A Secure and Optimal Method of Steganography Using Bit Combination and Dynamical Rotation over Addresses.

Authors Name 1: Moses Okello  
Permanent Affiliation: Jaso Skynet Technology,  
Permanent Address: Gulu, Uganda  
E-mail: mosesokellomoses@gmail.com

ABSTRACT: Steganography is the branch of cryptology that deals with concealing information inside another information. It’s prefer over cryptography as it hides a message inside another innocent looking message, while cryptography shows scrambled message that might arouse attention of an attacker, however the challenges in steganography is the modification of carrier that causes some abnormalities which is detectable and often the methods are not optimize. This paper presents a novel approach in cryptography and steganography, which utilizes mathematical concept of bits combination to optimize time of transmission using sets of multiple transmitter, and receiver’s addresses where each abstractly presents a bits combination without itself modified. In addition, the paper introduces a concept of dynamical clockwise and anti-clockwise rotation of bits combination over addresses after every transmission to reinforce resistivity against steganalysis.

Keywords: General Mathematics, Combinatorial, Algebra, Steganography, Probability, Cryptography.

1. BACKGROUND

Steganography[1,2,3] is the science of hiding secret data in a cover medium in order to produce a stego-medium in which data are undetectable to all except the recipient. 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. Steganography is derived from a finding by Johannes Trithemius meaning, “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. 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 attendant 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]. Moreover, some organization or government may not allow encrypted communication [8]. Therefore, steganography is the best way for secret communication. Furthermore, it is an ever-growing research topic as the idea of hiding information 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 technique using dynamical key [11] either Asymmetric or symmetric encryption [12] and applying concept of multichannel and cryptography [13] becomes extremely powerful. However, in my previous work [14], 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 hid a single bit one after another, it is extremely difficult to attained optimal transmission base on time for transmitting this flood of binaries and the proposed method introduces bit combination to combat such problem. Network steganography is prefer over 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 recipients does not get the right message as for example Least Significant Bits (LSB) method is very vulnerable to image modification.

2. Steganography Methods

This paper presents a theoretical mathematical approach in security(Original 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 BITS COMBINATORIAL. This section presents the concept of combination of bits such that maximum bits can be send at once by making a given address represent such bit combination. Below shows examples of bits combination. In addition, how a formula for calculating total number of possible bits combination “C” given that a number of bits to be combine is “n”. Let \( \Psi \) be set of string of bit combinations resulting from a given combinatorial such that \( \Psi = \{\psi_0, \psi_1, \psi_2, \psi_3, \ldots \psi_{C-2}, \psi_{C-1}\} \) and \( C \) is the total number of elements of \( \Psi \) excluding empty set \( \Psi = \{\emptyset\} \).

One by one binary combination; \( \Psi = \{0, 1\} \) so total combination are \( C=2 \) and \( n=1 \) however for Two binary combination, the possibility are \( \Psi = \{00, 01, 10, 11\} \) and total combination \( C=2 \) where \( n=2 \)

For three binary combination, \( \Psi = \{000, 001, 010, 011, 100, 101, 110, 111\} \) total combination \( C = 2^3 \) where \( n=3 \)

For four binary combination are; \( \Psi = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\} \) and total combination \( C = 16 \) where \( n=4 \)

Therefore, the pattern keep on increasing such that the exponential of possible binary combination “\( C'\)” for a given bit combine “\( n'\)” can be can 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 \( \Psi = \{\psi_0, \psi_1, \psi_2, \ldots, \psi_{(2^n-2)}, \psi_{(2^n-1)}\} \). Total element of set \( \Psi = C \)

\[ C = 2^n \]

**Prove of Equation \( C = 2^n \) above**

By using Kleene Star formulae [15] derived from Kleene plus

Let Kleene plus be \( V^+ \) and Kleene star be \( V^* \)

Therefore,

\[ V^+ = \bigcup_{n=1}^{\infty} V_n \quad \text{implies that} \quad V^* = \bigcup_{n \in \mathbb{N} \cup \emptyset} V_n \]

\[ V^+ = \{V_1 \cup V_2 \cup V_3 \ldots V_n\} \quad \text{and} \quad V^* = \{V_0 \cup V_1 \cup V_2 \cup V_3 \ldots V_n\} \]

Given that \( V_0 = \emptyset \) an empty set and Kleene Star is given as \( V^* = V_0 \cup V^+ \)

\[ V^* = \{V_0 \cup V_1 \cup V_2 \cup V_3 \ldots V_n\} \]

If \( V = \{"z", "y"\} \), then \( V^* = \{\emptyset, "z", "y", "zz", "zy", "yz", "yy", "zzz", "zyz", "yzy", "yyyy" \ldots \} \)

If \( V = \{"x", "z", "y"\} \), then \( V^* = \{\emptyset, "x", "z", "y", "xxx", "xzz", "xyy", "xxz", "xyz", "xyz", "yyy", "yxz", \ldots \} \)

Now substituting \( z=0, y=1 \) as binary character of string, the above can be rewritten as below.

If \( V = \{0,1\} \), then \( V^* = \{\emptyset, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \ldots \} \)

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 \( V^* = \{\emptyset, "zz", "zy", "yz", "yy"\} \).

