Modern Symmetric Key Ciphers

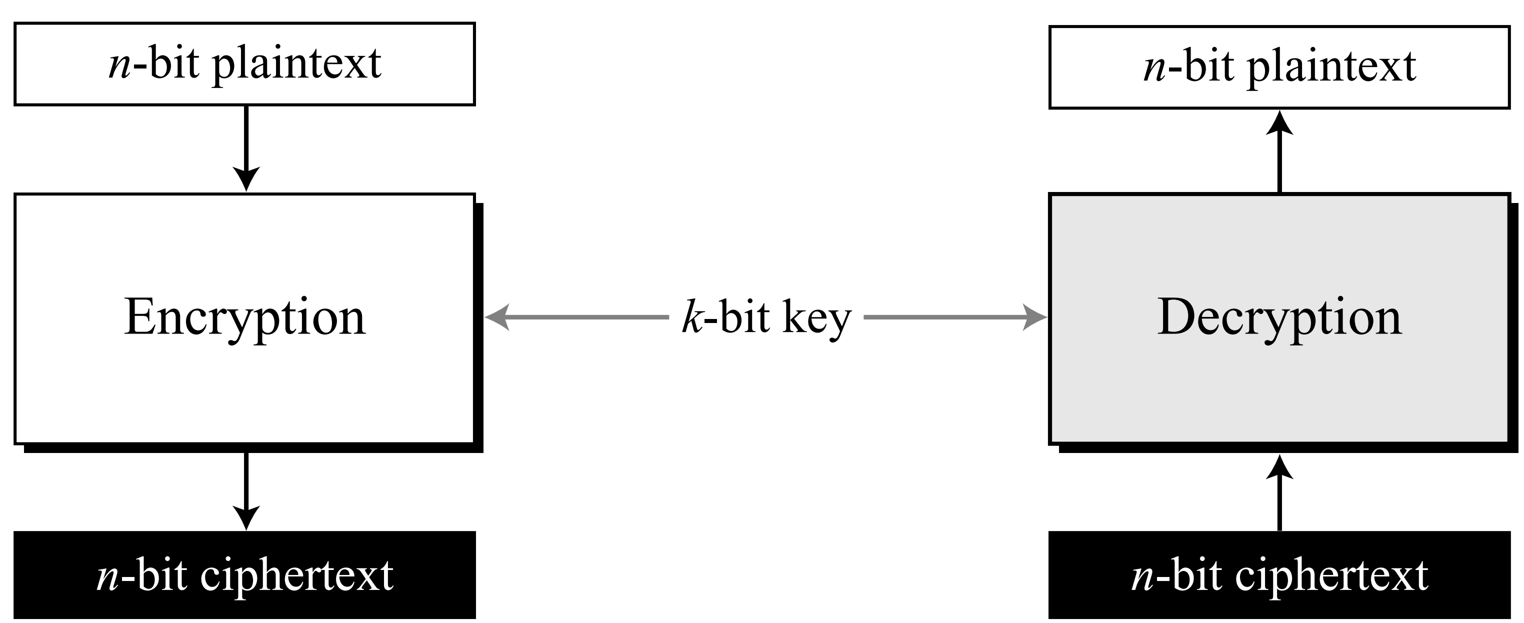
Part I: Foundational Blocks: Permutations and Substitutions

THE UNIVERSITY OF UTAH

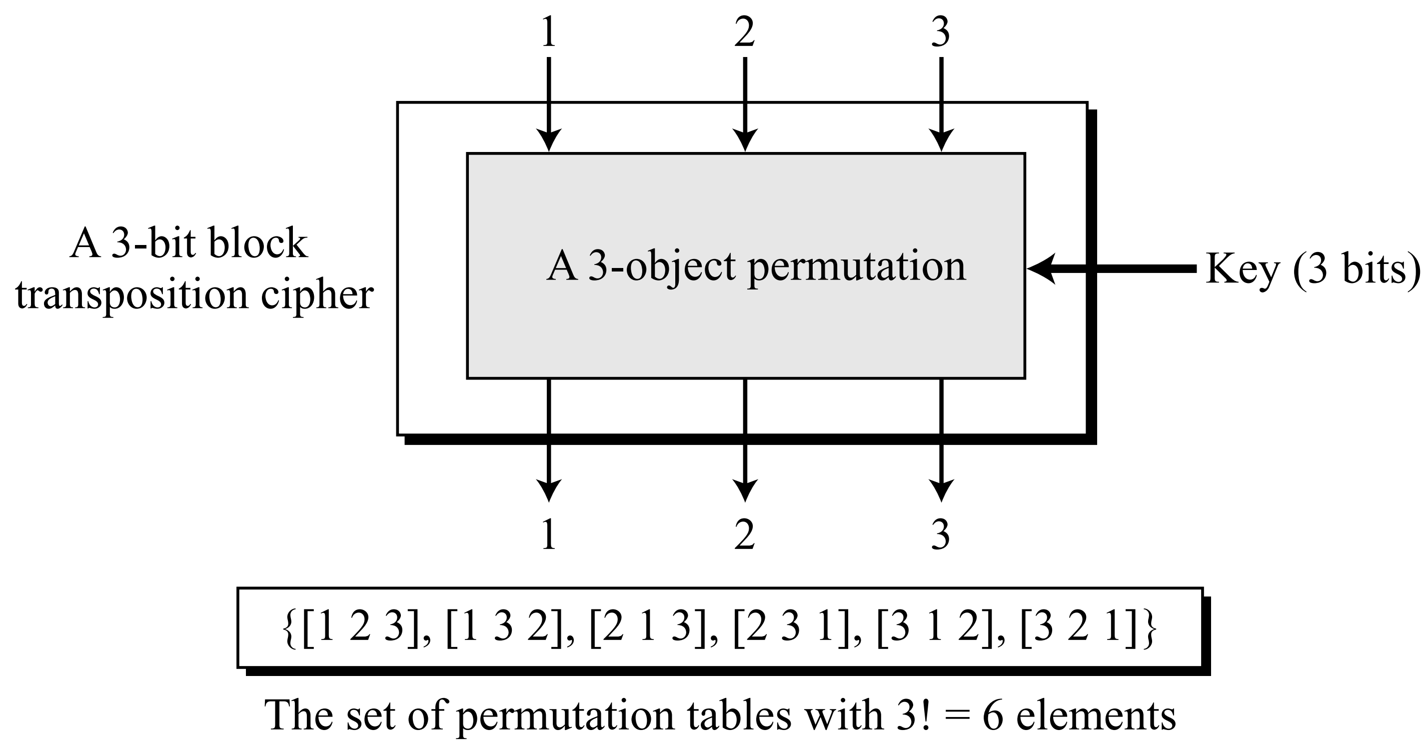
Priyank Kalla
Professor
Electrical & Computer Engineering

