QOS E SICUREZZA NELLE RETI
30.05.2016
DES ALGORITHM (A TUTORIAL)

Ing. Amilcare Francesco Santamaria, Ph.D. – DIMES Dpt. University of Calabria

CRITTOGRAFIA
DES

- DES Algorithm works on bit or binary number, this mean that if we have to face with an HEX number such as 1 it needs to be converted in a binary number:
  - 1 – hex = 0001 (4 bits – 1 nibble)
  - 9 hex = 1001
  - A hex = 1010
- DES works on 64 bits (64/4) = 16 hex numbers;
- DES uses a KEY with a length of 64 bits. The DES key (K) is composed of 2 components
  - 56 bits (used as effective key's values)
  - 8 bits (used for control the each 8-th bit of the word is given by the XOR (logic operator) of the previous 7 bits of the word). Example
    - 1100111 – 1 – Therefore, the whole word shall be 11001111

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

CRITTOGRAFIA
DES

- **DES** works on 64-bits length blocks, therefore, we can summarize two cases
  - One block (length < 64 bits) – The block shall be filled with 0s in order to reach the length of 64 bits
  - One block (legth > 64 bits) – In this case we shall obtain more than one block to encrypt (Total block number = total_length[bits] / 64). Last obtained block length must be composed of 64 bits. Follow the same rule explained in the previous point.
- Convert String to hex number – «Your»
  - Your = 59 6F 75 72
  - 59 6F 75 72 = [0101 1001] [0110 1111] [0111 0101] [0111 0010]
- Caratteri Speciali
  - Space = 20
  - CR = 0D
  - Line Feed = 0A

WORKING WITH DES

- Starting with M = 0123456789ABCDEF
  - First step is to achieve the binary form of M
- M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
- L = 0000 0001 0010 0011 0100 0101 0110 0111
- R = 1000 1001 1010 1011 1100 1101 1110 1111
- We have to read from Left ro Right

WORKING WITH DES – FIRST PERMUTATION

- K = 133457799BBCDFF1
- K = 0001 0011 0011 0100 0101 0111 0111 1001 1001 1011 1011 1100 1101 1111 1111 0001

PC-1

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

- K(Permutata) = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

WORKING WITH DES – 16 SUBKEYS

- K(Permutata) = **1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111**
  - C0 = 1111000 0110011 0010101 0101111 [28 bits]
  - D0 = 0101010 1011001 1001111 0001111 [28 bits]
- At this point we have to obtain 16 subkeys bloks following the follow table

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

- For a generic iteration n of the keys (Cn, Dn) each bit has to be moved by x left position. Only the first bit shall take place in the last position of the word.

WORKING WITH DES

$C_0 = 11110000110011001010101011111$
$D_0 = 01010101011001100111100011111$

$C_1 = 11100001100110010101010111111$
$D_1 = 10101010110011001111000111110$

$C_2 = 11000011001100101010101111111$
$D_2 = 01010101100110011110001111101$

$C_3 = 00001100110010101010111111111$
$D_3 = 01010110011001111000111101101$

$C_4 = 00110011001010101011111111100$
$D_4 = 01011001100111100011110101011$

$C_5 = 11001100101010101111111110000$
$D_5 = 01100110011110001111010101011$

$C_6 = 00110010101010111111111000011$
$D_6 = 10011001111000111101010101011$

$C_7 = 11001010101011111111100001100$
$D_7 = 01100111100011111010101010110$

$C_8 = 00101010101111111110000110011$
$D_8 = 10011110001111010101011001111$

$C_9 = 01010101011111111100001100110$
$D_9 = 00111100011110101010101100111$

$C_{10} = 01010101111111110000110011001$
$D_{10} = 11110001111010101010110011001$

$C_{11} = 01010111111111000011001100101$
$D_{11} = 11000111101010101011001100111$

WORKING WITH DES

$C_{12} = 010111111110000110011001010 1$

$D_{12} = 000111101010101011001100 1111$

$C_{13} = 011111111000011001100 1010101$

$D_{13} = 011110101010101100110011 1100$

$C_{14} = 111111100001100110010 1010101$

$D_{14} = 111010101010110011001 1110001$

$C_{15} = 111110000110011001010 1010111$

$D_{15} = 101010101011001100111 1000111$

$C_{16} = 111100001100110010101 0101111$

$D_{16} = 010101010110011001111 0001111$

WORKING WITH DES – KEYS PERMUTATION (PC-2)