Substituting that with binary bit of string, becomes \( V = \{0, 1\} \) then \( V^* = \{\emptyset, 0, 1, 00, 01, 10, 11\} \).

However, \( V_2 = \{0,1,00,01,10,11\} \), and then the total element of the sets in \( V_n \) is given as \( C = 4, C = 2^2, C = 2^n \) 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 \( C = 2^n \)
End of prove.

For more example, see Table (I, II, III) where a binary combination is assign to addresses.

2.2 MULTI-ADDRESS RELATIONSHIP: Given two sets of addresses A and B with an equal number of elements \(\{(a, b): a \in A, b \in B\}\) such that each element of one set is maximally relating to all the elements of the other sets and the inverse relationship holds true \((A = \{a_0, a_1, a_2, ..., a_k\}: i \in \mathbb{N} \text{ and } B = \{b_0, b_1, b_2, ..., b_j\}: j \in \mathbb{N}\) The relationship of the two sets "A" and "B" can be describe as: \(\mathcal{R} \subseteq A \times B = \{(a, b): a \in A, b \in B\}\). For the inverse case where the receiver wants to reply to the sender, the relationship inverse hold true as express \(\mathcal{R}^{-1} = \{(a, b): (a, b) \in \mathcal{R}\}\)

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 \(\mathcal{L} = (q \times x) \implies 0 < \mathcal{L} \leq (q \times x); \mathcal{L} \in \mathbb{N}\) and to calculate total number of cross transmission is below, provided relationship holds true.

2.2.1 CONDITION 1: when \(q \neq x\) and \(q = \mathcal{C}, x = \mathcal{C}\)

For \(x < q; x = 2^n; q = 2^m\) given that \(m \in \mathbb{N}\) Therefore \(q = 2^{n+m}\)

When condition one is encounter, it depict relationship of many to one or one to many or in other word a relation where total from one side does not equal to the total from the other side but to accommodate same bits combination from both side of sender and receiver such that equations holds true, "\(q\)" must be equal to "\(x\)". Which certifies in the below condition 2: So, when \(x < q\), it can be express as \(x = 2^n\), \(q = 2^{n+m}\) given that \(m \in \mathbb{Z}\)

\[(2^{n+m}) \geq 0\]. Therefore \(\mathcal{L} = xq\) and by substituting value of \(q\) and \(x\); \(\mathcal{L} = 2^n \times (2^{n+m})\)

\[\mathcal{L} = 2^{2n+m}\]

However, the total addresses "\(d\)" should be equal to the number of the maximum possible binary combination where each side of sender’s and receiver’s addresses.

\(x = \mathcal{C}, q = \mathcal{C}\) Therefore; \(q = 2^n; x = 2^{n+m};\) since \(q \neq x\) total addresses or ports "\(\Lambda\)" is as below.

\[d = x + q \implies d = 2^n + 2^{n+m}\]

Factoring out \(2^n\) from the above equation, then below is the final equation for total addresses base on Condition one above.

\[d = 2^n(1 + 2^m)\]

2.2.2 CONDITION 2: when \((q = x)\) and \((q = \mathcal{C}, x = \mathcal{C})\)

The above condition 2 shows where all sender set of elements are equally relating to all the other elements of the receiving set. Since \(\mathcal{L} = xq\) can be express as \(\mathcal{L} = q^2\) or \(\mathcal{L} = x^2\) by making either \(q = x\) or \(x = q\).

But the total addresses should be equal to the number of the maximum possible binary combination where \(x = q = \mathcal{C}\) Therefore, \(q = 2^n, x = 2^n\)

So \(q = 2^n; x = 2^n\) therefore total addresses or ports "\(d\)" is as below.

\[d = q + x; d = 2^n + 2^n\]

\[d = 2^{(n+1)}\]

Alternatively, from condition 1: where \(d = 2^n(1 + 2^m)\) by making \(m = 0; d = 2^n(1 + 2^0): m = 0\) therefore \(d = 2^{(n+1)}\)
Substituting $q = 2^n, x = 2^n$ into $L$ and a new equation, which only hold when condition 2 is true such that $q = x$ and $q = 2^n, x = 2^n$ and $L = (2^n * 2^n)$

$$L = 2^{2n}.$$ 

Or, from condition 1, by letting $m = 0$ and substituting into equation $L = 2^{2n+m}$ therefore, $L = 2^{2n+0}$

$$L = 2^{2n}$$

2.2.3 CONDITION 3: when $(q > C, x > C)$ and $(x = q)$

In this scenario when $(q, x) \in 2^n$, is when the available addresses are more than the bit combination $x > 2^n$ and $q > 2^n$. Here I introduce a new two variables $(\ell, \lambda) \in \mathbb{N}$. $0 < \ell < (x - 2^n)$ and $0 < \lambda < (q - 2^n)$