Modern Symmetric Key Ciphers

- Modern Ciphers: “Overall” Substitution based Cipher
  - Transposition components are also used, but we have to be careful
  - The concept of permutation groups, and related issues
- Can be designed as Block ciphers as well as Stream ciphers
- Used as bit-oriented ciphers, as opposed to purely character oriented ciphers
  - 8-bit ASCII encoding, or other k-bit encodings (think $\mathbb{F}_{2^k}$ fields)

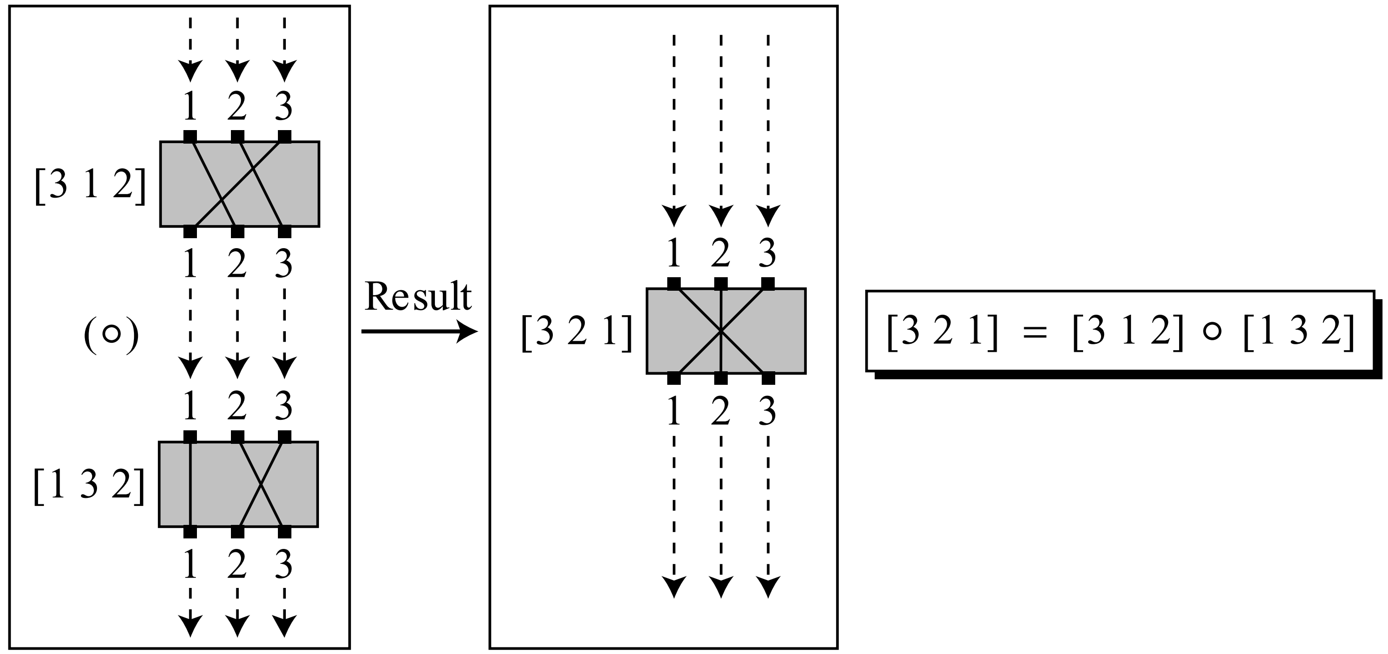


Transposition Blocks as Permutations



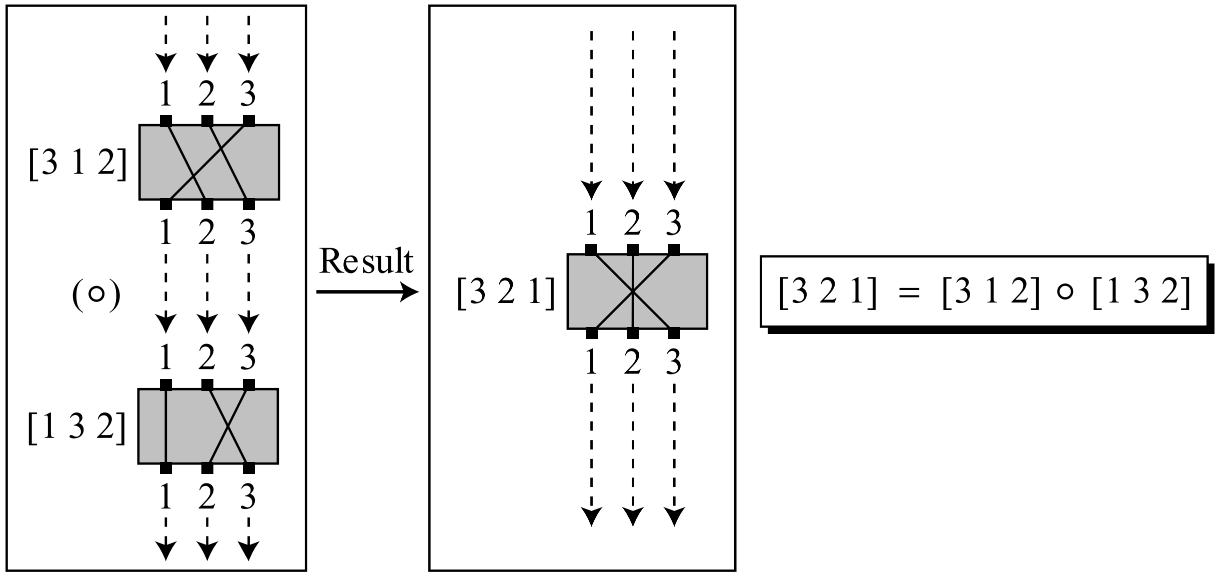
- 6 elements = 3-bits of key. Key defines the permutation
- This is a full-size (3 bits) key. In practice, keys are smaller
- 3-bit block: data is n=3 bits.
- For a transposition cipher, key length = $\lceil log_2\ n! \rceil$ bits

Transposition/Permutation Cipher Issues



- Can we perform multiple rounds of permutations?
- Permutation Groups: Group = $(G, \odot\ )$
  - G = elements of the set = permutations
  - $\odot =$ composition
  - If G_3 can be composed as $G_3 = G_1 \odot G_2$, then multiple rounds are useless, they don't add to security
  - Instead of doing 2 rounds G_1, G_2, just use one round G_3
- Should not use ciphers that can modeled as permutation groups! [How to make these decisions, more on this later…]

