Cryptography and Network Security Chapter 3

Block Ciphers, Stream Ciphers, RC4 Stream cipher, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), IDEA.

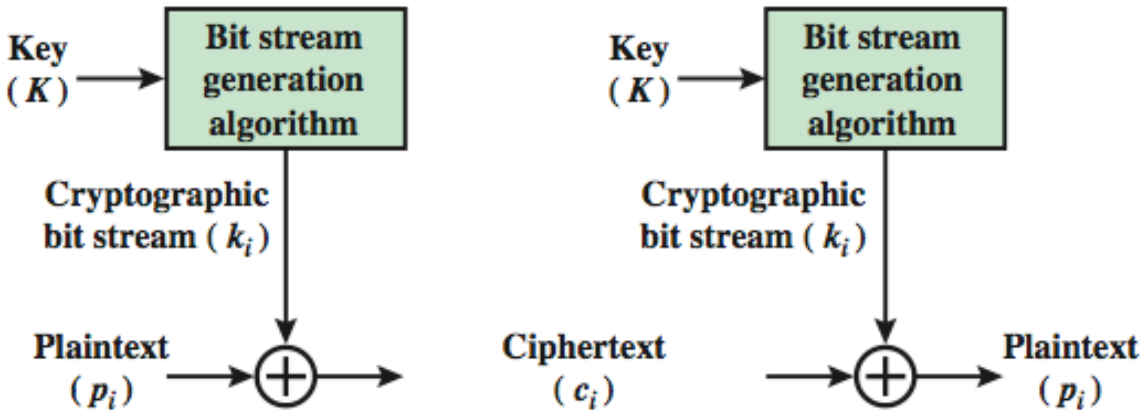
Block vs Stream Ciphers

- block ciphers process messages in blocks, each of which is then en/decrypted
- like a substitution on very big characters
  - 64-bits or more
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers
  - better analyzed
  - broader range of applications

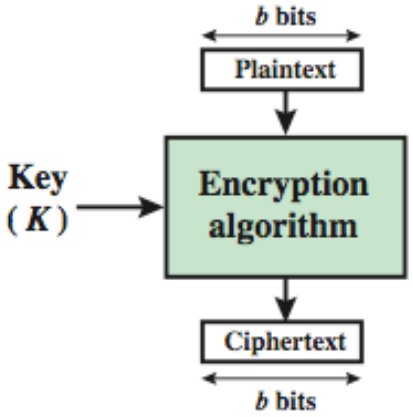
Block vs Stream Ciphers

- Q: What is a block cipher we have already seen?
- A: Playfair cipher. What is its block size?
- A: 2 characters
- Q: What are some stream ciphers we have already seen?
- A: Autokey cipher, Vigenere cipher, Vernam cipher, OneTime Pad (OTP)

Block vs Stream Ciphers



(a) Stream Cipher Using Algorithmic Bit Stream Generator

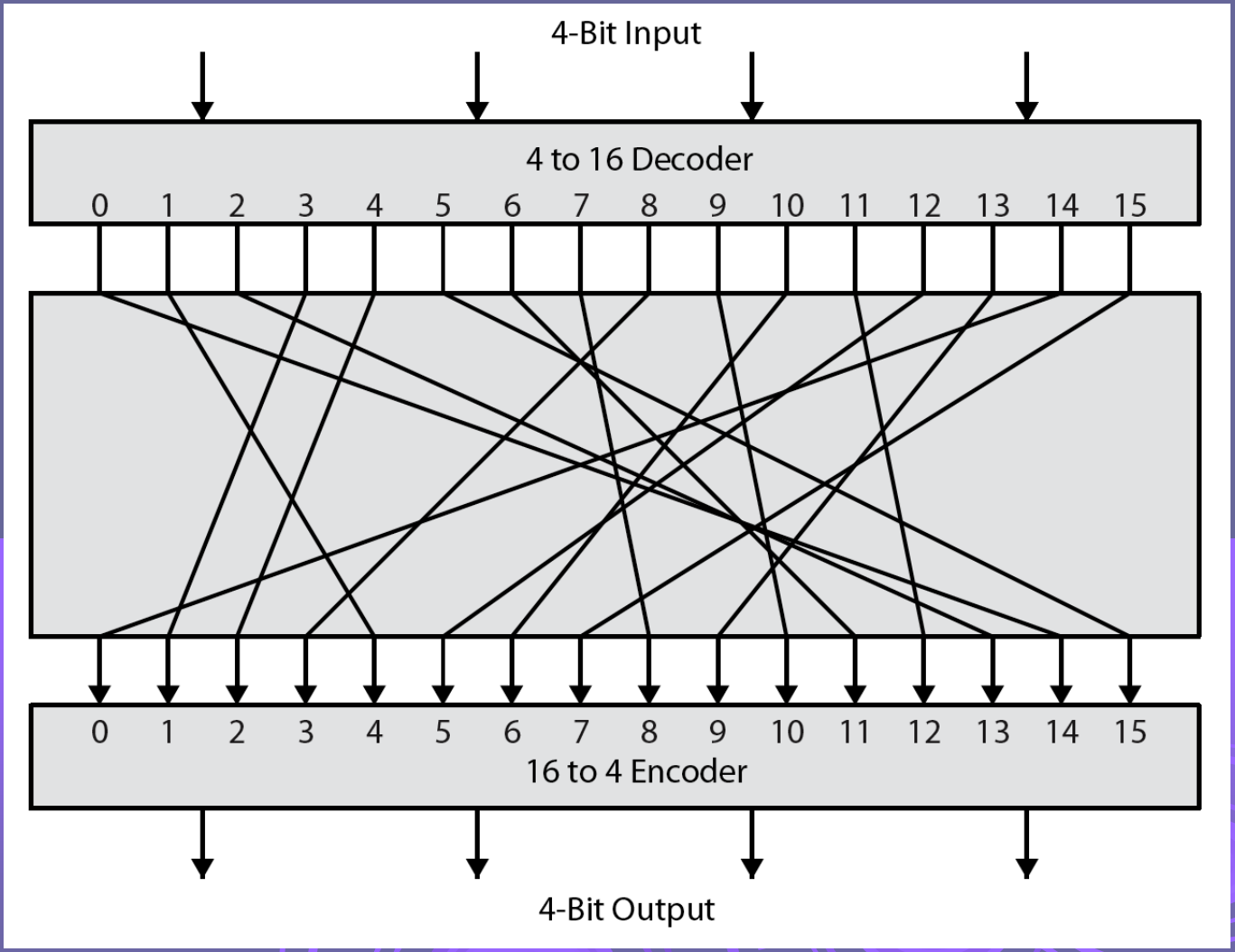


(b) Block Cipher

Block Cipher Principles

- most symmetric block ciphers are based on a **Feistel Cipher Structure**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
- instead create from smaller building blocks
- using idea of a product cipher

Ideal Block Cipher



Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* & *diffusion* of message & key