Therefore, $x = 2^n + \ell$ and $q = 2^n + \lambda$ given that $L = xq$

$$L = (2^n + \ell) * (2^n + \lambda)$$

$$L = 2^{2n} + 2^n(\ell + \lambda) + \ell \lambda$$

Given that $d = x + q$ therefore $d = (2^n + \ell) + (2^n + \lambda)$

$$d = 2^{n+1} + \ell + \lambda$$

2.2.4 CONDITION 4: when $(q > C, x > C)$ and $(x \neq q)$

Base on condition 3 above, and condition 1, therefore, the below condition are made. Combination $x > 2^n$ and $q > 2^n+m$. Therefore, when $x < q$, it can be express as $x = 2^n + \ell$ given that; $0 < \ell < (x - 2^n)$ and $q = 2^{n+m} + \lambda$, given that $0 < \lambda < (q - 2^{n+m})$

Given that $L = xq$ and $L = (2^n + \ell) * (2^{n+m} + \lambda)$

$$L = 2^{2n+m} + 2^n(\lambda + \ell 2^m) + \ell \lambda$$

Given the addresses formulae as $d = x + q$ therefore $d = (2^n + \ell) + (2^{n+m} + \lambda)$

$$d = 2^n(1 + 2^m) + \ell + \lambda$$

Prove of the Formulae $L = 2^{2n}$

The formulae is precise and base on condition when $(q, x) \in 2^n$ or when $q = C, x = C$ and $x = q$ such that, $m = 0, \ell = 0, \lambda = 0$

Given that $L = xq$ and by making “$q$” subject of the equation. $q = \frac{L}{x} \implies q = x^{-1}L$

Substituting value of, $L = 2^{2n}$ and $x = C = 2^n$ into equation above

$$q = (2^n)^{-1} * 2^{2n} \implies q = 2^{(-n+2n)} \implies q = 2^n$$

On the other hand, since $q = x$ given that; $q = \sqrt{L}$, by substituting value of $L = 2^n$.

$$q = \sqrt{(2)^{2n}} \implies (\sqrt{(2)^{2n}})^2 = (q)^2$$

$$\sqrt{(2)^{2n}} = q^2$$

Introducing logarithm on both side of the above equation such that:
\[
\log(2^{2n}) = \log(q^2) \quad \Rightarrow \quad 2n \cdot \log(2) = 2 \cdot \log(q)
\]

Making “\(n\)”, the subject of the equation.

\[
n = \frac{2 \cdot \log(q)}{2 \cdot \log(2)} \quad \text{Given that} \quad q = 2^n
\]

Substituting the value of \(q\) into equation and a new equation is form below.

\[
n = \frac{\log(2^n)}{\log(2)}
\]

\[
n = \frac{n \cdot \log(2)}{\log(2)} \quad \Rightarrow \quad n = n \log_2 2 \quad \Rightarrow \quad n = n
\]

**End of Prove.**

Determination of maximum value of \(n\) that can accommodate any given value of \(x, y, C, \mathcal{L}\) or \(d\)

i. Given \(C = 2^n\); \(n = \lfloor \log_2 C \rfloor\)

ii. Given \(\mathcal{L} = 2^{2n}\); \(n = 0.5 \log_2 \mathcal{L}\)

iii. Given \(d = 2^{(n+1)}\); \(n = \lfloor \log_2 d \rfloor - 1\)

iv. \(x = 2^n\); \(n = \lfloor \log_2 x \rfloor\)

v. \(q = 2^n\); \(n = \lfloor \log_2 q \rfloor\)

\([x]\) Is the floor function.

Expression of \(C\) in term of \(\mathcal{L}\) Or "d" from above.

\[
C = 2^{[0.5 \log_2 \mathcal{L}]} \quad \text{Or} \quad C = 2^{[\log_2 d] - 1}
\]

Expression of \(\mathcal{L}\) in term of \(C\) Or "d"

\[
\mathcal{L} = 2^{[\log_2 C]} \quad \text{Or} \quad \mathcal{L} = 2^{[\log_2 d] - 2}
\]

Expression of "d" in term of \(C\) Or \(\mathcal{L}\)

\[
d = 2^{[\log_2 C] + 1} \quad \text{Or} \quad d = 2^{[0.5 \log_2 \mathcal{L}] + 1}
\]

Example 1: Given total sender addresses is \(x = 23\) and receiver addresses is \(q = 35\)

i. Determine the maximum value of \(n\) that can accommodate in the addresses.

Solution:

For \(x = 23\), \(n = \lfloor \log_2 x \rfloor\), maximum \(n = 4\), \(x = 2^4 + 7\)

For \(q = 35\), \(n = \lfloor \log_2 q \rfloor\), maximum \(n = 5\), \(q = 2^5 + 3\)

Therefore, expression of \(x\) and \(q\) are as below;

\[
x = 2^n + \ell \quad \text{given that} \quad n = 4, \quad \ell = 7
\]

\[
q = 2^{n+m} + \lambda \quad \text{given that} \quad n = 4, m = 1, \quad \lambda = 3
\]

ii. Using the above solution from i., it can be shown that the formula \(\mathcal{L} = xq\) is the same as

\[
\mathcal{L} = 2^{2n+m} + 2^n(\lambda + \ell 2^m) + \ell \lambda \quad \text{And} \quad d = x + q \quad \text{the same as} \quad d = 2^n(1 + 2^m) + \ell + \lambda
\]

