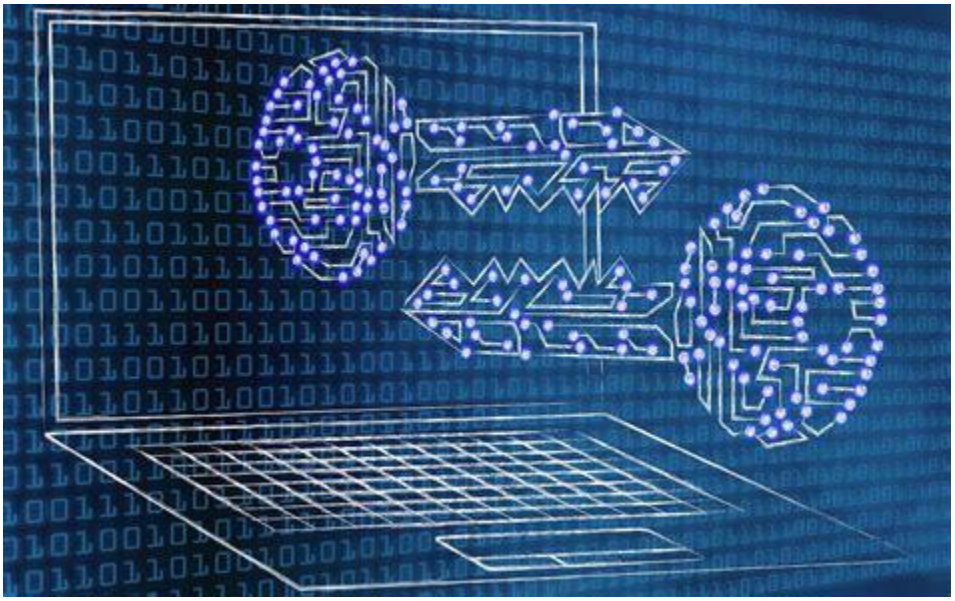
DATA ENCRYPTION ALGORITHM



1827572
School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
3 May 2021

Introduction

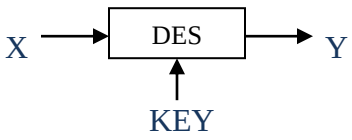
Data Encryption Standard (DES) is a symmetric-key algorithm for the encryption of digital data. DES was developed in the early 1970s by IBM. In 1977 the NSA released a public standard for DES. There was widespread academic scrutiny as many thought the NSA's involvement meant they could have a backdoor. Having a relatively short key length raised suspicion too.

However, it was later found that the design of the S-boxes was designed by the NSA to remove a backdoor they knew about. In today's times DES is insecure due to the short 56-bit key size and since the late 1990s DES has publicly been broken. There are some analytical results which shows theoretical weaknesses in the cipher. DES has since been removed from the National Institute of Standards and Technology.

Triple DES and AES has since replaced the old DES encryption method. The use of encryption in current times is crucial in keeping information safe as cyberattacks have become increasingly effective. Triple DES for example is used across the banking industry and even to the point where the new digital German passports are encrypted using Triple DES.

The Basics of DES

DES is a symmetrical block cipher used in cryptography.



With a stream cipher you encrypt one byte at a time however, with block ciphers you encrypt a block of data at a time. In DES 8 bytes (64 bits) are encrypted at a time. Two atomic operations occur during DES.

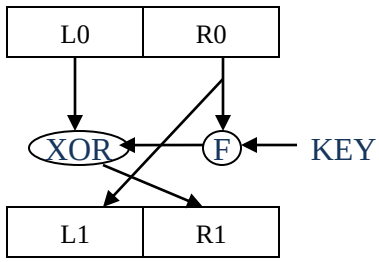
a) Confusion, the relationship between plain and ciphertext is obscured (e.g., substitution table).

b) Diffusion, the influence of each plaintext bit is spread over many ciphertext bits (e.g., permutation table).

When you combine confusion and diffusion together many times, you build a strong block cipher, namely a 'product cipher'. If you have, x input into a block cipher with y output, a bad block cipher is when one bit flip in x only causes one bit flip in y. In a good block cipher when you flip one bit in the input x, the output y has many bit flips.

