P2 and K2

P2 XNOR K2

1110101	XNOR	1100110	=	1101100
1110010	XNOR	1101111	=	1100010
1111001	XNOR	1101111	=	1101001
1111001	XNOR	1100100	=	1100010
1100010	XNOR	1111000	=	1100101

1101010	XNOR	1100101	=	1110000
1100010	XNOR	1100001	=	1111100
1100101	XNOR	1110100	=	1101110
1111001	XNOR	1100101	=	1100011
1110001	XNOR	1110010	=	1111100

Let the value be addressed as P3

P3 (binary) = 1101100 1100010 1101001 1100010 1100101
1110000 1111100 1101110 1100011 1111100

Converting this binary value in characters using ASCII table.

P3 = lbibe p|nc|

Decryption Process:

Cypher-text (P3) = lbibe p|nc|

P3 (binary): l = 1101100

b = 1100010

i = 1101001

b = 1100010

e = 1100101

p = 1110000

| = 1111100

n = 1101110

c = 1100011

| = 1111100

Key (K1) = food eater

Analysing the length of key and cypher-text.

P3 > K1

So,

We add the necessary characters to equal the length of K1 and P3

i.e. Key (K2) = foodx eater

Converting K2 into binary using ASCII table.

K2 (binary): f = 1100110

o = 1101111

o = 1101111

d = 1100100

x = 1111000

e = 1100101

a = 1100001

t = 1110100

e = 1100101

r = 1110010

Applying XNOR gate to K2 and P3

K2 XNOR P3

1100110	XNOR	1101100	=	1110101
1101111	XNOR	1100010	=	1110010
1101111	XNOR	1101001	=	1111001

1100100 XNOR 1100010 = 1111001

1111000 XNOR 1100101 = 1100010

1100101 XNOR 1110000 = 1101010

1100001 XNOR 1111100 = 1100010

1110100 XNOR 1101110 = 1100101

1100101 XNOR 1100011 = 1111001

1110010 XNOR 1111100 = 1110001

Let the value be addressed as P4.

Converting the binary value into characters using ASCII table.

P4 = uryyb jbeyq

Shifting the value of P4 by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

The decrypted cipher-text is hello world.

6.5. Test 5

Given,

Plain-text (P1) = coursework is done

Key (K1) = you do it

Encryption Process:

Shifting the value of plain-text by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

Now,

Shifted plain-text (P2) = pbbhefrjbrx vf qbar

Then,

Key (K1) = you do it

Analysing the length of both key and plain-text

Number of characters in Plain-text > Number of characters in Key

Therefore,

Add the extra characters:

Key (K2) = youxxxxxxx do itxx

Converting P2 and K2 to binary using ASCII table,

P2 (binary): p = 1110000

b = 1100010

h = 1101000

e = 1100101

f = 1100110

r = 1110010

j = 1101010

b = 1100010

r = 1110010

x = 1111000

v = 1110110

f = 1100110

q = 1110001

b = 1100010

a = 1100001

r = 1110010

K2 (binary): y = 1111001

o = 1101111

u = 1110101

x = 1111000

x = 1111000

x = 1111000

x = 1111000

x = 1111000

x = 1111000

x = 1111000

d = 1100100

o = 1101111

i = 1101001

t = 1110100

x = 1111000

x = 1111000

Applying XNOR gate to P2 and K2

P2 XNOR K2

1110000	XNOR	1111001	=	1110110
1100010	XNOR	1101111	=	1110010

1101000 XNOR 1110101 = 1100010

1100101 XNOR 1111000 = 1100010

1100110 XNOR 1111000 = 1100001

1110010 XNOR 1111000 = 1110101

1101010 XNOR 1111000 = 1101101

1100010 XNOR 1111000 = 1100101

1110010 XNOR 1111000 = 1110101

1111000 XNOR 1111000 = 1111111

1110110 XNOR 1100100 = 1101101

1100110 XNOR 1101111 = 1110110

1110001 XNOR 1101001 = 1100111

1100010 XNOR 1110100 = 1101001

1100001 XNOR 1111000 = 1100110

1110010 XNOR 1111000 = 1110101

Let the value be addressed as P3

P3 (binary) = 1110110 1110010 1100010 1100010 1100001 1110101 1101101
1100101 1110101 1111111
1101101 1110110

1100111 1101001 1100110 1110101

Converting this binary value in characters using ASCII table.