- Now we have 16 subkeys, for a generic subkey n we have to apply the following permutation
- This permutation takes into account only 48 bits on 56 that are available. Therefore the result of this operation will be 16 bloks of 48 bits.
- We use n=1 for example
  - K1 = C1D1 =
  - 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

WORKING WITH DES

$K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001\ 110010$

$K_2 = 011110\ 011010\ 111011\ 011001\ 110110\ 111100\ 100111\ 100101$
$K_3 = 010101\ 011111\ 110010\ 001010\ 010000\ 101100\ 111110\ 011001$
$K_4 = 011100\ 101010\ 110111\ 010110\ 110110\ 110011\ 010100\ 011101$
$K_5 = 011111\ 001110\ 110000\ 000111\ 111010\ 110101\ 001110\ 101000$
$K_6 = 011000\ 111010\ 010100\ 111110\ 010100\ 000111\ 101100\ 101111$
$K_7 = 111011\ 001000\ 010010\ 110111\ 111101\ 100001\ 100010\ 111100$
$K_8 = 111101\ 111000\ 101000\ 111010\ 110000\ 010011\ 101111\ 111011$
$K_9 = 111000\ 001101\ 101111\ 101011\ 111011\ 011110\ 011110\ 000001$
$K_{10} = 101100\ 011111\ 001101\ 000111\ 101110\ 100100\ 011001\ 001111$
$K_{11} = 001000\ 010101\ 111111\ 010011\ 110111\ 101101\ 001110\ 000110$
$K_{12} = 011101\ 010111\ 000111\ 110101\ 100101\ 000110\ 011111\ 101001$
$K_{13} = 100101\ 111100\ 010111\ 010001\ 111110\ 101011\ 101001\ 000001$
$K_{14} = 010111\ 110100\ 001110\ 110111\ 111100\ 101110\ 011100\ 111010$
$K_{15} = 101111\ 111001\ 000110\ 001101\ 001111\ 010011\ 111100\ 001010$
$K_{16} = 110010\ 110011\ 110110\ 001011\ 000011\ 100001\ 011111\ 110101$

WORKING WITH DES – MESSAGE PERMUTATION (IP)

- M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

IP

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

IP = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

- Follow IP table (fixing row index and moving on columns) we can compose the IP word. Starting from row 1, M's Bit in position 58 shall be the bit 0 of word IP, M's bit 50 shall be the bit 1 of word IP and so on

WORKING WITH DES – L0 AND R0

- Once IP word is achieved we can obtain two substring composed of 32 bits

$$L_0 = 1100\ 1100\ 0000\ 0000\ 1100\ 1100\ 1111\ 1111$$

$$R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$$

- We have to procede along 16 iterations. Each Iterations has to follow the herein rules:

$$L_n = R_{n-1}$$

$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

WORKING WITH DES – FINDING ENCRYPTED WORD

- Fixing n=1 we have to realize the following steps

$$K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001\ 110010$$

$$L_1 = R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$$

$$R_1 = L_0 \oplus f(R_0, K_1)$$

- To work with f(x) function we have to exapand R_{n-1} from 32 to 48 bits. This shall be made using the E-Table

E BIT-SELECTION TABLE

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

WORKING WITH DES – USING THE E-TABLE

- Considering R0 and apply on it the E-Table permutaion

$$\boldsymbol{R_0} = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$$

$$\mathbf{E}(\boldsymbol{R_0}) = 011110\ 100001\ 010101\ 010101\ 011110\ 100001\ 010101\ 010101$$

