 doi:10.20944/preprints201902.0252.v3

Optimal Covert Communication Technology

Moses Oyaro Okello
Self-employed, Gulu, Uganda
e-mail: mosesokellomoses@gmail.com

Due to advancement in hacking/reverse engineering tools, threat against transfer of sensitive data or highly classified information is always at risk of being intercepted by an attacker. Covert communication outwit this malicious breach of privacy act better than cryptography as it camouflage secret information inside another innocent looking information, while cryptography shows scrambled information that might arouse attention of an attacker. However, the challenges in Steganography are the modification of carrier that causes some abnormalities, which is detectable and often the methods are not optimize. This paper presents an approach in Covert communication Chanel, which utilizes mathematical concept of combination to optimize time of transmission using sets of multiple transmitter's, and receiver's addresses where each abstractly represents a set of bits or characters combination without modifying the address. To minimize the number of physical address to be use, a combinatorial and permutation concept of virtual address generation from physical address is introduce. The paper in addition presents some technique like relationship and their application in both re-enforcing resistivity against Steganalysis and generating combinations. Furthermore, a concept of dynamical clockwise and anti-clockwise rotation of combination over addresses after every transmission is introduce to further improve on resistivity against Steganalysis. A simple test was performed for demonstrating relay address, combination and permutation concepts. Based on test results and analysis, the method is effective as expected and it is quite easy to use as it can be implemented in different platform without much difficulties.

Keywords: Steganography; Cryptography; Algorithm; Combinatorial; Permutation

1. BACKGROUND STUDIES

1.1. Introduction

Information security threats come in many different forms. Some of the most common threats today are software attacks, theft of intellectual property, identity theft, theft of equipment or information, sabotage, and information extortion. In awake of increasing cyber warfare, threat against transfer of sensitive data or highly classified Information is always at great risk of being leak or tap by an attacker. One of the solutions to this is through steganography[1,2,3] which is the science of camouflaging secret data in a cover medium in order to produce a stego-medium in which the information is imperceptible to all except the recipient. Steganography is a word defined by Johannes Trithemus as "Covered writing"[4],[5]. It hides a message inside another message without drawing any suspicion to others so that its intended recipient [6] can only detect the message. Traditional steganography methods try to hide information in the Noise of the data by distorting original data just enough to embed a message without this distortion being noticeable. While steganography hide the existence of a message, It is not enough to simply encipher the traffic, as criminals detect, and react to the presence of encrypted communications, But when information hiding is used, even if an attender tap the transmitted object, he or she cannot surmise the communication since it is carried in a hidden way. Limitation of cryptography is that the third party is always aware of the communication because of the unintelligible nature of the text. Steganography overcomes this limitation by hiding message in an innocent looking object called carrier, cover, or stego-object. In addition, still attacker can be attracted towards encrypted data due to different form of data. Therefore, this limitation can be overcome by using steganography [7].

Some organization or government may not allow encrypted communication [8]. Therefore, steganography is the best-suited way for secret communication for such case in case of alarming situation where confidential is very crucial. Furthermore, Staganography is an ever-growing research topic as the idea of secret communication attracts many researchers [9]. Although steganography techniques provide us with some more secure communication, the modification is traceable by using some brand-new "Steganalysis" tools [10]. Good Steganography method when mixed with cryptography [11,12] becomes nearly impossible to intercept encrypted contents. Network steganography is preferred over other methods of steganography such as image steganography as image often may be modified during transmission due to either image filtering, resizing, scaling, transformation and many others. This may distort the hidden information such that the final recipient does not get the right message, as for example Least Significant Bits (LSB) method is very vulnerable to image modification.

1.2. Related Works

This sub-section introduces some few related work in the area of covert Communication channel and some method use for detecting those covert communication channel.

A paper presented by [13], introduces two methods base on packet length. In their first method, the sender encodes a bit of data in each pair that included two non-identical length and the second method where packets are separated into buckets. Only a packet within a bucket can be pairs as indicated by the authors. the drawback however is that the method is applicable in situation where packet length does not have a constant value and further since the method relies on swapping of the packets for instances in their first method, it's prone to being detected as minute abnormality can be detected. In addition, Guangjie Liu, Jiantao Zhai, Yuewei Dai in a paper [14], which in their methods "treats the network traffic as the flow with fixed-length fragment, and calculates the histogram of the packet delays in each fragment. They modulate a message bits into the delays by binary coding method, while keeping the histogram almost unchanged by assigning the matched distribution". However due to advances in technology in the area of detecting hidden methods in the covert channel, there are several methods which do detect such minute changes or modification of the network traffics flows. such as those presented by Steven Gianvecchio and Haining Wang [15] which presented an entropy base method for detecting covert channel and also another one which detect word-base algorithmically generated

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

Domains by using inter-word and inter-domains correlation using semantics analysis [16]. They took into account word embedding and same part of speech in addition to detecting using frequency distribution of words and part of speech.
Furthermore, a paper [17], presented a method based on time interval or delays, which takes the interval of the time such as $\Delta t = t_i - t_{i-1}$ and Δt is then compare with chosen sequences of keys to decode or encode hidden binary. The paper further discusses the drawback of most steganography methods, which are due to modification of carrier, and that it gives loophole for some sophisticated algorithm or statistical method to detect steganography flow. Nevertheless, this problem can be solve in this proposed novel methods, which does not modify anything. In addition, the fact that most method hides a single bit one after another, it is extremely difficult to attained optimal transmission based on time for transmitting these floods of bits and this proposed method introduces bit/string and address combination techniques to combat such problem.

2. METHODOLOGY

This paper presents theoretical mathematical approach in security(Network steganography), which utilizes mathematical concept of combinatorial, and binary string concatenation by combining bits to optimize time of transmission, using multiple transmitters and receivers addresses, for example email addresses, mail addresses, phone numbers, or network ports or addresses and many others where each is assigned a bits combination. In addition, the paper present the concept of dynamical clockwise and anti-clockwise rotation of bits combination over the given transmitter-receiver addresses after every transmission to reinforce resistivity against any form of security analysis like (Steganalysis) or cryptanalysis.

