Encryption
DES

Dr.Talal Alkharobi

The Data Encryption Standard (DES)

2

- A block cipher selected as an official Federal Information Processing Standard (FIPS) for the United States in 1976, and which has subsequently enjoyed widespread use internationally.
- The algorithm was initially controversial, with classified design elements, a relatively short key length, and suspicions about a National Security Agency (NSA) backdoor.
- DES consequently came under intense academic scrutiny, and motivated the modern understanding of block ciphers and their cryptanalysis.

The Data Encryption Standard (DES)

3

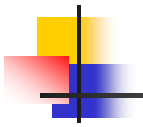
- DES is now considered to be insecure for many applications.
- This is chiefly due to the 56-bit key size being too small; DES keys have been broken in less than 24 hours.
- There are also some analytical results which demonstrate theoretical weaknesses in the cipher, although they are infeasible to mount in practice.
- The algorithm is believed to be practically secure in the form of Triple DES, although there are theoretical attacks.
- In recent years, the cipher has been superseded by the Advanced Encryption Standard (AES).

4

DES

- DES is a revised version of Lucifer; the first civilian block cipher, developed at IBM in the 1970s.
- DES was adopted as a US government FIPS standard, the Data Encryption Standard (DES).
- Chosen by the US National Bureau of Standards (NBS) after a public invitation for submissions and some internal changes by NBS (and, potentially, the NSA).
- DES was publicly released in 1976 and has been widely used.
- DES prompted a large amount of other work and publications in cryptography and cryptanalysis in the open community and it inspired many new cipher designs.

5



DES

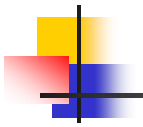
- DES was designed, among other techniques, to resist differential and linear cryptanalysis attacks
- Some attacks were known to the NSA and IBM, though unknown publicly until rediscovered again and published in late 1980s.
- Mitsuru Matsui showed some cryptanalysis techniques to DES.
- A special purpose machine designed to break DES was demonstrated in 1998 by the Electronic Frontier Foundation.
- On 19 May 2005, FIPS 46-3 was officially withdrawn, but NIST has approved Triple DES through the year 2030 for sensitive government information.

6

DES description

- Block size is 64 bits.
- The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm.
- Eight bits are used solely for checking parity, and are thereafter discarded.
- Hence the effective key length is 56 bits, and it is usually quoted as such.

7



DES Overall structure

- Initial and final permutation, IP and FP,
  - They are inverses: IP "undoes" the action of FP, and vice versa
  - IP and FP have almost no cryptographic significance, but were apparently included in order to facilitate loading blocks in and out of mid-1970s hardware, as well as to make DES run slower in software.
- 16 identical stages (rounds) of processing.
- Before the main rounds, the block is divided into two 32-bit halves and processed alternately; this criss-crossing is known as the Feistel scheme.

8

DES Overall structure

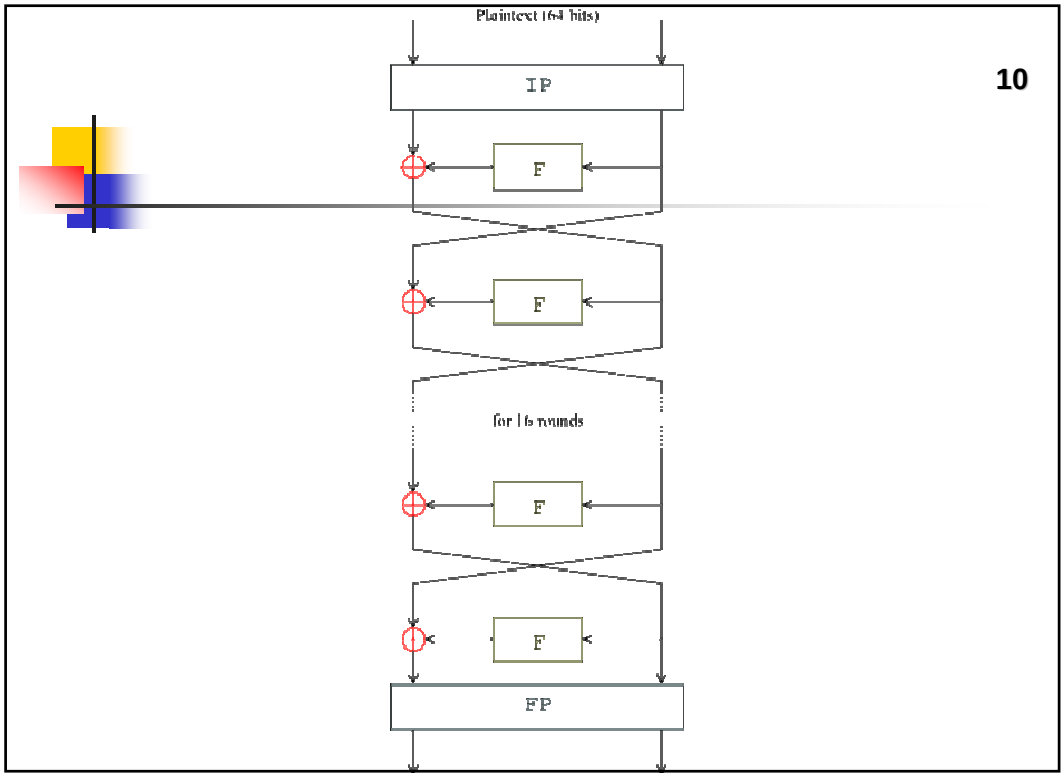
- The Feistel structure ensures that decryption and encryption are very similar processes — the only difference is that the subkeys are applied in the reverse order when decrypting.
- This greatly simplifies implementation, particularly in hardware, as there is no need for separate encryption and decryption algorithms.
- The red symbol denotes the exclusive-OR (XOR) operation. The F-function scrambles half a block together with some of the key.
- The output from the F-function is then combined with the other half of the block, and the halves are swapped before the next round.

