Triangle: Language, Literature, Computation, n. 8, 2012
Publicacions Universitat Rovira i Virgili · ISSN: 2013-939X
https://revistes.urv.cat/index.php/triangle

A Comparison of the DES and Dömösi Cryptosystems

Zoltan Pal Mecsei

Research Group on Mathematical Linguistics
Universitat Rovira i Vigili
Tarragona, Spain
E-mail: zoltan.mecsei@estudiants.urv.cat

Summary. In this paper we compare the well known DES cryptosystem with the recently introduced Dömösi system, which is based on finite automata. We do a time complexity analysis on both algrithms. We show that without making use of an auxiliary matrix the Dömösi cryptosystem is slower than DES. However, the use of auxiliary matrices makes the former perform better than its well known counterpart for some block lengths.

1 The Data Encryption Standard (DES)

First let us take a look at the DES cryptosystem. In particular, if we consider the following to be elementary steps (es): reading input, comparing two values, jumping, the system will have the following requirements:

1. First we perform the initial permutatincludegraphicion as seen in Figures 1 and 2. This phase consists of 64 elementary steps.
2. a) Make a copy of the current half of the 64 bits (1 es)
   b) Extend the current 32 bits to 48 bits (48 es) (Figure 3)

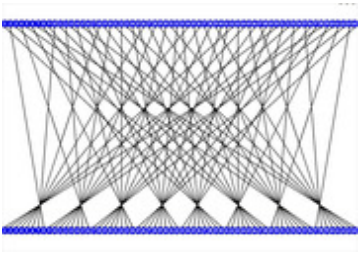

Fig. 1. Initial permutation

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Fig. 2. IP

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Fig. 3. Expansion function (E)

   c) Read the first key (1 es)
   d) XOR the 48 bits resulting from 2b with the key read (1 es)
   e) do 8 s-box $6 \mapsto 4$ bit mappings (3x8=24 es) (Figure 4)
   f) apply the 32 bit permutation (32 es) (Figure 5)
   g) XOR what we have so far and the remaining 32 bits (Figure 6)
3. Swap the two sides (1 es)
4. Perform the inverse permutation (64 es) as seen in Figures 7 and 8

We can now summarize the calculations on each of the steps. More exactly we will get the following:

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

Fig. 4. Substitution box (S-box) S_1

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Fig. 5. Permutation (P)

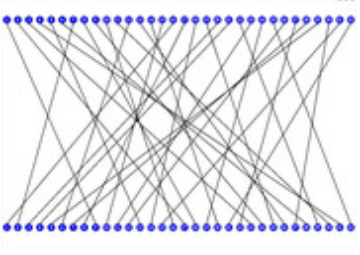

Fig. 6. XOR with the remaining 32 bits

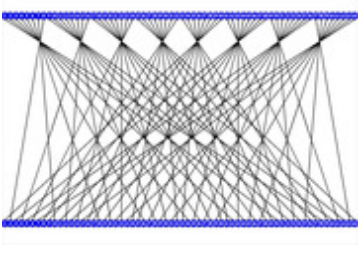

Fig. 7. Final permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Fig. 8. Inverse of IP

$1st step:$ $64es$

$2nd step:$ $(1 + 48 + 1 + 1 + 24 + 32 + 1) \times 16 = 1728es$

$3rd step:$ $1es$

$4th step:$ $64es$

The number of steps needed to perform the 64-bit DES are given by the table in Figure 9. Summing up this table gives us a total of 1857 steps. For a detailed and more formal time complexity analysis of the algorithm please refer to [3, 4, 5].

Step	Operation	Time	Equivalent total	Notes
1.	IP 64 bit transposition	1	64	
2.a	32 bit Copy	16	16	*16 steps
2.b	48 bit transposition	16	48x16	*
2.c	READ the key	16	16	*
2.d	48 bit XOR	16	16	*
2.e	$6 \mapsto 4$ bit two dimensional mapping	8x16	3x128	*
2.f	32 bit transposition	16	32x16	*
2.g	32 bit XOR	16	16	*
3.	32 bit swapping	1	1	
4.	IP^{-1}	1	64	

Fig. 9. Number of steps required by DES

2 The Dömösi System

Let us now move on to the Dömösi system and take a look at the number of elementary steps needed to process 64 bits with or without using an auxiliary matrix.

2.1 Without auxiliary matrix

As we move on to the Dömösi system first we will take a look at the number of elementary steps needed to process 64 bits without using an auxiliary matrix. When considering a Dömösi system without an auxiliary matrix, from a final state we will have the following phases to follow:

1. Read a character $8 \times [1]$, where the number between [] is the number of elementary steps; read a character from previously generated random number row $8 \times [1]$; in the transition matrix we find the state transition corresponding to the random number read $8 \times [Psurit]$, where $Psurit$ is the number of elementary steps of the logical and physical correspondence for the transition matrix
2. This phase is a longer one that depends on the length k of the given code word. The expected step count is $8 \times k \times [2 \times Psurit + 4]$
3. The expected cost of the 3rd step is $8 \times 2 \times Psurit + 4$.
4. Now that the parity is correct, we have to look for the input sign which will take the automaton into a final state. Based on the reference we can expect that this goes down in $8 \times [128 \times Psurit + 132]$ steps, that is with 4 "if"s and with final state compression and comparison reachable in 128 steps.