Feistel Network

Many of the ciphers in use today are Feistel ciphers, but not all. The Feistel Network is the implementation of encryption over many rounds, and for DES there is 16 rounds. This would be round 1's framework:



The main framework of how DES works is that is implements 16 rounds of the above configuration. The 64-bit input gets split into two halves of size 32-bit, L0 and R0. R0 and the KEY gets inputted into the Feistel function, F, the result of that gets XOR with

L0 which then gets stored as the next round's right-hand side, in the case of round 1, R1. L1 is simply the same as R0.

DES

The process of encrypting text using DES has many steps, but a lot of them is just repetition which secures the encrypted text.

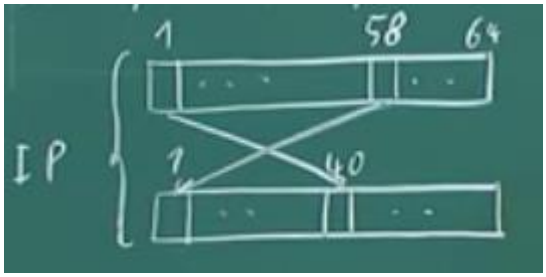
Initial and Final Permutation

Initial and Final Permutation is a simple bit permutation that occurs before DES enters the Feistel network and after all the rounds have been implemented.

Table: 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

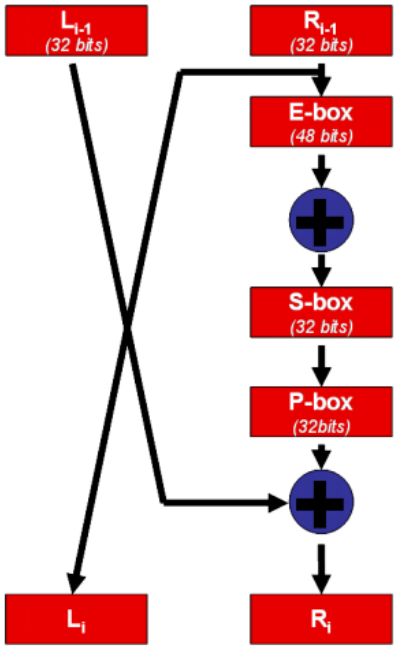
Bit position 58 now becomes position 1 and 1 becomes position 40, etc.



And the final permutation is simply the inverse of IP, where bit position 40 becomes position 1, etc.

Interestingly, IP does not add to the security at all, the only reason it is found in DES is because of electrical engineering reasons as the developers had problems getting the data in and out of the chip. It was not related to cryptography but rather technical reasons and with the development of DES cryptography it just became apart of the specifications, possibly unintentionally.

Details of the F-function



Expansion Permutation (E-box)

The subkey for each round, explained in detail later, is of size 48 bits. The E-box expansion changes the right halve from 32-bit to 48-bit which allows it to get XOR with the subkey. It achieves that by creating duplicates of certain bits in the order specified in the table below.

Table: E-box expansion

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

S-Box Substitution

Performs substitution and compression of the 48-bit output from the XOR mentioned at the E-box to 32-bit which then gets inputted into the P-Box. S-Box substitution introduces confusion and is arguably the most important section in ensuring the strength of DES. The 48-bit input into the S-Box substitution gets split into 8 different 6-bit boxes. Each box goes through an individual S-Box substitution and is a 4-bit output. Here is an example of S-Box 1:

Table: S-Box 1

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

This is 4 rows, 16 column table. The index position chosen from this table is determined with the value of the 6-bit input to it. The first and last bit of the 6-bit input (2 bits) selects the row (2^2=4, min of 0, max of 4). The middle 4 bits of the 6-bit inputted to the S-box determines the column (2^4=16, min of 0, max of 16). Therefore, you have a row and column selector. They select the value from the S-Box and that decimal value gets converted into 4-bit binary form. That procedure gets repeated for all 8 S-Boxes and thereafter you have 8x4=32-bit output.