- In order to calculate the f(x) function we have to make the following operations:

$$E\left(R_{n-1}\right) \oplus K_n$$

$$K_1 = 000110110000010110111111111100111000001110010$$

$$E\left(R_0\right) = 011110100001010101010101011110100001010101010101$$

$$K_1 + E(R_0) = 011000010001011110111010100011001100101001001111$$

WORKING WITH DES – S FUNCTION PERMUTAION

- As last permutation we have to apply the S(y) function on the result of

$$E(R_{n-1}) \oplus K_n = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$$

- Each B is composed of 6-bits therefore taking as input this consideration we obtain B1 as the first 6 bits of the previous word:

$$B_1 = 011000$$

- Other terms will be obtained in the same way. Regarding S function we obtain:

$$S_1(B_1) S_2(B_2) S_3(B_3) S_4(B_4) S_5(B_5) S_6(B_6) S_7(B_7) S_8(B_8)$$

WORKING WITH DES – S FUNCTION PERMUTAION

- In order to perform S-Permutation we have to specify some rules
- We will take under consideration the term B1 that we have already obtained:

1	2	3	4	5	6
0	1	1	0	0	0

S1

Row No.	Column Number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

WORKING WITH DES – S FUNCTION PERMUTAION

1	2	3	4	5	6
0	1	1	0	0	0

- Bits into positio {1,6} give us the row index to access in the table (i) in this case 00 = 0; i index can assume value in the range [0,3]
- Middle bits {2,3,4,5} give us the column index (j), in this case 1100 = 12
- Accessing in table S1 we obtain the value in decimal form of S1(B1) = 5
  - In binary => 5 = 0101
  - The result is composed of 4 bits

WORKING WITH DES - STABLES

S1

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

S2

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

WORKING WITH DES - STABLES

S3

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

S4

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

S5

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

WORKING WITH DES - STABLES

S6

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

S7

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

WORKING WITH DES - STABLES

S8

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

At the end for the first step (n = 1) we obtain :

$K_1 + \mathbf{E}(R_0) = 011000\ 010001\ 011110\ 111010\ 100001\ 100110\ 010100\ 100111.$

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8) = 0101\ 1100\ 1000\ 0010\ 1011\ 0101\ 1001\ 0111$

WORKING WITH DES – F FUNCTION COMPUTATION

- To achieve the result of S another permutation is needed using the P table

$$f = \mathrm{P}(S_1(B_1)S_2(B_2)\ldots S_8(B_8))$$

P

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

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111

we get

f = 0010 0011 0100 1010 1010 1001 1011 1011

WORKING WITH DES

- Coming back on Evaluation of

$$L_n = R_{n-1}$$
$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

$$R_1 = L_0 \oplus f(R_0, K_1)$$

$$= 1100\ 1100\ 0000\ 0000\ 1100\ 1100\ 1111\ 1111$$
$$+ 0010\ 0011\ 0100\ 1010\ 1010\ 1001\ 1011\ 1011$$
$$= 1110\ 1111\ 0100\ 1010\ 0110\ 0101\ 0100\ 0100$$

- We have to repeat all those steps until we reach the 16 step. At this point we shall report all in a 64 bits form, making the «Last»

LAST PERMUTATION

L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100

R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101

- We shall revert the Bits order by using the Last Permutation Table called IP^{-1}

$R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

LAST PERMUTATION TABLE (IP^{-1})

$R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

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

IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

which in hexadecimal format is

85E813540F0AB405.

This is the encrypted form of **M** = 0123456789ABCDEF: namely, **C** = 85E813540F0AB405.

CRYPTOGRAPHY
RSA

- is one of the first practicable public-key cryptosystems and is widely used for secure data transmission.
- In such a cryptosystem, the encryption key (e) is public and differs from the decryption key (d) which is kept secret.
- RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977.