9

DES Overall structure

- After the final round, the halves are not swapped; this is a feature of the Feistel structure which makes encryption and decryption similar processes.

10



11

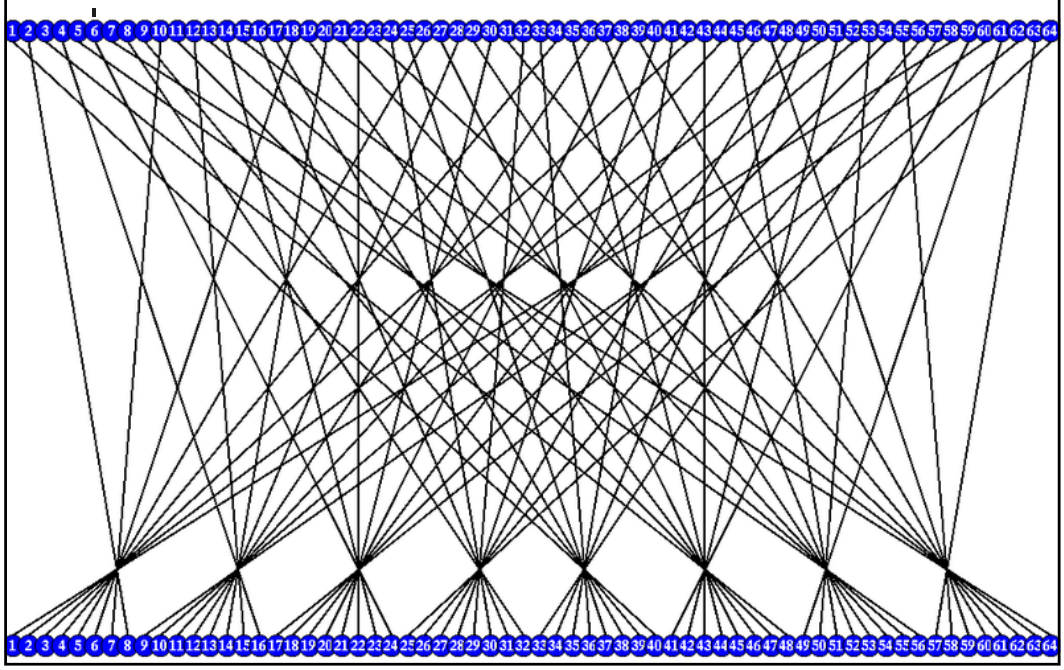
Initial permutation

- 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

- This table specifies the input permutation on a 64-bit block.
- The meaning is as follows:
- bit 1 of the out = bit 58 input
- bit 2 of the out = bit 50 input
- bit 64 of the out = bit 7 input
- and so on,