2.1. Combination

This section presents the concept of combination of bits or characters such that maximum bits or characters can be send at once by making a given address represent such bit combination. Below shows examples of bits combination. In addition, how a formulae for calculating total number of possible bits or character combination "*C*" given that a number of bits or character to be combine is *"n"*. Let W be set of string of possible combinations resulting from a given combinatorial such that $W = \{w_0, w_1, w_2, w_3, .. w_{C-2}, w_{C-1}\}$ and C is the total number of elements of w excluding empty set $w = \{\emptyset\}$.
Below indicates bit combination.
One by one bit combination; $w = \{0, 1\}$ so total combination are $c = 2$ and $n = 1$. However for
Two binary combination, the possibility are $W = \{00, 01, 10, 11\}$ and total combination C=2 where $n = 2$
For three binary combination, $W = \{000, 001, 010, 100, 011, 101, 110, 111\}$ total combination $C = 2^3$ where n=3
For four binary combination are; $W = \{0000, 0001, 0010, 0100, 1000, 0011, 0101, 1111, \dots\}$ and total combination $C = 16$ where n=4

Therefore, the pattern keep on increasing such that the power of possible binary combination "*C"* for a given bit combine "*n"* can be express as $C = 2^n$.
From the above, it is clearly evidence that, for binaries combination, the maximum possible binary combination "C" is as below where "*n" is* the total number of binaries combination. given that $W = \{w_0, w_1, w_2 \dots, w_{(2^n-2)}, w_{(2^n-1)}\}$. Total element of set W can be express as follows. We know that total elements (cardinality) of a set W can be express as $v = n(W)$.
$C = v^n$ For example $W = \{0,1\}$, total elements of set W is two $n(W) = 2$, meaning binary or base two number system, so $C = 2^n$
Therefore C can be express as;

$$C = v^n \tag{2.1}$$

Kleen Star Approach

By using Kleene Star formulae [18] derived from Kleene plus. Let Kleene plus be V^+ and Kleene star be V^*
Therefore,

$$V^+ = \bigcup_{n=1}^{\infty} V_n \text{ implies that } V^* = \bigcup_{n \in \mathbb{N} \cup \emptyset} V_n \tag{2.2}$$

$V^+ = \{V_1 \cup V_2 \cup V_3 \dots . V_n\}$ In addition, given that, $V_0 = \emptyset$ an empty set and Kleene Star is given as $V^* = V_0 \cup V^+$ therefore $V^* = \{V_0 \cup V_1 \cup V_2 \cup V_3 \dots . V_n\}$.
If $V = \{"z", "y"\}$, then $V^* = \{\emptyset, "z", "y", "zz", "zy", "yz", "yy", "zzz", "zzy", "zyz", "zyy", \dots\}$
If $V = \{"x", "z", "y"\}$, then $V^* = \{\emptyset, "x", "z", "y", "xx", "xy", "xz" \dots, "xxx", "xxz", "xxy", "xzz", \dots\}$
Now substituting z=0, y=1 as binary character of string, the above can be rewritten as below.
If $V = \{0,1\}$, then $V^* = \{\emptyset, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \dots\}$
Given that, $V_0 = \emptyset$ for any alphabet, the set of all strings over V of length n is denoted as V_n. If $V = \{"z", "y"\}$, then $V^* = \{\emptyset, "z", "y", "zz", "zy", "yz", "yy", \dots\}$. Substituting that with binary bit of string, becomes $V = \{0,1\}$ then $V^* = \{\emptyset, 0, 1, 00, 01, 10, 11 \dots\}$. For, $V_1 = \{0,1\}$. and then the total element of the sets in V_1 is given as $C = 2, C = 2^1$. However, $V_2 = \{00, 01, 10, 11\}$. and then the total element of the sets in V_2 is given as $C = 4, C = 2^2$ with the exception of the empty set $V_0 = \emptyset$. Therefore, the total is minus one. From above total summation of elements of sets of Kleene plus for a given "n" bits combination is. C Where $C = v^n$
For more examples, see Table 1 where a binary combination is assign to addresses.
Further, V' is defined as set of cardinality of all V_n so, $V' = \{n(V_1), n(V_2), n(V_3), \dots \dots, n(V_n)\}$ therefore, for an example, if $V = \{0,1\}$ so $V' = \{2^1, 2^2, 2^3, \dots, 2^n\}$ where $v = n(V)$ therefore $V' = \{v^1, v^2, v^3, \dots \dots, v^n\}$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

2.2. Relationship

Given two sets of addresses A, B such that $\{(a,b): a \in A, b \in B\}$ and each element of one set is maximally relating to all the elements of the other sets and the inverse relationship holds true$\big(A = \{a_0, a_1, a_2, \ldots . a_i\} : i \in \mathbb{N} \;\; and \; B = \{b_0, b_1, b_2, \ldots . b_j\}: j \in \mathbb{N}\big)$The relationship of the two sets "A" and "B" can be describe as; $\mathcal{R} \subseteq AxB = \{(a,b): a \in A, b \in B\}$. For the inverse case where the receiver wants to reply to the sender, the relationship inverse can be express as $\mathcal{R}^{-1} = \{(a,b): (a,b) \in \mathcal{R}\}x$ and q are total elements of sets A and B respectively.

Therefore since the transmission is related, such that,$\{a,b\} \in \mathcal{R}$, a $\mathcal{R}$ b, Maximum crossing among addresses" $\mathcal{L}$" for sending information can be express as $\mathcal{L} = n(A) * n(B) \implies 0 < \mathcal{L}; \; \mathcal{L} \in \mathbb{N}$ Or $\mathcal{L} = n(A)n(B)$

And to calculate total number of cross transmission is as $\mathcal{L}$. Please see Figure 1 below for the address relationship involving only Transmitters and Receiver addresses.

2.2.1. Homogenous Relationship

A relationship involving two or more set where each of the set have same base size and same element type for example a give set $\{V_0, V_1, V_2, .., V_i\}$ and base set of $n(V_1) = n(V_2) =, .., n(V_i)$ and their relationship results in homogenous combination. For example, set V can be set of binary number, so $\{V_1, V_2, .., V_i\}$ are all binary set, and their base set of all are $\{0,1\}$

So for an example, given $v_1 = \{00,01,10,11\}$ and $v_2 = \{0,1\}$. Therefore, relationship between $\{v_1, v_2\}$ are homogenous and produces homogeneous combination $v_3 = \{000, 001, 010, 100, 011, 101, 110, 111\}$

2.2.2. Non-Homogenous relationshi

A relationship involving two or more sets where each of the set have different base size and different element types for example a give set $\{v_1, v_2, .., v_i\} \notin V$ and base set of $(v_1) \neq (v_2) \neq, .., v_i$ and their relationships results in non-homogenous combination. For example set V can be set of mixture of number, character etc. $\{v_1, v_2, .., v_i\}$ their base set of all are different $v_1 = \{0,1\}, v_2 = \{x, y, t\}, \; v_3 = \{0, d, P\}, \ldots \ldots$

So for more example, given $v_1 = \{00,01,10,11\}$ and $v_2 = \{x, y, z\}$ so, the results is

$$v_3 = \{00x, 01x, 10x, 11x, 00y, 01y, 10y, 11y, \quad 00z, 01z, 10z, 11z\}$$

Therefore, relationship between $\{v_1, v_2\}$ are non-homogenous and produces non-homogeneous combination.

2.2.3. Transmitter's to Receiver's Address Relationship

Please see figure 1 for relationship without relay address directly from sender to recipient address without intermediate address.