Confusion and Diffusion

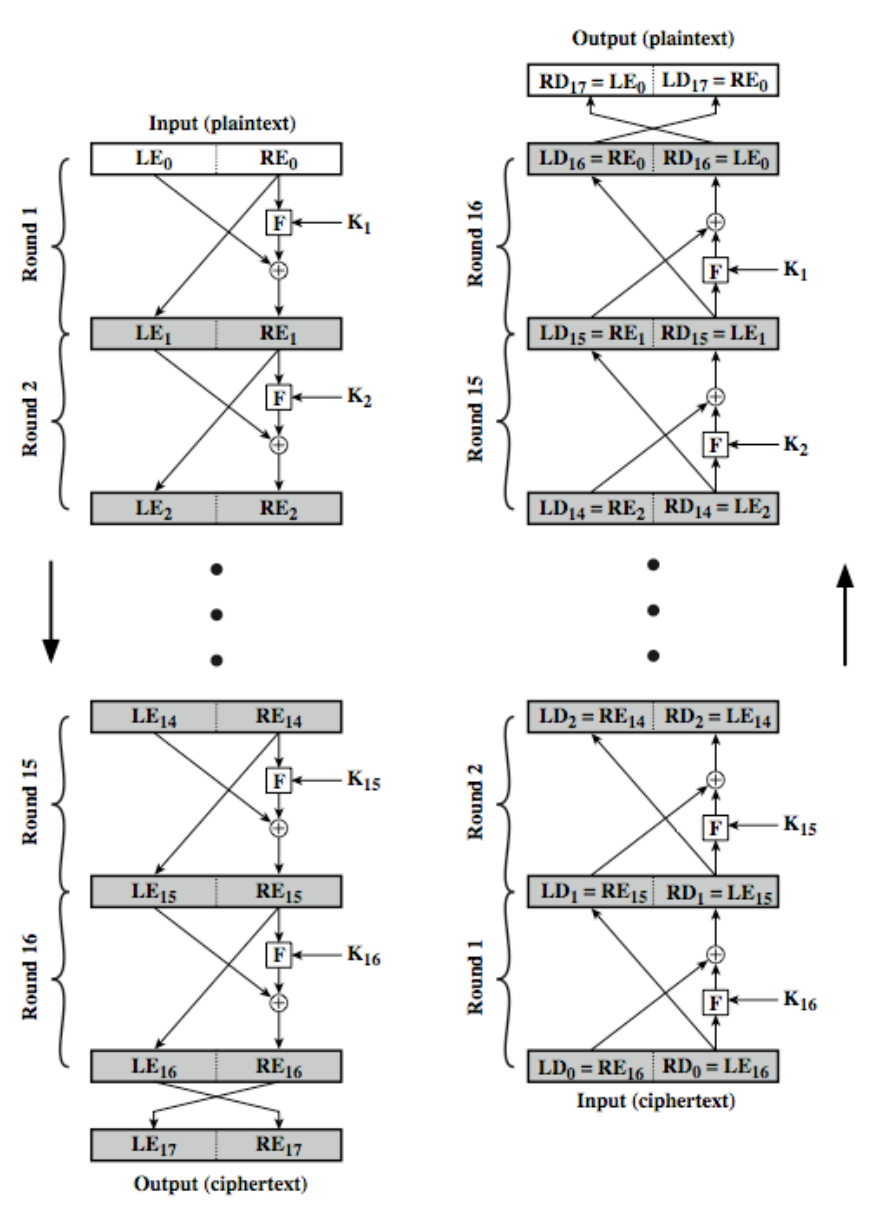
- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S & P elements to obtain:
- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as possible

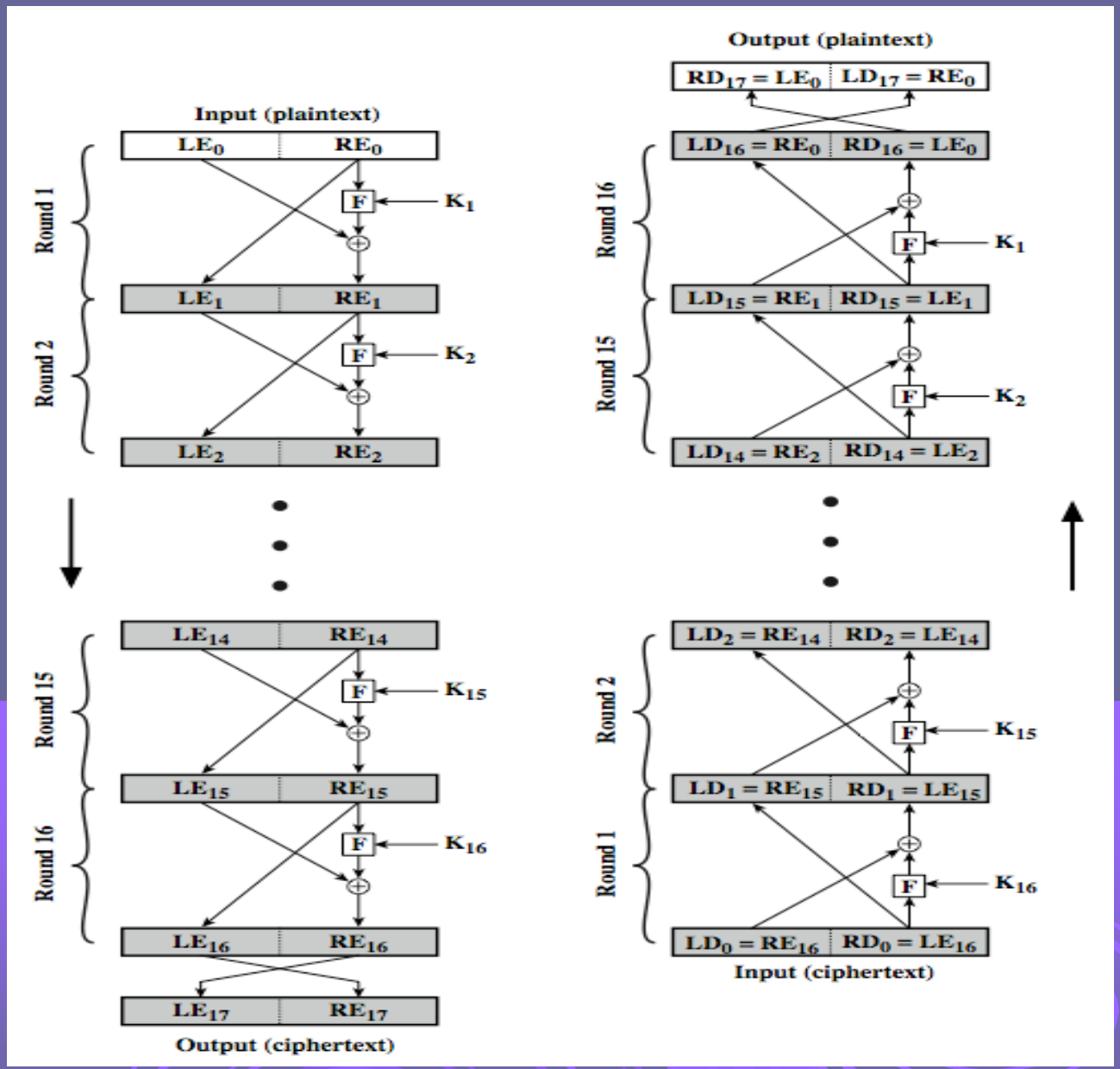
COMPARISON	CONFUSION	DIFFUSION
Basic	Utilized to generate vague cipher texts.	Utilized to generate obscure, plain texts.
Seeks to	Make a relation between statistics of the ciphertext and the value of the encryption key as complicated as possible.	The statistical relationship between the plaintext and ciphertext is made as complicated as possible.
Achieved through	Substitution algorithm	Transposition algorithm
Used by	Block cipher only.	Stream cipher and block cipher
Result in	Increased vagueness	Increased redundancy

