Симулација и практична имплементација на сите слаби и полуслаби клучеви на DES.

Автор:
Андреј Бардакоски

Мај 2023 година

Содржина

Вовед

Што е DES?

DES (Data encryption standard) е алгоритам за шифрирање кој спаѓа во групата на блок шифрувачи со симетричен клуч. DES е развиен од компанијата IBM и прв пат публикуван во 1975 година а во 1977 е поставен за FIPS стандард. DES и неговата посигурна верзија Triple DES бил користен и важел како стандар три децеении се до 2005 година кога бил заменет од AES (Advanced Encryption Standard).

Карактеристики на DES

DES е блок шифрувач кој шифрира блокови со големина од по 64 бита односно 8 бајти тоа значи дека DES на влез зема блок од 64 бита како порака (plain text), а на излез враќа блок од 64 како шифрирана порака (cypher text). Алгоритмот се состои од 16 рунди каде што во секоја рунда на блокот се применуваат функции на експанзија, пермутација, субституциј и додавање на рундовски клуч. Користи 56 битен клуч од кој што се генерираат шестнаесет различни 48 битни рундовски клучеви. DES користи Феистелова мрежа односно спаѓа во групата на Феистелови шифрувачи. Карактеристично за DES е тоа дека алгоритмот за декрипција е подетнакво ист како и алгоритмот за енкрипција единствената разлика е во распределбата на рундовските клучеви, се генерираат истите рундовски клучеви само во обратен редослед.

Како работи DES

Алгоритмот започнува со влез на 64 битен блок и првата операција што се извршува врз блокот е иницијална пермутација каде што битовите од блокот се мешаат пример седмиот бит доаќа на последно место додека на седмо место доаѓа десетиот бит.
Потоа почнува првата рунда каде блокот се дели на два дела од по 32 бита лев и десен дел, на десниот дел се применува Феистелова функција која се состои од четири дела експанзија, додавање на рундовски клуч, субституција и пермутација, потоа излезот од функцијата се додава на левиот дел и така се добива десниот дел од влезотниот блок во следната рунда, а левиод дел од влезниот блок во следната рунда всушност е десниот блок од тековната рунда. Ова се повторува вкупно 16 пати односно има 16 рунди.
Функцијата експанзија на влез прима блок од 32 бита а на излез враќа блок од 48 бита проширувањето се добива до дуплицирање на некои битови пример четвртиот бит од влезот доаѓа на пето и седмо место во излезниот блок.
Додавање на рундовски клуч е всушност XOR операција со рундовски клуч, во секоја рунда се користи различен 48 битен рундовски клуч кој се добива од тајниот 56 битен клуч. Функцијата субституцаја прима 48 битен блок а враќа 32 битен.
Субституцијата се врши со користење на 8 различни S-box-ови каде што 6 бита влез се заменуваат со 4 бита излез, ова представува единствена нелинеарна функција и е главен фактор за сигурноста на алгоритмот.
Функцијата пермутацијата ги меша битовите од блокот слично како и иницијалната пермутација, целта на оваа функција е битови коишто биле излез од еден S-box од тековната рунда да влијаат на повеќе S-box-ови во следната рунда.
После завршување на сите 16 рунди доаѓа финална пермутација која е инверзна на иницијалната пермутација, на седмо место доаѓа последниот бит, на десето место седмиот итн. Излезот од финалната пермутација е 64 битен блок со шифрираниот текст (cypher text) што е и излез од алгоритмот.

Колку е DES сигурен

DES денес се смета за недоволно сигурен шифрувач заради релативно краткиот 56 битен клуч. Односно вкупниот број на клучеви е 2^{56} што со користење на денешната технологија напад со груба сила односно пробување на сите клучеви е изводлив. Таквиот напад побарува голема пресметковна моќ и би траел релативно долго време но сепак е изводлив.

Клучеви во DES

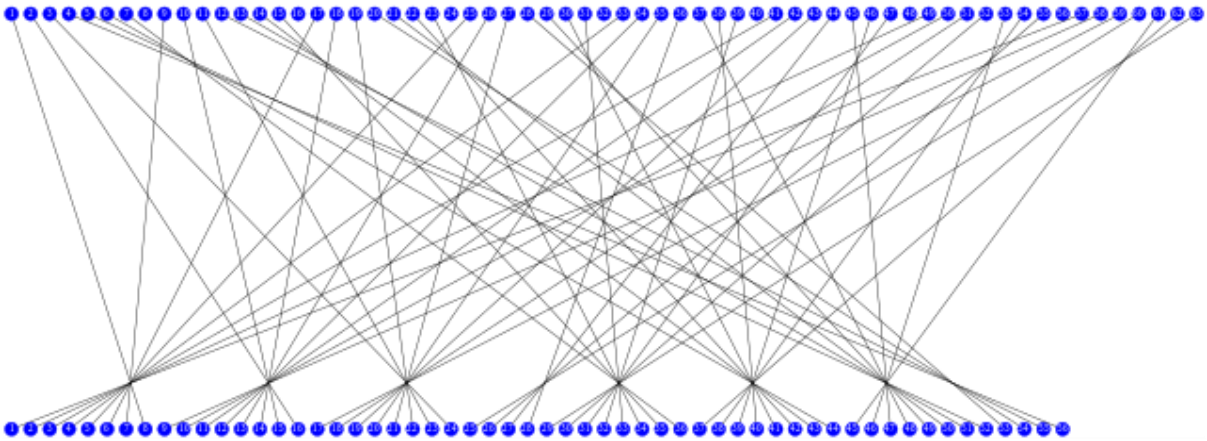
Официјално клучот во DES се чува и пренесува како 8 бајти од кој што секој е со непарна парност, односно се состои од 64 бита но само 56 од нив се користат од алгоритмот останатите осум се битови за проверка на парност. Оттука ефективната големина на клучот е 56 бита па и просторот на клучеви е 2^{56} односно вкупниот број на клучеви е 2^{56}.

Како се добиваат рундовските клучеви

Permuted Choice 1 (PC-1)

Првиот чекор е добивање на 56 битен клуч од иницијалниот 64 битен со функцијата Permuted Choice 1 (PC-1) тоа се прави така што најпрвин се отсрануваат битовите за проверка на парност а тоа е секој осми бит односно битовите на позиции 8, 16, ..., 56, 64, а потоа се врши пермутација односно мешање на останатите битови. Од кога ќе го добиеме 56 битниот клуч од него понатаму се генерираат рундовските клучеви.

i Permuted choice 1 (PC-1)



Кружно поместување (circular shift)

56 битниот клуч се дели на два блока лев и десен секој од по 28 бита. Потоа секој блок се се ротира односно кружно поместува во лево за 1 бит во 1ва, 2ра, 9та и 16та рунда а за 2 бита во сите останати рунди. Кружно поместување во лево за 2 бита значи дека сите битови во блокот ќе се поместат за две позиции во лево, првиот бит ќе дојде на предпоследната позиција, а вториот бит на последна позиција.

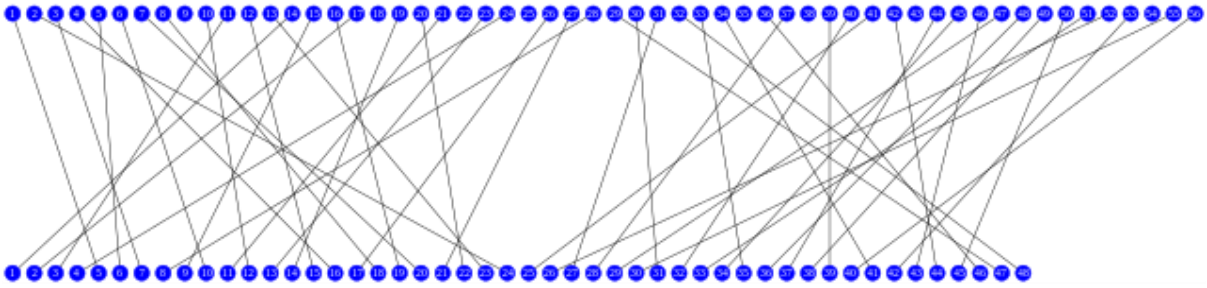
Од кога ќе се извршат соодветните поместувања левиот и десниот блок се спојуваат, конкатанираат, формираќи 56 битен блок врз кој се применува функцијата Permuted Choice 2 (PC-2) и така се добива 48 битниот рундовски клуч за соодветната рунда.

Permuted Choice 2 (PC-2)

PC-2 прави пермутација односно мешање на битовите, но и го намалува бројот на битови од 56 на 48 со тоа што отсранува 8 битови. Исто така интересно за PC-2 е тоа што не доаѓа до межање на битовите меѓу двете половини на блокот односно ако еден бит се наоѓал во првата