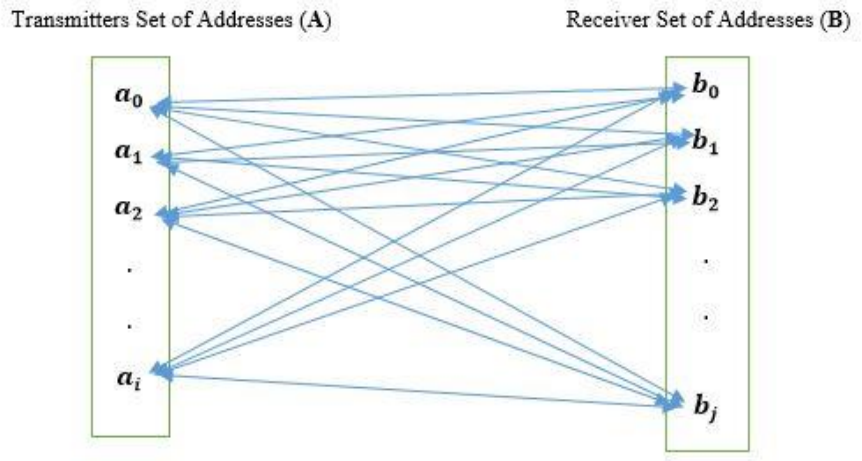

Figure 1: Shows Relationship Transmitter and Receivers Addresses

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

2.2.4. Transmitter's Receiver's Address Relationship with Relay Addresses

Below Fig 2 shows, relationship involving relay addresses where multiple relay addresses as intermediate address before final destination address.

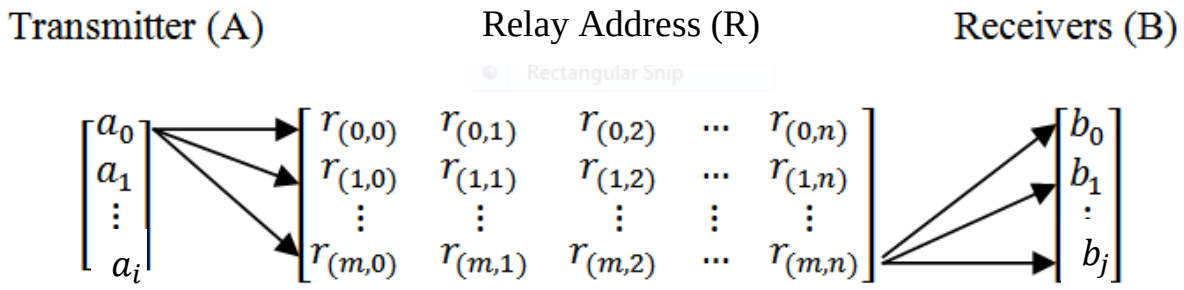

Figure 2: Shows Relationship Involving Relay Addreses

$$\mathcal{L} = f_0 * f_1 * f_2 * \dots * f_n \tag{2.3}$$

$$d = f_0 + f_1 + f_2 + f_3 + \cdots . + f_n \tag{2.4}$$

In (2.3) and (2.4), $f_i = n(r_i)$ and $n(A) = f_0, n(B) = f_n$ from figure 2

2.3. Maximization of Address

Here, an idea of how to maximize total number of address based on concept of combinatorial and permutation to produce more virtual addresses. This approach is idea that given a set of address (A) with more than one distinct element i.e.$n(A) \geq 2$, a given combination of virtual address can be generated. For example. Given address such as $A = \{a, b, c\}$, virtual address A' can be generated combinatorial as $A' = \{ab, ac, bc, abc\}$ and for permutation as

$A' = \{ab, ba, ac, ca, bc, cb, abc, acb, bca, bac, cab, cba\}$. These are all distinct element although some are virtual and others are real address, so total address available for use has increased to $A + A'$. Three real addresses has generated four virtual addresses and in total seven new addresses are available for use when using combinatorial concept, However, for permutation twelve virtual addresses are generated from three real physical addresses. 15 addresses available for us.

2.3.1. Combinatorial of Address

For combinatorial ${}_nC_r$ of address to generate virtual address, here, order of virtual address combination does not matter much as transmission is simultaneous, so indexing address is difficult if order of address combination is to be taken into account such as in permutation.

For an address with two real elements, virtual can be generated as $A = \{a, b\}$ where virtual is $A' = \{ab\}$ only one distinct virtual address element can be generated.

So formulae for finding total elements of virtual address A' i.e. $n(A')$ that can be used from a given real address A is shown here (2.5) where x number of real address total is $n(A)$ and u is Combination ${}_xC_r$ of address, r is the selected address in combination.

$$u = \frac{x!}{r!(x-r)!} \tag{2.5}$$

$$s.t\ 0 \leq r \leq x$$

Unlike permutation, in combinatorial order of elements or their arrangements does not matter. For instance, $\{ab\}$ is the same as $\{bc\}$. And for three elements, such as $\{abc\}$ is the same as $\{acb\}, \{cab\}, \{bca\}, \{bac\}, \{cba\}$ ... ...

So total virtual addresses generated from real distinct address can be as below in formulae (2.6)

$$n(A') = \sum_{r=2}^{x} \left(\frac{x!}{r!(x-r)!}\right) \tag{2.6}$$

However, for the total elements of real and virtual addresses see (2.7) also known as Grand address G

$$G = \sum_{r=1}^{x} \left(\frac{x!}{r!(x-r)!}\right) \tag{2.7}$$

$$G = n(A) + n(A') \tag{2.8}$$

When $(r = x)$ so equation (2.5) equals one. In addition, when $(r = 1)$ therefore, equation (2.5) equals (x) or $n(A)$.

So equation (2.7) can be rewritten as in (2.9)

$$G = (x + 1) + \sum_{r=2}^{x-1} \left(\frac{x!}{r!(x-r)!}\right) \tag{2.9}$$

Total possible Relation $\mathcal{L}$ from (2.3) involving virtual addresses generated from combination in (2.2) can be express as in (2.30)

$$\mathcal{L} = \left((x_0 + 1) + \sum_{r=2}^{x_0-1} \left(\frac{x_0!}{r!(x_0-r)!}\right)\right)\left((x_1 + 1) + \sum_{r=2}^{x_1-1} \left(\frac{x_1!}{r!(x_1-r)!}\right)\right) \tag{2.30}$$

For relationship involving relay addresses, please see equation (2.31). In addition, for $i = 0$ represent address A for $i = n$ represents Address B, for $0 \leq i \leq n$ represent relay addresses.

$$\mathcal{L} = \prod_{i=0}^{n} \left((x_i + 1) + \sum_{r=2}^{x_i-1} \left(\frac{x_i!}{r!(x_i-r)!}\right)\right) \tag{2.31}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

For address can be express as in (2.32)

$$d = \left((x_0 + 1) + \sum_{r=2}^{x_0-1}\left(\frac{x_0!}{r!(x_0-r)!}\right)\right) + \left((x_1 + 1) + \sum_{r=2}^{x_1-1}\left(\frac{x_1!}{r!(x_1-r)!}\right)\right) \quad (2.32)$$

However, for address involving relay address please see (2.33)

$$\mathrm{d} = \sum_{i=0}^{n}\left((x_i + 1) + \sum_{r=2}^{x_i-1}\left(\frac{x_i!}{r!(x_i-r)!}\right)\right) \quad (2.33)$$

2.3.2. Permutation of Address

In permutation**,** the order combination of physical address forming virtual address does matter very much because transmissions are sequential not simultaneous, and it is time index i.e. ab is different from ba. So two or more address combination representing one virtual address can be rearrange in such a way that the order of those address distinctively represents different address for example. A transmission from physical address a and b can be from the virtual address ab transmission received at $t_0$and t_1 respectively given that $t_0 < t_1$. For virtual address ba transmission received at $t_2$and t_3 respectively given that $t_2 < t_3$.