Feistel Cipher Structure

- Horst Feistel devised the **Feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon's S-P net concept

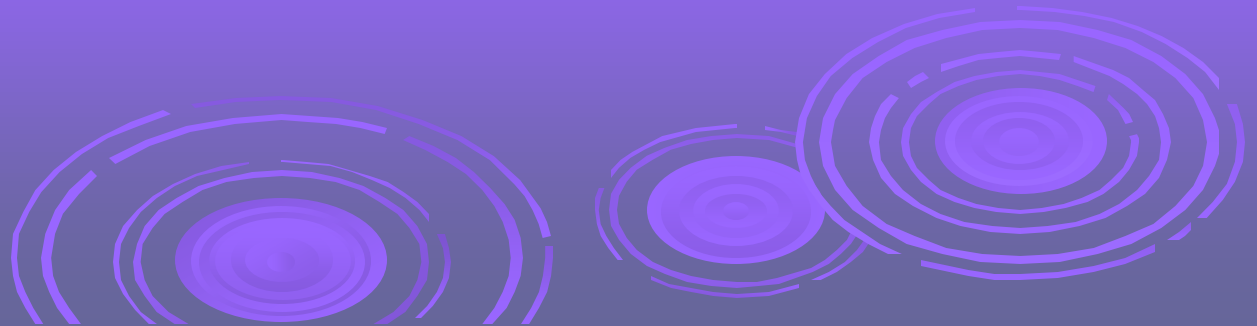
Feistel Cipher Structure




Input (plaintext)
LE_0 RE_0
Round 1
F
K_1
LE_1 RE_1
Round 2
F
K_2
LE_2 RE_2
LE_{14} RE_{14}
Round 15
F
K_{15}
LE_{15} RE_{15}
Round 16
F
K_{16}
LE_{16} RE_{16}
LE_{17} RE_{17}
Output (ciphertext)
Output (plaintext)
$RD_{17} = LE_0$ $LD_{17} = RE_0$
$LD_{16} = RE_0$ $RD_{16} = LE_0$
Round 16
F
K_1
$LD_{15} = RE_1$ $RD_{15} = LE_1$
Round 15
F
K_2
$LD_{14} = RE_2$ $RD_{14} = LE_2$
$LD_2 = RE_{14}$ $RD_2 = LE_{14}$
Round 2
F
K_{15}
$LD_1 = RE_{15}$ $RD_1 = LE_{15}$
Round 1
F
K_{16}
$LD_0 = RE_{16}$ $RD_0 = LE_{16}$
Input (ciphertext)

Feistel Cipher Design Elements

- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease of analysis



Feistel Decryption

- After the last iteration of the encryption process, the two halves of the output are swapped, so that the ciphertext is *RE*16 ‘*LE*16.

- The output of that round is the ciphertext. Now take that ciphertext and use it as inp ut to the same algorithm.

- The input to the first round is *RE*16 ‘*LE*16, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process.

$LE16 = RE15$

$RE16 = LE15 \oplus F(RE15, K16)$

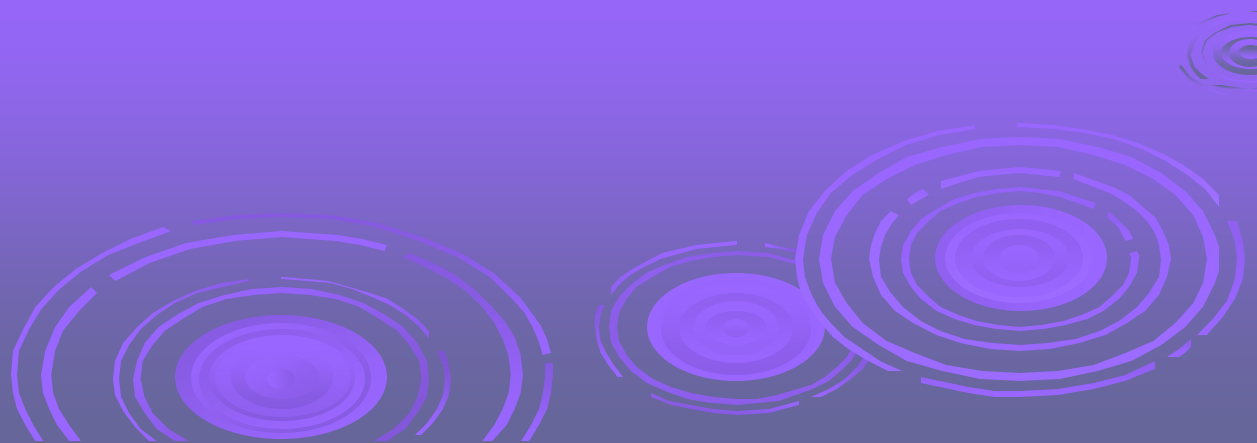
On the decryption side,

$LD1 = RD0 = LE16 = RE15$

$RD1 = LD0 \oplus F(RD0, K16)$

$= RE16 \oplus F(RE15, K16)$

$= [LE15 \oplus F(RE15, K16)] \oplus F(RE15, K16)$



Data Encryption Standard (DES)

- most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security
- Now deprecated due to short key