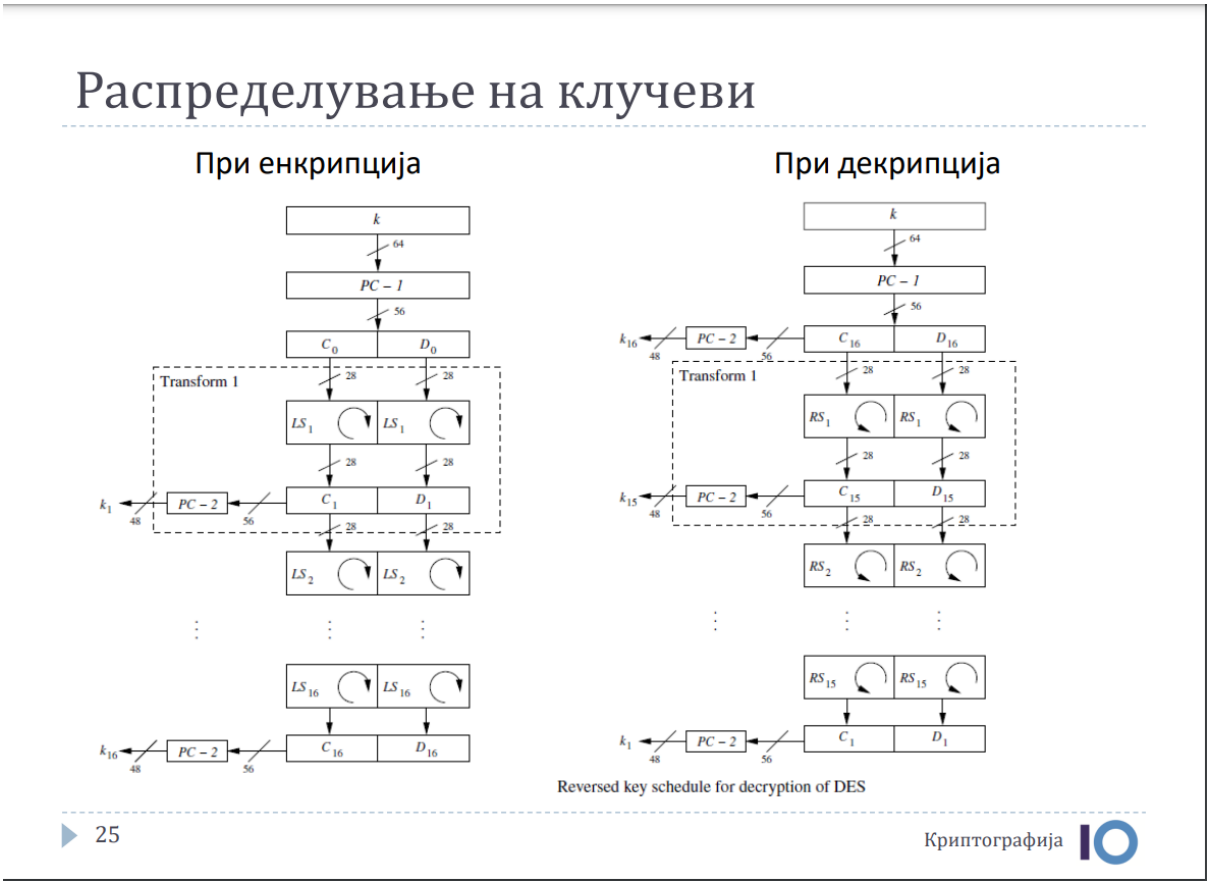
половина од блокот пред PC-2 тогаш повторно ќе се наоѓа во првата половина од блокот и после PC-2.

ii Permuted choice 2 (PC-2)



При декрипција се генерираат истите кличеви но само во обратен редослед, за да се постигне тоа се применуваат истите чекори како и при енкрипција само кружното ротирање на левиот и десниот дел од 56 битниот клуч е во десна насока и тоа за еден бит во 2-ра, 9-та и 16-та рунда во 1-ва рунда нема ротација, а во сите останати, за 2 бита во десно.

iii Распределба на клучеви



Слаби и полу слаби клучеви

Во криптографијата, слаб клуч е клуч, кој употребен со одреден шифрвач, прави шифрувачот да се однесува на некој непожелен начин. Слабите клучеви обично претставуваат многу мал дел од целокупниот простор на клучеви, што значи дека клучот добиен со генерирање на случаен број има многу мала веројатност да е слаб клуч а со тоа и да доведе до безбедносен проблем.

Слаби клучеви во DES

Слаби клучеви во DES се сметаат клучевите за кој што важи дека операцијата шифрирање е идентична со операцијата дешифрирање. Односно $E(x) = E^{-1}(x)$ од ова следува дека $E\big(E(x)\big) = x$ односно дека доколку некоја порака два пати се шифрира се добива самата порака. За да важи ова правило мора рундовските клучеви генерирани при енкрипција да се исти со тие генерирани при декрепција односно првиот рундовски клуч да биде еднаков на последниот вториот на предпоследниот итн. Но заради природата на алгоритам за генерирање на рундовски клучеви ова е возможно само доколку сите рундовски клучеви се еднакви. Тоа единствено ќе се случи доколку клучот по доведување во 56 битна форма односно после PC-1 е составен само од 0, само од 1 или една половина само од 0 а друга само од 1. Па така добиваме вкупно 4 слаби клучеви и тоа:

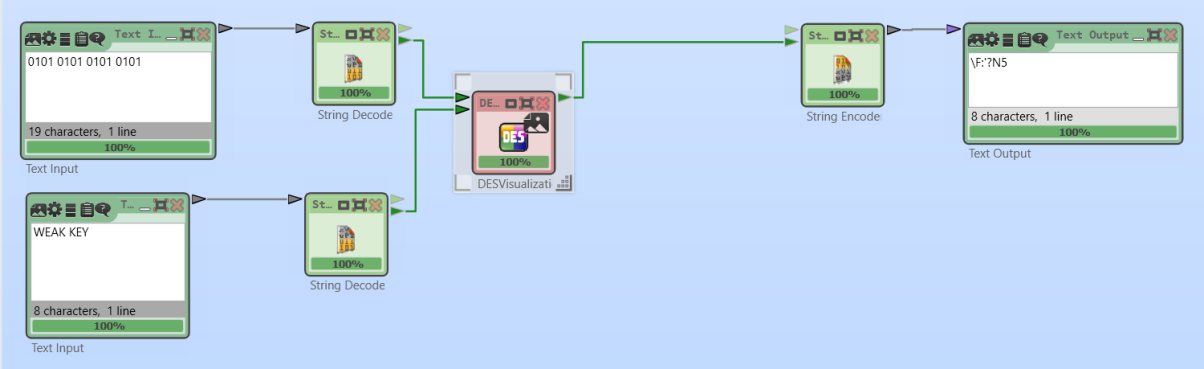
64 битен клуч во хексадецимален запис	56 битен клуч во хексадецимален запис
0101 0101 0101 0101	0000000 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000

Симулација на слаби клучеви

За симулација на алгоритмот DES ќе ја искористиме алатката [CrypTool 2]

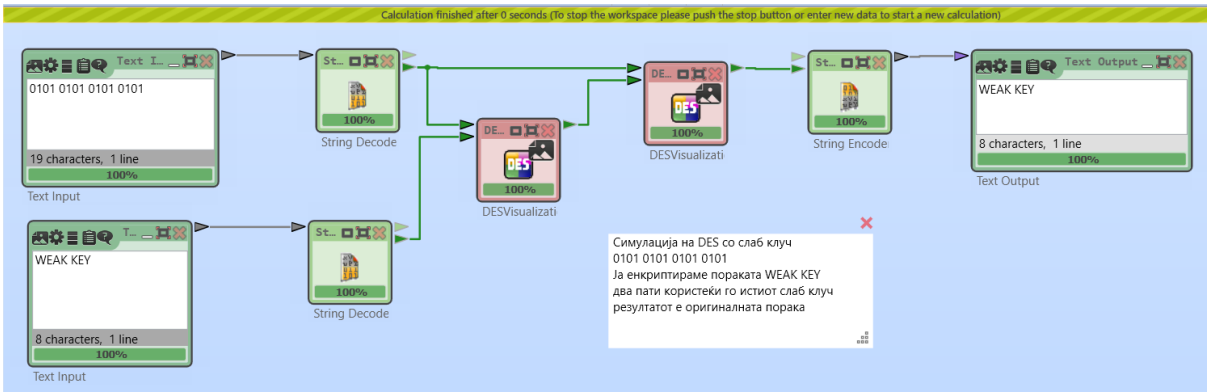
Симулација на 0101 0101 0101 0101

iv симлуација 1 со клич 0101 0101 0101 0101



Сакаме да ја енкриптираме пораката “ WEAK KEY “ користеќи го шифрувачот DES со слаб клуч 0101 0101 0101 0101. Резултатот односно шифрираната порака е “ \F:’?N5 “ , што на прв поглед изгледа дека е добро.

v симлуација 2 со клич 0101 0101 0101 0101



Но додоколку пораката “ WEAK KEY “ ја енкриптираме два пати користејќи го истиот слаб клуч добиваме резултат односно шифрирана порака “ WEAK KEY “ што е всушност оригиналната порака.

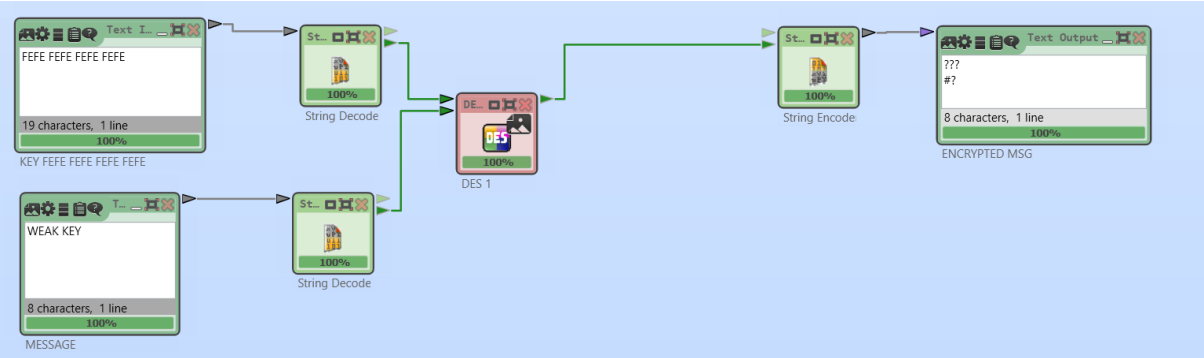
vi рундовски клучови генерирани од 0101 0101 0101 0101

Round	Round Key
1	000000000000000000000000000000000000000000000000
2	000000000000000000000000000000000000000000000000
3	000000000000000000000000000000000000000000000000
4	000000000000000000000000000000000000000000000000
5	000000000000000000000000000000000000000000000000
6	000000000000000000000000000000000000000000000000
7	000000000000000000000000000000000000000000000000
8	000000000000000000000000000000000000000000000000
9	000000000000000000000000000000000000000000000000
10	000000000000000000000000000000000000000000000000
11	000000000000000000000000000000000000000000000000
12	000000000000000000000000000000000000000000000000
13	000000000000000000000000000000000000000000000000
14	000000000000000000000000000000000000000000000000
15	000000000000000000000000000000000000000000000000
➔ 16	000000000000000000000000000000000000000000000000

Дополнително доколку ги видиме рундовските клучови генерирани во секоја рунда ќе забележиме дека всушност тие се сите исти.