It should be noted that to differentiate between virtual address sequential transmissions, time of transmission from the same combination should be within a defined range Δt or δt see (2.34). Where ω values set such that $\Delta t \leq \omega$

$$\Delta t = t_i - t_{i-1} \quad (2.34)$$

From equation (2.5), permutation, ${}_{\mathrm{n}}\mathrm{P}_{\mathrm{r}}$ of such combination is the permutation of entire virtual address plus physical address generated from combination can be written or express in (2.35).

$$G = \sum_{r=1}^{x}\left(\frac{x!}{(x-r)!}\right) \quad (2.35)$$

Where x is total number of address and r number of address chosen. Just like in (2.30~2.33) total relationship and addresses involving relay address can be written as in (2.36) and (2.37).

$$\mathcal{L} = \prod_{i=0}^{n}\left(\sum_{r=1}^{x_i}\left(\frac{x_i!}{(x_i-r)!}\right)\right) \quad (2.36)$$

$$d = \sum_{i=0}^{n}\sum_{r=1}^{x_i}\frac{x_i!}{(x_i-r)!} \quad (2.37)$$

Please note: handling zero factorial in address here is when its zero.

Zero factorial is defined as a mathematical expression for the number of ways to arrange a data set with no value in it, which equals one by definition ($0! = 1$). So, for this address, since total number of address is greater than zero, so $r \geq 0$ so (r) much be greater than zero.

2.4. Concatenation

The concept of concatenation is mainly use in formal language theory like in programming languages and pattern. Concatenation of two strings a and b is often denoted as ab, $a||b$, or, in the Wolfram Language, $a <> b$ [19]. However, throughout this text, it is denoted as a$||b$. From the two sets of strings of binary assigned to addresses *A* and *B*, the concatenation $A||B$ consists of all strings of the form $a||b$ where "a*"* is a binary string from *A* and "b*"* is a string from *B*, or formally

$A||B$= {$a||b$: $a \in A$, $b \in B$} for concatenation of a string set and a single string, and vice versa. $A||b$ = {$a||b$: $a \in A$} and $a||B$ = {a$||b$: $b \in B$}

However, as given by the work [19], the concatenation of two or more numbers is the number formed by concatenating their numerals. He gave an example, the concatenation of 1, 234, and 5678, whichare12345678. In addition, the value of the result depends on the numeric base. He further presented the formula for the concatenation of numbers p and q in base β as in (2.38).

$$p||q = p\beta^{l(q)} + q \text{ Where } l(q) = \lfloor \log_\beta q \rfloor + 1 \quad (2.38)$$

$l(q)$Is the number length of "q" in base "β" and $\lfloor x \rfloor$ is the floor function.

The above work well when "p" is a non-floating point number or number without decimal point like $p = 23, p = 12$ etc but for floating point number like $p = 5.56,\ q = 34.03$ etc., it yield different result when compared with a number concatenation treated as string. In addition, numbers with zero in front like for example $q = 0045, q = 034,\ q = 00018$, yield less floor function not as intended as zero before a number is disregarded unless it is before a decimal point.

Therefore, throughout this paper, binaries, or stream of bits are treated as string and string concatenation formulae and rules/law are applied as below**.**

2.5. Associative Law

Rules of Binary operation applicable to string Concatenation here below. For the binary operation, is associative and repeated application of the operation produces the same result regardless how valid pairs of parenthesis are inserted in the expression. A product of two elements (addresses or bits combination) $((a, b): a \in A, b \in B)$may be written in five possible ways as below.

1. $((a||b)||a)||b$ 2. (a||b)||(a||b) 3. (a||(b||a))||b 4. a||((b||a)||b) 5. a||(b||(a||b))

Since the product operation is associative, the generalized associative law says that all these formulas will yield the same result, making the parenthesis not relevant. Thus, "the" product is as below:

a||b||a||b

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

For example involving bits combination, Binary concatenation and port assignment, for four by four bits, combination and concatenation see Table 1.

2.5.1. Order of Concatenation

Below, we discussed some of the order of concatenation that needs to be followed in order not to lose track of the transmitted code of combination. See Table 1 showing sample bit combination

Table 1: Four by four bit combination assigned to addresses

Sender Addresses	Sender's Bits	Receiver Addresses	Receiver's Bit
a_0	0000	b_0	0000
a_1	0001	b_1	0001
a_2	0010	b_2	0010
a_3	0100	b_3	0100
a_4	1000	b_4	1000
a_5	0011	b_5	0011
a_6	0110	b_6	0110
a_7	1100	b_7	1100
a_8	1001	b_8	1001
a_9	0101	b_9	0101
a_{10}	1010	b_{10}	1010
a_{11}	0111	b_{11}	0111
a_{12}	1110	b_{12}	1110
a_{13}	1101	b_{13}	1101
a_{14}	1011	b_{14}	1011
a_{15}	1111	b_{15}	1111

Given time series of transmission, as $T = \{t_0, t_1, t_2, \dots, t_n\}$ such that $(t_n > t_{n-1} > t_{n-2} > t_{n-3} > \cdots > t_1 > t_0)$
By defining sender to receiver order as O_{SR} and receiver to sender order as O_{RS}
Therefore from sender to receiver order $O_{SR} = (a||b)_{t_0}||(a||b)_{t_1}||(a||b)_{t_2}|| \dots \dots ||(a||b)_{t_n}$
And from receiver to sender order $O_{RS} = (b||a)_{t_0}||(b||a)_{t_1}||(b||a)_{t_2}|| \dots \dots ||(b||a)_{t_n}$
Above is an example of bit combination in table form shown in Table 1
Base on combination concept, to send letter 'H'=01101001, Can be separated into two 4 by 4 bits combination and sent at once in a single transmission so each transmission will carry one character of 8-bits (1byte)

$$O_{SR} = (a_6||b_8)_{t_0} \quad \Longrightarrow a_6||b_8 = \overbrace{01101001}^{a_6||b_8}$$

For reply,

$$O_{RS} = (b_6||a_8)_{t_0} \quad \Longrightarrow b_6||a_8 = \overbrace{01101001}^{b_6||a_8}$$

Application of Relationship for Generating Combination
From the rule of string concatenation given above.

