

Attacking Attackers in Relevant to Information Security

Prasun Chakrabarti, Dr. B. C. Roy

Engineering College , Durgapur – 713206 , West Bengal , India

Abstract-The paper points out a brief idea of some techniques for secured data transmission. Message extraction based on comparison analysis has been illustrated. The net value is determined by accumulated sum of the respective product of offset value and weight corresponding to each character in the original message. The examples taken into consideration are mainly based on cipher production in the light of fuzzy rule and theory of computation , Merkle's puzzle of key agreement and security analysis using logarithmic rule. Hence attacking attackers by security enhancement can be done efficiently.

Keywords:-Comparison analysis, automatic variability , fuzzy rule, theory of computation , Merkle's puzzle of key agreement

I. THEORY OF COMPARISON ANALYSIS

Let original message is "MOTHER"

For the first alphabet, $\mu_{\text{value}} = 1/((\text{position of that}) + \pi/100)$
 Hence its offset value = ceiling of (the product of μ_{value} and 10)

The weight is given by its position in alphabet string
 Therefore total_value = offset value * weight

From the next character onwards,

$\mu_{\text{value_next}} = 1/(\text{mod value of (position of next-position of previous)} + \pi/100)$

Hence total_value is calculated in similar manner.

Now, bias value will be equal to total number of characters in the message.

Compute net_value as (total_value_ first char + total_value_ last char) - (bias value) and let it be x (say).

Mode	Operation
$0 \leq x < 100$	Reverse the message
$100 \leq x < 150$	Circular left shift of message by n/2 bits where n= bias value
$150 \leq x < 200$	Circular right shift of message by n/2 bits

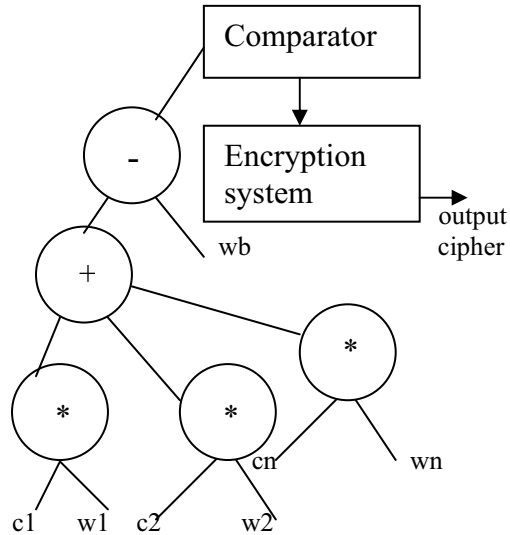
c_i = offset value, for $i = 1$ to n , w_i = weight , w_b = bias value

II. NUMERICAL ANALYSIS

Iteration 1: $\mu_M = 1/(\text{position of M in alphabet list} + \pi/100) = 0.077$. Offset value = ceiling of $(0.077 * 10) = 1$. Weight = position of M in alphabet list = 13. Thus , total_value = $1 * 13 = 13$.

Iteration 2: $\mu_O = 1/((\text{position of O} - \text{position of M}) + \pi/100) = 1/2.031 = 0.492$. Offset value = ceiling of $(0.492 * 10) = 5$. Weight = 15. Thus, total_value = $5 * 15 = 75$

Iteration 3: $\mu_T = 1/((\text{position of T} - \text{position of O}) + \pi/100) = 0.199$. Offset value = 2 Weight = 20. Thus total_value = $2 * 20 = 40$



Iteration 4: $\mu_H = 1/((\text{position of H} - \text{position of T}) + \pi/100) = 0.083$. Offset value = 1 Weight = 8. Thus total_value = $1 * 8 = 8$

Iteration 5: $\mu_E = 1/((\text{position of E} - \text{position of H}) + \pi/100) = 0.33$. Offset value = 4 Weight = 5. Thus total_value = $4 * 5 = 20$

Iteration 6: $\mu_R = 1/((\text{position of R} - \text{position of E}) + \pi/100) = 0.077$. Offset value = 1 Weight = 18. Thus total_value = $1 * 18 = 18$

Now, w_b = bias value = number of bits in MOTHER = 6
 So net_value = accumulated sum of all total_value - $w_b = (13 + 75 + 40 + 8 + 20 + 18) - 6 = 168$. It falls in the range $150 < x < 200$. So, "MOTHER" is circular right shifted by $6/2 = 3$ bit

Therefore resultant cipher is "HERMOT".

