International Data Encryption Standard (IDEA)

Introduction

- A Symmetric Block Cipher
- developed by XuejiaLai and James Massey in 1991
- General description
  - Operates on 64-bit plaintext block.
  - Uses 128 bit key
  - Same algorithm is used for encryption and decryption (like DES).
  - Considered by some to be superior to the DES
- Why IDEA?
  - The algorithm was designed to achieve high data throughput for use in real-time communications system, especially for wireless communication

Introduction (contd.)

- The International Data Encryption Algorithm (IDEA) is one of a number of conventional encryption algorithms that have been proposed in recent years to replace DES.

- In terms of adoption, IDEA is included in PGP (Pretty Good Privacy) which means that it already is becoming a de-facto standard for encryption worldwide.

Overview

- DES algorithm has been a popular secret key encryption algorithm and is used in many commercial and financial applications. However, its key size is too small by current standards and its entire 56 bit key space can be searched in approximately 22 hours

- IDEA is a minor revision of an earlier cipher, PES (Proposed Encryption Standard)

- IDEA was originally called IPES (Improved PES) and was developed to replace DES

Overview (cont')

- It entirely avoids the use of any lookup tables or S-boxes

- IDEA was used as the symmetric cipher in early versions of the Pretty Good Privacy cryptosystem

Detailed description of IDEA

- IDEA operates with 64-bit plaintext and cipher text blocks and is controlled by a 128-bit key
- Completely avoid substitution boxes and lookup tables used in the block ciphers
- The algorithm structure has been chosen such that when different key sub-blocks are used, the encryption process is identical to the decryption process

Design Principles

- The two properties that a good encryption algorithm should have are diffusion, and confusion.

- IDEA utilizes three basic operations to achieve them. They are:

  – Bit-by-bit exclusive OR, denoted as $\oplus$

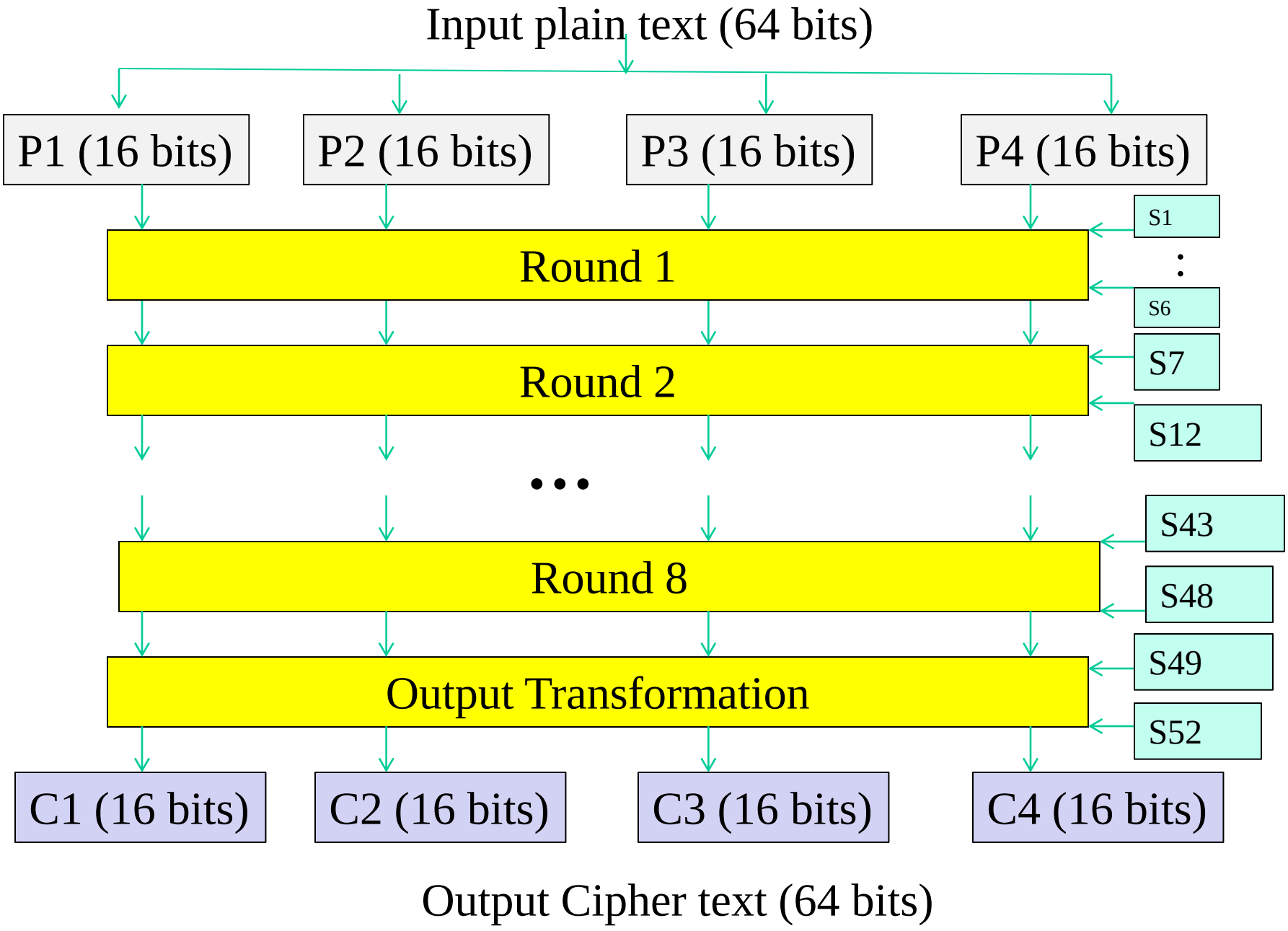
  – Addition of unsigned integers modulo 2^16 (65536), denoted as $\boxplus$