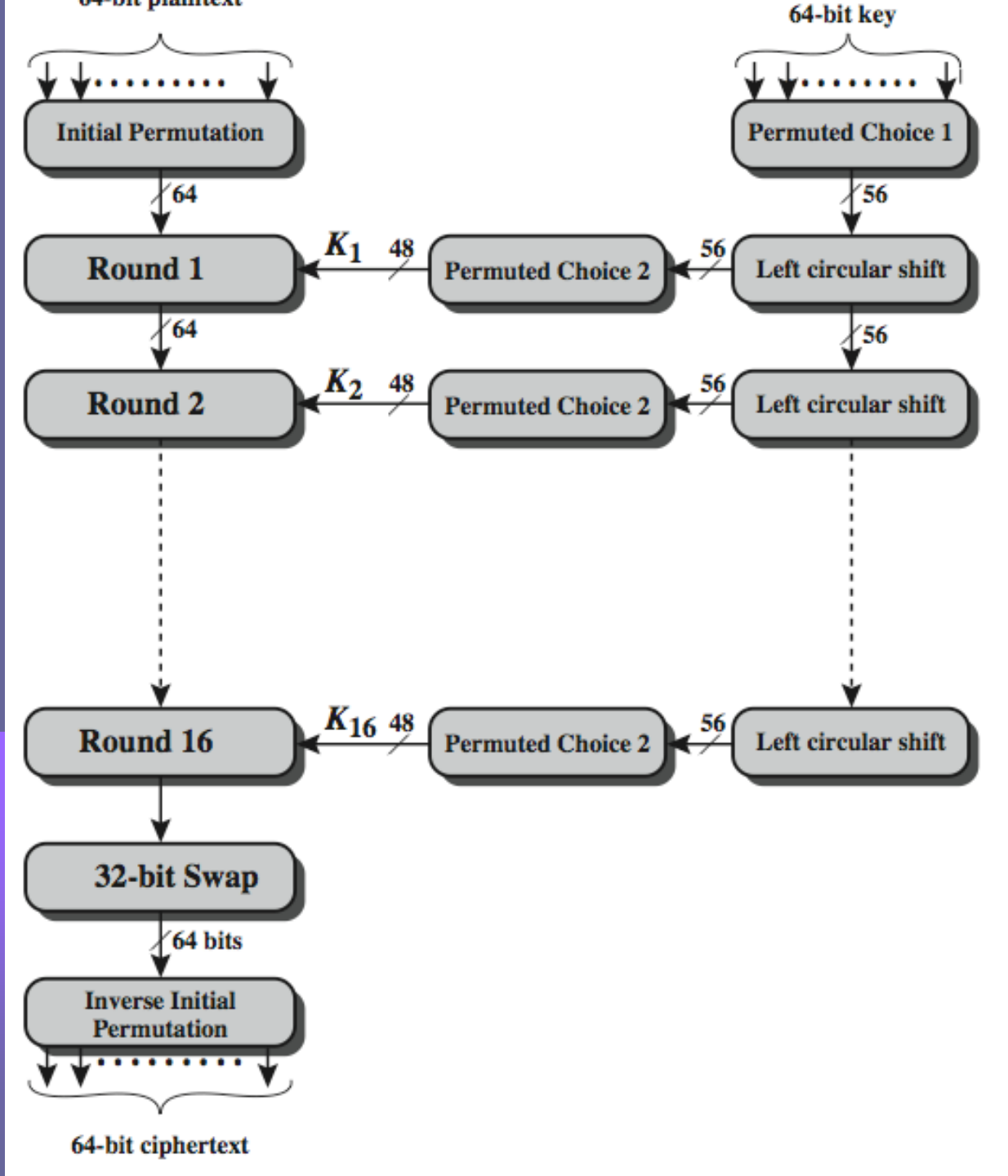
DES History

- IBM developed Lucifer cipher
  - by team led by Feistel in late 60's
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES

DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (vs Lucifer 128-bit)
  - and because design criteria were classified
- subsequent events and public analysis show in fact design was appropriate
- use of DES has flourished
  - especially in financial applications
  - still standardized for legacy application use
  - 3DES still strong (112 bit key)

view


64-bit plaintext
64-bit key
Initial Permutation
Permuted Choice 1
64
56
Round 1
K_1
48
Permuted Choice 2
56
Left circular shift
64
56
Round 2
K_2
48
Permuted Choice 2
56
Left circular shift
Round 16
K_{16}
48
Permuted Choice 2
56
Left circular shift
32-bit Swap
64 bits
Inverse Initial Permutation
64-bit ciphertext

Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- even bits to LH half, odd bits to RH half
- quite regular in structure (easy in h/w)
- no cryptographic value
- example:

```
IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)
```