III. ENCRYPTION BASED ON PROPOSITIONAL LOGIC

Let message = $m_1 = 110111$. Key = 111010. If m_1 XOR $k = 111010$ is the encrypted result based on linear property, then if hacker knows k , then by XOR operation m_1 will be revealed. To solve this, an intermediate result has to be

found out and it will be XOR-ed with m_1 to get resultant cipher. The necessary bit-padding is also applied. Let the intermediate result be based on the truth value of $(\sim m_1 U_k) \cap (m_1 \leftrightarrow k)$

m_1	k	$\sim m_1$	$(\sim m_1 U_k)$	$(m_1 \leftrightarrow k)$	$(\sim m_1 U_k) \cap (m_1 \leftrightarrow k)$
0	0	1	1	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	1	0	1	1	1

Therefore intermediate result is = 001001
 Resultant cipher = m_1 XOR intermediate result = 111110

IV. ENCRYPTION SCHEME BASED ON FUZZY OPERATORS

Let message = 010100111001000. Let 000→0.1, 001→0.2, 010→0.3, 011→0.4, 100→0.5, 101→0.6, 110→0.7, 111→0.8. Therefore if we can denote message as a fuzzy set and replace each of 3 bits by its corresponding value, then $\tilde{M} = \{(x_1, 0.3), (x_2, 0.5), (x_3, 0.8), (x_4, 0.2), (x_5, 0.1)\}$.

Let the encrypted key will be generated based on each value of x and it will be governed by $\{[(0.3)^{1/2} + ((0.5)^2 + (0.8)^2)] + (0.2/0.6 + 0.6/0.6)\}$ in binary form. It should be noted that dilation = $\sqrt{\mu_A(x)}$, concentration = $[\mu_A(x)]^2$, normalization = $\mu_A(x)/\max_x \mu_A(x)$.

V. ENCRYPTION SCHEME BASED ON OPTIMIZATION TECHNIQUE

Let the message = $m_1 = 110101$. The values of the possible keys are $k_1 = 100001$, $k_2 = 110111$, $k_3 = 10010$, $k_4 = 111100$. For each key its corresponding optimized value is obtained and the one, whose value is largest, is the ultimate encrypted key. $m_1 = 110101$, $k_1 = 100001$, $m_1 \text{ XOR } k_1 = k_1'$
 Objective_function1 = $(m_i - k_i')^2 + (m_i - k_{i-1}')^2 + \dots$
 $= 1+0+0+0+0+1 = 2$.

Similarly, objective_function2 = $1+1+0+1+1+1 = 5$;
 objective_function3 = $1+0+0+0+1+0 = 2$;
 objective_function4 = $1+1+1+1+0+0 = 4$. So, objective_function2 is the largest. Hence cipher = m_1 XOR (k_2 in reverse form) = 110101 XOR $111011 = 001110$.

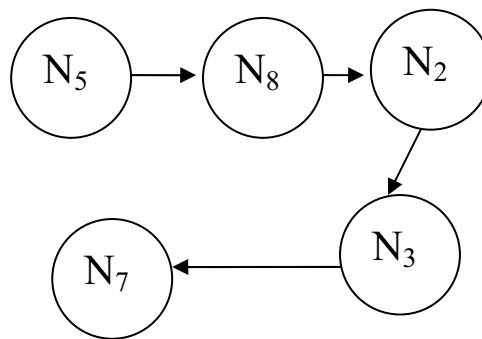
VI. CIPHER GENERATION BASED ON THEORY OF COMPUTATION

Let the message be = 100111001010110 breaking the message into frames of 3 bits length and assigning values we get,

message = 100 111 001 010 110 = $\{N_5, N_8, N_2, N_3, N_7\} = \{0.5, 0.8, 0.2, 0.3, 0.7\}$

Code	Value	Node _i	Octal value
000	0.1	N ₁	0
001	0.2	N ₂	1
010	0.3	N ₃	2
011	0.4	N ₄	3
100	0.5	N ₅	4
101	0.6	N ₆	5
110	0.7	N ₇	6
111	0.8	N ₈	7