ALGORITHM 1: Homogenous Relationship an example see Table 1

Function: Combination(V, n)

```
START
    Read: T1, T2=V, z=length(V), j=0,i=0,y=0,w=z, u=length(V), x=u*z, M=0, Array_Size(T1, x),R=0
     while( Y<n) loop
          while(I<x) loop
               While(J<w) Loop
                    While(M<u)loop
                        If(R==1)
                              T1[i]=T2[m]||V[j]
                        Else
                              T1[i]=T2[m]||T2[j]
                          End_If
                              M++
                    End_while
                      J++
               End_while
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

```
                    I++
                End_while
                    If($u^2 < z^n$)
                            R=1
                            W=u
                     Else
                  R=0
                 W=z
            End_if
                  U=length(T1)
                 T2=T1
                 T1=null
               X=u*z
                  Array_Size(T1,x)
        Y++
End_while
 Return T2
 STOP
End_Function
```

Please note from the above algorithm 1, function Array_size(x,y) is for pre-allocation of array x of size y however in some programming language, array size are allocate dynamically and use of this function is not needed. See Figure 3 for example of above algorithm 1 implementation.

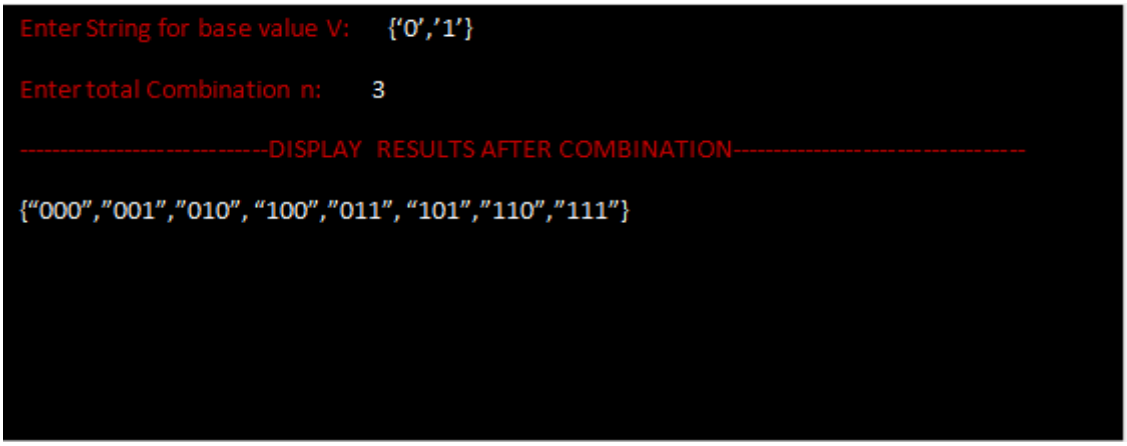

Figure 3: Shows Output of Algorithm 1

ALGORITHM 2: Iterative Algorithm for non-homogenous relationship

```
Function:Combination(V1, V2)
    START
 Read: h=0, i=0, j=0; C1 = lengtht(V1) C2 = length(V2),temp
            while(i <C2) loop
                  while(j <C1) loop
                      j=j+1
                   temp[h]= V2[i]|| V1[j];
                     h=h+1;
                end_while
                  i=i+1
          end_while
 Return temp
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

STOP

Algorithm 2 for input two relationship with non-homogenous see output in Figure 4

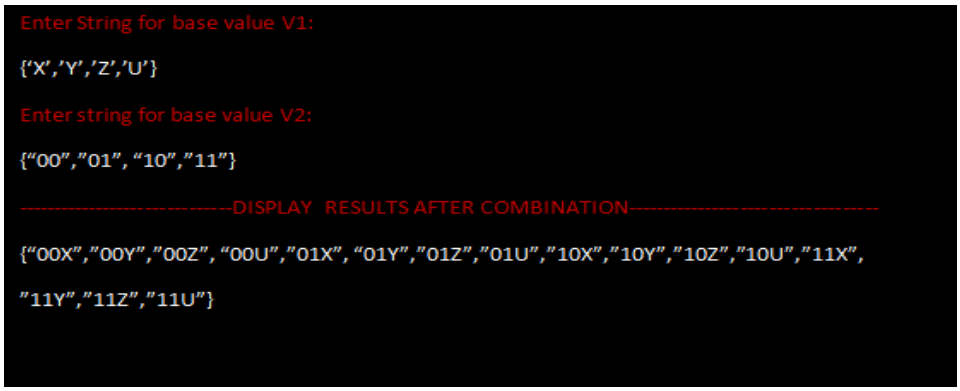

Figure 4: Shows sample output of the algorithm 2

2.6. Rotation over Addresses

Let W represent sets of bit combination over addresses *A*, or *B*. for subscribe of W_I "*I*" is the current position of bit combination over a given addresses A or B of index *I*, and *"J"* represents total transmission time such that($I \in \mathbb{N}$, $J \in \mathbb{Z}$).In addition, Q is the maximum addresses ($Q \in \mathbb{N}$) or in other word, totals elements of set A or B.for example if $w_2 = a_2 = b_2$

Given a bit combination, "*C"* can rotate over elements of either set A or B after every transmission such that the rotation is either clockwise or anti-clockwise. $A = \{a_{Q-1}, a_{Q-2}, .. a_2. a_1, a_0\}$ or $B = \{b_{Q-1}, b_{Q-2} \dots b_2, b_1, b_0\}$ so W can be express as $W = \{w_{Q-1}, w_{Q-2} \dots w_2, w_1, w_0\}$

Below is the general equation, and conditions for the rotation in the positive direction or clockwise direction and Negative or anti-clockwise direction. See (2.39) and (2.40) respectively.

Let modular (mod) or remainder operator for dividing a number by another is ($mod = \oslash$) to avoid confusing mod with other letter symbols presented here in the paper. For instance, $A \; mod \; B = R$ can be express as $A \oslash B = R \quad or \quad 5 \; mod \; 4 = 1$ can be written as $5 \oslash 4 = 1$ see Algorithm and 4 of (2.39) and (2.40)

Positive (Clockwise) Rotation

$$f(J, \mathrm{w}, Q, I) = \begin{cases} \mathrm{W}_{I+J}, & 0 \le J < Q : (I+J) < Q \\ \mathrm{W}_{((I+J) \oslash Q)}, & 0 \le J < Q : (I+J) \ge Q \\ \mathrm{W}_{(I+(J \oslash Q)) \oslash Q}, & J \ge Q : (I + (J \oslash Q)) \ge Q \\ \mathrm{W}_{(I+(J \oslash Q)) \oslash Q}, & J \ge Q : (I + (J \oslash Q)) \ge Q \end{cases} \tag{2.39}$$