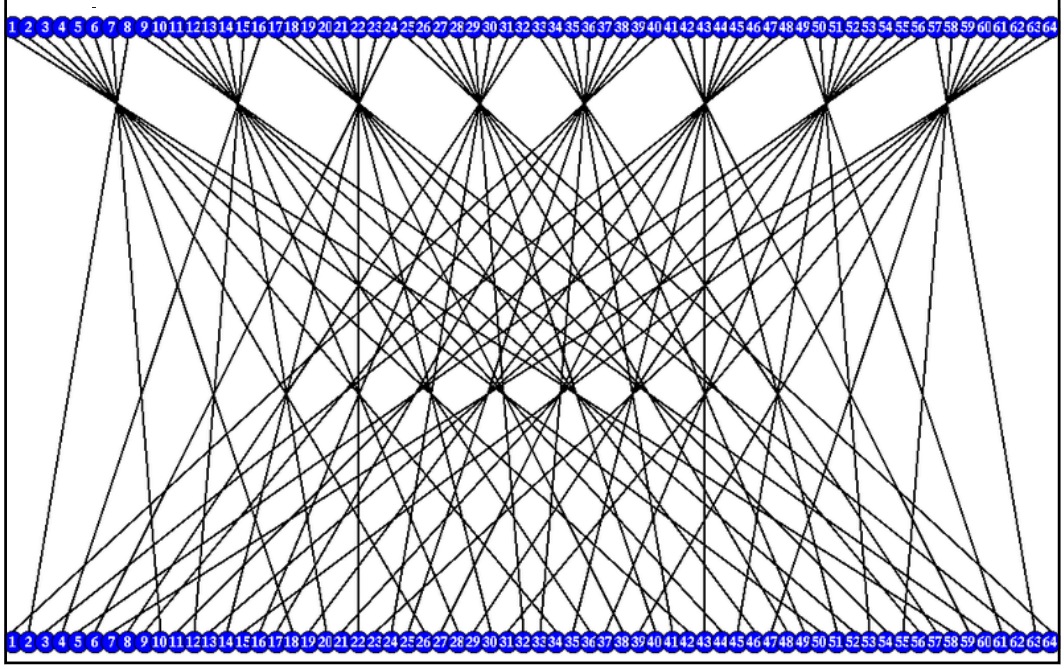
IP: Initial permutation

12



FP: Final permutation

13



Final permutation

14

- 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

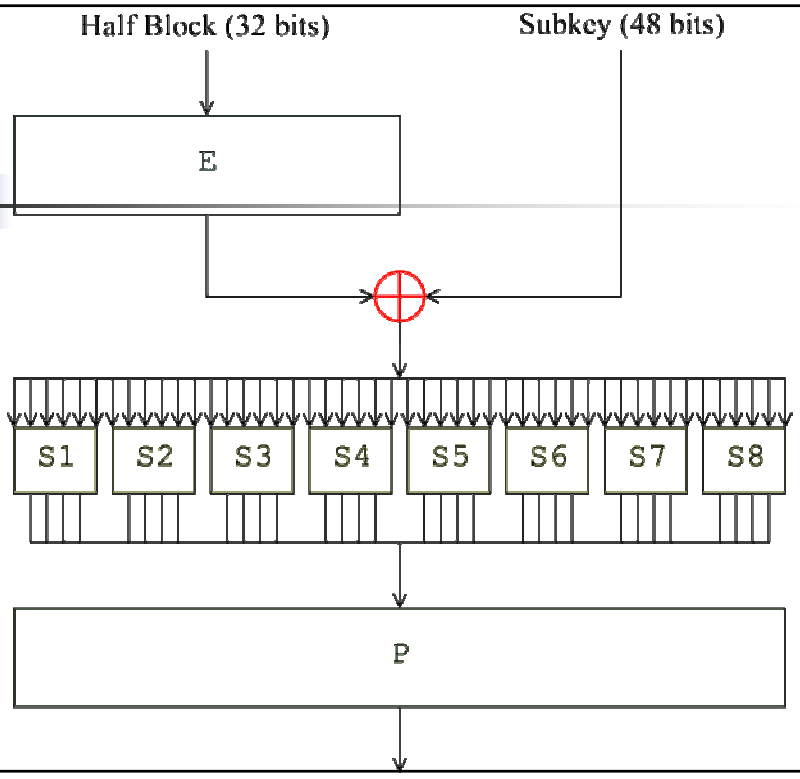
- The final permutation is the inverse of the initial permutation; the table is interpreted similarly.

15

The F-function

- Operates on half a block (32 bits) at a time and consists of four stages:
  - Expansion
  - Key mixing
  - Substitution
  - Permutation

16

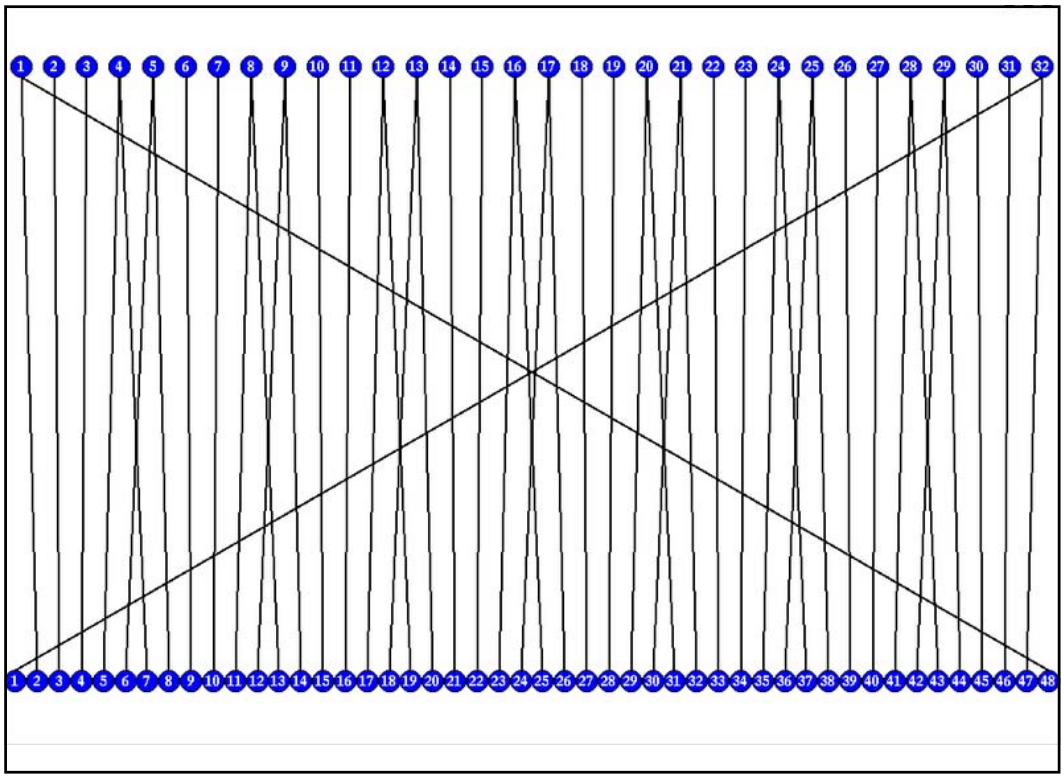


17

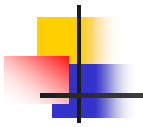
The F-function
Expansion

- The 32-bit half-block is expanded to 48 bits using the *expansion permutation*, denoted *E* in the diagram, by duplicating some of the bits.
- Note that some bits from the input are duplicated at the output; e.g. the fifth bit of the input is duplicated in both the sixth and eighth bit of the output.
- Thus, the 32-bit half-block is expanded to 48 bits.

- 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



19



Key mixing

- The result is combined with a subkey using an XOR operation.
- Sixteen 48-bit subkeys — one for each round — are derived from the main key using the key schedule
- Key schedule: Initially, 56 bits of the key are selected from the initial 64 by Permuted Choice 1 (PC-1) — the remaining eight bits are either discarded or used as parity check bits.
- The 56 bits are then divided into two 28-bit halves; each half is thereafter treated separately.
- In successive rounds, both halves are rotated left by one or two bits (specified for each round),

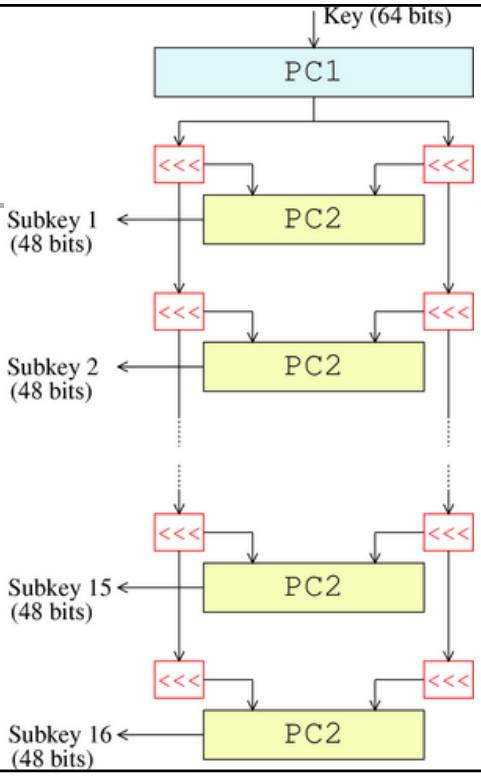
20

Key mixing

- Then, 48-bit subkey is selected by Permuted Choice 2 (PC-2) - 24 bits from the left half, and 24 from the right.
- The rotations (denoted by "<<<" in the diagram) mean that a different set of bits is used in each subkey; each bit is used in approximately 14 out of the 16 subkeys.
- The key schedule for decryption is similar — it must generate the keys in the reverse order.
- Hence the rotations are to the right, rather than the left