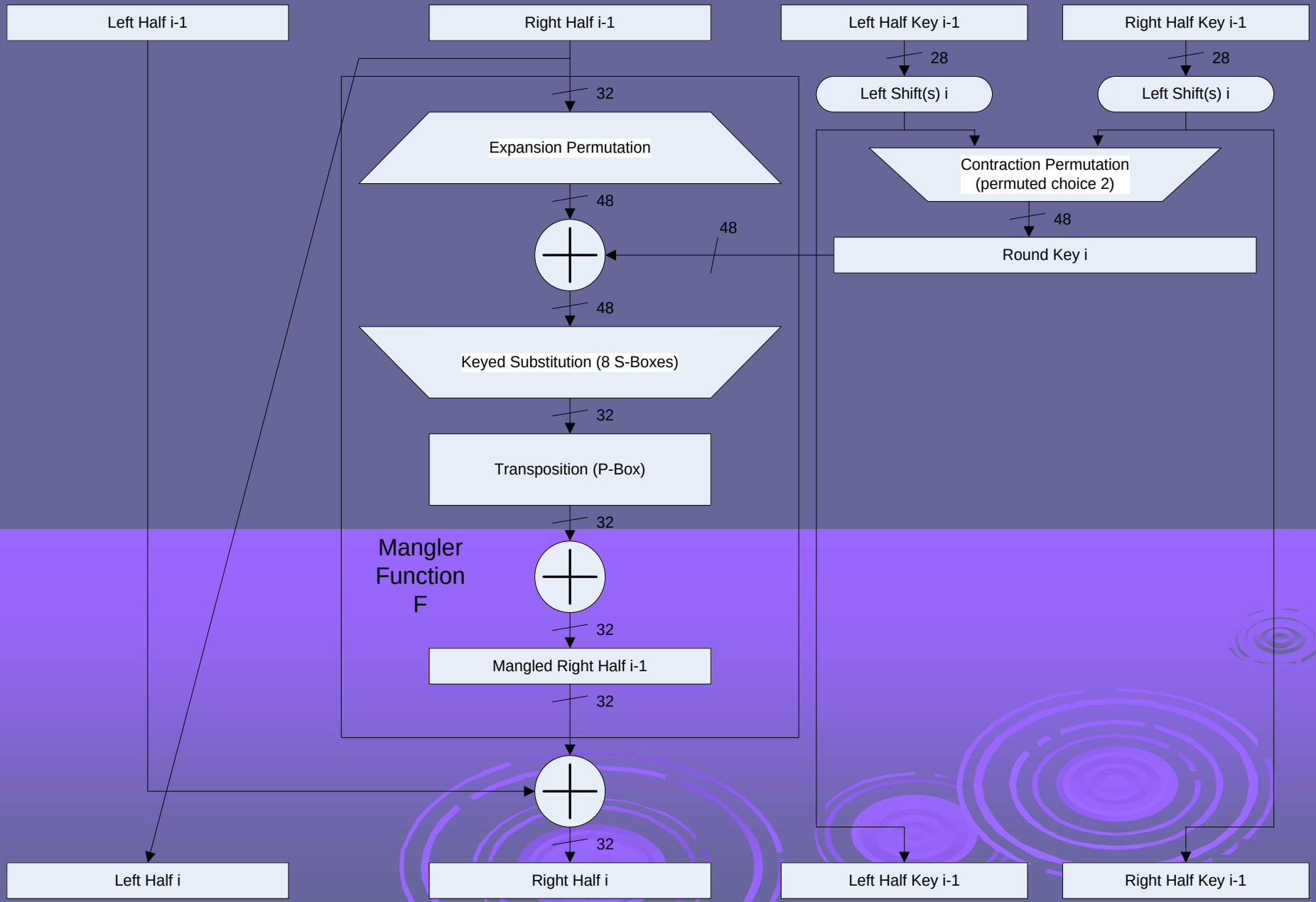
DES Round Structure

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

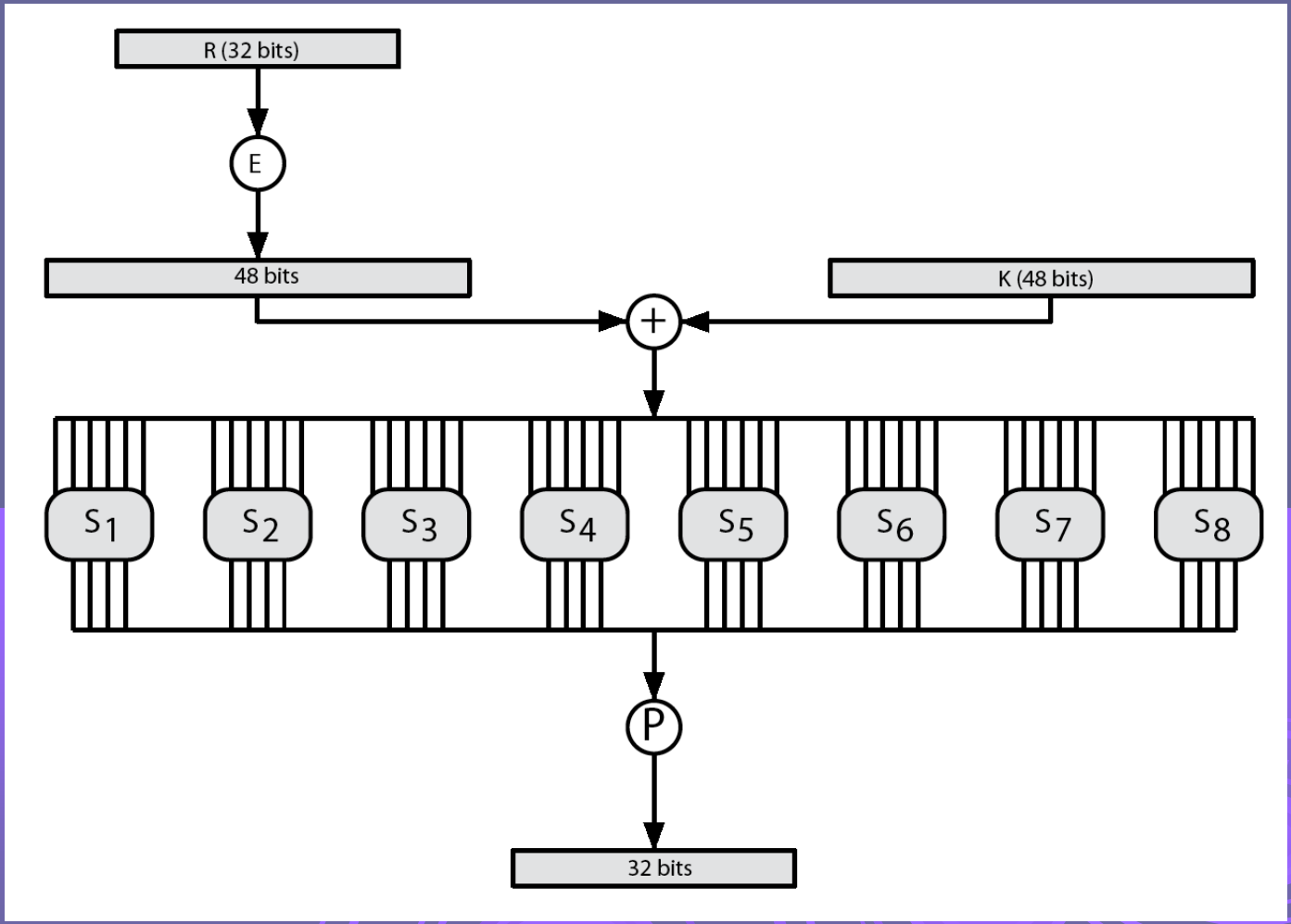
  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i)$
- F takes 32-bit R half and 48-bit subkey:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P

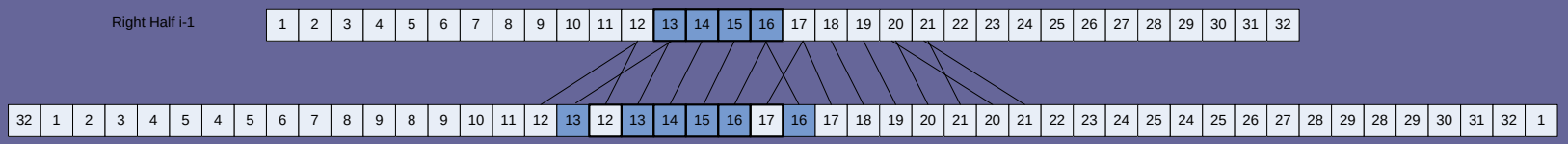
DES Round Structure



DES Round Structure



DES Expansion Permutation

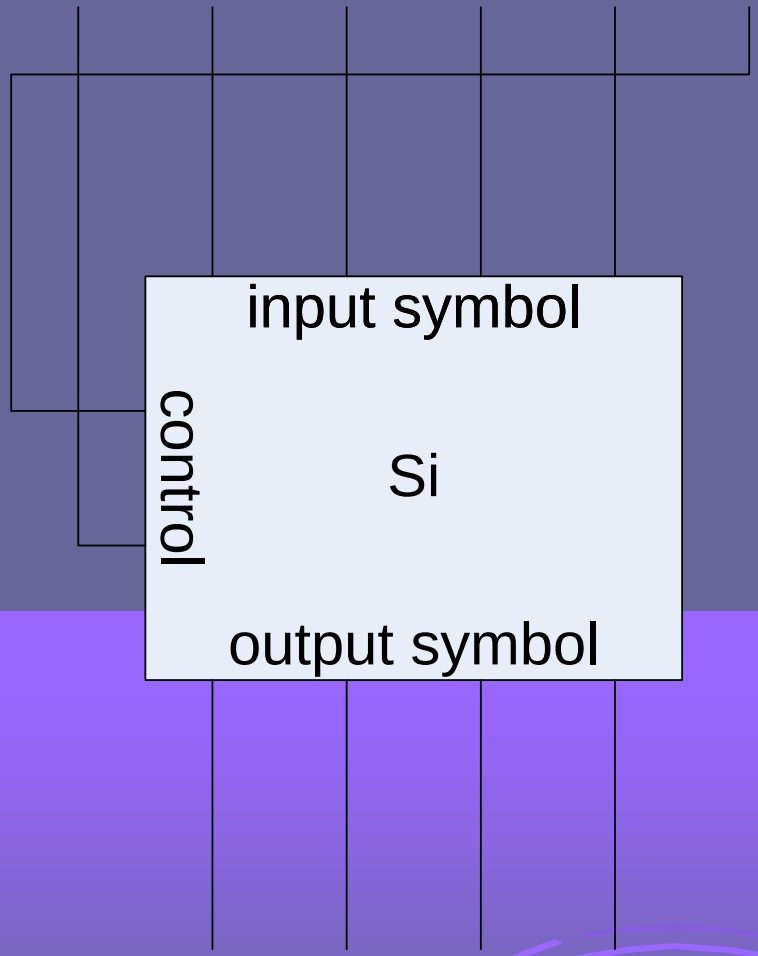


- R half expanded to same length as 48-bit subkey
- consider R as 8 nybbles (4 bits each)
- expansion permutation
  - copies each nybble into the middle of a 6-bit block
  - copies the end bits of the two adjacent nybbles into the two end bits of the 6-bit block