Negative (Anti-Clockwise) Rotation

$$f(J, \mathrm{w}, Q, I)^* = \begin{cases} \mathrm{W}_{I+J}, & -Q < J \le 0 : (J+I) \ge 0 \\ \mathrm{W}_{(I+J)+Q}, & -Q < J \le 0 : (J+I) < 0 \\ \mathrm{W}_{I+(J \oslash -Q)}, & J \le -Q : \big(I + (J \oslash -Q)\big) \ge 0 \\ \mathrm{W}_{(I+(J \oslash -Q))+Q}, & J \le -Q : \big(I + (J \oslash -Q)\big) < 0 \end{cases} \tag{2.40}$$

ALGORITHM 3: written based on Clockwise Rotation

Q;p;I;J;W_t;
for J from 1 to p loop
  if J>=0&& J<Q then
    if (I+J)>=Q then
      T_J ←((I+J∅Q)
        else
          T_J ← (I+J)
        endif
      else
      if((J∅Q)+I)>=Q then
      T_J ←(((J∅Q)+I)∅Q)
      else
      T_J ←(J∅Q)+I
    endif

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

```
   endif
endloop
```

For result of above algorithm 3, see figure 5

Data of Clockwise Rotation

8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5

Fig 5: Shows Data for Clockwise Rotation

From figure 5: initial Position is seven and total addresses are sixteen
Initialization of variables

ALGORITHM 4: Written base don Anticlockwise Rotation

```
Q;J;I;i;P;W_t;
for i from 1 to P  loop
 J←(-i)
    if J>-Q&&J<=0 then
          if (I+J)>=0 then
             T_i ←(I+J)
              else
                T_i ←((I+J)+Q)
                endif
             else
            if(I+ (J∅-Q))>=0 then
              T_i ←(I+ (J∅-Q))
            else
              T_i ←((I+ (J∅-Q)) +Q)
          endif
   endif
endloop
```

Below are data from the above algorithm 4.

Data of Anticlockwise Rotation

4 3 2 1 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 10 15 14 13 12 11 10 9 8 7

Fig 6: Shows Data for anticlockwise Rotation and initial Position is five and total addresses are sixteen

Application of Rotation Functions to Rotate Array of String

Below are pseudo-code created functions that uses the above two function for rotation, to rotate array of string. Since stream of bits (binary) are treated as string so it can be converted into it subsequence array and the array index manipulated such that it is rotated as prescribe above.

The idea creates two functions where each is input initial index I, J, Q and the function returns a numeric index after rotation for each element of the array.

Furthermore, another function which call the two rotation function is created which determine the value of J

Where if it is negative, it calls anti-clockwise rotation, else it calls clockwise rotation, and the function returns array of all string after its rotation see Fig 7 for the output of the algorithm being tested.

ALGORITHM 5: Iterative Algorithm for Rotation function

```
Function:clockwise(I,J,Q)
START
read: I, J, Q ;
   if (J>=0&& J<Q) then
     if ((I+J)>=Q)then
         T=(I+J) ∅Q
              else
         T=(I+J);
     end_if
   else
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

```
      if(((J∅Q)+I)>=Q) then
         T=(((J∅Q)+I) ∅Q);
      else
         T=(J∅Q)+I;
      end_if
     end_if
return T;
                    STOP
Function: Anti_clockwise(I, J,Q)
START
read:  I, J, Q
 if (J>-Q &&J<=0) then
    if ((I+J)>=0) then
      T=(I+J);
    else
     T=((I+J)+Q);
    end_if
  else
     if ((I+(J∅-Q))>=0) then
        T=(I+(J∅-Q));
     else
          T =(I+(J∅-Q)+Q);
       end_if
return  T;
STOP
Function: Rotation(W, Q,  J)