Key schedule

21

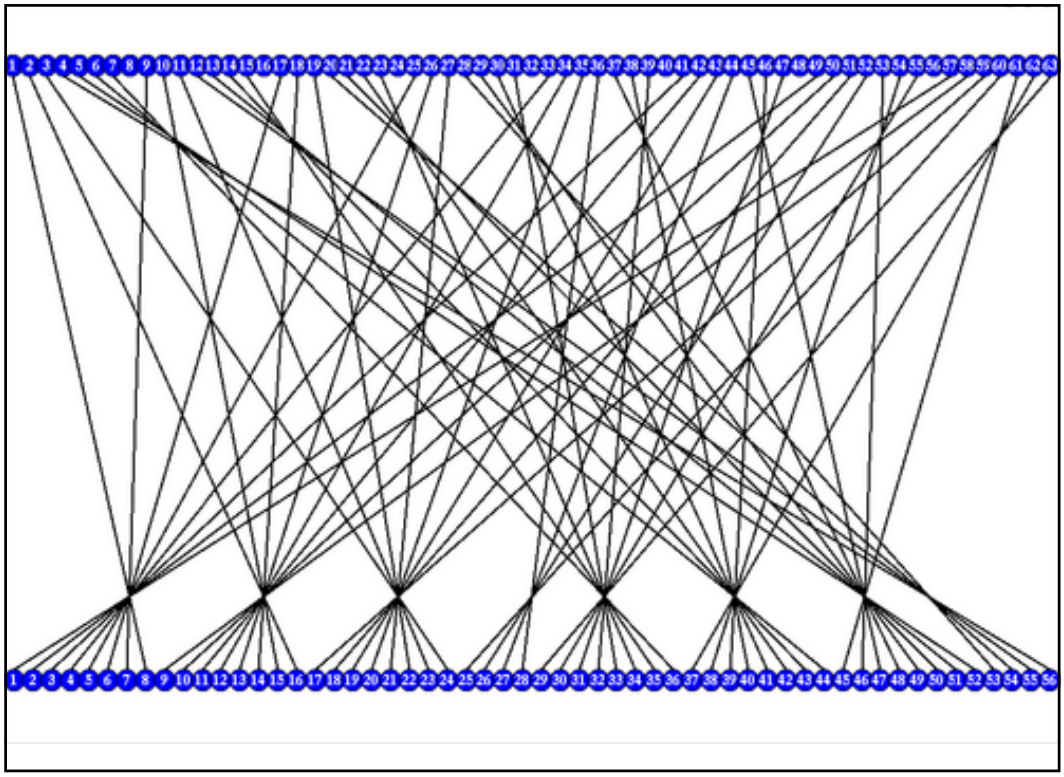


Permuted choice 1 (PC-1)

22

- Left
- 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
- Right
- 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

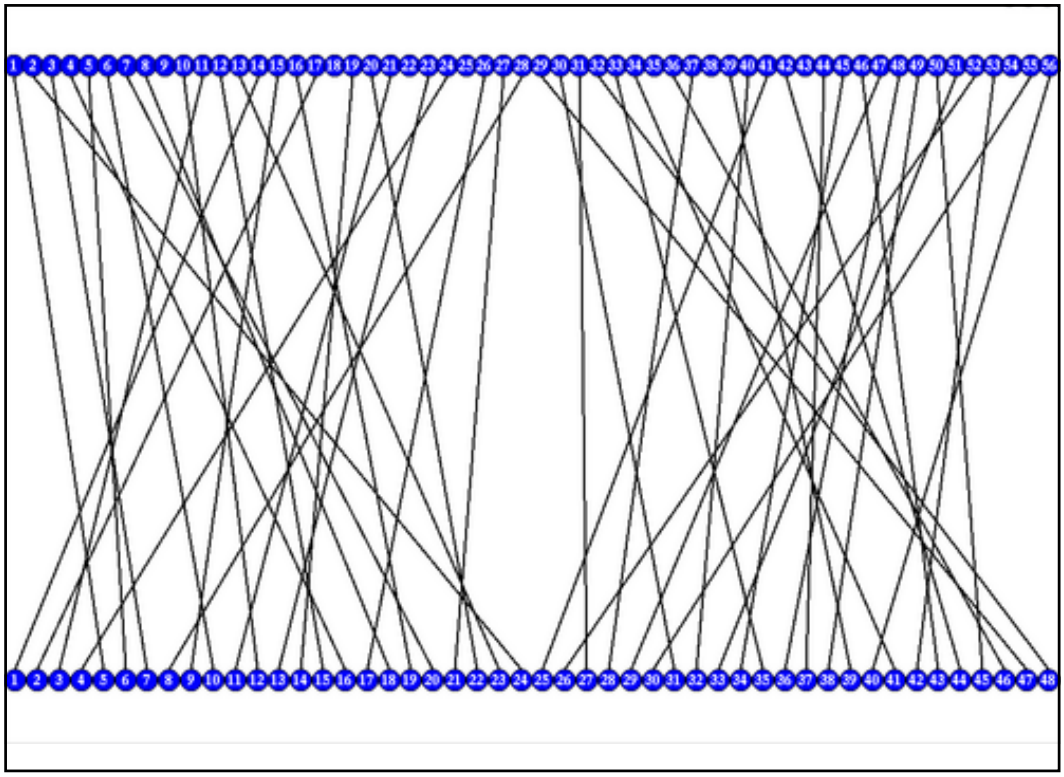
- The "Left" and "Right" halves of the table show which bits from the input key form the left and right sections of the key schedule state.
- Note that only 56 bits of the 64 bits of the input are selected; the remaining eight were specified for use as parity bits.