P3 = vrbbaumeuDEL mv gifu

Decryption Process:

Cypher-text (P3) = vrbbaumeuDEL mv gifu

P3 (binary): v = 1110110

r = 1110010

b = 1100010

b = 1100010

a = 1100001

u = 1110101

m = 1101101

e = 1100101

u = 1110101

DEL = 1111111

v = 1101101

f = 1110110

q = 1100111

b = 1101001

a = 1100110

r = 1110101

Key (K1) = you do it

Analysing the length of key and cypher-text.

P3 > K1

So,

We add the extra characters to K1 to equal the length of K1 and P3

i.e. Key (K2) = youxxxxxxx do itxx

Converting K2 into binary using ASCII table.

K2 (binary): y = 1111001

o = 1101111

u = 1110101

x = 1111000

x = 1111000

x = 1111000

x = 1111000

x = 1111000

x = 1111000

x = 1111000

x = 1111000

d = 1100100

o = 1101111

i = 1101001

t = 1110100

x = 1111000

x = 1111000

Applying XNOR gate to K2 and P3

K2 XNOR P3

1111001	XNOR	1110110	=	1110000
1101111	XNOR	1110010	=	1100010
1110101	XNOR	1100010	=	1101000
1111000	XNOR	1100010	=	1100101
1111000	XNOR	1100001	=	1100110
1111000	XNOR	1110101	=	1110010
1111000	XNOR	1101101	=	1101010
1111000	XNOR	1100101	=	1100010
1111000	XNOR	1110101	=	1110010
1111000	XNOR	1111111	=	1111000

1100100	XNOR	1101101	=	1110110
1101111	XNOR	1110110	=	1100110

1101001	XNOR	1100111	=	1110001
1110100	XNOR	1101001	=	1100010
1111000	XNOR	1100110	=	1100001
1111000	XNOR	1110101	=	1110010

Let the value be addressed as P4.

Converting the binary value into characters using ASCII table.

P4 = pbhefrjbrx vf qbar

Shifting the value of P4 by 13 characters,

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j
11	12	13	14	15	16	17	18	19	20
k	l	m	n	o	p	q	r	s	t
21	22	23	24	25	26				
u	v	w	x	y	z				

The decrypted cipher-text is coursework is done.

7. Evaluation

A more secure method of encryption for non-sensitive messages is the goal of the ROT 2.0 cypher, which is a modified version of the ROT-13 cypher. Stronger encryption is one of the ROT 2.0 cipher's many advantages over the ROT-13 cypher, which is evaluated below. Nevertheless, there are a number of drawbacks to this approach, which will be discussed below along with some examples of when to use it and when to avoid using it.

7.1. Strengths of ROT 2.0 cipher

The strengths of this modified cipher are discussed below:

1. Stronger encryption**:** The ROT 2.0 cipher provides stronger encryption compared to the original ROT-13 cypher by applying the XNOR gate to the binary forms of the plain text and key. This makes it more difficult for an attacker to decrypt the cipher text.

2. Customizable key: The user of the ROT 2.0 cipher is free to select their own key, which can be any word. This makes it possible to customize and be more flexible with security.

3. Adjustment of Key-length: The ROT 2.0 cipher uses adjustment criteria to guarantee that the length of the plain text and the key are the same. This removes the potential for errors brought on by mismatched lengths.

4. Human-readable output: The ROT 2.0 cipher produces cipher text that can be easily read by humans, which makes it more user-friendly.

7.2. Weaknesses of ROT 2.0

The weaknesses of ROT 2.0 are discussed below:

1. Limited security: Though it offers enhanced encryption than the original ROT-13 cypher, the ROT 2.0 cipher is still not a secure encryption technique and isn't recommended to use when sending private or sensitive data.

2. Vulnerability to key discovery: If an attacker is able to discover the key used in the ROT 2.0 cipher, they will be able to easily decrypt the cipher text.

3. Limited use cases: The ROT 2.0 cipher is not suitable for transmitting sensitive or information and is only suitable for exchanging above-normal messages.

4. Limited key length: Since the ROT 2.0 cipher's key is only as long as the plain text, it might not be as secure as longer keys combined with other encryption techniques.