RSA

- Step 1 – Choosing of Two large prime numbers (p,q)
- Step 2 – Compute n = p * q
  - N is called modulus for both public and private keys.
- Step 3 – Compute the Euler's Totient function of n
  - $\varphi(n) = (p-1) * (q-1)$
- Step 4 – Choose an integer «e» such that 1<e<$\varphi(n)$
  - «e» and $\varphi(n)$ are coprime.
  - «e» is released as public key exponent
- Step 5 – gcd(e, $\varphi(n)$) = 1
- Step 6 – d is the multiplicative inverse of e mod $\varphi(n)$ and it is the private key exponent keeped secret

RSA – EXTENDED EUCLIDEAN ALGORITHM

- Step 6 is commonly solved using the extended Euclidean Algorithm
- Considering the following assumption to better explain the algorithm
  - m = p
  - n = $\varphi(n)$
- Algorithm starts as follow
- $q_0 = \left\lfloor \frac{n}{m} \right\rfloor$ $r_0 = n - m * q_0$

$n = q_0$*(m)+r_0	$p_0 = 0$

- $q_1 = \left\lfloor \frac{m}{r_0} \right\rfloor$ $r_1 = m - r_0 * q_1$

$m = q_1$*(r_0)+r_1	$p_1 = 1$

$r_0 = q_2$*(r_1)+r_2	$p_2 = p_0 - p_1 * q_0 mod\ n$

RSA – EXTENDED EUCLIDEAN ALGORITHM

- *Most generally, after, the iterations 0 and 1 we can generalize the iteration i-th as follow*
- $q_i = \left\lfloor \frac{r_{i-2}}{r_i - 1} \right\rfloor \qquad r_i = r_{i-2} - r_{i-1} * q_i$
  - $r_{i-2} = q_i*(r_{i-1})+r_i$
  - $p_i = p_{i-2} - (p_{i-1} * q_{i-2}) mod\ n$
- **Algorithm ends when**
  - $r_i = 0$
- **At this point, it is possible to compute the d-key in the following way:**
  - $d = p_{i+1} = p_{i-2} - (p_{i-1} * q_{i-2}) mod\ n$

RSA – EXAMPLE 1

- p = 13
- q = 53
- n = p * q = 13 * 53 = 689
- $\varphi(n) = \varphi(689) = (p-1) * (q-1) = 12 * 52 = 624$
- e t.c. 1<e<624
  - Let choose e as a prime number that is not a divisor of 624

RSA – EXAMPLE 1 - ITERATIONS

Iterazione		
0	$q_0 = \left\lfloor \frac{f(n)}{e} \right\rfloor = \left\lfloor \frac{624}{5} \right\rfloor = 124$	$\boldsymbol{r_0 = f(n) - (e * q_0)}$ $\boldsymbol{= 624 - (5 * 124) = 4}$
		$p_0 = 0$
1	$q_1 = \left\lfloor \frac{e}{r_0} \right\rfloor = \left\lfloor \frac{5}{4} \right\rfloor = 1$	$\boldsymbol{r_1 = e - (r_0 * q_1) = 5 - (4 * 1)}$ $\boldsymbol{= 1}$
		$p_1 = 1$
2	$q_2 = \left\lfloor \frac{r_{i-2}}{r_{i-1}} \right\rfloor = \left\lfloor \frac{4}{1} \right\rfloor = 4$	$\boldsymbol{r_2 = r_{i-2} - (r_{i-1} * q_i)}$ $\boldsymbol{= 4 - (1 * 4) = 0}$
		$p_2 = p_{i-2} - (p_{i-1} * q_{i-2}) mod\ n$ $= 0 - 124 mod 624 = 500$
3		$p_3 = p_{i-2} - (p_{i-1} * q_{i-2}) mod\ n =$ $1 + (-500 * 1 \bmod 624) = d = 125$

RSA – EXAMPLE 1

- we encrypt 111 follow the equation herein shown:
  - C = $111^5 mod\ 689 = 687$
  - P = $687^{125} mod\ 689 = 111$

Just a simple reminder : e = 5, n = 689, d=125.