Permuted choice 2 (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

- This permutation selects the 48-bit subkey for each round from the 56-bit key-schedule state.



26

Substitution

- After mixing in the subkey, the block is divided into eight 6-bit pieces before processing by the S-boxes, or substitution boxes.
- Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table.
- The S-boxes provide the core of the security of DES — without them, the cipher would be linear, and trivially breakable.

Substitution boxes (S-boxes)

27

- The below tables lists the eight S-boxes used in DES.
- Each S-box replaces a 6-bit input with a 4-bit output.
- Given a 6-bit input, the 4-bit output is found by selecting the row using the outer two bits, and the column using the inner four bits. For example, an input "011011" has outer bits "01" and inner bits "1101"; the corresponding output for S-box S5 would be "1001".

28

S-Box

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0010	1100	0100	0001	0111	1100	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1100	0011	1001	1000	0110
10	0100	0010	0001	1011	1100	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1100	0100	0101	0011

Substitution boxes (S-boxes)

29

- 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

Substitution boxes (S-boxes)

30

- 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

Substitution boxes (S-boxes)

31

- 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
```

Substitution boxes (S-boxes)

32

- 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

Substitution boxes (S-boxes)

33

- 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

Substitution boxes (S-boxes)

34

- 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

Substitution boxes (S-boxes)

35

- 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

Substitution boxes (S-boxes)

36

- 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

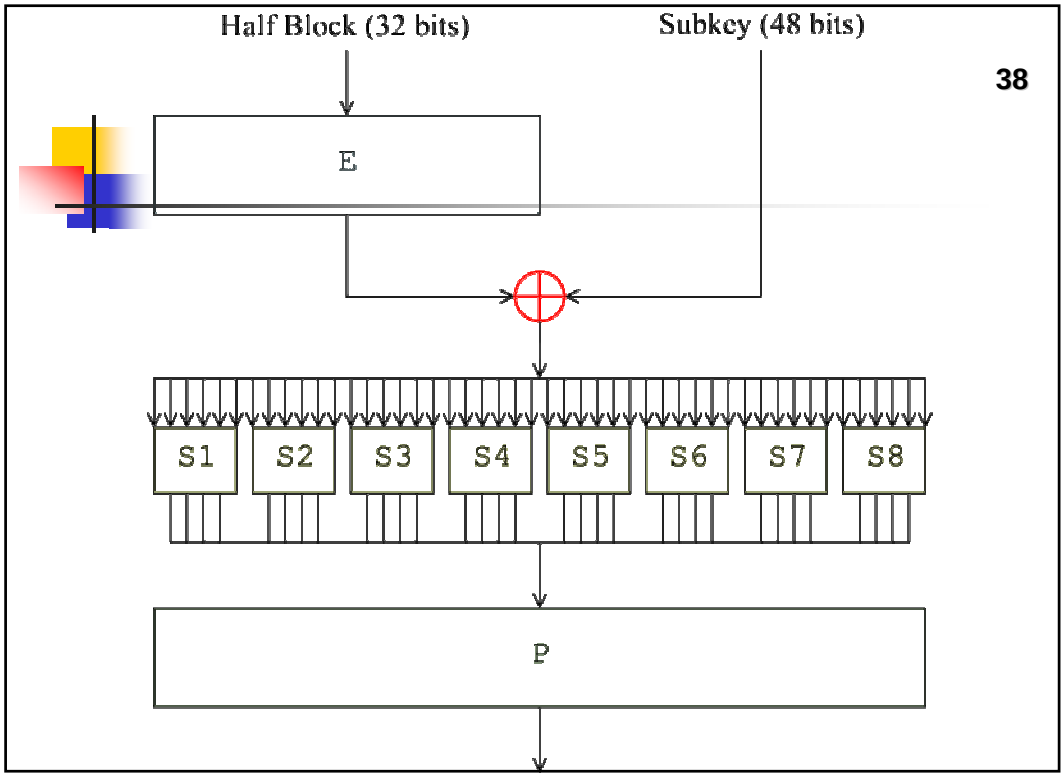
37

Permutation

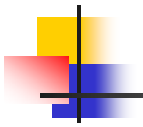
- 16 7 20 21
- 29 12 28 17
- 1 15 23 26
- 5 18 31 10
- 2 8 24 14
- 32 27 3 9
- 19 13 30 6
- 22 11 4 25

- Finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation

38



39



confusion and diffusion

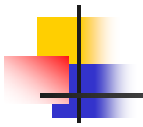
- The alternation of substitution from the S-boxes, and permutation of bits from the P-box and E-expansion provides so-called "confusion and diffusion" respectively, a concept identified by Claude Shannon in the 1940s as a necessary condition for a secure yet practical cipher.

40

Attacks on DES

- Although more information has been published on the cryptanalysis of DES than any other block cipher, the most practical attack to date is still a brute force approach.
- Various minor cryptanalytic properties are known, and three theoretical attacks are possible which, while having a theoretical complexity less than a brute force attack, require an unrealistic amount of known or chosen plaintext to carry out, and are not a concern in practice.

41



Brute Force Attack

- For any cipher, the most basic method of attack is brute force — trying every possible key in turn.
- The length of the key determines the number of possible keys, and hence the feasibility of this approach.
- For DES, questions were raised about the adequacy of its key size early on, even before it was adopted as a standard, and it was the small key size, rather than theoretical cryptanalysis, which dictated a need for a replacement algorithm.
- It is known that the NSA encouraged, if not persuaded, IBM to reduce the key size from 128 to 64 bits, and from there to 56 bits

42

Brute Force Attack

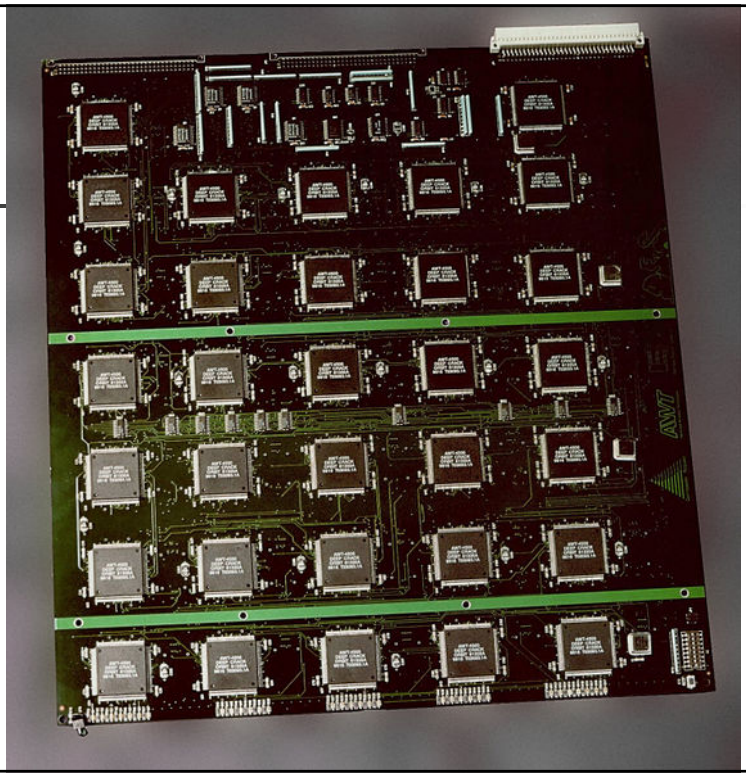