Solution:

\[
\mathcal{L} = xq; \quad \mathcal{L} = 23 \times 35 = 805
\]

\[
\mathcal{L} = 2^{2^4+1} + 2^4(3 + 7 \times 2^1) + (7 \times 3) = 805
\]

\[
d = 23 + 35 = 58
\]
Therefore \( f \) be

\[ p > \]

\[ \equiv \]

\[ q > \]

\[ \lambda \]

\[ L \equiv xq = 2^{2n+m} + 2^n(\lambda \ell 2^m) + \ell \lambda \]

2.3 CONCATENATION OF BITS COMBINATION.

The concept of concatenation is mainly used in formal language theory like in programming languages and pattern matching (including regular expressions). Concatenation of two strings \( a \) and \( b \) is often denoted as \( ab \), \( a\|b \), or, in the Wolfram Language, \( a < b \) \[16\]. 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.

\[ A\|b = \{ a\|b: a \in A \} \]

\[ a\|B = \{ a\|b : b \in B \} \]

However, as given by the work of Weisstein and Eric W \[16\], the concatenation of two or more numbers is the number formed by concatenating their numerals. They gave an example, the concatenation of 1, 234, and 5678, which is 12345678. And that the value of the result depends on the numeric base. They then gave the formula for the concatenation of numbers \( p \) and \( q \) in base \( \beta \) is

\[ p\|q = p\beta^{l(q)} + q \quad \text{Where} \quad l(q) = \left\lfloor \log_{\beta} q \right\rfloor + 1 \]

\( l(q) \) is the number length of “\( q \)” in base “\( \beta \)” and \([x]\) 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 \( p = 5.56, p = 34.03 \) etc., it yield different result when compared with a number treated as string.

In addition, numbers with zero in front like \( q = 0045, q = 034, q = 00018 \), 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 applied are as below.

2.3.1 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. \((a\|b)||(b)||(a)\)  
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:

\[ a\|b||a\|b \]

For example involving bits combination, Binary concatenation and port assignment, for one by one combination and concatenation see Table I, for two by two bits combination and concatenation see Table II and for four by four bits, combination and concatenation see Table III. In addition, since we are dealing with 8-bits binary system, “\( n \)” should be \{1, 2, 4\} or factors of 8 eg. \{8, 16, 24, 32, 40, 48\}. however the combination here and bits assignment are

2.3.2 ORDER OF CONCATENATION AFTER TRANSMISSION.

Given time series of transmission, as

\[ T = \{ t_0, t_1, t_2, \ldots, t_n \} \] such that \( t_n > t_{n-1} > t_{n-2} > t_{n-3} > \ldots > t_1 > t_0 \)

By defining sender to receiver order as \( O_{SR} \) and receiver to sender order as \( O_{RS} \)

Therefore from sender to receiver order. \( O_{SR} = (a\|b)_{t_0}||(a\|b)_{t_1}||(a\|b)_{t_2}|| \ldots ||(a\|b)_{t_n} \)

\[ d = 2^4(1 + 2^1) + 7 + 3 = 58 \]

Hence \( d \equiv q + x = 2^n(1 + 2^m) + \ell + \lambda \) and \( L \equiv xq = 2^{2n+m} + 2^n(\lambda \ell 2^m) + \ell \lambda \)
And from receiver to sender order \( O_{RS} = (b\|a)_{t_0}\|(b\|a)_{t_1}\|(b\|a)_{t_2}\| ... \|(b\|a)_{t_n} \) 

<table>
<thead>
<tr>
<th>BIT ASSIGN TO SENDER AND RECEIVER ADDRESSES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sender Addresses</strong></td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>( a_0 )</td>
</tr>
<tr>
<td>( a_1 )</td>
</tr>
</tbody>
</table>

Possible Crossing and bit combination sent per line.

\[ a_0||b_0 = 00; \quad a_0||b_1 = 01; \quad a_1||b_0 = 10; \quad a_1||b_1 = 11; \]

Base on the above combination, to send letter ‘H’=01101001. Can be such that each transmission carries two bits by reference from the above table, addresses relation can be as below to send the bits.

\[ O_{SR} = (a_0||b_1)_{t_0}\|(a_1||b_0)_{t_1}\|(a_1||b_0)_{t_2}\|(a_0||b_1)_{t_3} \Rightarrow 0\overline{1} | \overline{1}0 | \overline{1}0 | \overline{0}1 \Rightarrow 01101001 \]

For reply reverse condition holds true as below;

\[ O_{RS} = (b_0||a_1)_{t_0}\|(b_1||a_0)_{t_1}\|(b_1||a_0)_{t_2}\|(b_0||a_1)_{t_3} \Rightarrow \overline{0}1 | \overline{1}0 | \overline{1}0 | \overline{0}1 \Rightarrow 01101001 \]

Joining the binaries, its then 01101001. However, calculating maximum crossing is below:

\( L = 2^{2n}; \quad n = 1 \quad therefore \quad L = 2^2 = 4 \)

Total addresses: \( d = 2^{(n+1)}; \quad n = 1 \quad , \Rightarrow \quad d = 2^{(1+1)} \quad therefore \quad d = 4 \)

For more example, see Table II, where two bits combine to make maximum bits.