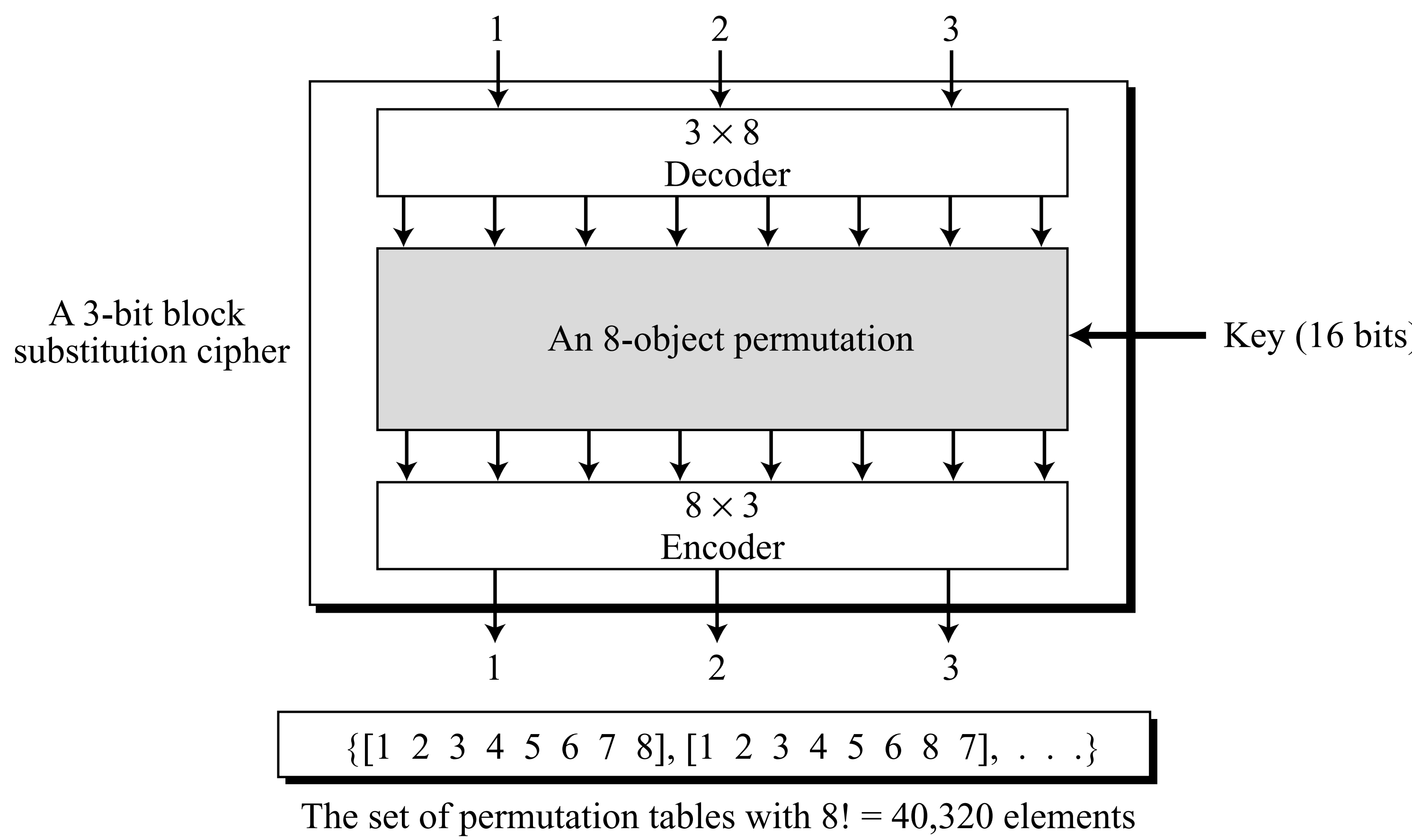
Transposition/Permutation Cipher Issues



∘	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 2 3]	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 3 2]	[1 3 2]	[1 2 3]	[2 3 1]	[2 1 3]	[3 2 1]	[3 1 2]
[2 1 3]	[2 1 3]	[3 1 2]	[1 2 3]	[3 2 1]	[1 3 2]	[2 3 1]
[2 3 1]	[2 3 1]	[3 2 1]	[1 3 2]	[3 1 2]	[1 2 3]	[2 1 3]
[3 1 2]	[3 1 2]	[2 1 3]	[3 2 1]	[1 2 3]	[2 3 1]	[1 3 2]
[3 2 1]	[3 2 1]	[2 3 1]	[3 1 2]	[1 3 2]	[2 1 3]	[1 2 3]

- [1 2 3] = identity element = I
- The above table = non-commutative group = non-abelian group
- Inverse: $P + P^{-1} = I$
- Multiple rounds of permutations DO NOT add to security here

Substitutions modeled as Permutations with Encoder/Decoder Combinations



- Key = $\lceil log_2\ (2^n)! \rceil$ bits long
- Here n = 3, $2^n = 8, \ \ 8! = 40320$
- Key creates a “mapping”