  – Multiplication of unsigned integers modulo 2^16+1 (65537), with the provision of treating a block of zeros as 2^16. This operation is denoted as $\odot$

IDEA Design Principles Contd..

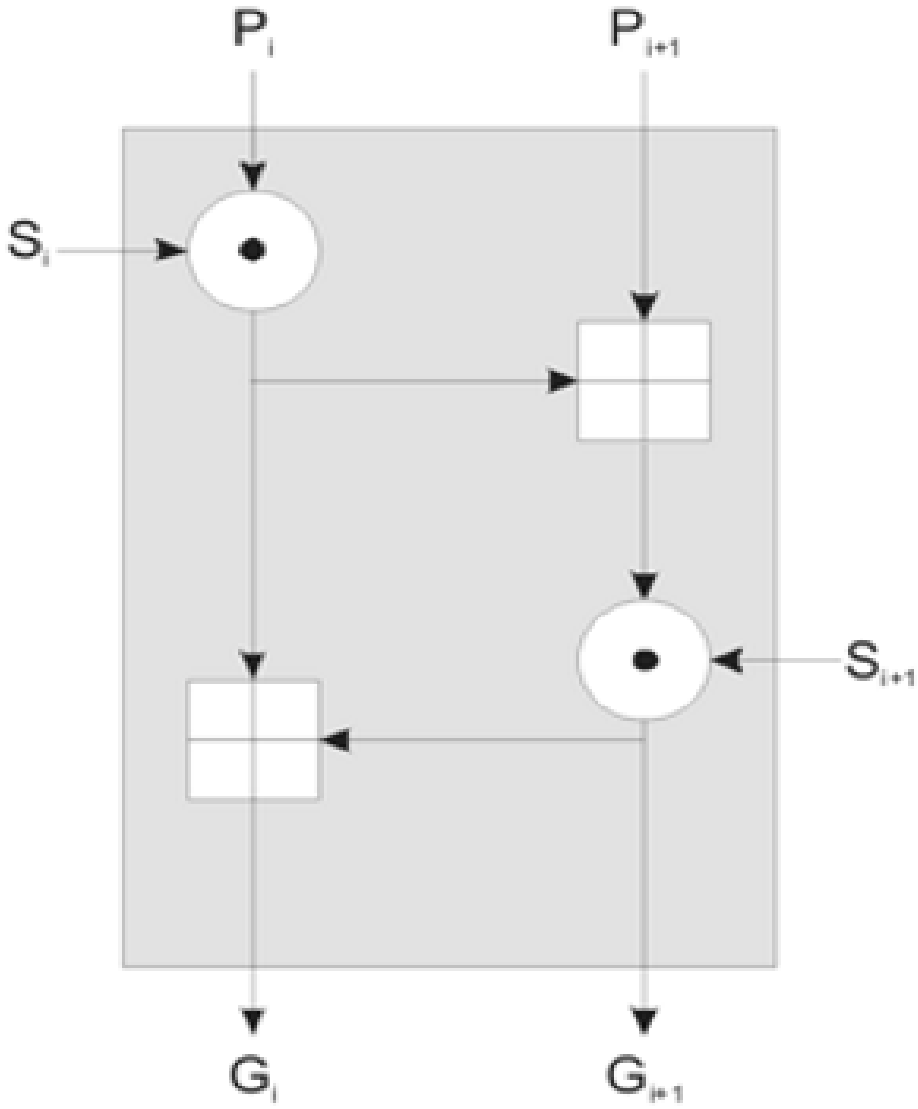
- Confusion is achieved by mixing these three incompatible operations.
- The use of these three separate operations in combination gives a complex transformation and makes cryptanalysis much more difficult than with a single XOR function as is the case in DES.