Substitution Boxes S

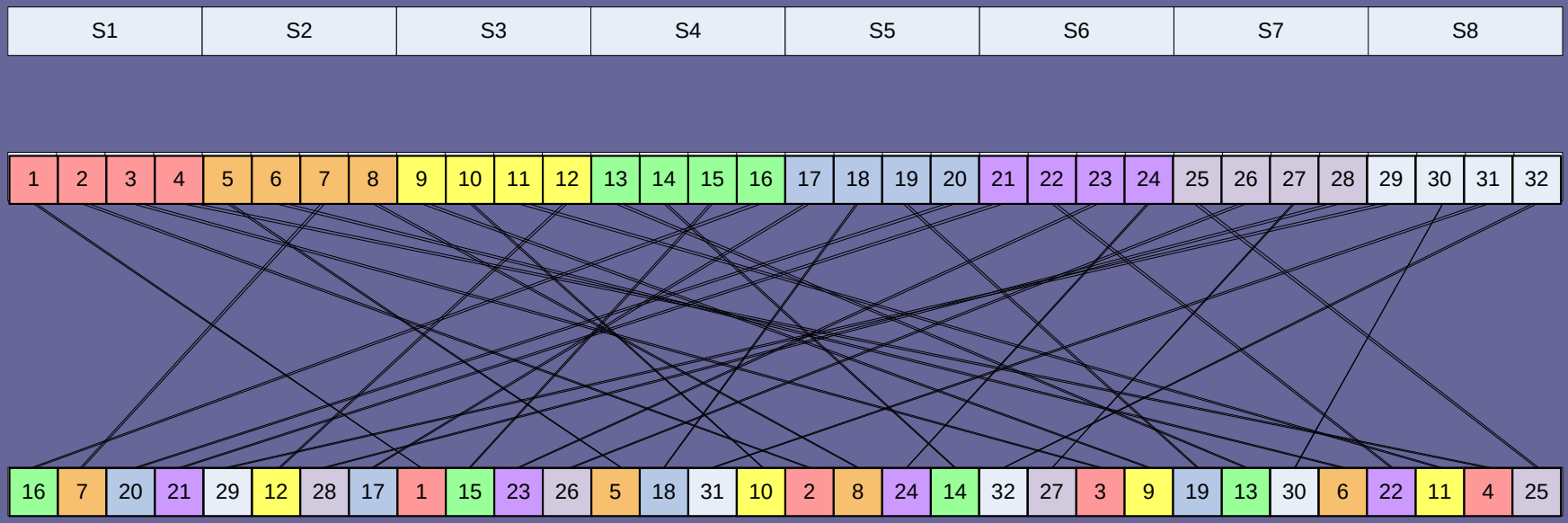
- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) select one row of 4
  - inner bits 2-5 (**col** bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

Substitution Boxes S



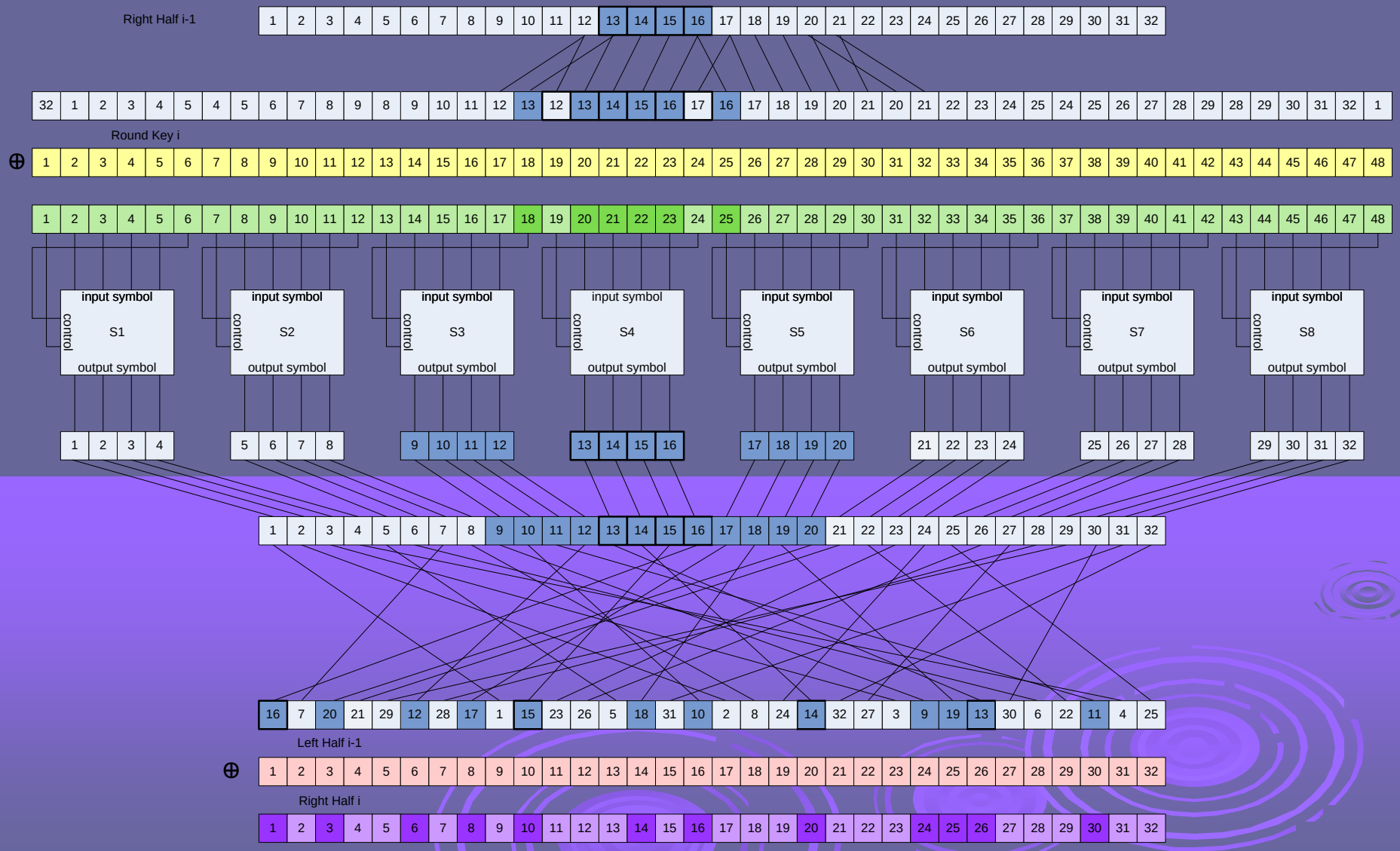
- each of the eight s-boxes is different
- each s-box reduces 6 bits to 4 bits
- so the 8 s-boxes implement the 48-bit to 32-bit contraction substitution