<table>
<thead>
<tr>
<th>BITS COMBINATION ASSIGN TO SENDER AND RECEIVER ADDRESSES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sender Addresses</strong></td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>( a_0 )</td>
</tr>
<tr>
<td>( a_1 )</td>
</tr>
<tr>
<td>( a_2 )</td>
</tr>
<tr>
<td>( a_3 )</td>
</tr>
</tbody>
</table>

Possible Crossing and bit combination sent per line.

\[ a_0||b_0 = 0000; \quad a_0||b_1 = 0001; \quad a_0||b_2 = 0010; \quad a_0||b_3 = 0011; \quad a_1||b_0 = 0100; \quad a_1||b_1 = 0101; \quad a_1||b_2 = 0110; \quad a_1||b_3 = 0111; \quad a_2||b_0 = 1000; \quad a_2||b_1 = 1001; \]
Base on the above combination, to send letter ‘H’=01101001 Separating into two four by four bits for each transmission carries four bits.

\[ Q_{SR} = (a_1||b_2)_{t_0}|| (a_2||b_1)_{t_1} \implies a_1||b_2 = 0110 \quad a_2||b_1 = 1001 = 01101001 \]

For reply from receiver back to sender,

\[ Q_{RS} = (b_1||a_2)_{t_0}|| (b_2||a_1)_{t_1} \implies b_1||a_2 = 0110 \quad b_2||a_1 = 1001 = 01101001 \]

Total port required for assigning 4 bits combination at once is as below.

\[ L = 2^n ; \quad n = 2 \quad therefore \quad L = 2^4 = 16 \]

Table III. Showing four by four bits combination Assignment

<table>
<thead>
<tr>
<th>BITS COMBINATION</th>
<th>ASSIGN TO SENDER AND RECEIVER ADDRESSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender Addresses</td>
<td>Sender’s Bits</td>
</tr>
<tr>
<td>a_0 0000</td>
<td>b_0 0000</td>
</tr>
<tr>
<td>a_1 0001</td>
<td>b_1 0001</td>
</tr>
<tr>
<td>a_2 0010</td>
<td>b_2 0010</td>
</tr>
<tr>
<td>a_3 0100</td>
<td>b_3 0100</td>
</tr>
<tr>
<td>a_4 1000</td>
<td>b_4 1000</td>
</tr>
<tr>
<td>a_5 0011</td>
<td>b_5 0011</td>
</tr>
<tr>
<td>a_6 0110</td>
<td>b_6 0110</td>
</tr>
<tr>
<td>a_7 1100</td>
<td>b_7 1100</td>
</tr>
<tr>
<td>a_8 1001</td>
<td>b_8 1001</td>
</tr>
<tr>
<td>a_9 0101</td>
<td>b_9 0101</td>
</tr>
<tr>
<td>a_{10} 1010</td>
<td>b_{10} 1010</td>
</tr>
<tr>
<td>a_{11} 1011</td>
<td>b_{11} 1011</td>
</tr>
<tr>
<td>a_{12} 1110</td>
<td>b_{12} 1110</td>
</tr>
<tr>
<td>a_{13} 1101</td>
<td>b_{13} 1101</td>
</tr>
<tr>
<td>a_{14} 1011</td>
<td>b_{14} 1011</td>
</tr>
<tr>
<td>a_{15} 1111</td>
<td>b_{15} 1111</td>
</tr>
</tbody>
</table>

Since the total bit for sender is sixteen, and for receiver is sixteen, therefore, total maximum crossing is as below:

\[ L = 2^{2n} : \quad n = 4 \quad therefore \quad L = 2^8 = 256 \]

Base on the above combination, 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 (1 byte)

\[ Q_{SR} = (a_6||b_8)_{t_0} \implies a_6||b_8 = 01101001 \]

For reply,

\[ Q_{RS} = (b_6||a_8)_{t_0} \implies b_6||a_8 = 01101001 \]
Total addresses are as below.

\[ d = 2^{(n+1)} : n = 4 \implies d = 2^{(4+1)} \implies d = 2^5 = 32 \]

As shown above in Table (I, II, III), the number of possibilities in a binaries combination is given as \( C = 2^n \) and therefore number of crossing in transmission is given as \( L = 2^{2n} \) base on condition 2 when \( q = x = C \).

2.4 BIT ROTATION OVER ADDRESSES: Let \( \Psi \) represent sets bit combination over addresses \( A, B \). for subscribe of \( \Psi \) "I" is the current position of bit combination over a given addresses \( A \) or \( B \) of index \( I \), “J” represents total transmission time such that \( (J \in \mathbb{N}, J \in \mathbb{Z}) \). In addition, \( Q \) is the maximum addresses \( (Q \in \mathbb{N}) \) or in other word, totals elements of set \( A \) or \( B \). example if \( \psi_2 = a_2 = b_2 \)

However, for Circum-rotation, \( Q \) is the total circumference of a circle \( Q = 2\pi r \); such that radius of a circle can be express as

\[ r = \frac{Q}{2\pi} \]