- DES is a “partial size key cipher”: 64 bit block = $\lceil log_2\ (2^{64}!) \rceil = 2^{70}$ bits!!! Uses only a 56-bit key

Multi-Stage Ciphers and Security

- Let $(G, \odot)$ be a permutation group, where G is the set of permutations, under the composition $\odot$ operation

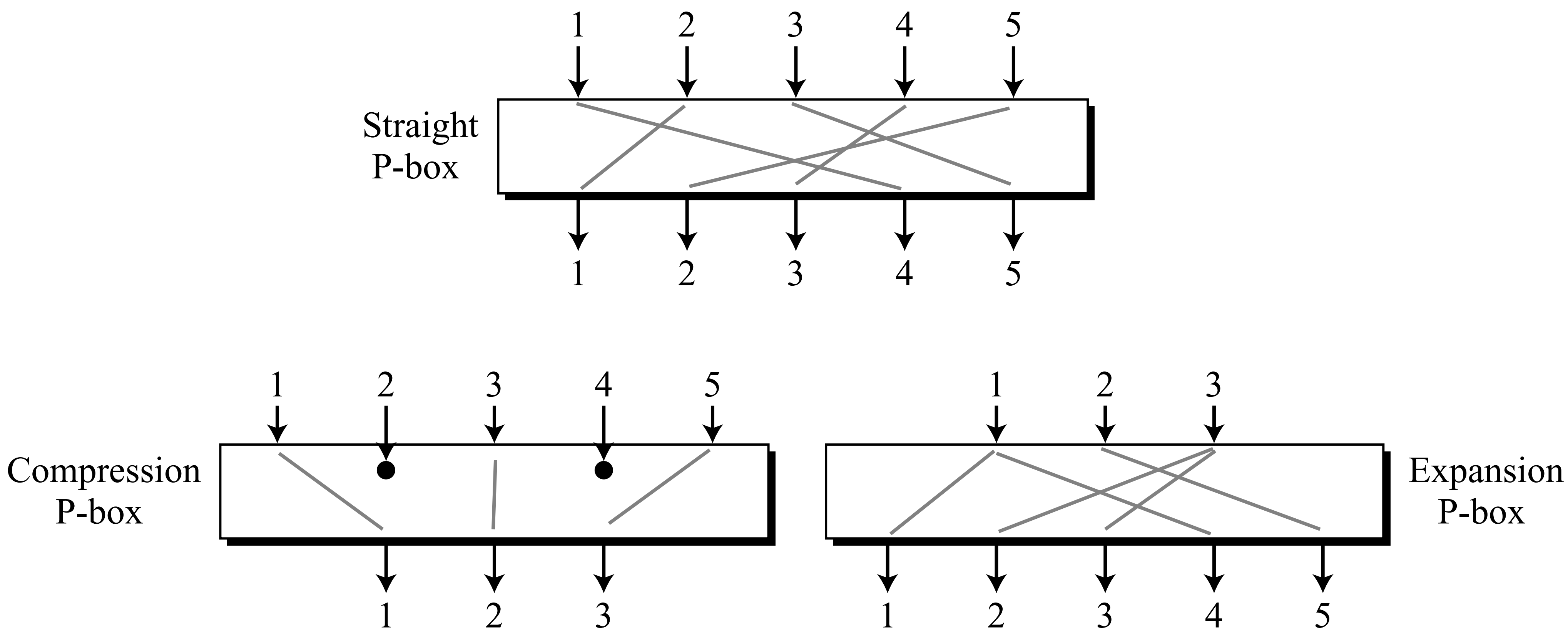
- Let $(M, \odot)$ be a permutation group, where M is the set of permutations, under the composition $\odot$ operation. Let $M \subset G$, then $(M, \odot)$ is a subgroup of $(G, \odot)$

- A partial-key cipher is a ***group*** if it is a ***subgroup*** of the full-size key cipher

- To obtain more security using multiple rounds of the same cipher, ensure that the partial-key ciphers are NOT subgroups of the full-size ciphers

- Multi-stage DES w/ 56-bit key is NOT a group: No subgroup with 2^{56} mappings can be created with $2^{64}!$ mappings/permutations