START
Read:I=0;W = {w0, w1, w2 ... wQ−1};  J.
  While( I<Q) loop
     if(J<0)
       R[Anti_clockwise(I, J, Q)] = W[I];
     else
      R[clockwise(I, J, Q)] = W[I];
     end_if
   end_while
 I++;
return R:
STOP
```

The figure below see Fig 7 which display results after rotating array string of addresses

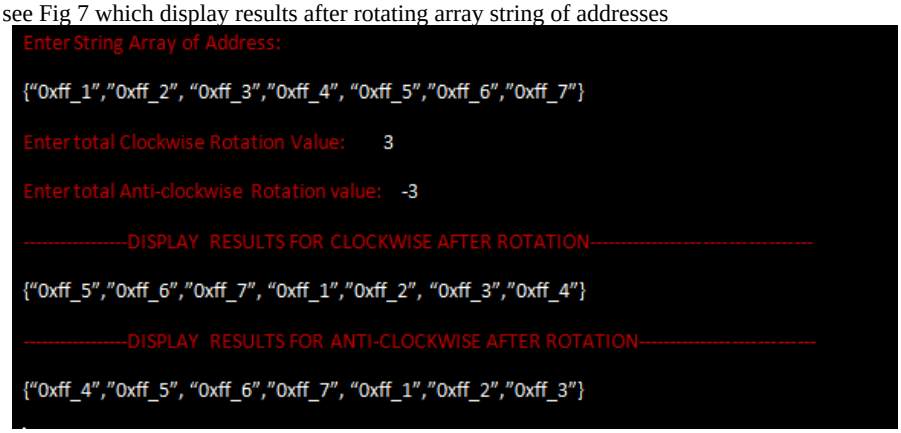

Figure 7: Shows Clockwise and Anti-clockwise Rotation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

3. EXPERIMENTAL RESULT

In this section, three experimental results are presented where the first one is done in a very simple environment to make it easily understandable by non-specialist is covert channel communication and easy to perform the experiment. It is based on the idea of using many phone numbers from two different locations where the confidential information is to be transmitted from and to a given location with those numbers representing the address. The second experiment was done using multiple client machines located in given location where information is to be send from to another location where recipient is located, in the recipient location are locate multiple server machine through this use of email forwarding functionality and use of intermediate proxy server is used.

3.1. Experimental Result Based on Rotation

In this, at the sender side, six phone numbers was set up as sender addresses and receiver side six phone numbers also was set up as receiver addresses. At sender side, only two bit combination were used and the same four at the receiver side. See Table 2

Table 2: Phone number at sender and receiver assigned bit combination

Sender phone	Bit combination	Receiver Phone	Bit combination
+256-788-011	00	+256-789-001	Null
+256-789-081	Null	+256-777-222	00
+256-711-022	01	+256-772-412	10
+256-777-187	10	+256-777-011	01
+256-772-101	Null	+256-772-111	11
+256-777-787	11		

To withhold the identity of the user's phone number from exposing, only six digits are shown without their location of calls only just labeled as sender and receiver's location.

Furthermore an extra phone number is used in addition to the one assigned bit combination, those phone number are label as null which means it does not carries any bit combination and any phone called from or to such carries no bit combination whether it is directed to the one with assigned bit combination see Table 2 for more details.

The following phone calls were recorded at receiver sides with their respective time of call see Table 3

Table 3: Extracted time of calls from phone number

Sender phone	Receiver Phone	Time of call	Extracted Bit Combination
+256-711-022	+256-772-412	09:39	(null-01)
+256-772-101	+256-772-111	09:40	(01-null)
+256-777-187	+256-772-111	10:53	(00-00)
+256-788-011	+256-772-412	10:59	(01-00)
+256-777-787	+256-772-111	11:28	(00-11)
+256-777-18	+256-777-011	11:29	(10-11)

From table 3, the extracted bit combination can be concatenated excluding the any combination from null phone number
(Null-01)+ (01-null) + (00-00) + (01-00) + (00-11) + (10-11)
Therefore, the stream of bits is (0000010000111011)

3.2. Experimental Result 2 Based on Relay Address and Permutation Without rotation

Electronic Mail Forwarding functionality:

Here from the email address sender, send an open message to recipient via relay email by forwarding the email.

Please see table 5 below contains sample email for transmission through relay address. To withhold the identity of email address involve in this practically, notation is use to represent email address such as A1@email.com. In Table 4

Table 4: Number of Email address with email forwarding as relay address

Sender Email	Bits	Relay Email	Bits	Receiver	Bits
A1	00	R1	00	B1	00
A2	01	R2	01	B2	01
A1~A2	10	R1~R2	10	B1~B2	10
A2~A1	11	R2~R1	11	B2~B1	11

The following message were recorded from the above emails address in the order
(A2-R2~R1-B1)+ (A1-R2-B1~B2) + (A2-R1-B2~B1) + (A1~A2-R1-B2) decoding this into binary (011100000110010011010001)

4. ANALYSIS

4.1. Optimal Performance

Since this combine bits such that two or more bit can be send at once, so time of transmission reduces significantly, as many bits can be send at once.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

Let "t" be time required for sending total of "η" bit or character combination for both relay and or without relay address at once. And "k" is a constant unit time per one bit transmission such that: $\eta t \propto k \quad \Longrightarrow \quad \eta t = k$ and $t = \frac{k}{\eta}$

$$t = k\eta^{-1} \tag{4.1}$$

Therefore $\lim_{\eta\to\infty}(t = \frac{k}{\eta}) \approx 0$; given that $k = 1$

From the above, it is evidence that when number of bits combination increases so does the time required for transmission reduces.

4.2. Secure Analysis Using Probability

This section shows how resistive the method is from someone or a program that need to detect it by applying probability theory.

Probability without Relay Address

Let sample space be τ for a total address at either sender or receiver's side where a bit combination can occupy at a given time. Therefore, the probability that a chosen bit combination occupy such address is as below: $probability(P) = \frac{1}{\tau} \ |\tau = n(A), \ \tau = n(B), \ \Longrightarrow \ P = \frac{1}{\tau}$

For the probability that it is not is given as $P' = \left(1 - \frac{1}{\tau}\right) \ \lim_{\tau\to\infty}\left(1 - \frac{1}{\tau}\right) \approx 1$

However, the probability that a chosen transmission line carries the right bits combination is different from above as the sample space is from equation and equal to maximum crossing, a combination of bits between sender and receiver addresses send at once. Let sample space be maximum crossing between sender and receiver's addresses

$probability(P) = \frac{1}{\mathcal{L}}$ Probability that it is not, is given as $P' = (1 - P)$

For instance, if $\mathcal{L} = \frac{1}{n(A)n(B)}$ substitute in the above probability equation. $\lim_{\mathcal{L}\to\infty}\left(1 - \frac{1}{\mathcal{L}}\right) \approx 1$

This indicates that, increase in number of bit combination $n \to \infty$, makes probability $P' \approx 1$

Probability with Relay Address

In this part, we shows the probability that a chosen line of transmission from sender to receiver through relay addresses are as below, it's an expansion of the probability for a chosen line transmission above. Here find the probability within each relay from the sender up to receiver.

Consider the following: $A - R_1 - R_2 - R_3 - \cdots . - R_m - B$

From transmitter address A to relay address R_1 the probability that the line carries hidden information is p_A else the probability that it does not carries is denoted as p'_A for the next node of relay address it is given as p_{R_1} and p'_{R_1} respectively this continues up to the last point of receiver.

Overall probability P that the entire transmission carries hidden message involving relay addresses can expressed as an independent probability which is the product of all individual probability [20] and express as below (4.2);

$$P = (p_A)\ (p_{R_1})(p_{R_2}).. (p_{R_m}) \tag{4.2}$$

And the probability that it does not carries is given as (4.3)

$$P' = (p'_A)\left(p'_{R_1}\right)\left(p'_{R_2}\right) \ldots . \left(p'_{R_m}\right)\left(p'_B\right) \tag{4.3}$$

From above it can be noted that as more relay address is added, the probability that a chosen line of transmission caries hidden information tend to almost zero proving theoretically that this approach is good.

5. SAMPLE EXAMPLE

Example 1: Given a bits combination n=3 after J=11 times of transmission in clockwise rotation. In addition, an initial position of the bits I= 3.

a) What is the new position of the bits combination W_3?

Solution:

Therefore: $Q = 2^3 : \ n = 3 \ and \ Q < J$

$$f(J, w, n. I) = W_{(J \oslash 2^n)+I}, \quad J \geq 2^n$$
$$f(J, w, n, I) = W_{(11 \oslash 2^3)+3}$$
$$f(J, w, n, I) = W_6$$

The new position or index is six (6). So if on sender addresses is $a_3 \ to \ a_6$ or for receiver address, it is from $b_3 \ to \ b_6$.