Broad level steps in IDEA



Design Principles

- Diffusion is provided by the basic building block of the algorithm, known as the multiplication / addition (MA) structure as shown in the Figure.
- The structure takes as input two 16-bit values derived from the plaintext: Pi, Pi+1, and two 16-bit subkeys derived from the key: Si, Si+1.



Encryption

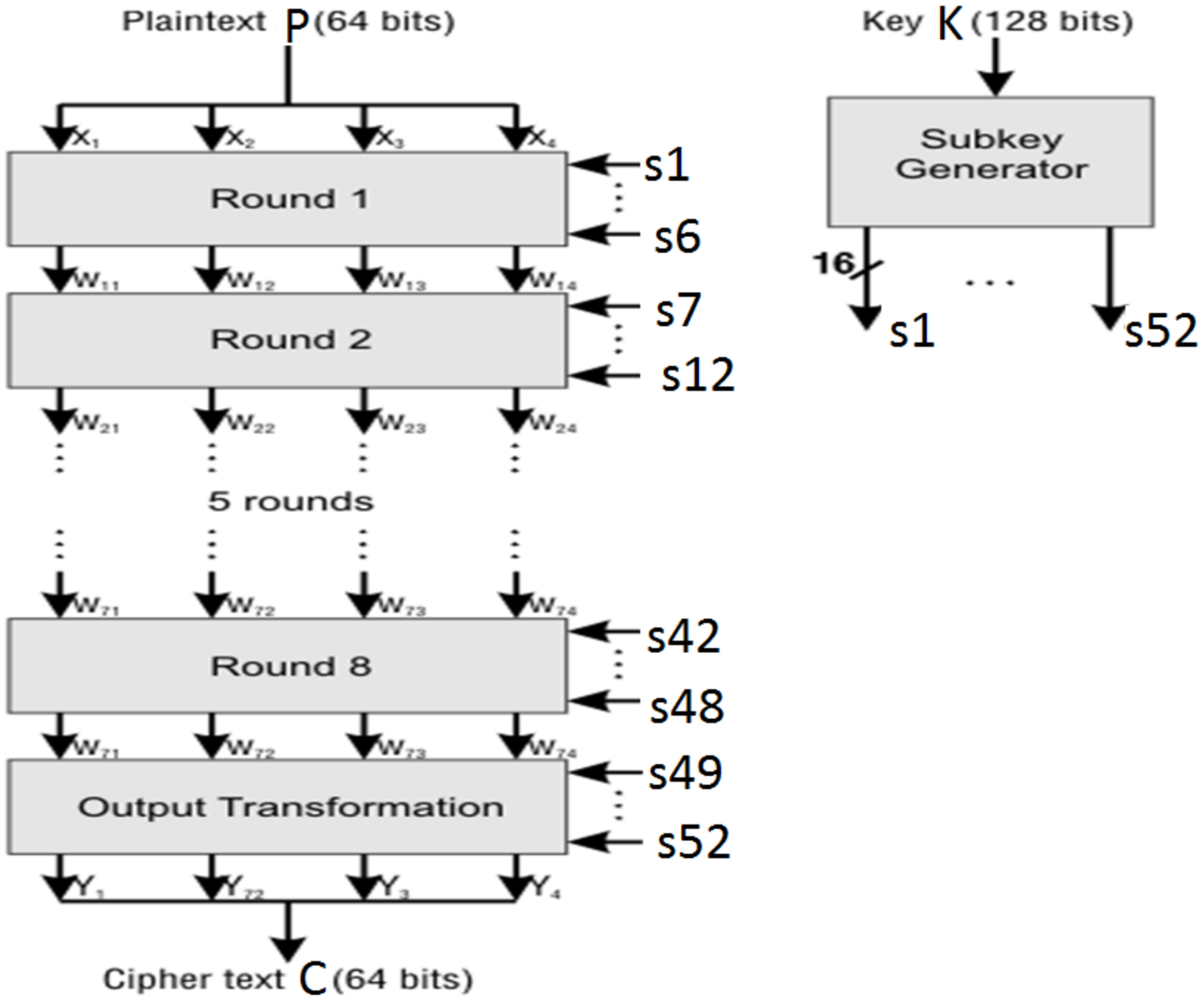
- The algorithm consists of eight identical rounds followed by a final transformation function.