In addition, substituting when \( Q \) is exactly equal to \( C \) such that \( Q = C = 2^n \) radius becomes \( 2^n = \frac{2\pi r}{\pi} \)

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.

\[
\begin{align*}
A &= \{a_{Q-1}, a_{Q-2} \ldots a_2, a_1, a_0\} & \text{OR} & & A &= \{a_0, a_1, a_2 \ldots, a_{Q-2}, a_{Q-1}\} \\
B &= \{b_{Q-1}, b_{Q-2} \ldots b_2, b_1, b_0\} & \text{OR} & & B &= \{b_0, b_1, b_2 \ldots b_{Q-2}, b_{Q-1}\} \\
\Psi &= \{\psi_{Q-1}, \psi_{Q-2} \ldots \psi_2, \psi_1, \psi_0\} & \text{OR} & & \Psi &= \{\psi_0, \psi_1, \psi_2 \ldots \psi_{Q-2}, \psi_{Q-1}\}
\end{align*}
\]

Below is the general equation, and conditions for the rotation in the positive direction or clockwise direction and Negative or anti-clockwise direction.

Let modular (mod) or remainder operator for dividing a number by another be \( \text{ (mod } = \ominus \) ) to avoid confusing mod with other letter symbols presented here in the paper.

For instance \( A \mod B = R \) can be express as \( A \odot B = R \) or \( 5 \mod 4 = 1 \) can be written as \( 5 \ominus 4 = 1 \)

2.4.1 POSITIVE (CLOCKWISE) ROTATION

\[
f(J, \Psi, Q, I) = \begin{cases} 
\psi_{I+j}, & 0 \leq J < Q : (I+j) < Q \\
\psi_{\{I+(i+J) \odot Q\}}, & 0 \leq J < Q : (I+j) \geq Q \\
\psi_{I+(J+Q)\odot Q}, & J \geq Q : (I+(J \odot Q)) < Q \\
\psi_{\{I+(J+Q)\odot Q\}}, & J \geq Q : (I+(J \odot Q)) \geq Q 
\end{cases}
\]

2.4.2 NEGATIVE (ANTI-CLOCKWISE) ROTATION

\[
f(J, \Psi, Q, I) = \begin{cases} 
\psi_{I+j}, & -Q < J \leq 0 : (J+I) \geq 0 \\
\psi_{\{I+(J+Q)+Q\}}, & -Q < J \leq 0 : (J+I) < 0 \\
\psi_{I+(J+Q)\ominus Q}, & J \leq -Q : (I+(J \ominus Q)) \geq 0 \\
\psi_{\{I+(J+Q)\ominus Q\}}, & J \leq -Q : (I+(J \ominus Q)) < 0 
\end{cases}
\]

Prove of the Equation above

The proved is by symmetrical which means if a given rotation ends with maximum number such as \( Q \) therefore \( Q+1 \) is equals to initial position which is one; and given as \( ((Q + 1) \odot Q) = 1 \) implying the rotation repeats itself time and again. However for rotation that starts from zero as initial position, \( (Q \odot Q) = 0 \) and its maximum rotation is \( (Q - 1) \) not \( Q \) which the entire proof is base on.
Prove of Clockwise Rotation by symmetry.

Substituting Q in the equation with $2^n$ given that $Q=q=x=C$ so:

$$f(J, \Psi, n, I) = \begin{cases} 
\Psi_{J+I}, & 0 \leq J < 2^n: (l + J) < 2^n \\
\Psi_{((l+J) \oplus 2^n)} ,& 0 \leq J < 2^n: (l + J) \geq 2^n \\
\Psi_{(J (\oplus 2^n) +I)} ,& J \geq 2^n: ((I \oplus 2^n) + I) < 2^n \\
\Psi_{(I+(J \oplus 2^n)) \oplus 2^n} ,& J \geq 2^n: (I + (J \oplus 2^n)) \geq 2^n 
\end{cases}$$

After every Q rotation, index should be equal to zero where $J = Q = 2^n$. In addition, substituting into above equation gives the below new equation.

$$f(J, \Psi, n, I) = \begin{cases} 
\Psi_{2^n+I}, & 0 \leq J < 2^n: (l + 2^n) < 2^n \\
\Psi_{((l+2^n) \oplus 2^n)} ,& 0 \leq J < 2^n: (l + 2^n) \geq 2^n \\
\Psi_{(2^n \oplus 2^n) + I} ,& J \geq 2^n: ((2^n \oplus 2^n) + I) < 2^n \\
\Psi_{(I+(2^n \oplus 2^n)) \oplus 2^n} ,& J \geq 2^n: (2^n + (2^n \oplus 2^n)) \geq 2^n 
\end{cases}$$

Now making initial position of a given bits any number from zero to a number less than Q or $0 \leq I < 2^n$ the resultant rotation should be exactly the same as that initial position.

