Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 15 Issue 1 Version 1.0 Year 2015
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Dynamic Permutations

By Dr. Saleh N. Abdullah & Dr. Sharaf A. Alhomdy

Faculty of Computer and Information Technology, Sanaa University, Yemen

Abstract- The confidentiality, integrity and authentication of anelectronic documentare necessary in many application systems. The security of confidentiality, integrity and authentication of an electronic document are based on nonlinear functions, in which there is no directrelationship between the inputs and the outputs. This means that the inputs cannot be extracted from the outputs.

Indeed, all modern cyphers are based on the concept of substitution transposition. In data encryption standard algorithm, DES, which consists of many functions, only one nonlinear function is used in the algorithm, called substitution boxes, and all other functions are linear, one of these linear functions is called IP, initial permutation function, which performs static permutations. The permutations are replaced by transpositions, based on predefined positions, and the permutation function is used several times in DES algorithm.

Keywords : *confusion, diffusion, linear function, nonlinear function, static permutations, dynamic permutations, one-way functions, hash table and complexity.*

GJCST-C Classification : *G.2.1*

Strictly as per the compliance and regulations of:

Dynamic Permutations

Dr. Saleh N. Abdullah [α] & Dr. Sharaf A. Alhomdy [σ]

Abstract- The confidentiality, integrity and authentication of anelectronic documentare necessary in many application systems. The security of confidentiality, integrity and authentication of an electronic document are based on nonlinear functions, in which there is no directrelationship between the inputs and the outputs. This means that the inputs cannot be extracted from the outputs.

Indeed, all modern cyphers are based on the concept of substitution transposition. In data encryption standard algorithm, DES, which consists of many functions, only one nonlinear function is used in the algorithm, called substitution boxes, and all other functions are linear, one of these linear functions is called IP, initial permutation function, which performs static permutations. The permutations are replaced by transpositions, based on predefined positions, and the permutation function is used several times in DES algorithm.

The permutation is an essential factor in many security systems or cryptosystems. That is because of the fact that every language has its own structure; the language structures disappearvia the permutation factors.

In this paper, we will try to develop dynamic permutations instead of static permutations, nonlinear factors, which in turn enhance the security system.

Keywords: confusion, diffusion, linear function, nonlinear function, static permutations, dynamic permutations, one-way functions, hash table and complexity.

I. Introduction

In any cryptosystem or message integrity and authentication, the nonlinear functions are the cornerstones because the inputs to the nonlinear functions cannot be extracted from the outputs. In linear function it is possible to obtain the output if both the inputs & the operation are known; also the second input can be obtained if one input & output are known (e.g. XOR function).

A function is called nonlinear if one solution can be retched from several inputs; in other words, if the operations and the outputs of a function are known, and the inputs to a function are not known, the function is callednonlinear. Moreover, if such outputs are produced via nonlinear functions, it becomes difficult to obtain the inputs to the nonlinear functions in a suitable time. For example, the operation *mod* acts as nonlinear function, because 20 *mod* 6 =2, also 20 *mod* 9=2, and 20 *mod* 3= 2. The value 2 comes from 20 *mod* 6, 20 *mod* 9, and 20*mod* 3. So, if we know one of the inputs and the output along with the operation *'mod',* we cannot know the second input.

In this paper, section two provides details about literature review. Section three describes our proposal technique to enhance the security in the confidentiality, integrity and authentication. The conclusion and future works will be found in section four.

II. Literature Review

In any cryptography systems, permutation (transposition) is an essential element to remove the relations between the alphabets which formulate the sentences because every language has its own characteristics.

Permutation: refers to mapping a block of length L1 into a block of length L1 [1].

Definition: Permutation denotes Π_p.

Π_p: {1,..,L_m}→{1,…,L_{m1}}is a permutation,

where L and m are positive integers.

Shannon [2, 3] suggests two methods for frustrating statistical cryptanalysis: Diffusion and Confusion. In diffusion, the statistical structure of the plaintext is dissipated into a long range statistics of the cipher text. On the other hand, confusion seeks to make the relationship between the statistics of the cipher text and the value of encryption key as complex as possible. Confusion can be achieved by the use of a complex substitution algorithm via using substitution boxes [1].For example, if we have the following inputs:
10101101 01001110 10000100 10101111.

The corresponding values in hexadecimal system are AC4E84AF.So every value will take a predefined position as shown in table 1.

Table1 : Shows the Values and Indexes

1	2	3	4	5	6	7	8	← Index input
A	C	4	E	8	4	A	F	
4	8	C	A	E	F	4	A	
3	5	2	7	4	8	6	1	← Index output

The first 4-bit input will be transferred into position 8 of output, and so on.