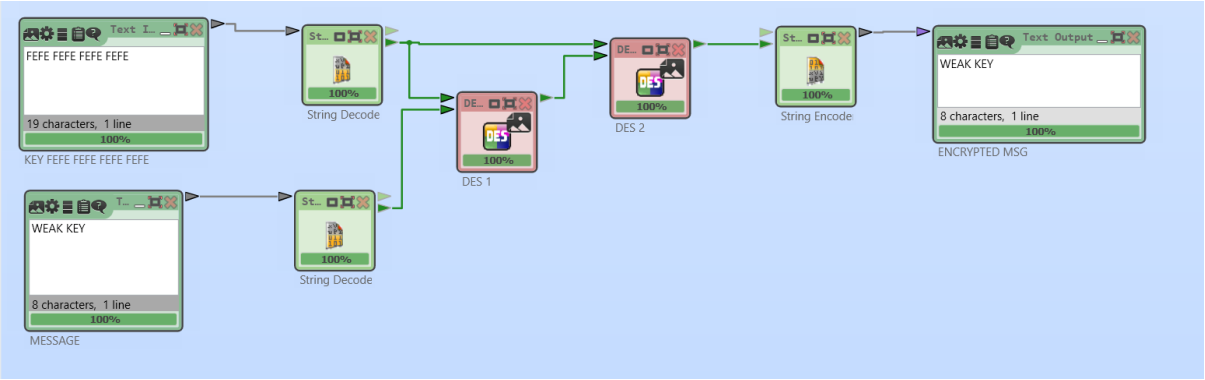
Симулација на FEFE FEFE FEFE FEFE

vii симлуација 1 со клич FEFE FEFE FEFE FEFE



Сакаме да ја енкриптираме пораката “ WEAK KEY “ користеќи го шифрувачот DES со слаб клуч FEFE FEFE FEFE FEFE. Резултатот односно шифрираната порака е “ ???\n#? “.

viii симлуација 2 со клич FEFE FEFE FEFE FEFE



Но додоколку пораката “ WEAK KEY “ ја енкриптираме два пати користејќи го истиот слаб клуч повторно ја добиваме оригиналната порака “ WEAK KEY “.

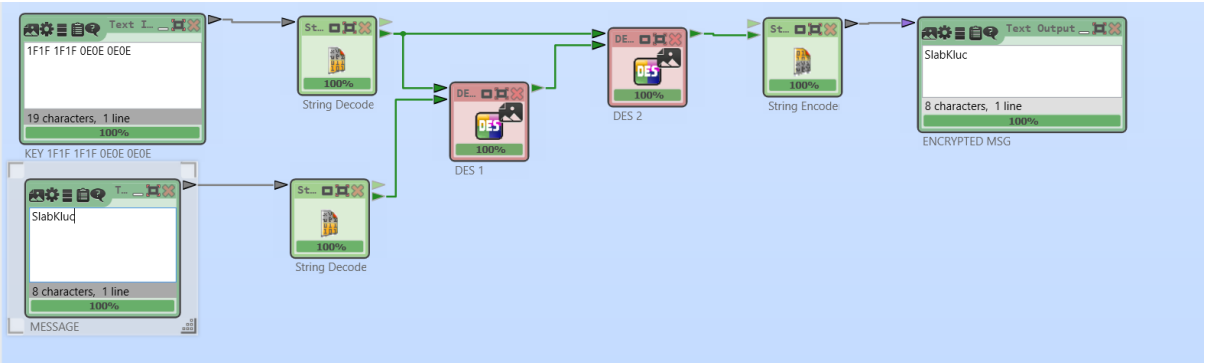
ix рундовски клучеви генерирани од FEFE FEFE FEFE FEFE

Round	Round Key
1	111111111111111111111111111111111111111111111111
2	111111111111111111111111111111111111111111111111
3	111111111111111111111111111111111111111111111111
4	111111111111111111111111111111111111111111111111
5	111111111111111111111111111111111111111111111111
6	111111111111111111111111111111111111111111111111
7	111111111111111111111111111111111111111111111111
8	111111111111111111111111111111111111111111111111
9	111111111111111111111111111111111111111111111111
10	111111111111111111111111111111111111111111111111
11	111111111111111111111111111111111111111111111111
12	111111111111111111111111111111111111111111111111
13	111111111111111111111111111111111111111111111111
14	111111111111111111111111111111111111111111111111
15	111111111111111111111111111111111111111111111111
→ 16	111111111111111111111111111111111111111111111111

Дополнително доколку ги видиме рундовските клучови генерирани во секоја рунда повторно ќе забележиме дека тие се сите исти.

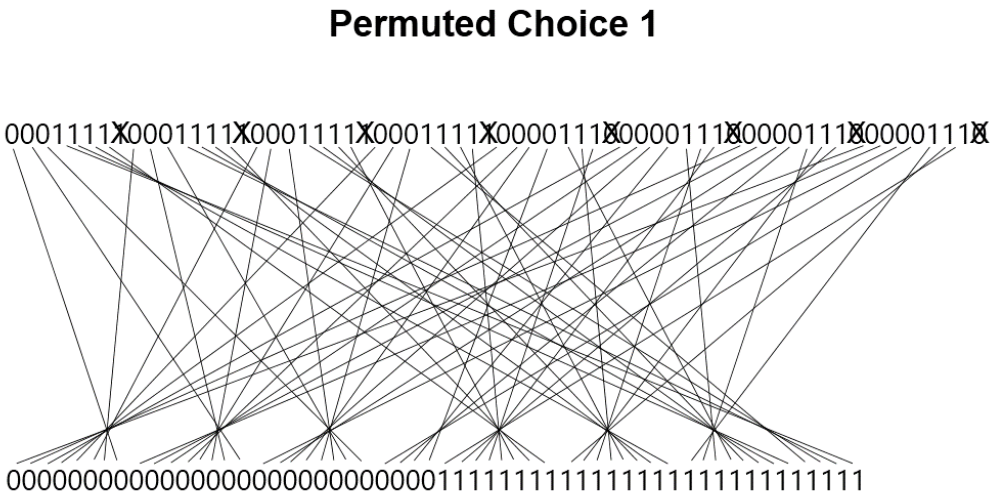
Симулација на 1F1F 1F1F 0E0E 0E0E

x симлуација 1 со клич 1F1F 1F1F 0E0E 0E0E



Да пробаме со друга порака и друг слаб клуч. Овој пат ќе ја шифрираме пораката “ SlabKluc “ ја енкриптираме два пати користејќи го слабиот клуч 1F1F 1F1F 0E0E 0E0E резултатот е оригиналната пораката “ SlabKluc “.

xi PC-1 со клуч 1F1F 1F1F 0E0E 0E0E



Може да забележиме дека по Permuted Choice 1, од 64 битниот клуч се добива 56 битен чиј што први 28 бита се нули а останатите 28 се единици.

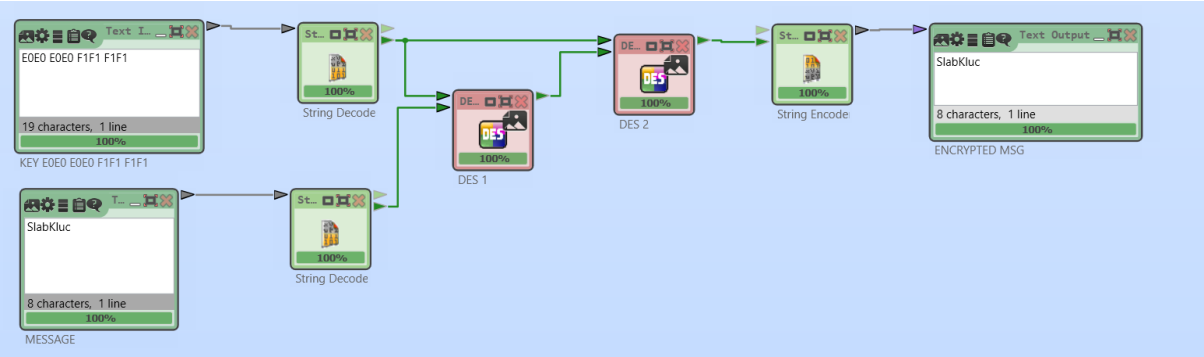
xii рундовски клучеви генерирани од 1F1F 1F1F 0E0E 0E0E

Round	Round Key
1	000000000000000000000000111111111111111111111111
2	000000000000000000000000111111111111111111111111
3	000000000000000000000000111111111111111111111111
4	000000000000000000000000111111111111111111111111
5	000000000000000000000000111111111111111111111111
6	000000000000000000000000111111111111111111111111
7	000000000000000000000000111111111111111111111111
8	000000000000000000000000111111111111111111111111
9	000000000000000000000000111111111111111111111111
10	000000000000000000000000111111111111111111111111
11	000000000000000000000000111111111111111111111111
12	000000000000000000000000111111111111111111111111
13	000000000000000000000000111111111111111111111111
14	000000000000000000000000111111111111111111111111
15	000000000000000000000000111111111111111111111111
➔ 16	000000000000000000000000111111111111111111111111

Сите рундовски клучеви генерирани од 1F1F 1F1F 0E0E 0E0E се исти.