5. Vulnerability to known-plaintext attacks: Due to its vulnerability to known-plaintext attacks, the ROT 2.0 cipher allows an attacker to obtain both the plain text and identifying encrypted text They can now figure out the key and decode the cipher text as a result.

7.3. Application areas of ROT 2.0

In situations where users wish to communicate above-normal messages without worrying about robust security, the ROT 2.0 cipher can be implemented. Personal information communication or non-critical business communication may be included. However, it is not a safe encryption method, it should not be used to send sensitive or private information.

Some of the examples of the application areas of ROT 2.0 are:

1. Personal communication: People who want to increase the security of their personal communications such as emails or messaging apps can utilize the ROT 2.0 cipher.

2. Non- critical business communication: Businesses may use the ROT 2.0 cipher for non-essential communications like internal memos or announcements to maintain secure communication within the organization.

3. Business trading: The traders doing business can use ROT 2.0 for the trade information such as bill, receipts, invoices and so on to encrypt the shipping details for security purpose.

It is important to note that the ROT 2.0 cypher should not be used for transmitting sensitive or confidential information, as it is not a secure encryption method. It is only suitable for exchanging above-normal messages.

8. Conclusion

In conclusion, cryptography is a vital method for guaranteeing data security and communication in the modern world. You can prevent unauthorized parties from obtaining sensitive information by encrypting it via symmetric and asymmetric encryption, among other forms. Confidentiality, integrity, and availability (the primary elements of information security) can all be understood using the CIA triad as a framework. Furthermore, by comprehending logic gates and how to apply them to modify current cyphers, even more secure encryption techniques may be developed. But it's crucial to keep in mind that no security approach is flawless, and in order to guarantee its effectiveness, security measures must be periodically evaluated and tested. In today's increasingly connected society, the importance of cybersecurity cannot be overstated, and it is important for individuals and organizations to take steps to protect themselves against potential threats.

9. References

Academic. (2024). Retrieved from https://academickids.com/encyclopedia/index.php/ROT13

Academy, K. (2024). Retrieved from https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:computers/xcae6f4a7ff015e7d:logic-gates-and-circuits/a/logic-gates

ASCII. (2024). Retrieved from https://theasciicode.com.ar/

Dahl, Ø. N. (2022, September 15). Retrieved from https://www.build-electronic-circuits.com/xnor-gate/

Danial. (2021). Retrieved from https://www.trentonsystems.com/blog/symmetric-vs-asymmetric-encryption

Deep. (2024). Retrieved from https://deepai.org/machine-learning-glossary-and-terms/xnor

Frulinger, J. (2022, May 22). *CSO*. Retrieved from https://www.csoonline.com/article/569921/what-is-cryptography-how-algorithms-keep-information-secret-and-safe.htm

GeeksForGeeks. (2024). Retrieved from https://www.geeksforgeeks.org/rot13-cipher/

IBM. (2023). Retrieved from https://www.ibm.com/blog/

Irwin, L. (2023, 2 14). *IT Governance*. Retrieved from itgovernance.co.uk: https://www.itgovernance.co.uk/blog/what-is-the-cia-triad-and-why-is-it-important

Javatpoint. (2024). Retrieved from https://www.javatpoint.com/xnor-gate-in-digital-electronics

Nagaraj, K. (2023, February 21). Retrieved from https://infosecwriteups.com/understanding-rot13-encryption-method-2023-cfea53b21770

Okeke, F. (2022, August 9). Retrieved from https://www.techrepublic.com/article/asymmetric-vs-symmetric-encryption/

Prakash, M. (2023, October 12). *knowledgehut* . Retrieved from https://www.knowledgehut.com/blog/security/cia-in-cyber-security#components-of-the-cia-triad%C2%A0

Sidhpurwala, H. (2023, January 12). *redhat*. Retrieved from https://www.redhat.com/en/blog/brief-history-cryptography

SSL. (2023). Retrieved from https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

Table, A. (2024). Retrieved from https://www.asciitable.xyz/static/0df1900e683d0151d0a0c94d390c02e6/2bef9/ascii-table.png

Thakkar, J. (2020, May 23). *SecurityBoulevard*. Retrieved from https://securityboulevard.com/2020/05/types-of-encryption-5-encryption-algorithms-how-to-choose-the-right-one/

Tutorialspoint. (2024). Retrieved from https://www.tutorialspoint.com/cryptography/origin_of_cryptography.htm