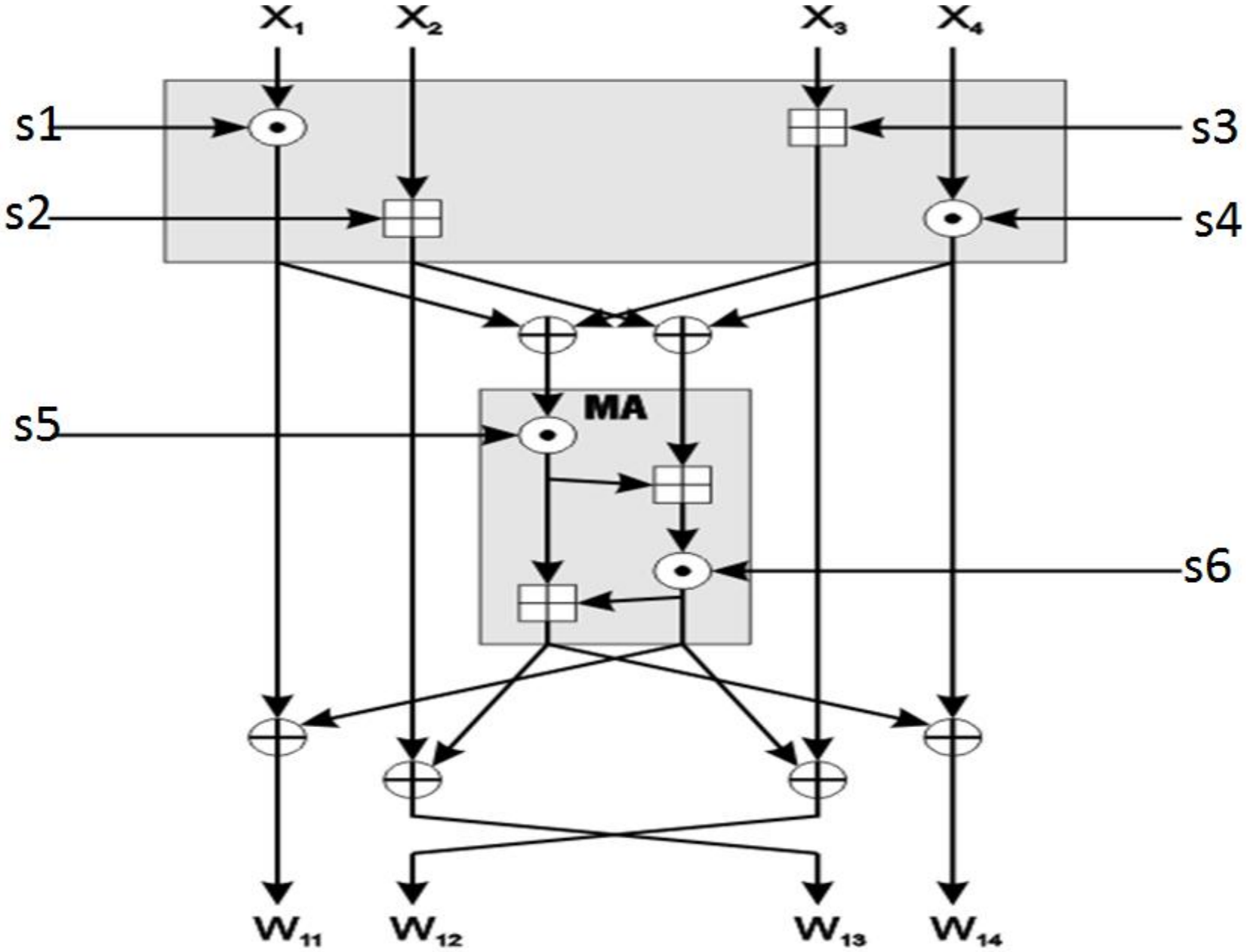
- The 64-bit plain text input is divided into four blocks of 16 bits each.