b) Assuming after example 1, and rotating in anti-clockwise for 19 times transmission. What is the new position of the bits?

Solution:

Therefore: $Q = -2^3$; $n = 3 \ and \ J \leq -Q$ and position I=6 is a new position after clockwise rotation.

And since anti-clockwise, J=-19

$$f(J, w, n, I) = W_{(J \oslash -2^n)+I}, \quad J \leq -2^n$$
$$f(J, w, n, I) = W_{(-19 \ \oslash \ -2^3)+6}$$

$f(J, w, n, I) = W_{-3+6=3}$ So new position= 3

The new position or index is six (6). So if on sender addresses is $a_6 \ to \ a_3$ or for receiver address, it is from $b_6 \ to \ b_3$.

Example 2: A person want to send a hidden message by post office mail, from country "A" to country "B" using 8-bits binary system such that the mail carries normal message without being modified:

a) How many mail addresses are requires from sender and receiver country so that at least each mail sent carries a character of 8-bits?

Solution:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 January 2022 doi:10.20944/preprints201902.0252.v3

This can be separated into two 4 by 4 bits where $n = 4$ see Table 1 for reference, such that at sender carries 4-bits and receiver's carries 4-bits. Given total address is $\mathrm{d} = n(A) + n(B)$ and substituting it with $n(A) = n(B) = 2^n$ $\mathrm{d} = 2^n + 2^n \Longrightarrow 2 * 2^4 = 16 + 16 = 32\ addresses.$

Thirty-two addresses, sixteen mail addresses at both sender and receiver's side.

b) From Table 1 As a references for the initial bit combination position for both address at sender and receiver's, what is the new position of the bits such that at sender address, bits are rotated clockwise while at the receiver's address, bits are rotated anti-clockwise? Sender wants to send a word of 50 characters ending with character "o".

Solution:

8-bit for letter "o"=01101111 breaking into 4 by 4 its 0110 and 1111

From Table 1, initial position of $0110 = a_6 = 6$ and for $1111 = b_{15} = 15$. Sender side Rotation Clockwise.

I=6, Q=C= 16, J=50 times.

$$f(J, \mathrm{w}, n, I) = \mathrm{W}_{(J \oslash Q)+I}, \qquad J \geq Q \ ; \ (J \oslash Q) < Q$$

So, $f(J, \mathrm{w}, n, I) = \mathrm{W}_{(50 \oslash 16)+7} = \mathrm{W}_9 \ \ new\ position = 8$

Receiver side Rotation Anti-Clockwise.

I=15, Q= -C=-16, J=-50 times.

$$f(J, \mathrm{w}, n, I) = \mathrm{W}_{I+(J \oslash Q)}, \qquad J \leq -Q \ \ ; \ (J \oslash Q) < 0$$

So $f(J, \mathrm{w}, n, I) = \mathrm{W}_{15-(-50 \oslash -16)} = \mathrm{W}_{15-2} \ \ new\ position = 13$

Overall position for the bit combination over address after 50 times transmission bits rotation is as below:

$from\ (a_6 || b_{15})\ \ to\ (a_8 || b_{13})$.

c) What is the percentage reduction in time of transmitting the bits?

Solution: $t = \frac{k}{2n}$; $n = 4,\ k = 1$. Time of transmission percentage decrease= $\frac{1}{2*4} = 0.125$ percentage.

d) What is the probability that a chosen address contains the right bits combination given that total bits combination $n = 4$?

Solution:

Sample spaces (total addresses over which a bit combination can occupy is as below)

$$C = n(A) = 2^n \ \ ; \ \ n = 4 \qquad \Longrightarrow \qquad n(B) = 2^4 = 16$$

Probability=$\frac{1}{n(B)}$; $n(B) = 2^n \ \ \Longrightarrow \ \ \frac{1}{2^n} = \frac{1}{16} = 0.0625$.

e) How many address and bit combination is required such that each mail sent carries two characters of 8-bit system at once or at a single transmission?

Solution:

Total bits for two character implies one-character at sender and another one at receiver side

At sender and receiver each, need a total of $n = 1 * 8 = 8$ bits combination

Bit combination is given $C = 2^n$; $n = 8 \ \Longrightarrow \ 2^8 = 256$-bits combination and addresses needed at both side of sender and receiver

6. CONCLUSION

Base on proves, examples and analysis, it is practically impossible to detect hidden information flow. Decrypting hidden message is extremely hard as nothing modifies like in most steganography methods, which involve modification of the carrier. Besides that, due to the rotation of bits, characters or strings combination over given addresses after every transmission makes it difficult to decipher the hidden contents, In addition, the use of methods such as relay address, homogenous and non-homogenous combination further strengthen this methods in term of security and optimal performance. Lastly, by using combination technique enable transmitting many bits at once unlike in other method where a single bit is transmitted one after another. Also combination technique allows generation of many virtual addresses from few physical addresses hence reducing number of physical addresses in use also improving on security. However transmission is based on multiple Sender-receiver's addresses, it should be sequential not simultaneous to avoid confusion.

7. CONFLICT OF INTEREST

Author declare that there is no conflict of interest

8. REFERENCES

1. D. Artz, "Digital steganography: Hiding data within data," *IEEE InternetComputing*, vol. 5, no. 3, pp. 75–80, May/Jun. 2001.
2. F. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—Asurvey," *Proc. IEEE: Special Issue on Protection of Multimedia Content*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.
3. N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," IEEE Security and Privacy, vol. 1, no. 3, pp. 32–44, May/Jun. 20
4. L. Y. Por, and B. Delina, "Information hiding- a new approach in text steganography," *7th WSEAS Int. Conf. on Applied Computer and Applied Computational Science*, 2008, pp. 689-695.
5. L. Y. Por, T. F. Ang, and B. Delina, "WhiteSteg- a new scheme in information hiding using text steganography," *WSEAS Transactions on Computers*, vol.7, no.6, pp. 735-745, 2008.
6. S. Changder, D. Ghosh, and N. C. Debnath, "Linguistic approach for text steganography through Indian text," 2010 2nd*Int. Conf. on Computer Technology and Development*, 2010, pp. 318-322.
7. Kefa. Rabah, "Steganography-the art of hiding data," *Information Technology Journal*, vol.3, pp. 245-269, 2004.

8. Sandib Bobade, Rajeshawari Goudar, “Secure Data Communication Using Protocol Steganography in IPv6,2015” International Conference on Computing Communication Control and Automation *IEEE 2015*
9. Jun O Seo, Sathiamoorthy Manoharan, Aniket Mahanti “network Steganography and steganalysis a concise Review ” 2016 2nd IEEE international conference on Applied and theoretical Computing and Communication Technology.
10. Chengshen Yuan, Zhihua Xia,Xingming Sun “Coverless image Steganography base on SIFT and BOF”, Journal of Internet Technology 2017, P435-442
11. Huy Hoang Ngo, Xianping Wu, Phu Dung Le, Cambel Wilson, and Balasubramaniam Srinivasan “Dynamic key Cryptography and Applications” *International journal for network security*, Vol 10 N0.3, PP.161-174 may 2010.
12. Stallings, W., 2017. *Cryptography and network security: principles and practice* (pp. 92-95). Upper Saddle River: Pearson.
13. Sabeti, Vajiheh, and Minoo Shoaei. "New High Secure Network Steganography Method Based on Packet Length." *ISeCure* 12.1 (2020).
14. Liu, Guangjie, Jiangtao Zhai, and Yuewei Dai. "Network covert timing channel with distribution matching." *Telecommunication Systems* 49.2 (2012): 199-205.
15. Gianvecchio, Steven, and Haining Wang. "Detecting covert timing channels: an entropy-based approach." *Proceedings of the 14th ACM conference on Computer and communications security*. 2007
16. Yang, Luhui, et al. "Detecting word-based algorithmically generated domains using semantic analysis." *Symmetry* 11.2 (2019): 176.
17. M. Okello, "A New Timing Steganography Algorithm in Real-Time Transmission Devices," *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, Chongqing, 2018, pp. 880-884. doi: 10.1109/ICCT.2018.8600103
18. Ebbinghaus, H.-D., Flum, J., Thomas, Wolfgang,”Mathematical Logic” (2nd ed.). New York: Springer. p. 656. ISBN 0-387-94258-0
19. Weisstein, Eric W. “Concatenation.” From MatheWorld—A Wolfram WebResource. http://mathworld.wolfram.com/Concatenation.html
20. Stephanie Glen. "Probability of Two Events Occurring Together" From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/how-to-find-the-probability-of-two-events-occurring-together/