In DES algorithm [3, 4] the function is called IP initial permutation acts. This function performs static

Author α: Assistant Prof., Khawlan College, Sana'a University, Yemen. e-mail: saleh.alasali97@yahoo.com
Author σ: Assistant Prof.& Vice-Dean, Faculty of Computer and Information Technology, Sana'a University, Yemen. e-mail: sharafalhomdy@gmail.com

permutations; the permutations are replaced by transpositions, based on predefined positions as showed in Table (2) and Table (3).

Table 2 : Inputs to Function IP

V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}
V_{17}	V_{18}	V_{19}	V_{20}	V_{21}	V_{22}	V_{23}	V_{24}
V_{25}	V_{26}	V_{27}	V_{28}	V_{29}	V_{30}	V_{31}	V_{32}
V_{33}	V_{34}	V_{35}	V_{36}	V_{37}	V_{38}	V_{39}	V_{40}
V_{41}	V_{42}	V_{43}	V_{44}	V_{45}	V_{46}	V_{47}	V_{48}
V_{49}	V_{50}	V_{51}	V_{52}	V_{53}	V_{54}	V_{55}	V_{56}
V_{57}	V_{58}	V_{59}	V_{60}	V_{61}	V_{62}	V_{63}	V_{64}

Table 3 : Output to Function IP

V_{58}	V_{50}	V_{42}	v_{34}	V_{26}	v_{18}	v_{10}	v_2
v_{60}	V_{52}	V_{44}	v_{36}	V_{28}	v_{20}	v_{12}	v_4
V_{62}	V_{54}	V_{46}	v_{38}	V_{30}	v_{22}	v_{14}	v_6
v_{64}	V_{56}	V_{48}	v_{40}	V_{32}	v_{24}	v_{16}	v_8
V_{57}	V_{49}	V_{41}	v_{33}	V_{25}	v_{17}	v_9	v_1
v_{59}	V_{51}	V_{43}	v_{35}	V_{27}	v_{19}	v_{11}	v_3
V_{61}	V_{53}	V_{45}	v_{37}	V_{29}	v_{21}	v_{13}	V_5
v_{63}	V_{55}	V_{47}	v_{39}	V_{31}	v_{23}	v_{15}	v_7

III. Dynamic Permutations

So far all the processes of any permutations are static, i.e, the permutations are replaced by transpositions, based on predefined positions. However, in this paper we will suggesta new method "dynamic permutations" to enhance the security in cryptosystems. The main idea for the new method is as follows:

- Constructing a suitable hash table along with suitable hash key.
- Dividing the binary data into groups, each group consists of 8-bits; and each 8-bitscan take values from 00 to FF in hexadecimal system.
- Each group should be hashed into the corresponding value; this value is used as an index to store the group in the hash table. Since the values stored in the hash table are based on randomindexes, each group will take dynamic position.

In this case, the permutations of the inputs are dynamic permutations but not static. Figure (1) shows the suggested method for the construction of the hash table.

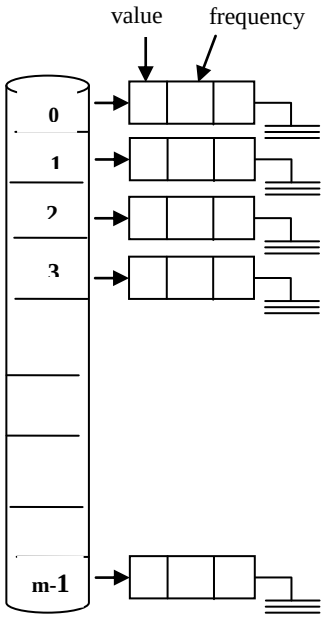

Figure1 : Shows the Construction of the Hash Table

Example: if we have the following inputs 10101101 01001110 10000100 10101111.The corresponding values in hexadecimal system are AC, 4E, 84, AF. So, every value will take a position in the hash table. If there is more than one value equals, the first one will take the correct position in the hash table and the others will increase the frequency field by 1, and so on, without taking extra positions in the hash table. If

there are more than one values hashed to the same index, the second value stays in another node with the same index in the hash table, and so on.

The length of the hash table is directly proportional to the S. That means, $L \alpha S$ (1)

such that S is the number of characters in the block simultaneously permuted and L is the length of the hash table.

The following equation:

$$pi=(pi-1+xi)\%m \quad (2)$$

Maybe used to produce the hash key, such that p0=7, pi is the index position in the hash table, x0= 11, xi is the value to be hashed, and m is prime number points to the size of the hash table.