- Each round takes as input
  - four 16-bit blocks, and
  - six 16-bit subkeys, and generates
  - four 16-bit blocks as illustrated.

IDEA Encryption Contd..

- Following the eighth round there is a final output transformation function that also takes as input
  - four 16-bit blocks, and
  - only four 16-bit subkeys, and
  - generates four 16-bit blocks that are concatenated together to form
  - the 64-bit block of cipher text.


Plaintext P (64 bits)
Key K (128 bits)
Subkey Generator
16
s1
s52
Round 1
Round 2
Round 8
Output Transformation
s1
s6
s7
s12
s42
s48
s49
s52
5 rounds
Cipher text C (64 bits)

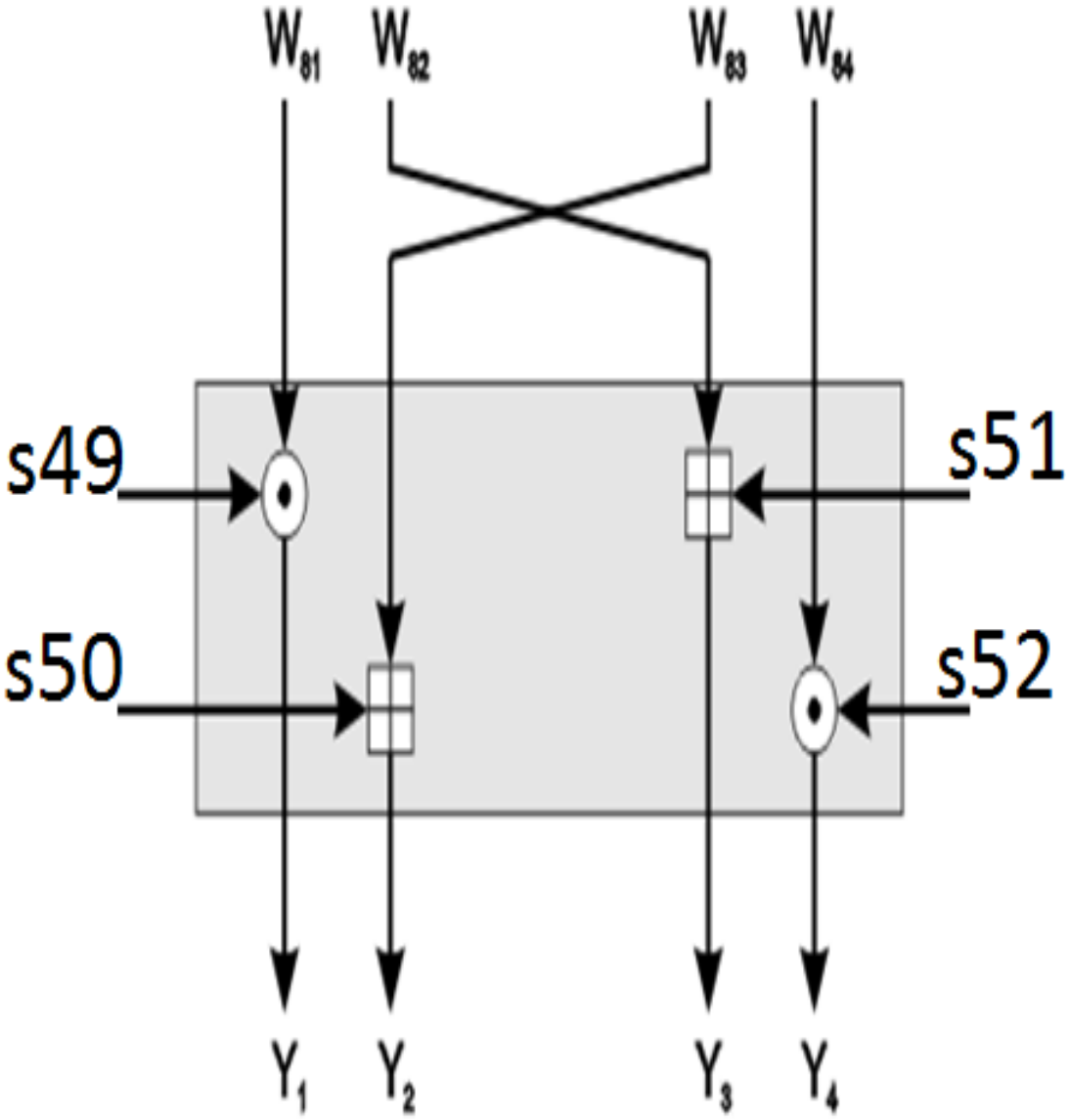


14 detail steps of IDEA

The Last Round

- At the end of the eighth round, there is an output transformation function, which has the same general structure as the beginning of a normal round.
  - Four subkeys are applied instead of 6
  - the second and third inputs are interchanged before being transformed.
- This effectively undoes the interchange at the end of the eighth round.
- This interchange is necessary, and exists so that encryption and decryption can be done using exactly the same algorithm.

The IDEA Last Round



Sub-Key Generation

- The eight rounds and the addition output transformation function require 52 subkeys, 16 bits each.

The function of generating sub keys

- The first eight subkeys are taken directly from the key, with S1 being the most significant 16 bits of the key, and S8 being the least significant 16 bits of the key. (16*8=128)

- Then a circular left shift of the key occurs by 25 bits, following which the next eight subkeys are extracted in the same way.

- This is repeated until the 52 required subkeys are generated.

- With this scheme, with the exception of the first round, some of the subkeys are generated before the 25 bit shift occurs, and some are generated after it occurs.