Let $I = 0$ and substituting in the above.

$$f(J, \Psi, n, I) = \begin{cases} 
\Psi_{J+0=0 \oplus 2^n=0} ,& 0 \leq J < 2^n: (l + 2^n) < 2^n \text{ Proved when } 2^n = 0 \\
\Psi_{((0+2^n) \oplus 2^n)=0} ,& 0 \leq J < 2^n: (l + 2^n) \geq 2^n \text{ Proved} \\
\Psi_{(2^n \oplus 2^n)+0=0} ,& J \geq 2^n: ((2^n \oplus 2^n) + I) < 2^n \text{ Proved} \\
\Psi_{(I+(2^n \oplus 2^n)) \oplus 2^n=0} ,& J \geq 2^n: (2^n + (2^n \oplus 2^n)) \geq 2^n \text{ proved} 
\end{cases}$$

End of prove

Algorithm 1: written base on Clockwise Rotation and Data proves rotation pattern as it increases see Fig 1

Initialization of variables

```
Q; p; I; J; \Psi_t;
for J from 1 to p loop
  if J>=0 & J<Q then
    if (I+J)>=Q then
      t_j ← ((I+J) \oplus Q)
    else
      t_j ← (I+J)
    endif
  else
    if((J\oplus Q)+I)>=Q then
      t_j ← (((J\oplus Q)+I) \oplus Q)
    else
      t_j ← (J\oplus Q) + I
    endif
  endif
endloop
```
Initialization of variables

**Algorithm 2**: Written base on Anticlockwise Rotation and Data proves rotation pattern as it decreases see Fig 2

Initial position of any number from zero to a number greater than \(-Q\). The resultant rotation should be the same as that initial position.

Let \( I = 0 \) and substituting into above equation.

\[
\begin{cases}
\Psi_{I=0-J=\pm 2^n} = 0, & -2^n < J \leq 0: (2^n + J) \geq 0, Proved \\
\Psi_{I=0-J=\pm 2^n} = 0, & -2^n < J \leq 0 : (2^n + J) < 0, Proved \\
\Psi_{I=0+(-2^n-J=\pm 2^n)} = 0, & J \leq -2^n : (I + (-2^n \otimes 2^n)) \geq 0, Proved \\
\Psi_{I=0+(-2^n-J=\pm 2^n)} = 0, & J \leq -2^n : (I + (-2^n \otimes 2^n)) < 0, Proved
\end{cases}
\]

End of prove.
Q;J;I;P;Ψ; 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.

<table>
<thead>
<tr>
<th>Data of Anticlockwise Rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4  3  2  1  0  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0  15  14  13  12  11  10  9  8  7</td>
</tr>
</tbody>
</table>

Fig 2: Shows Data for anticlockwise Rotation and initial Position is five

3. OPTIMAL TIME 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.

Let "t" be time required for sending total of “λ” bit combination at once. And “k” is a constant unit time per one bit transmission such that

\[ \lambda t \propto k \quad \Rightarrow \quad \lambda t = k \]

\[ t = \frac{k}{\lambda} \quad \text{but} \quad \lambda = n + n \]

\[ t = \frac{k}{2n} \]

Prove by Analysis

If: 1 bit takes “k” time for transmission.
    2 bits takes \( \frac{1}{2} \) of “k” time of transmission.
    3 bits takes \( \frac{1}{3} \) of “k” times of transmission.

What about “\( \lambda \)” given bits combination sent at once
\( \lambda \) Bits takes \( \frac{1}{k} k \) times of transmission, but from Table (I, II, III), one transmission carries \( 2n \) such that at
the sender port contains \( n \) bit while at the receiver port represent another \( n \) bits so a total of \( n + n = 2n \) bits per
transmission, therefore \( \lambda = n + n = 2n \) bits per transmission.

End of Prove.

If \( t = \frac{k}{2n} \). Therefore \( \lim_{n \to \infty} (t = \frac{k}{2n}) \approx 0 \); given that \( k = 1 \)

However, for cases where \( x \neq q \) at sender side carries \( n \) combination of bit while at receiver side carries \( n + m \) combination of bits.

\( \lambda = n + (n + m) \); \( t = \frac{k}{2n+m} \)

From the above, it is evidence that when number of bits combination increases so does the time required for
transmission reduces. See Fig 4 for graph showing reduction in percentage of transmission time.

![Graph of percentage decrease in time against Bit Combination](image)

**Fig 4: Graph of percentage decrease in time against Bit Combination**

4. **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.

Let sample space be \( \tau \) for a total addresses at either sender or receiver’s side where a bit combination can occupy at
a given time. Therefore, the probability that a chosen bit occupy such places is as below:

\[
\text{probability}(P) = \frac{1}{\tau} \mid \tau = q, \; \tau = x, \; \tau = 2^n \quad \Rightarrow \quad P = \frac{1}{2^n}
\]

For the probability that it is not is given as \( P' = \left(1 - \frac{1}{2^n}\right)\).
\[
\lim_{n \to \infty} \left(1 - \frac{1}{2^n}\right) \approx 1
\]

However, the probability that a chosen transmission 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

i. \( L = 2^{2n} \) when \((q, x) = C\) and \((x = q)\)

\[
\text{probability}(P) = \frac{1}{L}; \quad P = \frac{1}{2^{2n}}
\]

Probability that it is not, is given as \( P' = (1 - P) \)

\[
\lim_{n \to \infty} \left(1 - \frac{1}{2^n}\right) \approx 1
\]

ii. \( L = 2^{2n+m} \) when \((q, x) = C\) and \((x \neq q)\)

\[
\text{probability}(P) = \frac{1}{L}; \quad P = \frac{1}{2^{2n+m}}
\]