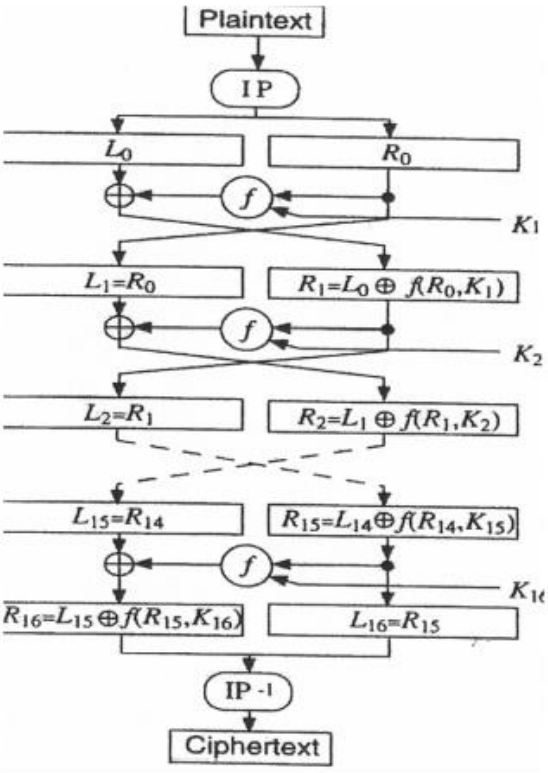
P-Box Permutation

Performs 1-1 bit mapping permutation. Like the IP permutation explained above, it is simply a swapping/moving of bit locations.

Table: P-Box Permutation

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

DES – Complete set of rounds



Plain texts get inputted into DES. IP is applied on the 64-bit text. It enters the Feistel network where the process gets repeated for 16 rounds, each round of the Feistel network the F-function gets implemented on the prior left and right halve 32-bit sets from the previous round along with that rounds subkey. At the end of the 16th round, the left and right halve swap before joining into one 64-bit block which goes through the final permutation (inverse of IP).

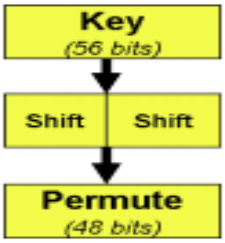
KEY SCHEDULE

The key is inputted in 64-bit size. There are parity bits within the key that have no effect on the encryption/decryption of DES, and they are therefore removed. The

remaining 56-bit key is split into two halves (each 28-bit length).

Dependent on which round of DES is being implemented at that time, each subkey goes through a left-shift or two. Rounds 1,2,9 and 16 only have one left shift on each halve whereas the rest of the rounds have two left shifts on both the left and right half.

After the left shift of that round has occurred the key joins together (56-bit) and then goes through a compression permutation (48-bit) which is then the input in the Feistel function. It is important to note that the left and right half after that rounds shift is the input to the next rounds key schedule.



Encryption vs Decryption

The only difference between the encryption process that I have explained so far and decryption is to do with the use of the key. If you have the key and left and right halve you can easily decrypt the ciphertext using the exact same components and procedure as before. Relating to Feistel Network explanation above, L1 is simply R0, so If you have L1, which is R0 and the key you can simply work backwards with the same function to solve and decrypt the cipher text.

Results

The results outputted by my DES code is correct as I tested and compared it to already worked out examples that can be found in the references [2]. The output of the encryption is used as the input of the decryption and the values match which means both method work.

Conclusions

DES is the most studied cipher in the world and has helped progress cryptography greatly in the past. DES will likely live on in most systems for the foreseeable future, in the form of Triple-DES. When DES is broken down into parts, there are very few complex implementations in itself, but rather the large number of differing operations combing over multiple rounds in order to encrypt the data. The combination of confusion and diffusion of multiple stages allows for this algorithm to have been the top standard of encryption for decades.

References

[1] Introduction to Cryptography by Christof Paar (2014). *Lecture 5: Data Encryption Standard (DES): Encryption by Christof Paar*. *YouTube*. Available at: https://www.youtube.com/watch?v=kPBJIhpcZgE&t=4870s [Accessed 3 May 2021].

[2] Tu-berlin.de. (2021). *The DES Algorithm Illustrated*. [online] Available at: http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm [Accessed 3 May 2021].

[3] Fe.up.pt. (2014). *How does DES works*. [online] Available at: https://paginas.fe.up.pt/~ei10109/ca/des.html [Accessed 3 May 2021].