Симулација на E0E0 E0E0 F1F1 F1F1

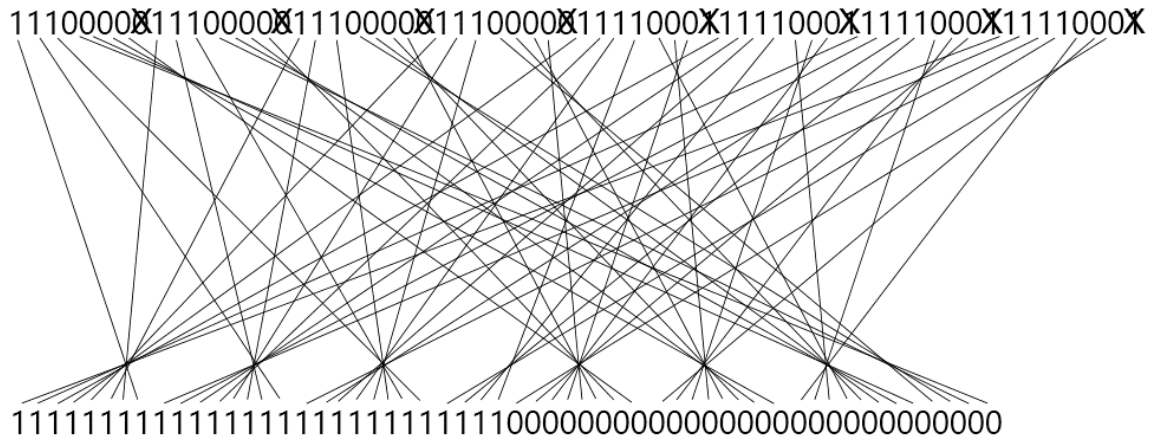
xiii симлуација 1 со клич E0E0 E0E0 F1F1 F1F1



Користеќи го преостанатиот слаб клуч E0E0 E0E0 F1F1 F1F1 повторно со двојна енкрипција се добива оригиналната порака.

xiv PC-1 со клуч E0E0 E0E0 F1F1 F1F1

Permuted Choice 1



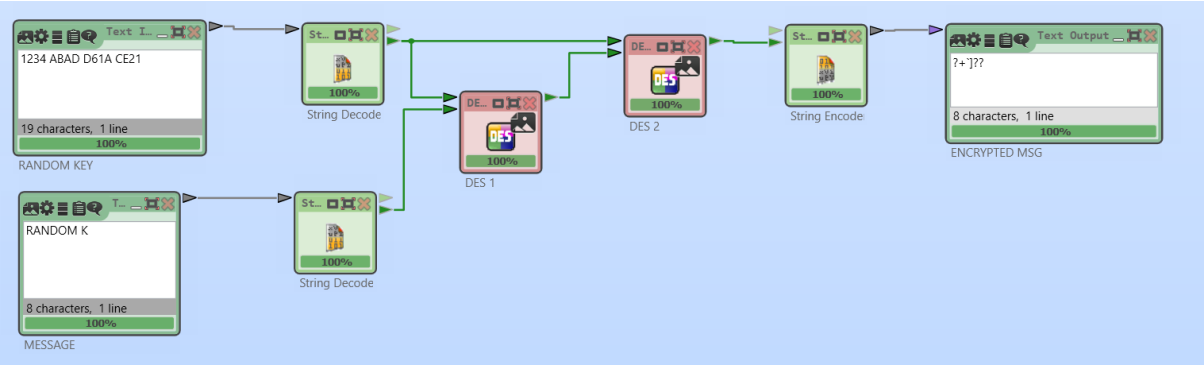
По Permuted Choice 1, од клучот E0E0 E0E0 F1F1 F1F1 се добива 56 битен клуч чиј што први 28 бита се единици а останатите 28 се нули.

xv рундовски клучеви генерирани од E0E0 E0E0 F1F1 F1F1

Round	Round Key
1	11111111111111111111111100000000000000000000000000
2	11111111111111111111111100000000000000000000000000
3	11111111111111111111111100000000000000000000000000
4	11111111111111111111111100000000000000000000000000
5	11111111111111111111111100000000000000000000000000
6	11111111111111111111111100000000000000000000000000
7	11111111111111111111111100000000000000000000000000
8	11111111111111111111111100000000000000000000000000
9	11111111111111111111111100000000000000000000000000
10	11111111111111111111111100000000000000000000000000
11	11111111111111111111111100000000000000000000000000
12	11111111111111111111111100000000000000000000000000
13	11111111111111111111111100000000000000000000000000
14	11111111111111111111111100000000000000000000000000
15	11111111111111111111111100000000000000000000000000
➔ 16	11111111111111111111111100000000000000000000000000

Симулација на двојна енкрипција со случаен клуч

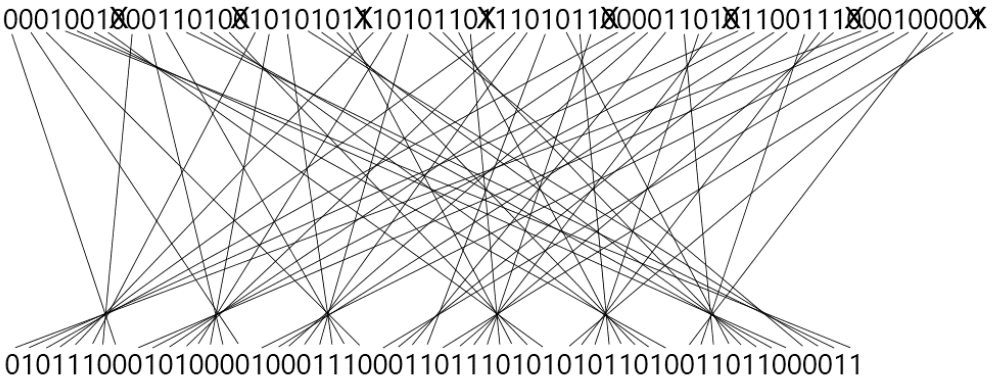
xvi симлуација 1 со случаен клуч



Ајде да пробаме ја енкриптираме пораката “ RANDOM K ” два пати користејќи го случајниот клуч 1234 ABAD D61A CE21. Резултатот од енкрипцијата е “ ?+`]?? ”.

xvii PC-1

Permuted Choice 1



xviii рундовски клучеви генерирани од случајниот клич

Round	Round Key
1	001011100010000110101100001101110111011000110010
2	000000111011010111000101100010100101101111000111
3	010110010100110011000011010101101100001110110101
4	011100011110000110101000110100110000110111001001
5	100100001000001011000011111001010101100110001100
6	011100010000101000010111011100110111011100101100
7	001001011011000010100100011110000000110011010101
8	100100100000010011111011011000100011110000011111
9	010000001100111001010011101010011100110111000001
10	011001011111100100100010110010101110011000010011
11	101000101010010111000011111111110100011100001100
12	011110010100011000010011100110000101001111001010
13	011001011001000110011000110101001111001000100101
14	000101101000000011010111111100100010111011101000
15	001111110100100000010010101110001011101100011011
→ 16	010001010111101100001001101011110000010110101101

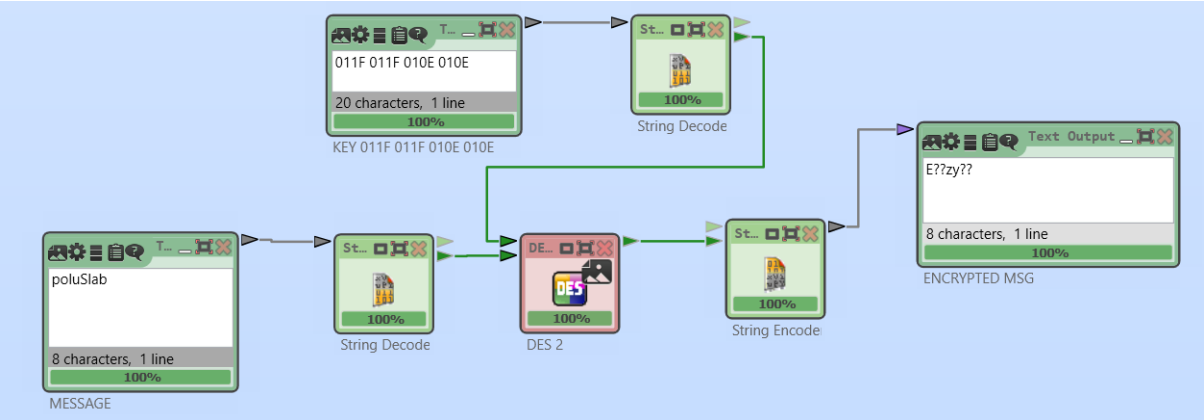
Полу слаби клучеви во DES

Полу слаби клучеви во DES доаѓаат во парови односно, пар од два клуча K_1 и K_2 се смета за пар од полу слаби клучеви доколку операцијата на енкрипција со коритење на едниот клуч е идентична со операцијата на дешифрирање со користење на другиот клуч. Односно $E_{k1}(x) = E_{k2}^{-1}(x);\ E_{k2}(x) = E_{k1}^{-1}(x)$ од ова следува дека $E_{k1}\left(E_{k2}(x)\right) = x$ односно дека доколку некоја порака ја енкриптираме со клучот k2 а пота ја енкриптираме уште еднаш со k1 се добива самата порака. За да важи ова правило мора рундовските клучеви добиени со користење на првиот клуч да се еднакви со рундовските клучеви добиени со користење на вториот клуч само во обратен редослед. Постојат само 6 вакви парови полу слаби клучеви односно вкупно 12 полу слаби клучеви. Секој полу слаб клуч генерира само два рундовски клучеви кои се повторуваат по 6 пати.