- In academia, various proposals for a DES-cracking machine were advanced.
- In 1977, Diffie and Hellman proposed a machine costing an estimated US$20 million which could find a DES key in a single day.
- By 1993, Wiener had proposed a key-search machine costing US$1 million which would find a key within 7 hours.
- However, none of these early proposals were ever implemented, at least no implementations were publicly acknowledged.

43

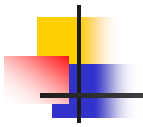
Brute Force Attack

- In 1997, RSA Security sponsored a series of contests, offering a $10,000 prize to the first team that broke a message encrypted with DES for the contest.
- That contest was won by the DESCHALL Project using idle cycles of thousands of computers across the Internet.
- The feasibility of cracking DES quickly was demonstrated in 1998 when a custom DES-cracker was built by the Electronic Frontier Foundation (EFF), a cyberspace civil rights group, at the cost of approximately US$250,000
- DES cracking machine contained 1,536 custom chips and could brute force a DES key in a matter of days

44



45



Brute Force Attacks

- Their motivation was to show that DES was breakable in practice as well as in theory:
- "There are many people who will not believe a truth until they can see it with their own eyes.
- Showing them a physical machine that can crack DES in a few days is the only way to convince some people that they really cannot trust their security to DES."
- The machine brute-forced a key in a little more than 2 days' search;

46

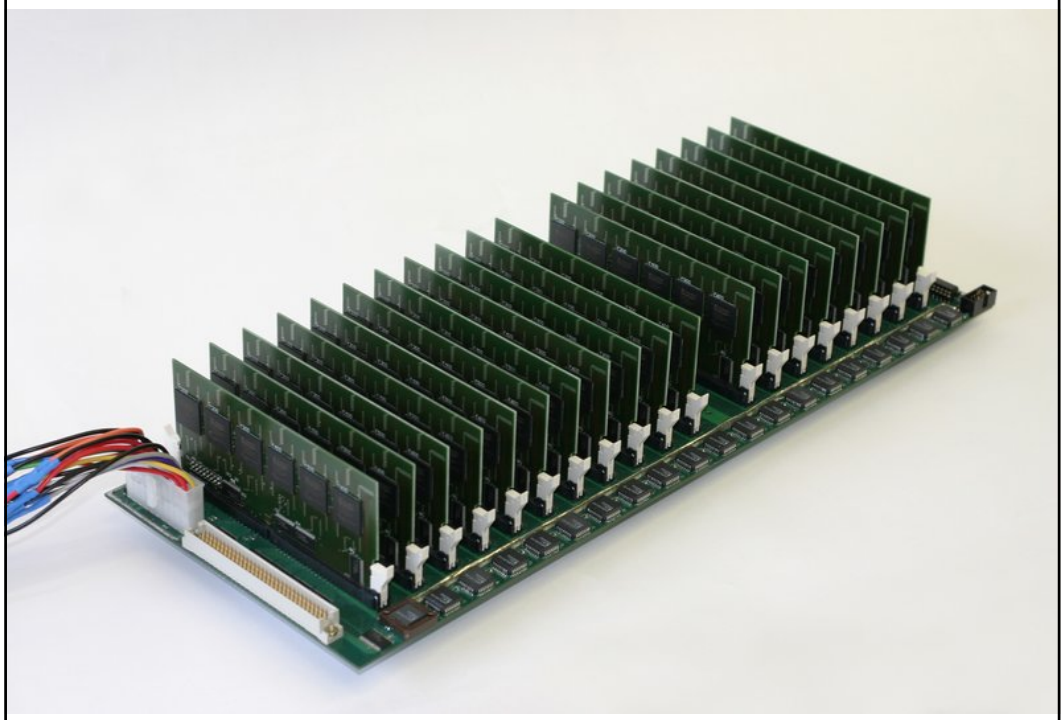
Brute Force Attacks

- The only other confirmed DES cracker was the COPACOBANA machine (Cost-Optimized Parallel COde Breaker) built more recently by teams of the Universities of Bochum and Kiel, both in Germany.
- Unlike the EFF machine, COPACOBANA consist of commercially available, reconfigurable integrated circuits.
- 120 of these FPGAs of type XILINX Spartan3-1000 run in parallel.
- They are grouped in 20 DIMM modules, each containing 6 FPGAs.

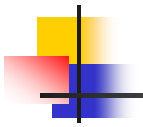
47

Brute Force Attacks

- . The use of reconfigurable hardware makes the machine applicable to other code breaking tasks as well.
- One of the more interesting aspects of COPACOBANA is its cost factor.
- One machine can be built for approximately $10,000. The cost decrease by roughly a factor of 25 over the EFF machine is an impressive example for the continuous improvement of digital hardware.



49



Replacement algorithms

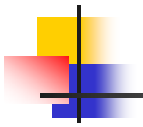
- Concerns about security and the relatively slow operation of DES in software motivated researchers to propose a variety of alternative block cipher designs, which started to appear in the late 1980s and early 1990s;
- Example RC5, Blowfish, IDEA, NewDES, SAFER, CAST5 and FEAL.
- Most of these designs kept the 64-bit block size of DES, and could act as a "drop-in" replacement, although they typically used a 64-bit or 128-bit key.
- In the USSR the GOST 28147-89 algorithm was introduced, with a 64-bit block size and a 256-bit key, which was used in Russia later.

50

Replacement algorithms

- DES itself can be adapted and reused in a more secure scheme.
- Many former DES users now use Triple DES (TDES) which was described and analyzed by one of DES's patentees
- TDES involves applying DES three times with two (2TDES) or three (3TDES) different keys. TDES is regarded as adequately secure, although it is quite slow.
- A less computationally expensive alternative is DES-X, which increases the key size by XORing extra key material before and after DES.

51



Replacement algorithms

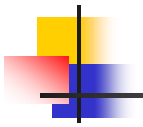