- This means that the sub-keys are not continuous, so there isn’t a simple relationship between them that can be exploited.

- The reason for this is that only six sub-keys are used in each round, whereas eight sub-keys are extracted with each rotation of the key.

Decryption

- The process of decryption in IDEA is virtually the same as encryption.
- The difference is in the decryption subkeys that are applied to each round.

IDEA Decryption

- The decryption subkeys are derived from the main key as follows:
  - The first four subkeys of decryption round i are derived from the first four subkeys of encryption round (10 – i), where the transformation stage is counted as round 9.
  - The first and fourth decryption subkeys are equal to the multiplicative inverse modulo of the corresponding first and fourth encryption subkeys.

- For rounds 2 through 8, the second and third decryption subkeys are equal to the additive inverse modulo of the corresponding third and second encryption subkeys.

- For rounds 1 and 9, the second and third decryption subkeys are equal to the additive inverse modulo of the corresponding second and third encryption subkeys.

- For the first eight rounds, the last two subkeys of decryption round i are equal to the last to subkeys of encryption round (9 – i).

- For the multiplicative inverse, the notation Zj-1 is used, and we have
  - Zj ⊙ Zj-1 = 1
- For the additive inverse the notation –Zj is used, and we have
  - - Zj ⌖ Zj = 0

Characteristics of IDEA

- To date, no method of cracking IDEA faster than exhaustive key search brute force has been discovered.
- Comparison of the algorithms
  - DES 56-bit key in 1 second
  - NSA Skipjack 80-bit key in 194 days
  - IDEA 128-bit key in 149745258842898 years
  - Software implementation speeds are comparable with those for DES.
  - Hardware implementations are just slightly faster.
- Keep the IDEA key safely, you lose it, you DESTROYED your data