64 битен клуч 1 во хексадецимален запис	64 битен клуч 2 во хексадецимален запис
011F 011F 010E 010E	1F01 1F01 0E01 0E01
01E0 01E0 01F1 01F1	E001 E001 F101 F101
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

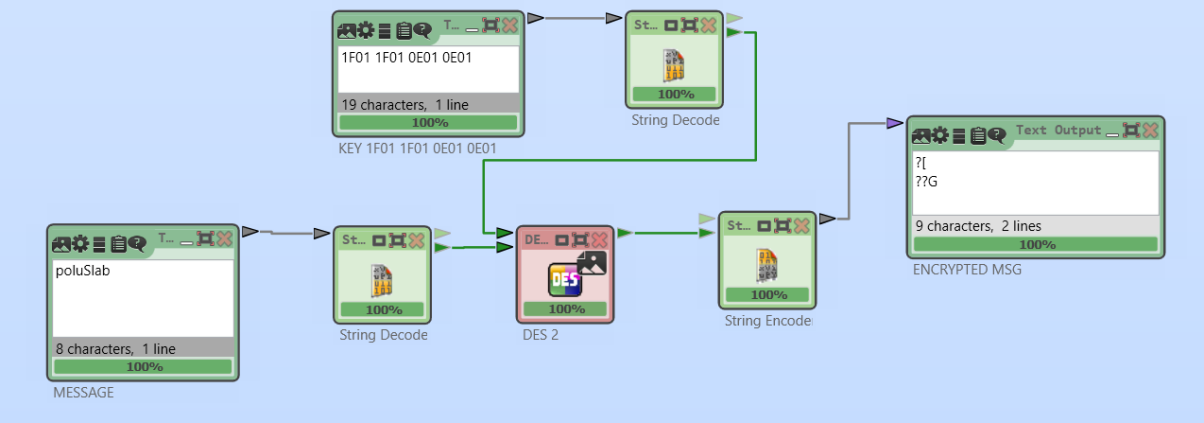
Симулација на 011F 011F 010E 010E и 1F01 1F01 0E01 0E01

xix Симулација 1 со 011F 011F 010E 010E



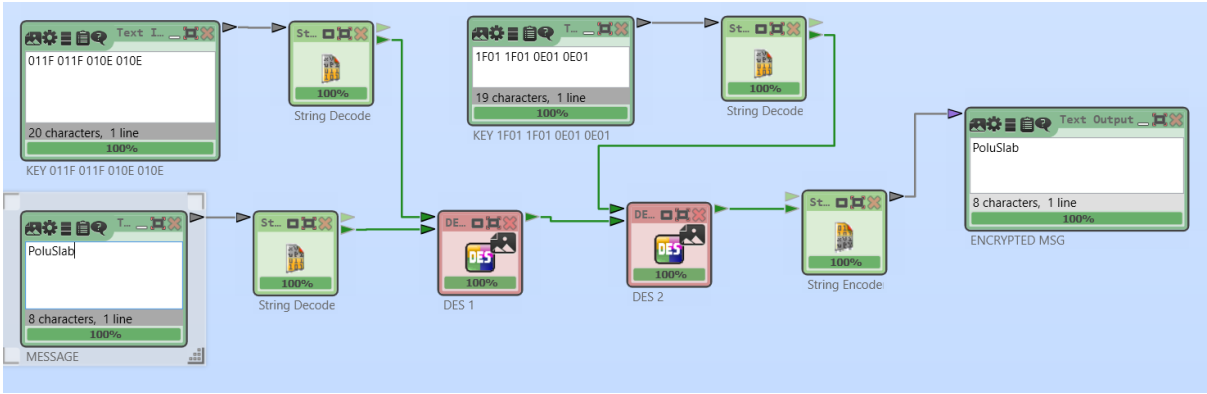
Сакаме да ја енкриптираме пораката “ poluSlab “ користеќи го шифрувачот DES со полу слаб клуч 011F 011F 010E 010E. Резултатот односно шифрираната порака е “ E??zy?? “ , што на прв поглед изгледа дека е добро.

xx Симулација 1 со 1F01 1F01 0E01 0E01



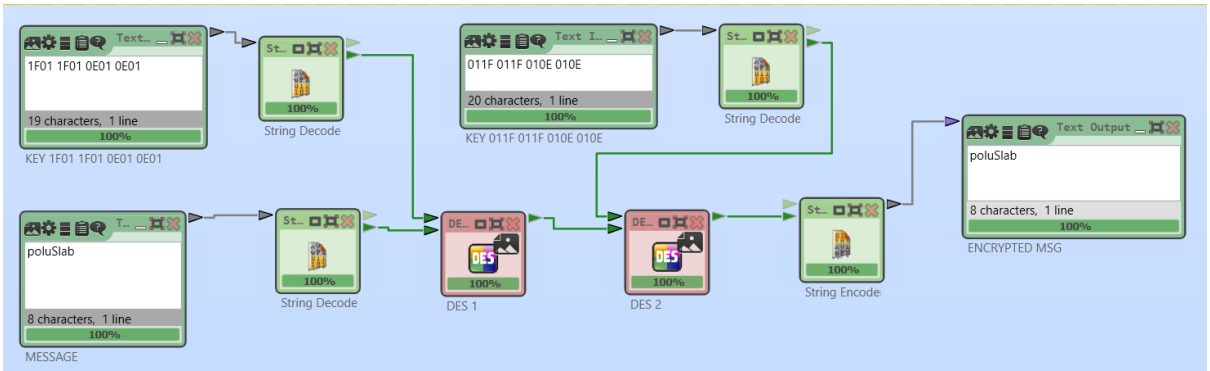
Ајде да пробаме да ја шифрираме истата порака “ poluSlab “ користеќи го другиот полу слаб клуч 1F01 1F01 0E01 0E01. Резултатот односно шифрираната порака е “ ?[\n??G “.

xxi Симулација 1 со 011F 011F 010E 010E и 1F01 1F01 0E01 0E01



Ајде да видиме што ќе се случи доколку пораката “ poluSlab “ првин ја шифрираме користејќи го клучот 011F 011F 010E 010E а потоа коритејќи го 1F01 1F01 0E01 0E01. Резултатот е оригиналната порака “ poluSlab “.

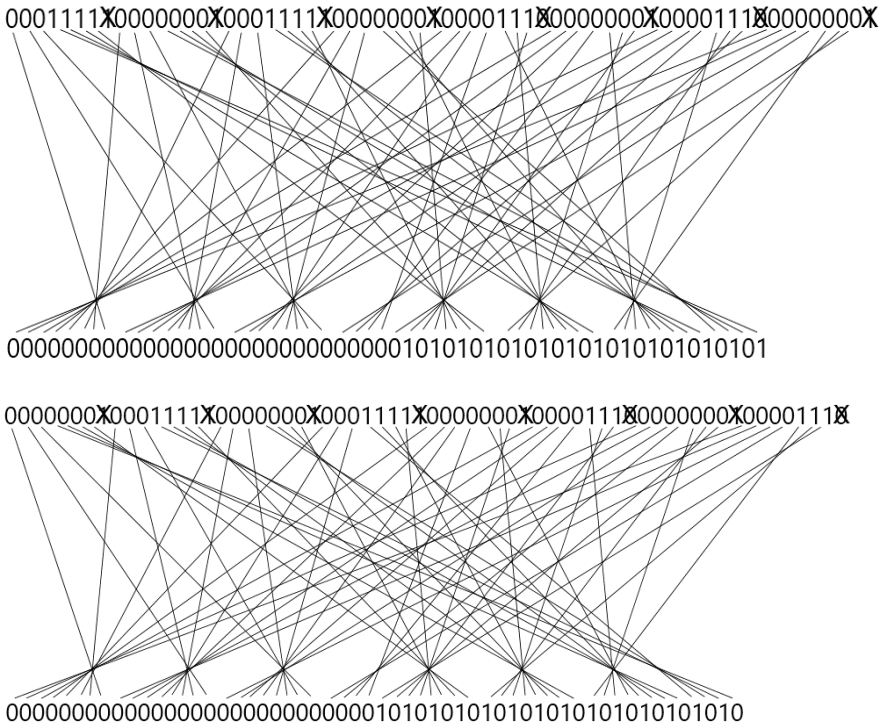
xxii Симулација 2 со 011F 011F 010E 010E и 1F01 1F01 0E01 0E01



Ајде да пробаме обратно првин ќе шифрираме користејќи го клучот 1F01 1F01 0E01 0E01 а потоа коритејќи го 011F 011F 010E 010E. Резултатот повторно е оригиналната порака “ poluSlab “.

xxiii PC-1 за 011F 011F 010E 010E и 1F01 1F01 0E01 0E01

Permuted Choice 1



Доколку ги разгедаме 56 битните клучеви добиени по PC-1 ќе забележиме дека десниот дел и во двата случаи е составен од наизменични нули и единици, разликата е во тоа што во првиот случај завршува со 1 а во вториот со 0.

xxiv Рундовски клучеви генерирани од 011F 011F 010E 010E и 1F01 1F01 0E01 0E01

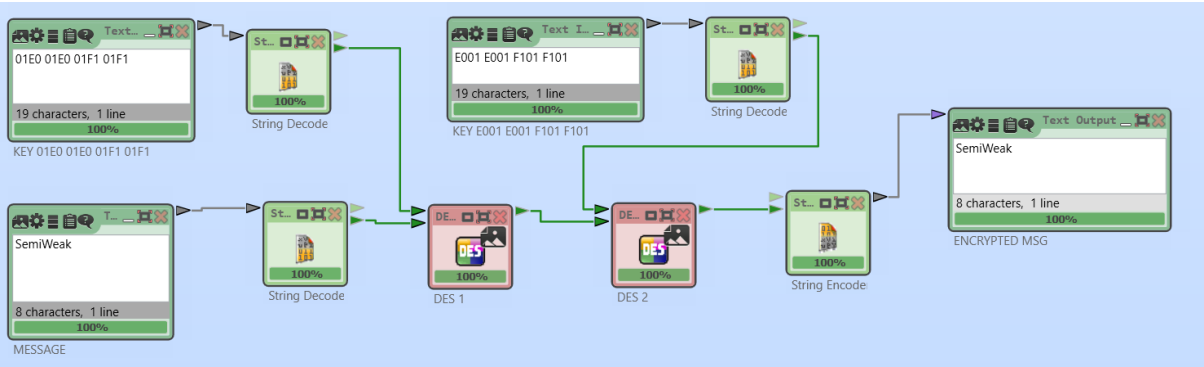
Round	Round Key	Round Key
1	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
2	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010
3	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010
4	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010
5	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010
6	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010
7	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010
8	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010
9	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
10	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
11	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
12	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
13	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
14	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
15	0000000000000000000000000101111001110011001000010	0000000000000000000000000100001100011001101111101
➔ 16	0000000000000000000000000100001100011001101111101	0000000000000000000000000101111001110011001000010

Ако ги анализираме рундовските клучеви добиени од клучевите 011F 011F 010E 010E и 1F01 1F01 0E01 0E01 ќе забележиме дека има само два различни рундовски клуча кои се

повторуваат. Дополнително распределбада на клучеви добиени во првиот случај е иста како и распределбата во вториот случај само во обратен редослед.