Permutation Box P



- P-box applied at end of each round
- Increases diffusion/avalanche effect

DES Round in Full



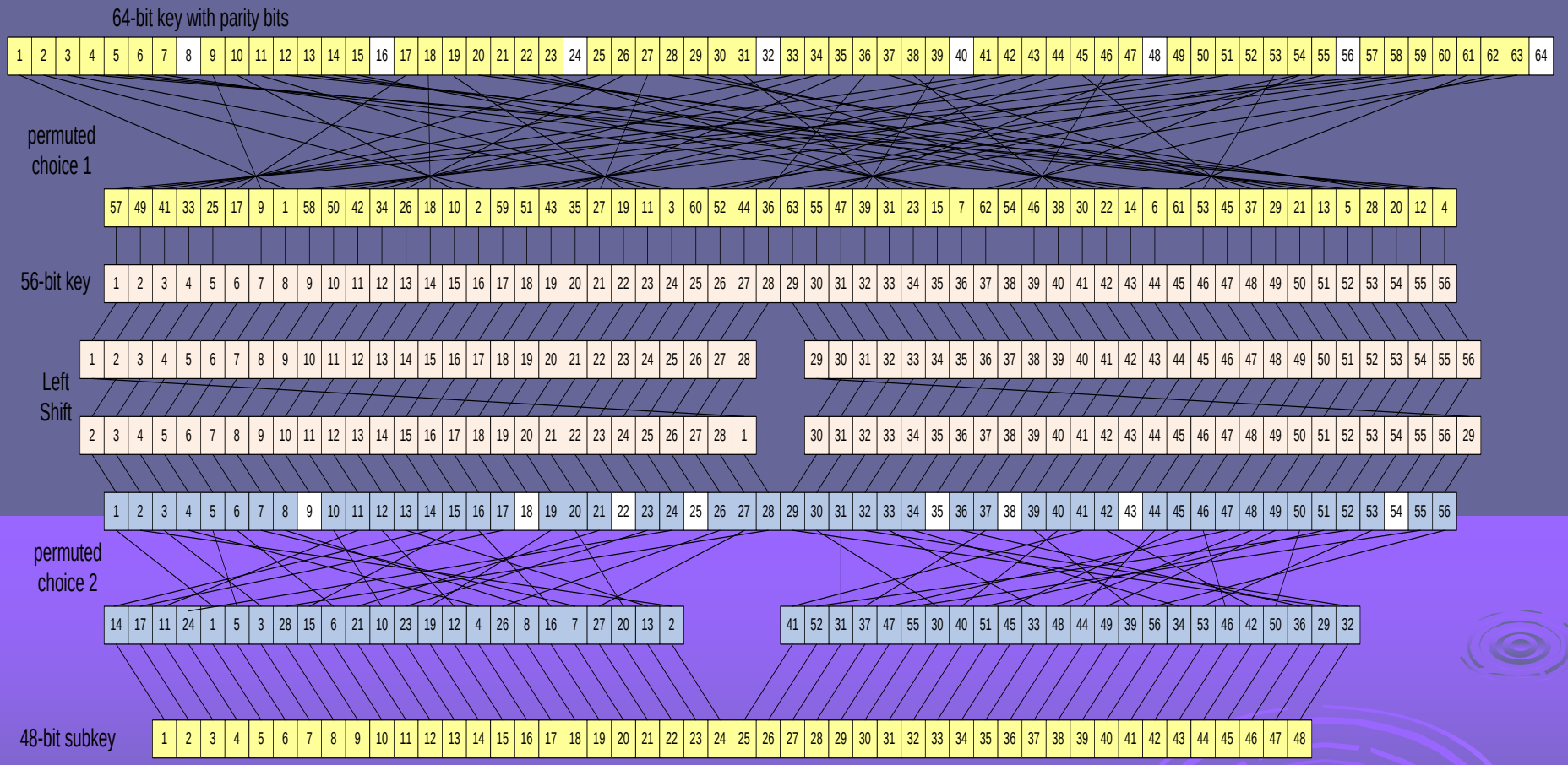
Interpreting Permutations

- In the textbook, permutations are given as tables
- The “inputs” are numbered from 1 to N
- The output positions are given by the position of the input in the table
- Tables are in rows of 8 numbers per row
- So for example, the P-Box permutation that takes bit 1 and moves it to bit 9 shows this by having a “1” in the 9th position (the first number on row 2)

DES Key Schedule

- forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K
    - selecting 24-bits from each half & permuting them by PC2 for use in round function F
- note practical use issues in h/w vs s/w

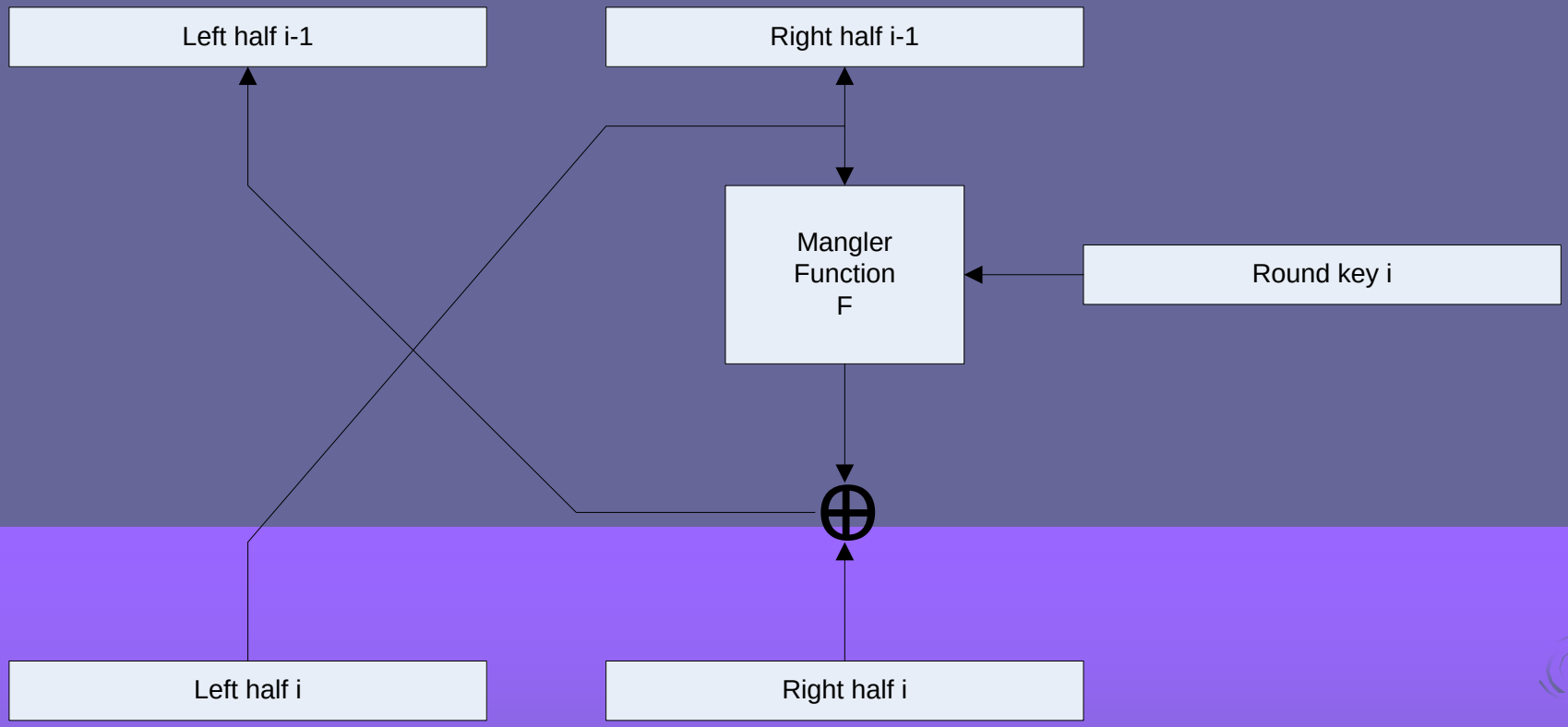
DES Key Schedule



DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again using subkeys in reverse order (SK16 … SK1)
  - IP undoes final FP step of encryption
  - 1st round with SK16 undoes 16th encrypt round
  - ….
  - 16th round with SK1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