The phases are explicitly described in the Figure 10:

These four cases consist altogether of k steps of wandering, parity change and state identifying. Together with the cost of the first steps this gives a total of elementary steps described by:

$$8 \times [Psurit \times (2k + 131) + 144 + k \times 4]$$

If we suppose c to be the extra cost for every k steps, in other words the upper approximation for the cost of the extra steps performed by the algorithm, then the formula is equivalent to $8 \times [Psurit \times (2k + 131) + 144 + k \times (4 + c)]$. When implementing in Windows, this extra cost is large for any algorithm, depending on Windows' event handler and scheduler. However, for the purpose of theoretical time complexity analysis we can disregard this cost, as it is insignificant compared to the other factors.

Comparison: In the case of the Dömösi system, the processing of 8-byte generating code words of length k takes $8 \times (Psurit \times (2k + 131) + 144 + k \times (4 + c))$ elementary steps. Furthermore, we can take $Psurit$ to be equal to 3 (as it is considered in the Figure 10) and c to be 0. This way the time cost of processing a 8-byte DES block becomes:

Step	Operation	Times	Equivalent total	Notes
1a	Read a character	8	8	
1b	Read the next random value	8	8	
1c	Mapping	8	8*Psurit	
2a	Read the next random value	8*k*2	8*k*2	*probably in two steps
2b	Mapping	8*k*2	8*k*2*Psurit	*probably in two steps
2c	Comparison(random in non-final state)	8*k*2	8*k*2	*probably in two steps
3a	Read the next random value	8*2	8*2	*probably in two steps
3b	Mapping	8*2	8*2*Psurit	*probably in two steps
3c	Comparison (random in non-final state with correct parity)	8*2	8*2	*probably in two steps
4a	Checking	8	8*4	*check the cases
4b	Mapping to the right final state	8*128	8*128*Psurit	*probably in 128 steps
4c	Comparison (map value, final state)	8*128	8*128	*probably in 128 steps

Fig. 10. No auxiliary matrix

$$8 \times (3 \times (2k + 131) + 144 + k \times 4)) = 10k + 537$$

By looking at the ratio $80k + 4296/1857$ we see that without the auxiliary matrix the algorithm is slower than the DES.

2.2 With auxiliary matrix

In this section we compare the DES cryptosystem with the Dömösi system that uses auxiliary matrices.

With the introduction of the auxiliary matrix the cost becomes minimal in Case 4, as it can be seen in the table. In the case of the DES this cost is 6. Thus, the formula is transformed into $Psurit \times (3 + 2k) + 4k + 12 + 6$. Considering once more $Psurit = 3$ we get $80k + 168 = 1857$. Since the equality holds for $k \approx 21$, it follows that for block lengths 21 the Dömösi cryptosystem with auxiliary matrix performs at same level as the DES cryptosystem. Thus, for blocks of shorter length the Dömösi

Step	Operation	Times	Equivalent total	Notes
1a	Read a character	8	8	
1b	Read the next random value	8	8	
1c	Mapping	8	8*Psurit	
2a	Read the next random value	8*k*2	8*k*2	* probably in two steps
2b	Mapping	8*k*2	8*k*2*Psurit	*probably in two steps
2c	Comparison(random in non-final state)	8*k*2	8*k*2	* probably in two steps
3a	Read the next random value	8*2	8*2	* probably in two steps
3b	Mapping	8*2	8*2*Psurit	* probably in two steps
3c	Comparison (random in non-final state with correct parity)	8*2	8*2	* probably in two steps
4a	Checking	8	8*4	*check the cases
4b	Mapping to the right final state	8*6	8*6*Psurit	*Exactly in 6 steps

Fig. 11. Using auxiliary matrices

cryptosystem is faster than DES while, naturally, when taking longer blocks it is slower. It is still a question, how secure the system remains when assuming these block lengths.

References

1. Dömösi, P. (2007). *Symmetric key cryptographic method and apparatus for information encryption and decryption.*
2. Dömösi, P. (2007). *A practical stream cipher based on finite automata without outputs*, manuscript.
3. Elkeelany, O., M.M. Matalgah, K.P Sheikh, M. Thaker, G. Chaudhry, D. Medhi and J. Qaddour (2002). Performance analysis of IPSec protocol: encryption and authentication. *Communications, 2002. ICC 2002. IEEE International Conference*, volume 2, pp. 1164–1168.
4. http://en.wikipedia.org/wiki/DES_supplementary_material
5. http://en.wikipedia.org/wiki/Data_Encryption_Standard#Overall_structure