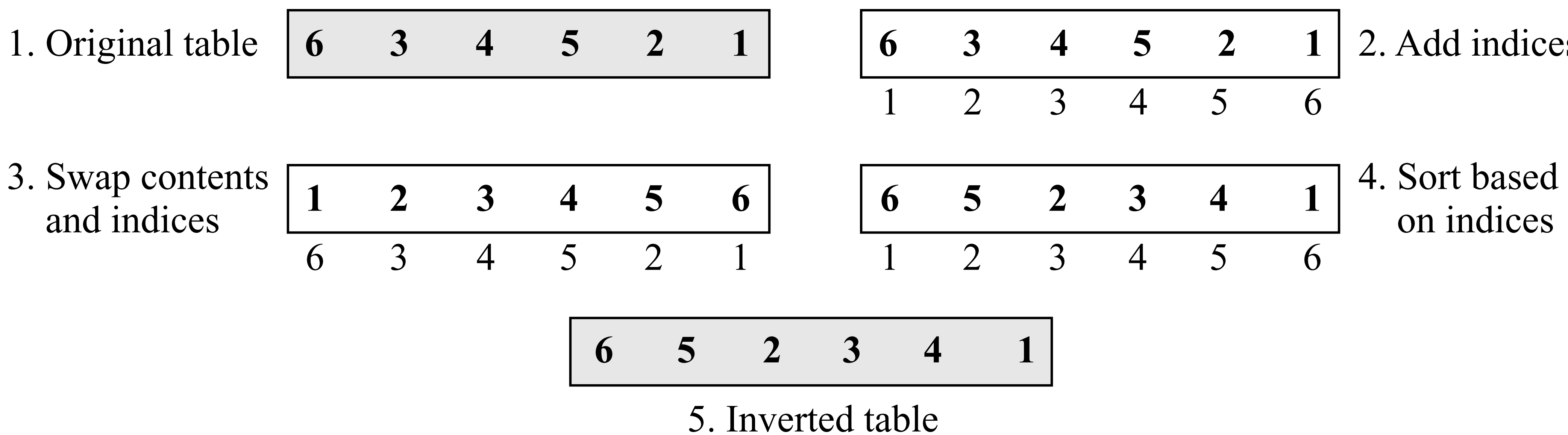
3-types of P-Boxes in Modern Ciphers



- Straight P-boxes are invertible (encryption and decryption)
- The above is a keyless straight P-Box
- Compression and Expansion are non-invertible [but, sometimes their combinations can cancel out, so they can be used in enchipherment]

Permutation table for the above straight P-Box: [2 5 4 1 3]
How to compute Inverses?

Inverting a Permutation Table



- Original table for encryption, Inverted tables used for decryption
- Hardware implementation is trivial wiring

S-Boxes

- An S-Box substitutes an n-bit input word with a m-bit output word:
- May or may not be invertible
- May or may not be linear

$$
\begin{gathered}
y_1 = f_1\,(x_1, x_2, \ldots, x_n) \\
y_2 = f_2\,(x_1, x_2, \ldots, x_n) \\
\ldots \\
y_m = f_m\,(x_1, x_2, \ldots, x_n)
\end{gathered}
$$

Linear S-Box: Invertible?

$$
\begin{gathered}
\boldsymbol{y_1 = a_{1,1}\,x_1 \oplus a_{1,2}\,x_1 \oplus \cdots \oplus a_{1,n}\,x_n} \\
\boldsymbol{y_2 = a_{2,1}\,x_1 \oplus a_{2,2}\,x_1 \oplus \cdots \oplus a_{2,n}\,x_n} \\
\ldots \\
\boldsymbol{y_m = a_{m,1}\,x_1 \oplus a_{m,2}\,x_1 \oplus \cdots \oplus a_{m,n}\,x_n}
\end{gathered}
$$

Invertible S-Box:

3 bits

	00	01	10	11
0	011	101	111	100
1	000	010	001	110

Table used for encryption

3 bits

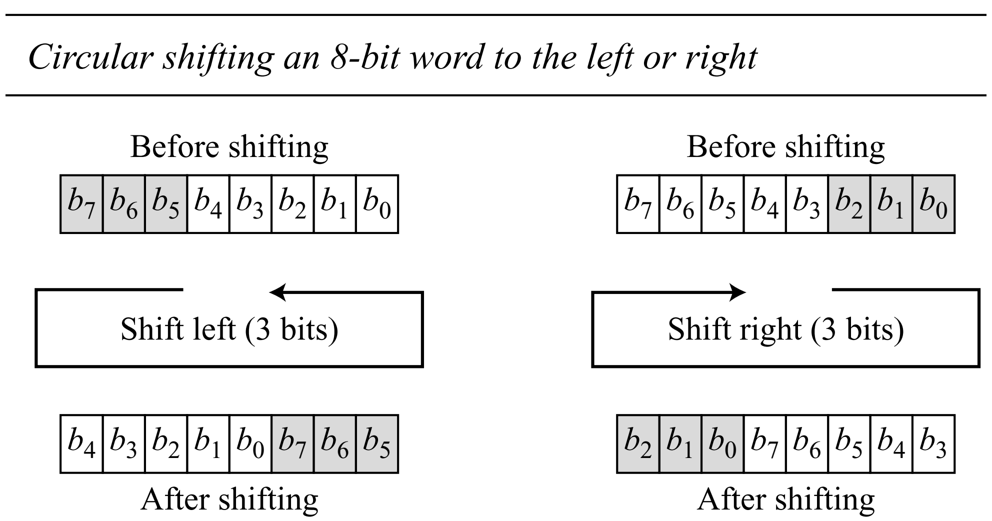
3 bits

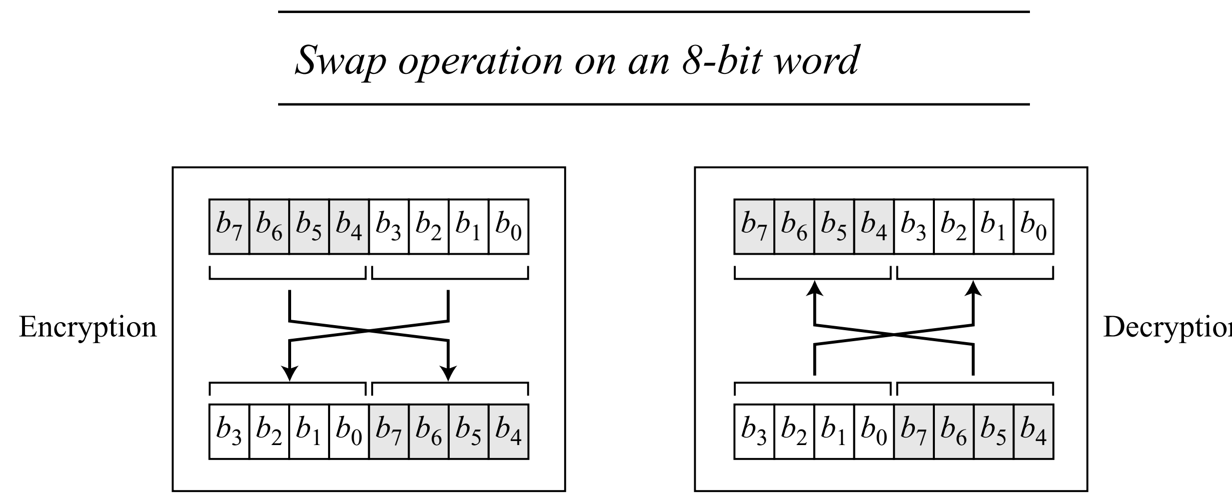
	00	01	10	11
0	100	110	101	000
1	011	001	111	010

Table used for decryption

3 bits

Shift and Swap Operations also used





- Circular left shift = inverse of circular right shift operations
- Circular shifts under composition = group operation!
- Circular shifting more than once = shifting once…

Diffusion and Confusion

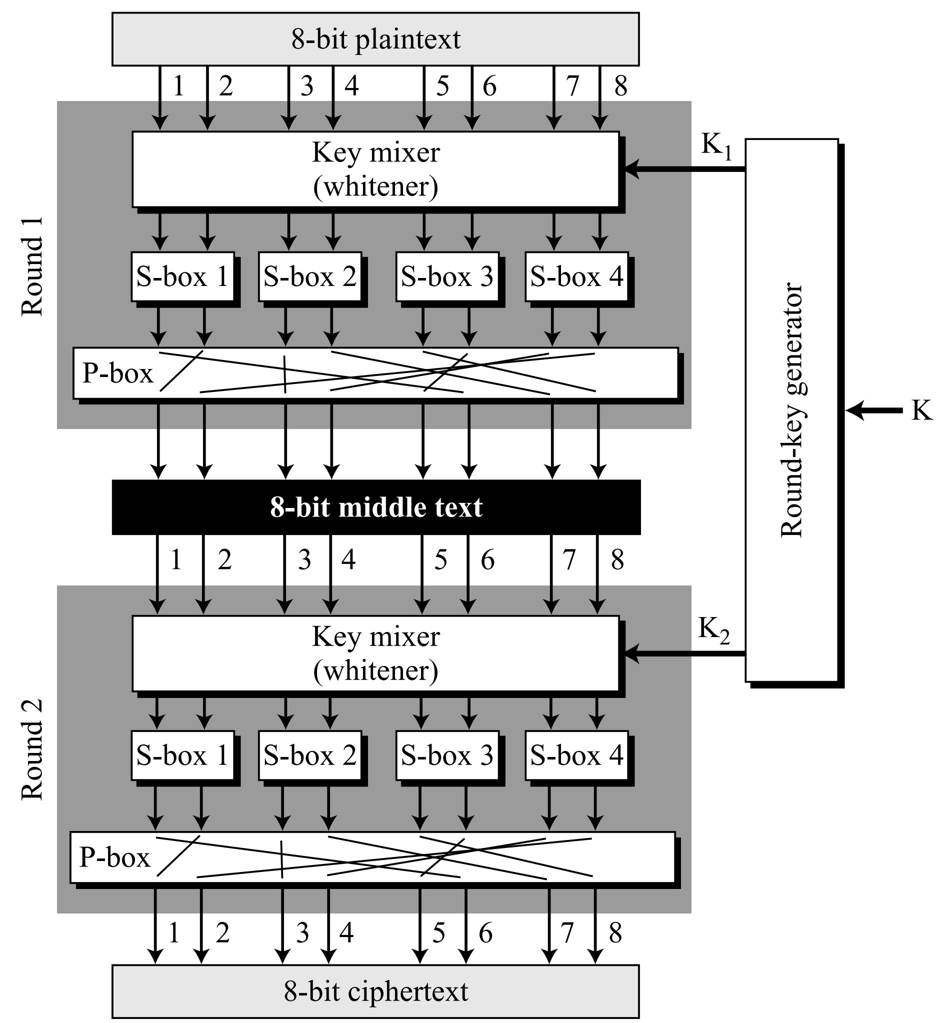
- Diffusion hides the relationship between Ciphertext C and Plaintext P
- Confusion hides the relationship between Ciphertext C and the Key K
- Rounds:
  - Diffusion and confusion achieved using iterations of S-Box, P-Box and other operations: **Product ciphers**
  - Create a block cipher using a key schedule/generator: create different keys for each round from the cipher key
  - N-round cipher: P is encrypted N times to create C, and then C is decrypted N times to create recreate P