DES Round Decryption



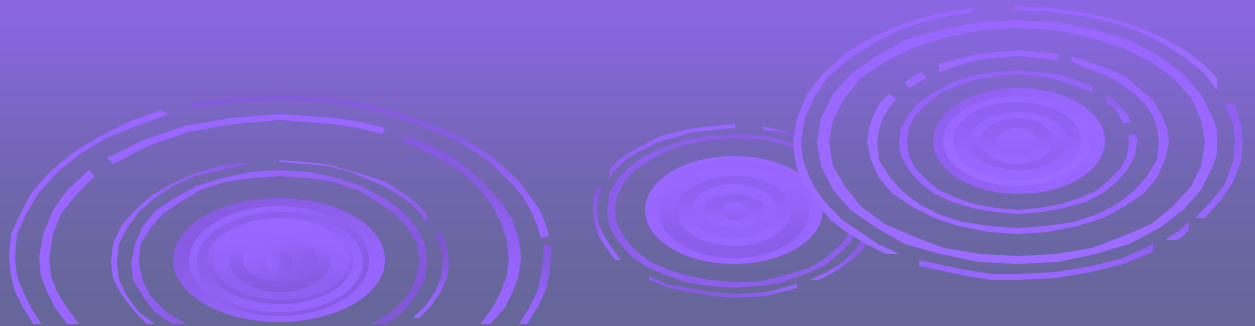
Decryption

DES Example

Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

Avalanche Effect

- key desirable property of encryption alg
- where a change of **one** input or key bit results in changing approx **half** output bits
- making attempts to “home-in” by guessing keys impossible
- DES exhibits strong avalanche



Avalanche in DES

original

modified

Round		δ
	02468aceeca86420	1
	12468aceeca86420	
1	3cf03c0fbad22845	1
	3cf03c0fbad32845	
2	bad2284599e9b723	5
	bad3284539a9b7a3	
3	99e9b7230bae3b9e	18
	39a9b7a3171cb8b3	
4	0bae3b9e42415649	34
	171cb8b3ccaca55e	
5	4241564918b3fa41	37
	ccaca55ed16c3653	
6	18b3fa419616fe23	33
	d16c3653cf402c68	
7	9616fe2367117cf2	32
	cf402c682b2cefbc	
8	67117cf2c11bfc09	33
	2b2cefbc99f91153	

Round		δ
9	c11bfc09887fbc6c	32
	99f911532eed7d94	
10	887fbc6c600f7e8b	34
	2eed7d94d0f23094	
11	600f7e8bf596506e	37
	d0f23094455da9c4	
12	f596506e738538b8	31
	455da9c47f6e3cf3	
13	738538b8c6a62c4e	29
	7f6e3cf34bc1a8d9	
14	c6a62c4e56b0bd75	33
	4bc1a8d91e07d409	
15	56b0bd7575e8fd8f	31
	1e07d4091ce2e6dc	
16	75e8fd8f25896490	32
	1ce2e6dc365e5f59	
IP^{-1}	da02ce3a89ecac3b	32
	057cde97d7683f2a	

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looked hard
- advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - in 1999 above combined in 22hrs!
- still must be able to recognize plaintext
- Forced to consider alternatives to DES

Strength of DES – Analytic Attacks

- now have several analytic attacks on DES
- these utilize some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Strength of DES – Timing Attacks

- attacks actual implementation of cipher
- use knowledge of consequences of implementation to derive information about some/all subkey bits
- specifically use fact that calculations can take varying times depending on the value of the inputs to it
- particularly problematic on smartcards

Seven Criteria for DES S-boxes

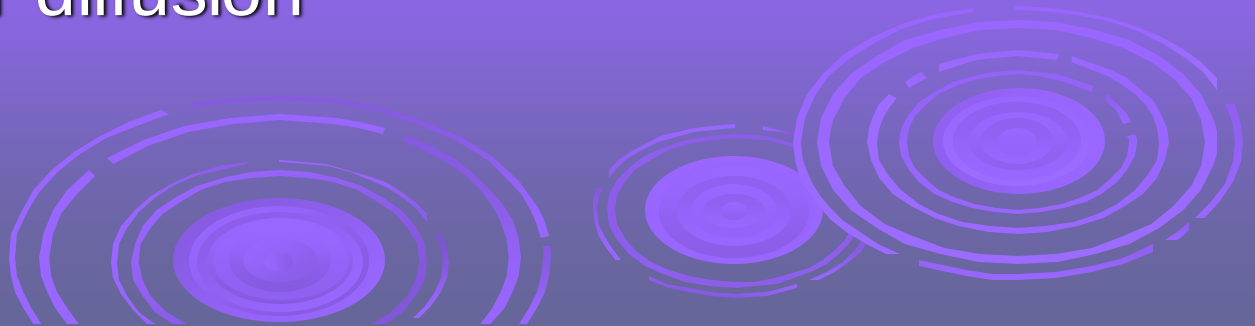
- No output bit of any S-box should be too close to a linear function of the input bits
- Each row of an S-box should include all 16 possible output bit combinations
- If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits
- If two inputs to an S-box differ in the two middle bits, the outputs must differ in >= 2 bits
- If two inputs differ in their first two bits and are identical in last two bits, the two outputs must not be the same
- For any nonzero 6-bit difference in inputs, no more than 8 of the 32 pairs with that difference may result in the same output difference

Three Criteria for DES P-box

- The four output bits from each S-box are distributed so that two are “middle” bits and two are “outer” bits in their nybbles
- The four output bits from each S-box affect six different S-boxes in the next round, and no two affect the same S-box
- For two S-boxes, j and k, if an output bit from Sj affects a middle bit of Sk in the next round, then an output bit from Sk cannot affect a middle bit of Sj (so Sj output can't be a middle bit of Sj in next round)

DES Design Criteria

- as reported by Coppersmith in [COPP94]
- 7 criteria for S-boxes provide for
  - non-linearity
  - resistance to differential cryptanalysis
  - good confusion
- 3 criteria for permutation P provide for
  - increased diffusion



Block Cipher Design

- basic principles still like Feistel's in 1970's
- number of rounds
  - more is better, exhaustive search best attack
- function f:
  - provides "confusion", is nonlinear, avalanche
  - have issues of how S-boxes are selected
- key schedule
  - complex subkey creation, key avalanche