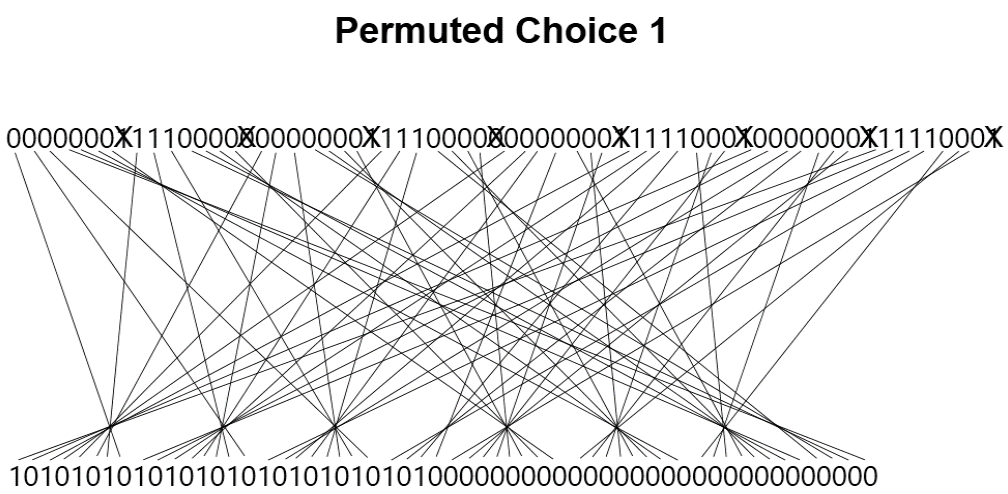
Симулација на 01E0 01E0 01F1 01F1 и E001 E001 F101 F101

xxv Симулација 1 со 01E0 01E0 01F1 01F1 и E001 E001 F101 F101



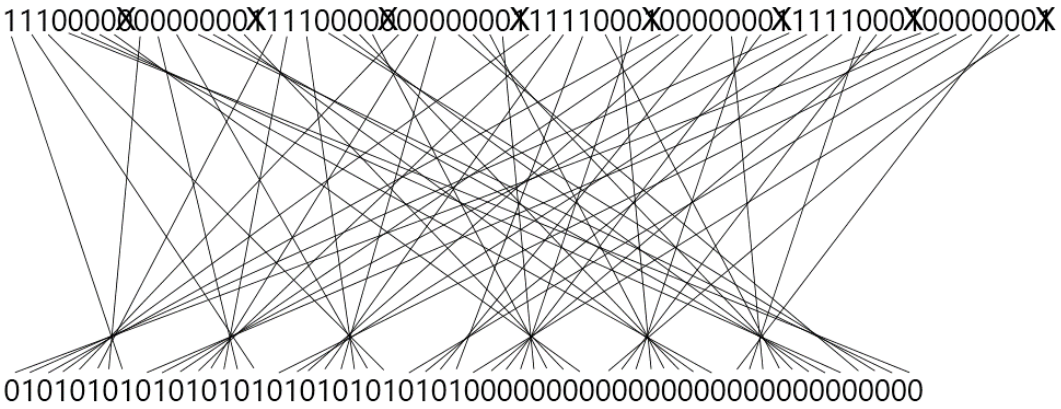
Да пробаме да шифрираме со друг пар на полу слаби клучеви овој пат првин со 01E0 01E0 01F1 01F1 па со E001 E001 F101 F101 и друга порака “ SemiWeak “ резултатот е оригиналната порака

xxvi PC-1 за 01E0 01E0 01F1 01F1



xxvii PC-1 за E001 E001 F101 F101

Permuted Choice 1



Од PC-1 за клучевите 01E0 01E0 01F1 01F1 и E001 E001 F101 F101 може да забележиме дека се случува истото како и кај претходниот пар на полу слаби клучеви но овој пат наизменичното повторување на 1 и 0 е во првите 28 бита од 56 битниот клуч односно во левата половина.

xxviii Рундовски клучеви генерирани од 01E0 01E0 01F1 01F1

	Round	Round Key
	1	100100010101001111100101000000000000000000000000
	2	011011101010110000011010000000000000000000000000
	3	011011101010110000011010000000000000000000000000
	4	011011101010110000011010000000000000000000000000
	5	011011101010110000011010000000000000000000000000
	6	011011101010110000011010000000000000000000000000
	7	011011101010110000011010000000000000000000000000
	8	011011101010110000011010000000000000000000000000
	9	100100010101001111100101000000000000000000000000
	10	100100010101001111100101000000000000000000000000
	11	100100010101001111100101000000000000000000000000
	12	100100010101001111100101000000000000000000000000
	13	100100010101001111100101000000000000000000000000
	14	100100010101001111100101000000000000000000000000
	15	100100010101001111100101000000000000000000000000
→	16	011011101010110000011010000000000000000000000000

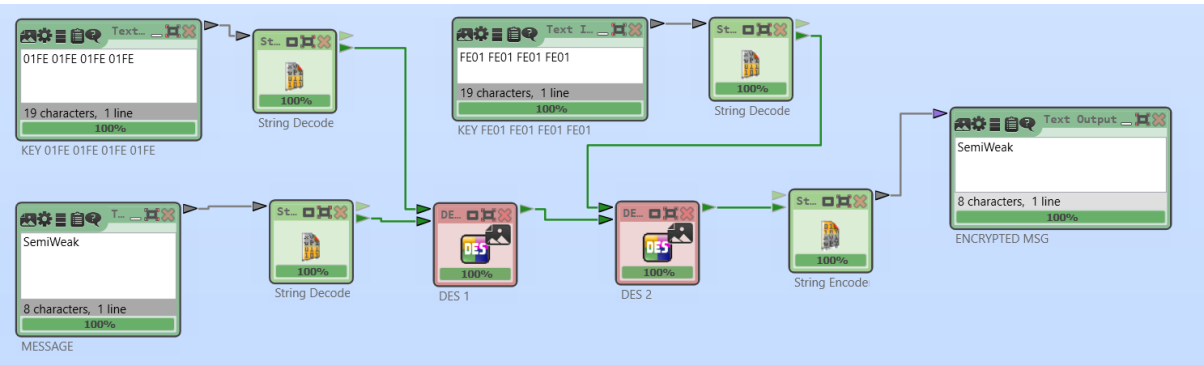
xxix Рундовски клучеви генерирани од E001 E001 F101 F101

Round	Round Key
1	011011101010110000011010000000000000000000000000
2	100100010101001111100101000000000000000000000000
3	100100010101001111100101000000000000000000000000
4	100100010101001111100101000000000000000000000000
5	100100010101001111100101000000000000000000000000
6	100100010101001111100101000000000000000000000000
7	100100010101001111100101000000000000000000000000
8	100100010101001111100101000000000000000000000000
9	011011101010110000011010000000000000000000000000
10	011011101010110000011010000000000000000000000000
11	011011101010110000011010000000000000000000000000
12	011011101010110000011010000000000000000000000000
13	011011101010110000011010000000000000000000000000
14	011011101010110000011010000000000000000000000000
15	011011101010110000011010000000000000000000000000
→ 16	100100010101001111100101000000000000000000000000

Повторно од парот полу слаби клучевите се генерираат само два различни рундовски клучеви. Дополнително распределбада на клучеви добиени во првиот случај е иста како и распределбата во вториот случај само во обратен редослед.