Probability that it is not, is given as \( P' = (1 - P) \)

iii. \( L = 2^{2n} + 2^n(\lambda + \ell) + \ell\lambda \). when \((q, x) = C\) and \((x = q)\)

\[
\text{probability}(P) = \frac{1}{L}; \quad P = \frac{1}{2^{2n} + 2^n(\lambda + \ell) + \ell\lambda}
\]

Probability that it is not, is given as \( P' = (1 - P) \)

iv. \( L = 2^{2n+m} + 2^n(\lambda + \ell 2^m) + \ell\lambda \) \( \) when \((q, x) = C\) and \((x \neq q)\)

\[
\text{probability}(P) = \frac{1}{L}; \quad P = \frac{1}{2^{2n+m} + 2^n(\lambda + \ell 2^m) + \ell\lambda}
\]

Probability that it is not, is given as \( P' = (1 - P) \)

**Example 2**: Given a bits combination \( n=3\) after \( J=11 \) times of transmission in clockwise rotation. In addition, an initial position of the bits \( l=3\).

a) What is the new position of the bits combination \( \Psi_3\)?

Solution:

Therefore: \( Q = 2^3 \); \( n = 3 \) and \( Q < J \)

\[
f(J, \Psi, n, l) = \Psi_{(J \cap 2^n) + l}; \quad J \geq 2^n
\]

\[
f(J, \Psi, n, l) = \Psi_{(11 \cap 2^3) + 3}
\]

\[
f(J, \Psi, n, l) = \Psi_6
\]

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 \( l=6 \) is a new position after clockwise rotation.

And since anti-clockwise, \( J=-19 \)
\[ f(J, \Psi, n, I) = \Psi_{(J \oplus 2^n) + \iota}, \quad J \leq -2^n \]
\[ f(J, \Psi, n, I) = \Psi_{(-19 \oplus -2^3) + 6} \]
\[ f(J, \Psi, n, I) = \Psi_{-3 + 6 = -3} \quad \text{So new position= 3} \]

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 wants 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 required from sender and receiver country so that at least each mail sent carries a character of 8-bits?

Solution:

This can be separated into two 4 by 4 bits where \(n = 4\) see Table III for reference, such that at sender carries 4-bits and receiver’s carries 4-bits.

Given total address is \(d = q + x\) and substituting it with \(q = x = 2^n\)

\[ d = 2^n + 2^n \Rightarrow 2 \times 2^4 = 16 + 16 = 32 \text{ addresses}. \]

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

b) From Table III As a reference 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 III, initial position of \(0110 = a_6 = 6\) and for \(1111 = b_{15} = 15\)

Sender side Rotation Clockwise.

\(I=6, \quad Q=C=16, \quad J=50\) times.

\[ f(J, \Psi, n, I) = \Psi_{(J \oplus Q) + \iota}, \quad J \geq Q ; (J \oplus Q) < Q \]

So, \(f(J, \Psi, n, I) = \Psi_{(50 \oplus 16) + \gamma} = \Psi_9\ new\ position\ = 8\)

Receiver side Rotation Anti-Clockwise.

\(I=15, \quad Q=-C=-16, \quad J=-50\) times.

\[ f(J, \Psi, n, I) = \Psi_{I + (J \oplus Q)}, \quad J \leq -Q ; (J \oplus Q) < 0 \]

So, \(f(J, \Psi, n, I) = \Psi_{15 - (50 \oplus -16)} = \Psi_{15 - 2}\ new\ position\ = 13\)

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:

\[ t = \frac{k}{2^n}; \ n = 4, k = 1 \]

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)

\[ C = q = 2^n; \ n = 4 \Rightarrow q = 2^4 = 16 \]

Probability \( \frac{1}{q}; \ q = 2^n \Rightarrow \frac{1}{2^n} = \frac{1}{16} = 0.0625. \)

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 \( n = 1 \times 8 = 8 \) bits combination

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

5. CONCLUSION

Base on proves, examples and analysis, it is practically impossible to detect hidden information flow. Decrypting a hidden message is extremely hard as nothing modifies like in other steganography methods, which involve modification of the carrier. Besides that, due to the rotation of bits combination over given addresses after every transmission makes it difficult to decipher the hidden contents, and lastly by using combination technique enable transmitting many bits at once unlike in other method where a single bits is transmitted one after another. However transmission should be sequential not simultaneous to avoid confusion.

6. ACKNOWLEDGEMENT

Special thanks to my Mother Mrs Adong Josephine and my father Mr Oyaro Silvio and my Siblings for their continual supports.

7. REFERENCE


Authors Biography

Author’s Name: Moses Okello
E-mail: mosesokellomoses@gmail.com or okellomosesokello@gmail.com
BSc, MSc
DOB: 19900228

NB. Author out of his own interest in the Subject wrote this work. It is not supported nor funded or endorse by any one, the author remains responsible for any material presented here in this paper. Any complaint related to this paper are highly welcome and can be addresses directly to the author.