- GDES was a DES variant proposed as a way to speed up encryption, but it was shown to be susceptible to differential cryptanalysis.
- In 2001, after an international competition, NIST selected a new cipher: the Advanced Encryption Standard (AES), as a replacement.
- The algorithm which was selected as the AES was submitted by its designers under the name Rijndael.
- Other finalists in the NIST AES competition included RC6, Serpent, MARS, and Twofish.

52

DES Chronology

- 1973: NBS publishes a first request for a standard encryption algorithm
- 1974: NBS publishes a second request for encryption algorithms
- 1975: DES is published in the Federal Register for comment
- 1976: First workshop on DES
- 1976: Second workshop, discussing mathematical foundation of DES
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46

53



DES Chronology

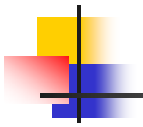
- 1983: DES is reaffirmed for the first time
- 1986: Videocipher II, a TV satellite scrambling system based upon DES begins use by HBO
- 1988: DES is reaffirmed for the second time as FIPS 46-1, superseding FIPS PUB 46
- 1992: Biham and Shamir publish the first theoretical attack with less complexity than brute force: differential cryptanalysis. It requires an unrealistic 2^{47} chosen plaintexts
- 1993: DES is reaffirmed for the third time as FIPS 46-2

54

DES Chronology

- 1994: The first experimental cryptanalysis of DES is performed using linear cryptanalysis (Matsui, 1994).
- 1997: The DESCHALL Project breaks a message encrypted with DES for the first time in public.
- 1998: The EFF's DES cracker (Deep Crack) breaks a DES key in 56 hours.
- 1999: Together, Deep Crack and distributed.net break a DES key in 22 hours and 15 minutes.

55



DES Chronology

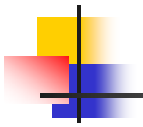
- 1999: DES is reaffirmed for the fourth time as FIPS 46-3, which specifies the preferred use of Triple DES, with single DES permitted only in legacy systems.
- 2001: The Advanced Encryption Standard is published in FIPS 197
- 2002: The AES standard becomes effective
- 2004: The withdrawal of FIPS 46-3 (and a couple of related standards) is proposed in the Federal Register
- 2005: NIST withdraws FIPS 46-3

56

Algorithmes based on DES

- | DES-X | DFC | E2 | FEAL | FROG | G-DES | GOST | Grand Cru | Hasty Pudding Cipher | Hierocrypt | ICE | IDEA | IDEA NXT | Iraqi | Intel Cascade Cipher | KASUMI | KHAZAD | Khufu and Khafre | KN-Cipher | Libelle | LOKI89/91 | LOKI97 | Lucifer | M6 | MacGuffin | Madryga | MAGENTA | MARS | Mercy | MESH | MISTY1 | MMB | MWA | MULTI2 | NewDES | NOEKEON | NUSH | Q | RC2 | RC5 | RC6 | REDOC | Red Pike | S-1 | SAFER | SC2000 | SEED | Serpent | SHACAL | SHARK | Skipjack | SMS4 | Square | TEA | Triple DES | Twofish | UES | Xenon | xmx | XTEA | Zodiac

57



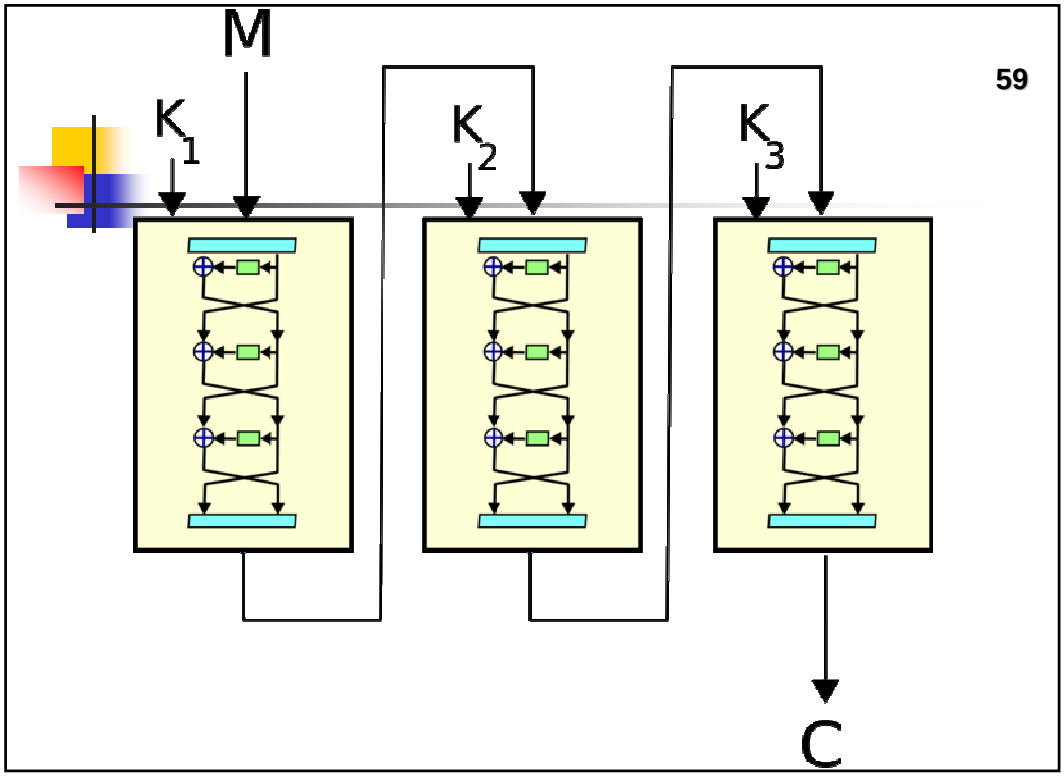
Triple DES

- Triple DES is a block cipher formed from the Data Encryption Standard (DES) cipher by using it three times.
- When it was found that a 56-bit key of DES is not enough to guard against brute force attacks, TDES was chosen as a simple way to enlarge the key space without a need to switch to a new algorithm.
- The use of three steps is essential to prevent meet-in-the-middle attacks that are effective against double DES encryption.
- Note that DES is not a group; if it were one, the TDES construction would be equivalent to a single DES operation and no more secure.

58

Triple DES

- The simplest variant of TDES operates as follows: DES(k3;DES(k2;DES(k1;M))), where M is the message block to be encrypted and k1, k2, and k3 are DES keys.
- This variant is commonly known as EEE because all three DES operations are encryptions.