In directed form, we can represent it by



$V = \{N_5, N_8, N_2, N_3, N_7\}$; $E = \{(N_5, N_8), (N_8, N_2), (N_2, N_3), (N_3, N_7)\}$

Sender can format cipher formation by the rule that $x = [(\{\text{Max}(N_5, N_8) \text{ in octal form} * \text{value}\} + \{\text{Min}(N_8, N_2) \text{ in octal form} * \text{value}\} + \{\text{Max}(N_2, N_3) \text{ in octal form} * \text{value}\} + \{\text{Min}(N_3, N_7) \text{ in octal form} * \text{value}\})]$ and it is transformed into binary form and then result is XORed with original message in reverse order to generate cipher. $X = 7*0.8 + 1*0.2 + 2*0.3 + 2*0.3 = 7 = (111)_2$ Cipher = 011010100111001 XOR $000000000000111 = 01101010011110 = 1101010011110$ (by 0-compression).

VII. CIPHER GENERATION APPLYING LOG RULE

Method 1

$y = \log_d k + \log_k d$. Hacker hacks y , k , known = $\log_d(\text{known}) + \log_{(\text{known})} d$

$= \log_d(\text{known}) + 1/\log_d(\text{known})$

Let $\log_d(\text{known}) = x$ (1)

Therefore known = $x + 1/x$ solve x put x in (1) and get d

Method 2:

Session 1: $d_1, k_1, y_1 = \log_{d_1} k_1 + \log_{k_1} d_1$

Session 2: $d_2, k_2, y_2 = \log_{d_2} k_2 + \log_{k_2} d_2$ and $k_2 = k_1 + d_2$. If the hacker hacks k_1 ,

y_2 known = $\log_{d_2} (k_1 + d_2) + \log_{(k_1 + d_2)} d_2$. Similarly we get d_2

Solution:

Applying automatic variable data $d_2 = d_2 \text{ XOR } k_1$ and this scheme is known to hacker. Another level of security can be performed by any one of the following

- (iii) $d_2 = (\text{left/right circular shift of } d_2 \text{ by } n \text{ bits}) \text{ XOR } k_1$

- (i) $d_2 = (\text{reverse } d_2) \text{ XOR } k_1$
- (ii) $d_2 = (\text{reverse } d_2) \text{ XOR } (\text{reverse } k_1)$

VIII. MERKLES' PUZZLE OF KEY AGREEMENT

A. Concept

A: Possesses n puzzles and its n solutions. It sends puzzles to B.

B: B solves any one puzzle at random and the solution is the key. It then encrypts its message with that key and sends it to A.

A: A decrypts by using all n keys and find message

Best Case: $O(1)$

Worst Case: $O(n)$

B. Illustration

Both sender and receiver agree the fuzzy values for the solution. If $n = 10$, then $f(1) = 0.1$

Suppose receiver has solved puzzle no. 6 So it will create a formula that net shift = (fuzzy value * 10) and shift criteria is that if puzzle number is odd, then left shift will occur else right. Sign is (+ve). From next character onwards, alternate sign:

if message is P R A S U N
 +6 -6 +6 -6 +6 -6
 6 V L G M A H

where the first bit (MSB) will denote puzzle number. For further security, a modified scheme can be applied where offset values of next character will be dependent on that of previous and formula is

$\text{off}_{\text{next}} = (\text{index position of next} + \text{previous off value}) \text{ mod } 26$

If mod operation is needed, sign bit is +ve else -ve.

CONCLUSION

The papers shows how efficiently a neuro-fuzzy approach can be used for information processing. Also it has been shown how encryption schemes can be generated using propositional logic , fuzzy operators , optimization technique , theory of computation , log rule. A key-agreement protocol has also been illustrated based on Merkle's puzzle of key agreement.

REFERENCES

S. Rajasekaran, G.A – V.Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithm”, PHI Pvt. Ltd., 2003.

Gerardo R Ungson et al, “The Emerging Knowledge Based Economy”, IEEE Spectrum, May’ 1999, pp 60-65.

Jang , Sun , Mizutoni , “Neuro Fuzzy and soft computing”, Pearson Education, 2005

T.Vishwanathan, “Telecommunication Switching Systems and Networks”, PHI Pvt Ltd, 2001

W.Stallings, “Data and Computer Communications”, PHI Pvt Ltd. 2001