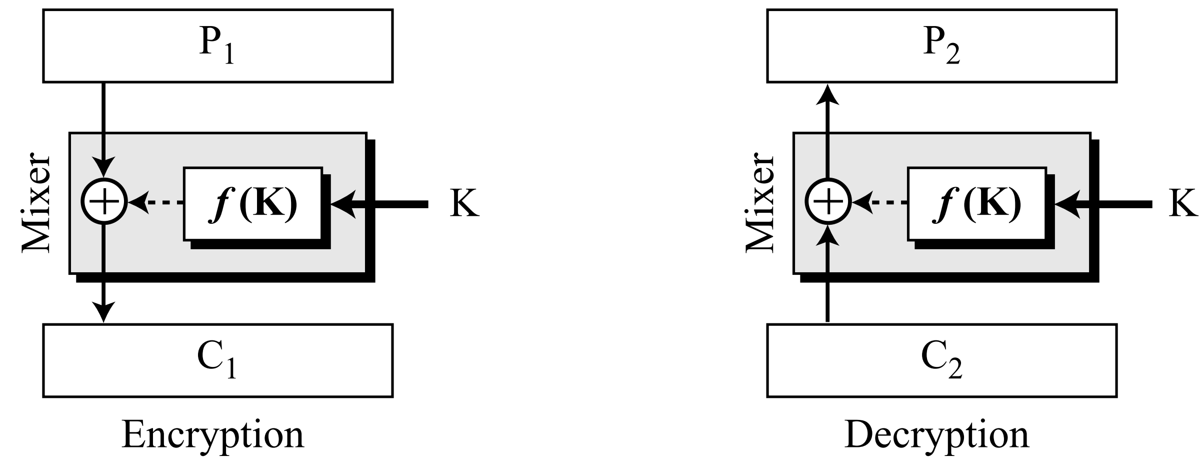
Product Ciphers

- An example of Product Cipher:
  - 8-bit text mixed with 8-bit Key: Whitening the text, to hide the bits of P with key K
    - E.g. XOR 8-bit text with 8-bit key
  - Outputs of the whitener are fed into 4 2-bit groups, and fed into 4 S-boxes
  - Outputs of S-boxes passed through a P-box
  - Do one more round of the same…

Example Product Cipher

A product cipher made of two rounds



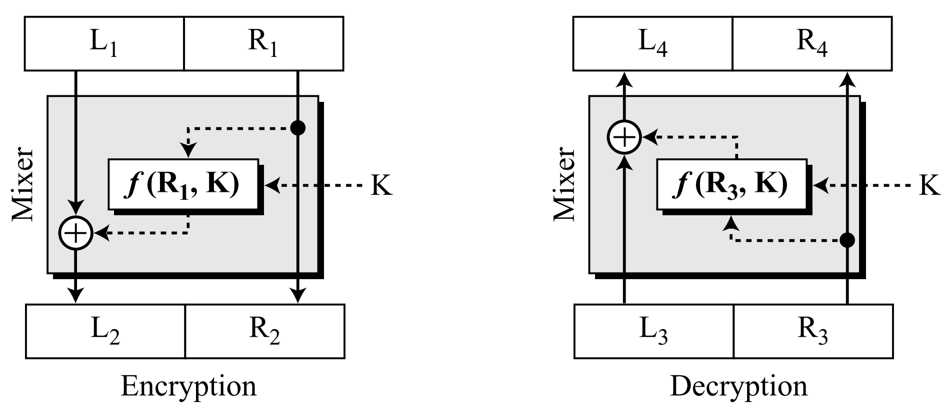


- Mixer: Use a non-invertible function $f(K)$: can be linear or polynomial in $\mathbb{F}_{2^k}$