- In order to simplify interoperability between DES and TDES the middle step is usually replaced with decryption (EDE mode): DES(k3;DES^{-1}(k2;DES(k1;M))) and so a single DES encryption with key k can be represented as TDES-EDE with k1=k2=k3= k.
- The choice of decryption for the middle step does not affect the security of the algorithm

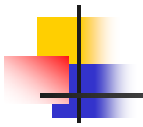


60

3TDES

- TDES with three different keys (3TDES) has a key length of 168 bits: three 56-bit DES keys (with parity bits 3TDES has the total storage length of 192 bits),
- Due to the meet-in-the-middle attack the effective security it provides is only 112 bits.
- A variant, called two-key TDES (2TDES), uses k1 = k3, thus reducing the key size to 112 bits and the storage length to 128 bits.
- As of 2005, the best attack known on 3TDES requires around 2^{32} known plaintexts, 2^{113} steps, 2^{90} single DES encryptions, and 2^{88} memory

61



TDES

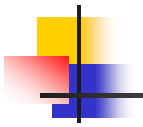
- TDES is slowly disappearing from use, largely replaced by its natural successor, the Advanced Encryption Standard (AES).
- One large-scale exception is within the electronic payments industry, which still uses 2TDES extensively and continues to develop and promulgate standards based upon it (e.g. EMV).
- This guarantees that TDES will remain an active cryptographic standard well into the future.
- By design, DES and therefore TDES, suffer from slow performance in software; on modern processors, AES tends to be around six times faster.

62

TDES

- TDES is better suited to hardware implementations, and indeed where it is still used it tends to be with a hardware implementation (e.g., VPN appliances and the Nextel cellular and data network), but even there AES outperforms it.
- Finally, AES offers markedly higher security margins: a larger block size, potentially longer keys, and as of 2006, no known public cryptanalytic attacks.
- Algorithms based on TDES: Twofish | UES | Xenon | xmx | XTEA | Zodiac.

63



Meet-in-the-middle attack

- Theis attack makes use of a space-time tradeoff.
- Attempts to find a value in each of the range and domain of the composition of two functions such that the forward mapping of one through the first function is the same as the inverse image of the other through the second function -- quite literally meeting in the middle of the composed function.
- It was first developed as an attack on an attempted expansion of a block cipher by Diffie and Hellman in 1977.

64

Meet-in-the-middle attack

- When trying to improve the security of a block cipher, one might get the idea to simply use two independent keys to encrypt the data twice.
- Naively, one might think that this would square the security of the double-encryption scheme.
- Certainly, an exhaustive search of all possible combination of keys would take 2^{2n} attempts if each key is n bits long, compared to the 2^n attempts required for a single key.
- Diffie and Hellman, however, devised a time-memory tradeoff that could break the scheme in only double the time to break the single-encryption scheme

65

Meet-in-the-middle attack

- The attack works by encrypting from one end and decrypting from the other end, thus meeting in the middle.
- Assume the attacker knows a set of plaintext and ciphertext: P&C.
- That is, C = E(E(P,K1),K2) ; K1 and K2 are the two keys.
- The attacker compute E(P,K) for all possible keys K
- compute D(C,K) for each K and compare with the previous results
- If a match is found; the K1 & K2 is discovered.
- To verify; the second set of plaintext and ciphertext can be sued.

66

Meet-in-the-middle attack

- If the keysize is n, this attack uses only 2^{n+1} encryptions (and $O(2^n)$ space)
- In contrast; the naive attack, needs 2^{2n} encryptions (but only O(1) space).