The following is a sample of values hashed to the some indexes.

indexvalue index value index value index value
94 199 123 206 173 8 32 225
181 152 186 4 93 164 12 88
140 254 194 90 67 60 7 169
28 125 175 231 17 174 168 136
150 89 56 112 70 110 95 166
23 140 105 147 149 160 70 124
92 7 8 56 94 113 30 87
182 46 181 52 72 183 108 164
53 142 195 64 42 201 151 126
162 225 0 26 132 160 129 118
144 68 167 249 33 46 177 162
103 183 167 55 51 254 145 232
11 213 202 58 152 14 157 140
28 214 11 163 88 234 5 248
62 194 88 187 156 9 201 204
114 247 43 26 162 108 38 38
109 222 53 153 78 117 88 51
93 234 108 122 11 5 190 246
121 210 61 215 11 218 130 88
191 15 170 72 155 39 116 113
57 152 67 216 95 117 48 21
209 220 23 42 165 185 26 163
148 1 100 232 186 183 109 10
206 92 7 109 210 150 108 28
76 49 142 179 1 197 131 23
160 172 58 126 67 13 78 113
38 31 160 202 152 247 142 228
156 126 61 114 46 97 95 186
6 24 174 35 67 91 49 90
34 152 125 129 20 72 94 20
55 59 48 151 201 50 175 41
63 83 132 132 76 109 182 30
131 253 27 100 47 112 89 67
126 31 171 125 147 239 136 126
118 165 66 171 178 62 203 93
6 252 7 14 24 226 210 92
83 206 21 241 171 121 8 115
152 0 171 254 133 204 136 228
168 234 77 161 136 50 68 67
150 214 177 185 99 40 166 198
198 187 100 134 171 46 186 206
111 58 181 207 78 228 111 182
187 218 209 82 140 101 199 220
83 128 67 114 182 228 123 238
137 50 157 28 206 134 148 119

208	13	122	250	40	127	171	170
185	132	85	220	198	111	207	109
123	219	18	99	113	255	80	169
121	227	202	239	199	183	9	209
104	247	149	204	60	35	79	177

a) Complexity Measurements

Complexity means studying each of execution time, input-data, language difficulties, mass storage required by the algorithm etc.

In this study we concentrate on complexity from only three points:

i. *Data complexity.*

The amount of data needed as input to the attack.

ii. *Processing complexity.*

The time needed to perform the attack. This is often called the work factor.

iii. *Storage requirements.*

The amount of memory needed to do the attack [6].

b) Complexity of Algorithms

An algorithm's complexity is determined by the computational power needed to execute the algorithm itself. The computation of an algorithm is often measured by two variables: T (for Time Complexity), and S (for Space Complexity). In general, the computational complexity of an algorithm is expressed in what is called "big O" notation: the order of magnitude of the computation complexity.

Generally, algorithms are classified according to their time or space complexity:

- An algorithm is a constant if its time complexity is independent of n: O(1).
- An algorithm is linear, if its time complexity is O (n).
- An algorithms can also be quadratic, cubic, and so on. Like those algorithms, their complexity are polynomial i.e. O (nm), where m is a constant.

Algorithms whose complexities are O(cf(n)), where c is a constant and f(n) is more than a constant but less than linear, are called "Supper polynomial"[6].

The suggested algorithm will take extra process more than static algorithm as the following:

- The process of conversion from binary todecimal O (n).
- The computation of indexes O (m).
- It needs also extra storage corresponding to the hash table.

IV. Conclusion and Future Work

The permutation is an essential factor in many security cryptosystems. Therefore, we developed a new method that uses dynamic permutation for enhancing the security of the system in a way better than using static permutations.

The future work, dynamic permutation can be used to produce one way hash function.

References Références Referencias

1. Douglasr. Stinson, "Cryptography: Theory and Practice", University of Waterloo, Waterloo, Ontario Canada, 2nd Ed., Chapman & Hall/CRC, (2002).
2. Bruce Schneier, "Applied Cryptography" 3rdEd. John Wiley & Sons. (ASIA) Pvt. Ltd. Singapore 129809, (2010).
3. D. Russell and G. T. Gangemi Sr, "Computer Security Basics" O'Reilly& Associates, Inc., New York, (2009).
4. Dennin, Dorothy E, "Cryptography and Data Security" Library of Congress Cataloging in Publication Data, Addison-Wesley, USA, .(1983).
5. Ellis Horowitz and Sanguthevar Rajasekran, "Computer Algorithms". Galgotia Publication Pvt. Lid., New Delhi, India, (2005).
6. Thomas H. Cormen, Charles E. Leiseroin and Ronald L. Rivest Clifford Stein, "Introduction to Algorithms" 2nd Ed. Prentice, Hall of India, Pvt. Ltd., New Delhi-110 001, (2002).
7. William Stallings, "Cryptography and Network Security: Principles and Practice" 3rd Ed. India, (2009).

Global Journals Inc. (US) Guidelines Handbook 2015

www.GlobalJournals.org