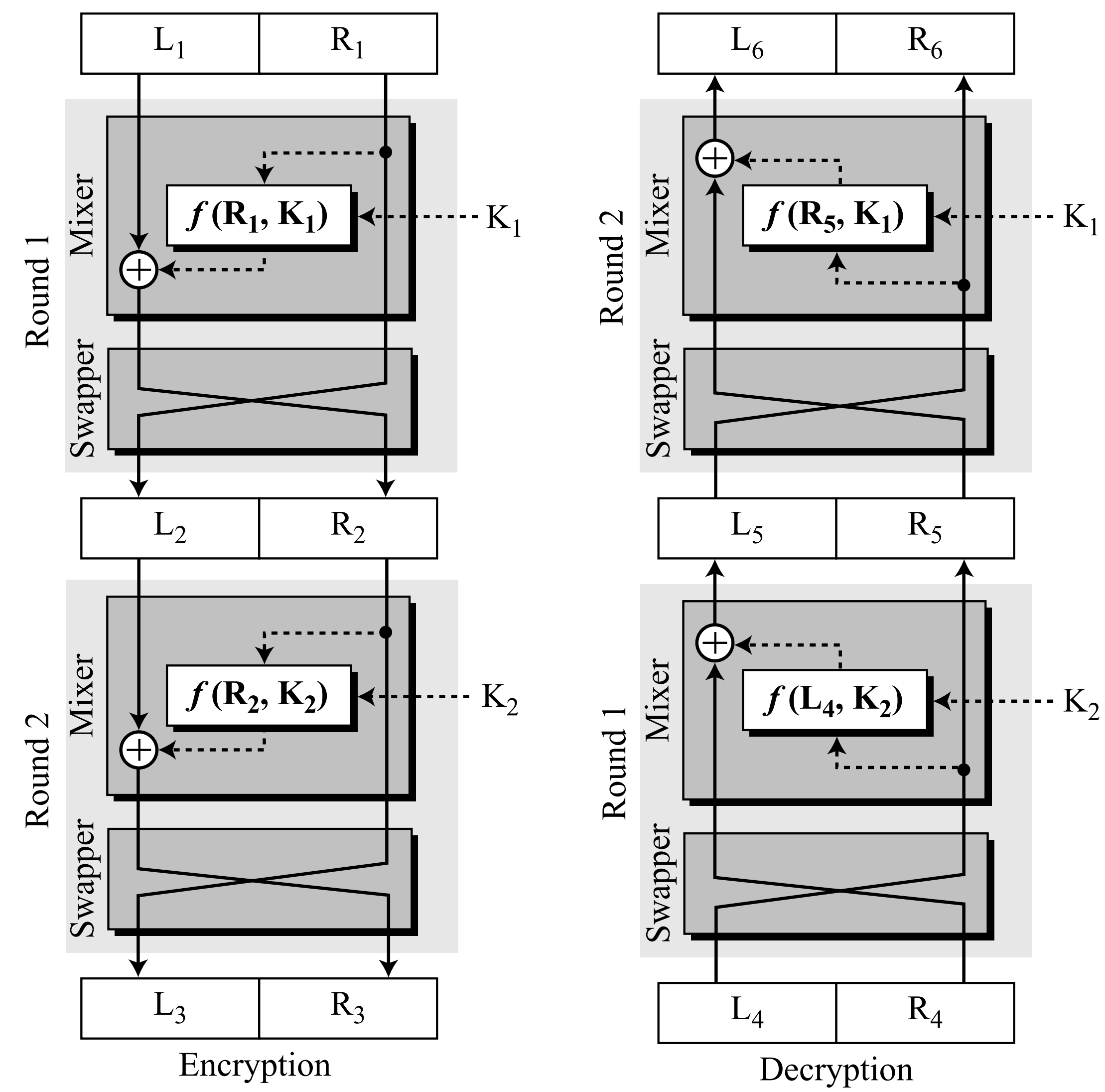
Encryption: $C_1 = P_1 \oplus f(K)$

Decryption: $P_2 = C_2 \oplus f(K) = C_1 \oplus f(K) = P_1 \oplus f(K) \oplus f(K) = P_1 \oplus (00\ldots0) = P_1$

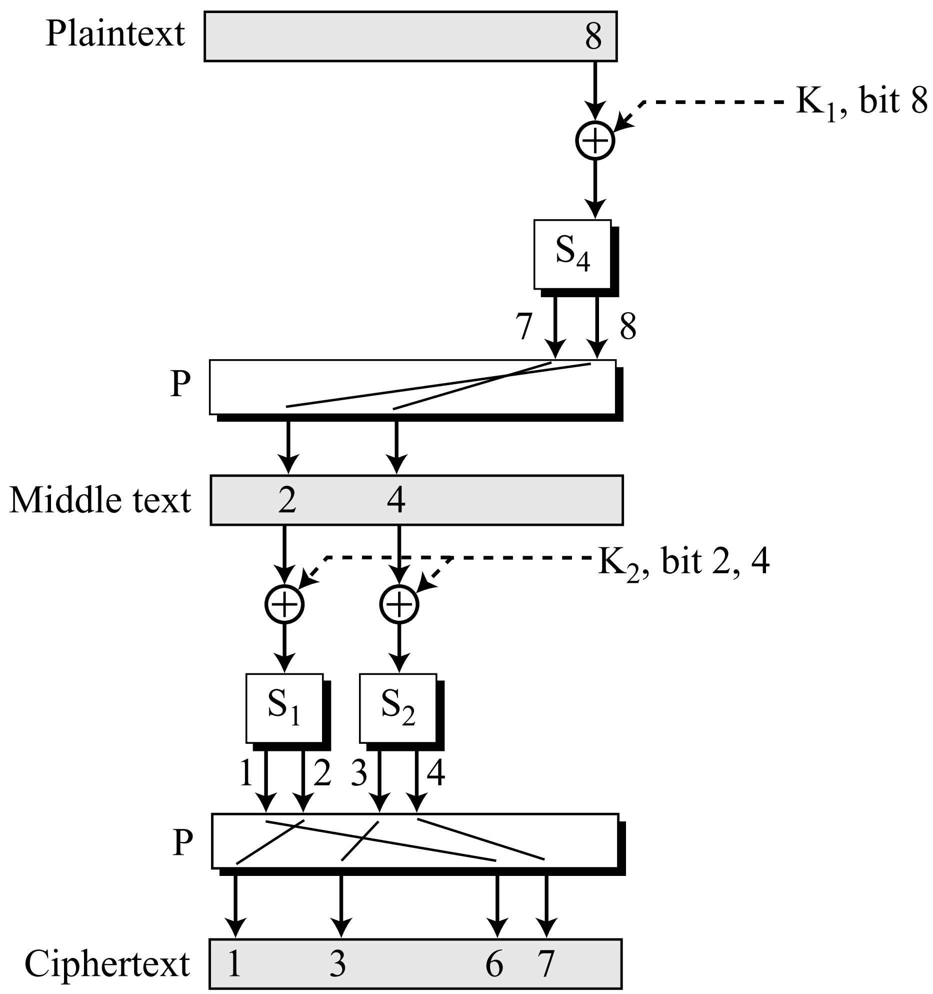
An Example Product Cipher: Feistel Cipher



- $L_3 = L_2, R_3 = R_2$



Example of Diffusion and Confusion



- Diffusion:
  - Bit-8 in P has affected bits 1, 3, 6,7 in C
  - Similarly, each bit in C is affected by several bits in P
- Confusion:
  - Bits 1, 3, 6, 7 in C affected by bit 8 in K_1 and bits 2, 4 in K_2