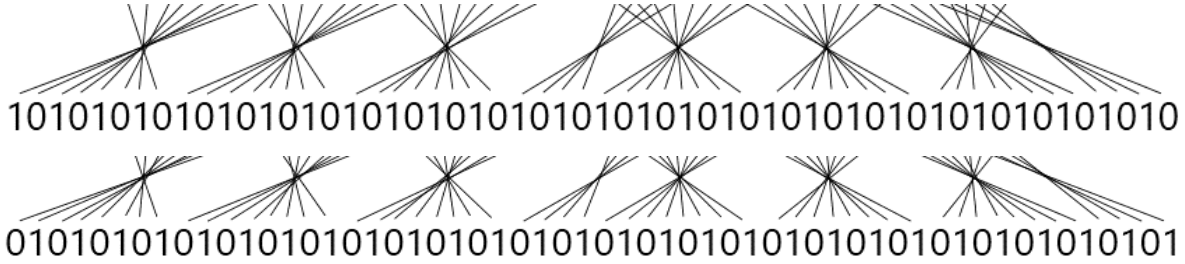
Симулација на 01FE 01FE 01FE 01FE и FE01 FE01 FE01 FE01

xxx Симулација 1 со 01FE 01FE 01FE 01FE и FE01 FE01 FE01 FE01



Енкриптираме порака со 01FE 01FE 01FE 01FE па со FE01 FE01 FE01 FE01 резултатот е самата порака

xxxi PC-1 за 01FE 01FE 01FE 01FE и FE01 FE01 FE01 FE01



Овојпат 56 битните клучеви се составени само од наизменични единици и нули.

xxxii Рундовски клучеви генерирани од 01FE 01FE 01FE 01FE

	Round	Round Key
	1	10010001010100111110010101000011000110011011110 1
	2	01101110101011000001101010111100111001100100001 0
	3	01101110101011000001101010111100111001100100001 0
	4	01101110101011000001101010111100111001100100001 0
	5	01101110101011000001101010111100111001100100001 0
	6	01101110101011000001101010111100111001100100001 0
	7	01101110101011000001101010111100111001100100001 0
	8	01101110101011000001101010111100111001100100001 0
	9	10010001010100111110010101000011000110011011110 1
	10	10010001010100111110010101000011000110011011110 1
	11	10010001010100111110010101000011000110011011110 1
	12	10010001010100111110010101000011000110011011110 1
	13	10010001010100111110010101000011000110011011110 1
	14	10010001010100111110010101000011000110011011110 1
	15	10010001010100111110010101000011000110011011110 1
→	16	01101110101011000001101010111100111001100100001 0

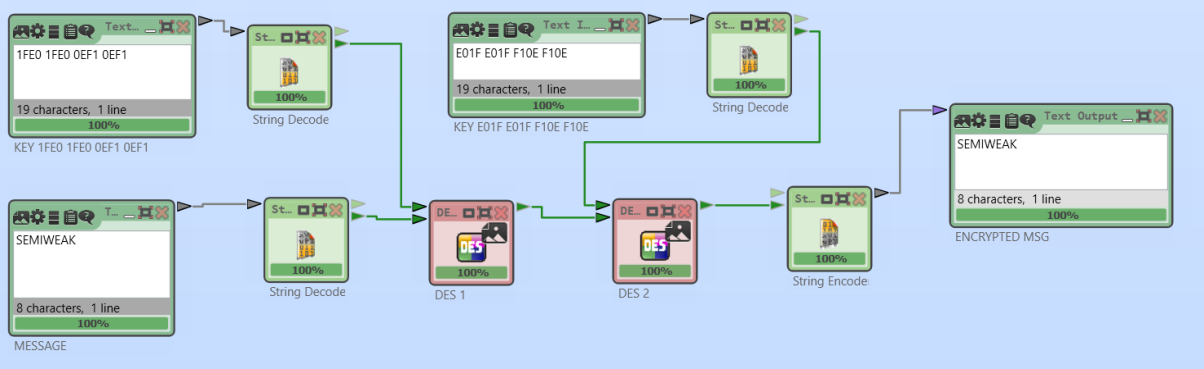
xxxiii Рундовски клучеви генерирани од FE01 FE01 FE01 FE01

Round	Round Key
1	011011101010110000011010101111001110011001000010
2	100100010101001111100101010000110001100110111101
3	100100010101001111100101010000110001100110111101
4	100100010101001111100101010000110001100110111101
5	100100010101001111100101010000110001100110111101
6	100100010101001111100101010000110001100110111101
7	100100010101001111100101010000110001100110111101
8	100100010101001111100101010000110001100110111101
9	011011101010110000011010101111001110011001000010
10	011011101010110000011010101111001110011001000010
11	011011101010110000011010101111001110011001000010
12	011011101010110000011010101111001110011001000010
13	011011101010110000011010101111001110011001000010
14	011011101010110000011010101111001110011001000010
15	011011101010110000011010101111001110011001000010
➔ 16	100100010101001111100101010000110001100110111101

Повторно од парот полу слаби клучевите се генерираат само два различни рундовски клучеви. Дополнително распределбада на клучеви добиени во првиот случај е иста како и распределбата во вториот случај само во обратен редослед.

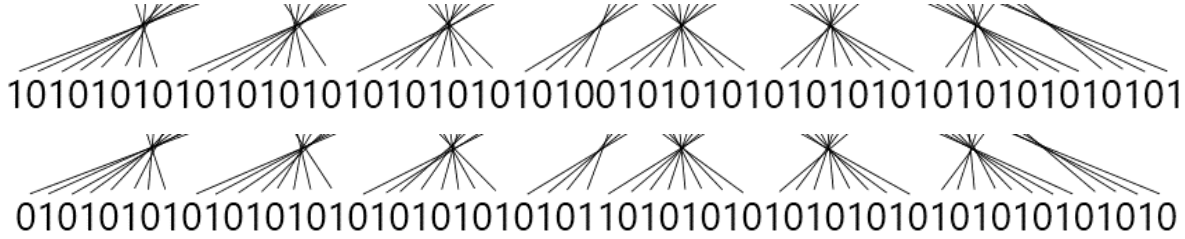
Симулација на 1FE0 1FE0 0EF1 0EF1 и E01F E01F F10E F10E

xxxiv Симулација 1 со 1FE0 1FE0 0EF1 0EF1 и E01F E01F F10E F10E



Повторно по примена на енкрипција на првиот полу слаб клуч па на вториот се добива оригиналната пораката

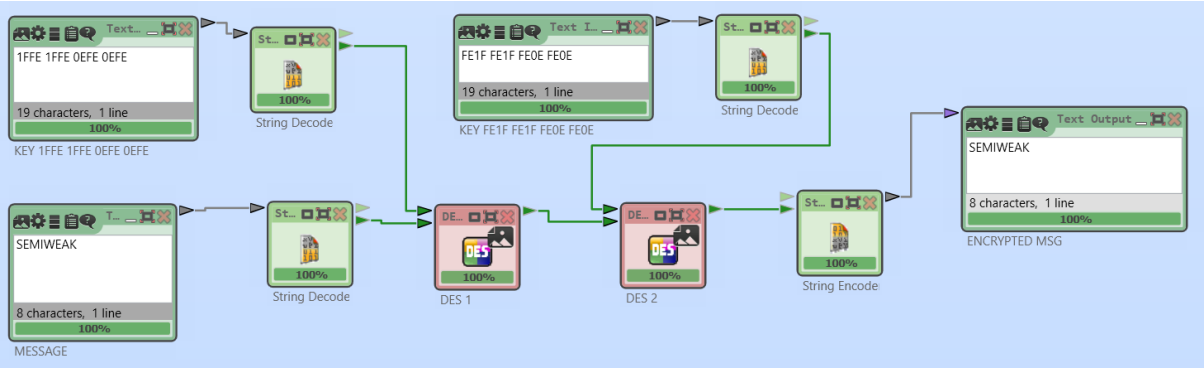
xxxv PC-1 за 1FE0 1FE0 0EF1 0EF1 и E01F E01F F10E F10E



56 битните клучеви изгледаат слично како и клучевите од предходниот пар. Сега се составени од наизменични нули и единици но во првиот случај клучот започнува и завршува на единица а во вториот започнува и завршува со 0.

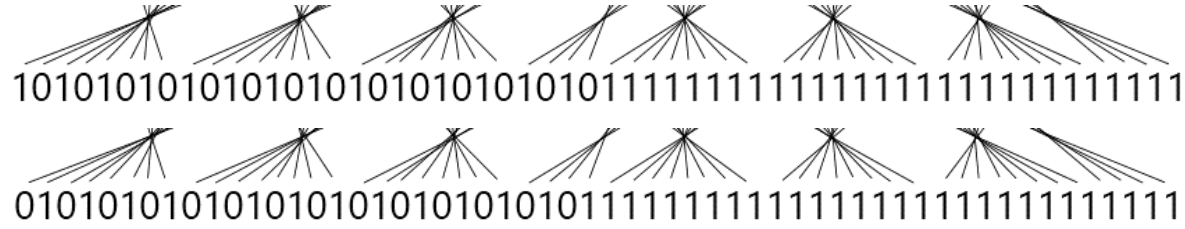
Симулација на 1FFE 1FFE 0EFE 0EFE и FE1F FE1F FE0E FE0E

xxxvi Симулација 1 со 1FFE 1FFE 0EFE 0EFE и FE1F FE1F FE0E FE0E



Повторно по примена на енкрипција на првиот полу слаб клуч па на вториот се добива оригиналната пораката

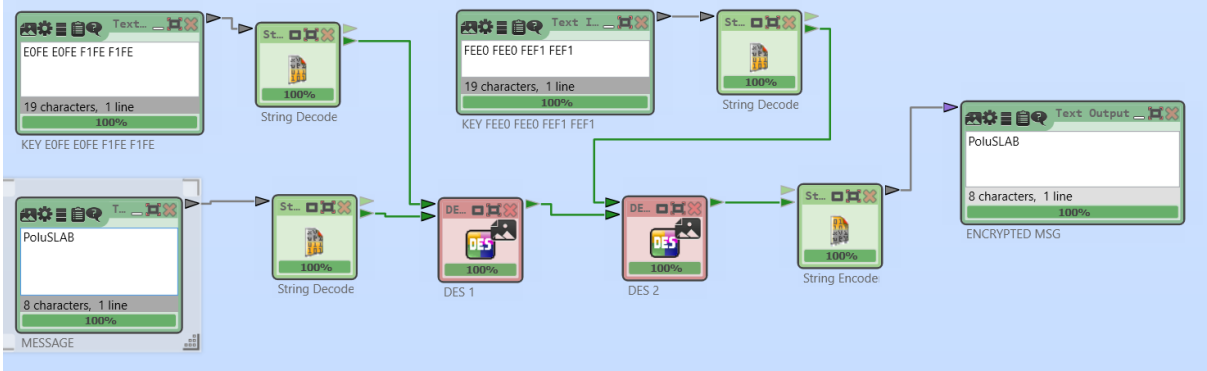
xxxvii PC-1 за 1FFE 1FFE 0EFE 0EFE и FE1F FE1F FE0E FE0E



56 битните клучеви што се добиваат по PC-1 се состојат од наизменични нули и единици во првата половина и само од единици во втората.

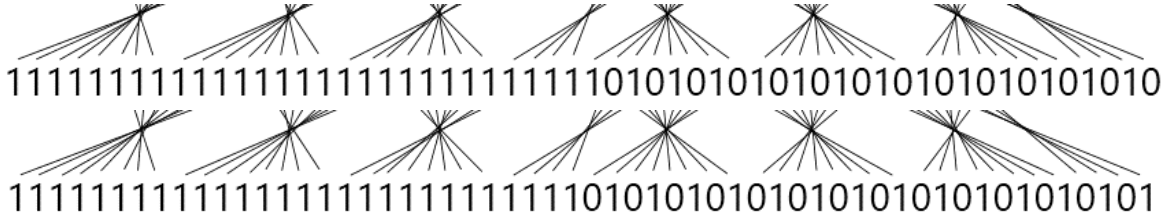
Симулација на E0FE E0FE F1FE F1FE и FEE0 FEE0 FEF1 FEF1

xxxviii Симулација 1 со E0FE E0FE F1FE F1FE и FEE0 FEE0 FEF1 FEF1



И за последниот пар на полу клучеви по примена на енкрипција на првиот полу слаб клуч па на вториот се добива оригиналната пораката

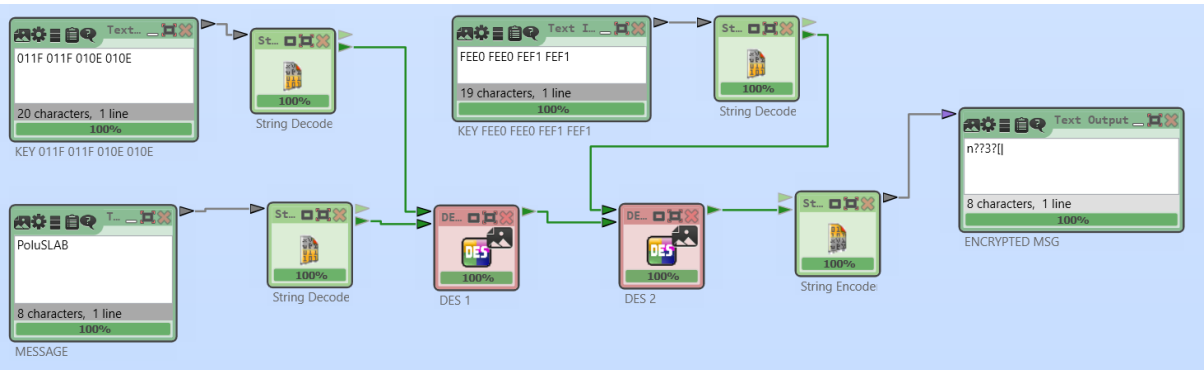
xxxix PC-1 за E0FE E0FE F1FE F1FE и FEE0 FEE0 FEF1 FEF1



56 битните клучеви што се добиваат по PC-1 од последниот пар на полу слаби клучеви се состојат само од единици во првата половина и од наизменични нули и единици во втората.

Симулација на два полу слаби клучеви кој не се пар

xl симулација на полу два полу слаби клучеви кој не се пар



Ајде да пробаме ја енкриптираме пораката “ PoluSLAB ” првин со клулот 011F 011F 010E 010E, потоа со FEE0 FEE0 FEF1 FEF1. И двата клучеви се полу слаби клучеви, но тие не се пар па така резултатот он енкрипцијата е “ n??3?[| ”.

Возможни слаби клучеви

Освен слабите и полу слабите клучеви во DES исто така постојат и возможни слаби клучеви. Тоа се клучеви кои што наместо да генерираат 16 различни рундовски клучеви тие генерираат само 4 различни рундовски клучеви од кои секој се повторува 4 пати.
Постојат 48 клучеви кои што го имаат ова својство на возможни слаби клучеви.

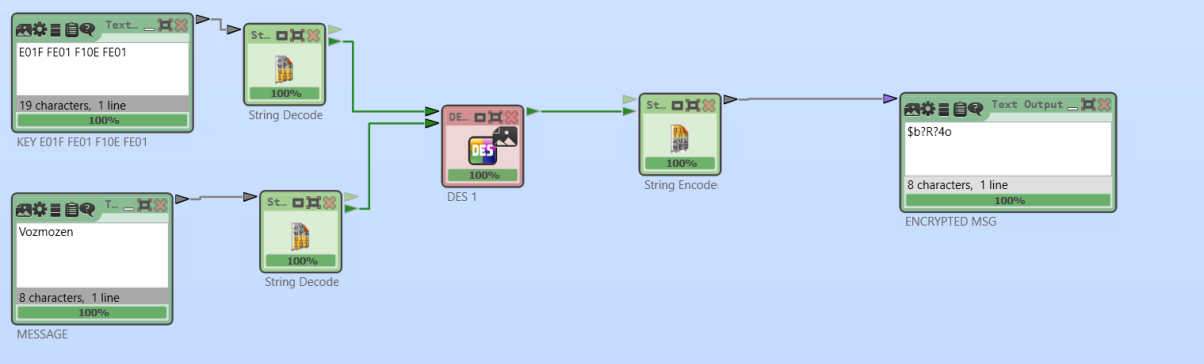
Овие клучеви немаат толку очигледни својства на симетрија за шифрирање/дешифрирање како слабите и полуслабите клучеви, но тие сепак произведуваат многу поедноставен распоред на клучеви од очекуваното, што можеби може некако да се искористи за напад на шифрувачот.

xli Табела со сите возможни слаби клучеви

1F1F 0101 0E0E 0101	E001 01E0 F101 01F1	011F 1F01 010E 0E01	FE1F 01E0 FE0E 01F1
1F01 011F 0E01 010E	FE01 1FE0 FE01 0EF1	0101 1F1F 0101 0E0E	E01F 1FE0 F10E 0EF1
E0E0 0101 F1F1 0101	FE01 01FE FE01 01FE	FEFE 0101 FEFE 0101	E01F 01FE F10E 01FE
FEE0 1F01 FEF1 0E01	E001 1FFE F101 0EFE	E0FE 1F01 F1FE 0E01	FE1F 1FFE FE0E 0EFE
FEE0 011F FEF1 010E	1FFE 01E0 0EFE 01F1	E0FE 011F F1FE 010E	01FE 1FE0 01FE 0EF1
E0E0 1F1F F1F1 0E0E	1FE0 01FE 0EF1 01FE	FEFE 1F1F FEFE 0E0E	01E0 1FFE 01F1 0EFE
FE1F E001 FE0E F101	0101 E0E0 0101 F1F1	E01F FE01 F10E FE01	1F1F E0E0 0E0E F1F1
FE01 E01F FE01 F10E	1F01 FEE0 0E01 FEF1	E001 FE1F F101 FE0E	011F FEE0 010E FEF1
01E0 E001 01F1 F101	1F01 E0FE 0E01 F1FE	1FFE E001 0EFE F001	011F E0FE 010E F1FE
1FE0 FE01 0EF1 FE01	0101 FEFE 0101 FEFE	01FE FE01 01FE FE01	1F1F FEFE 0E0E FEFE
1FE0 E01F 0EF1 F10E	FEFE E0E0 FEFE F1F1	01FE E01F 01FE F10E	E0FE FEE0 F1FE FEF1
01E0 FE1F 01F1 FE0E	FEE0 E0FE FEF1 F1FE	1FFE FE1F 0EFE FE0E	E0E0 FEFE F1F1 FEFE

Симулација на возможниот слаб клуч E01F FE01 F10E FE01

xlii Симулација на возможен слаб клуч E01F FE01 F10E FE01



Ајде да пробаме ја енкриптираме пораката “ Vozmozen ” користејќи го возможниот слаб клуч E01F FE01 F10E FE01. Резултатот од енкрипцијата е “ $b?R?4o ”.

xliiiРундовски клучеви генерирани од E01F FE01 F10E FE01

Round	Round Key
1	011011101010110000011010100100100110010011111010
2	100100010101001111100101110100010111110101000111
3	100100010101001111100101001011101000001010111000
4	100100010101001111100101110100010111110101000111
5	100100010101001111100101001011101000001010111000
6	100100010101001111100101110100010111110101000111
7	100100010101001111100101001011101000001010111000
8	100100010101001111100101110100010111110101000111
9	011011101010110000011010011011011001101100000101
10	011011101010110000011010100100100110010011111010
11	011011101010110000011010011011011001101100000101
12	011011101010110000011010100100100110010011111010
13	011011101010110000011010011011011001101100000101
14	011011101010110000011010100100100110010011111010
15	011011101010110000011010011011011001101100000101
➔ 16	100100010101001111100101001011101000001010111000

Може да забележиме дека од клучот E01F FE01 F10E FE01 се генерираат точно 4 рундовски клучеви и секој од нив се повторува по 4 пати.

Дали DES е слаб алгоритам бидејќи за него постојат слаби клучеви?

Постоењето на слаби и полу слаби клучеви во DES не е го прави алгоритмот послаб бидејќи вкупниот број на “лоши” клучеви во DES е 64 (4 слаби + 12 полу слаби + 48 возможни слаби) и ова е само малечко парче од вкупниот простор на клучеви во DES кој што е 2^{56}. Доколку избереме клуч по случаен избор шансите да се избере некој од овие “лоши” клучеви е многу мала $1/2^{50}$, што е скоро невозможно. Но сепак се препорачува при избор на клучот да се направи проверка за да се избегнат овие 64 клучеви.

Заклучок

Блоковскиот шифрувач DES користи 56 битен клуч од кој што се генерираат 16 различни рундовски клучеви кои што се користат во шеснаесете рунди на алгоритмот. Меѓутоа постојат 4, слаби 12, полуслаби и 48 возможни слаби клучеви кои што доведуваат шифрувачот да се однесува непожелно, односно не генерираат 16 различни рундовски клучеви туку само неколку клучеви кои што се повторуваат.