Principles of Cryptography: Part 2

Gaurav S. Kasbekar
Dept. of Electrical Engineering
IIT Bombay

References

- J. Kurose, K. Ross, "*Computer Networking: A Top Down Approach*", Sixth Edition, Pearson Education, 2013
- C. Kaufman, R. Perlman, M. Speciner, "*Network Security: Private Communication in a Public World*", Pearson Education, 2nd edition, 2002

Block Ciphers

- Symmetric key based
- Used in many Internet protocols
  - *e.g.*, PGP (for securing email), SSL (for securing TCP connections)
- Examples of popular block ciphers used in Internet:
  - Data Encryption Standard (DES), 3DES, Advanced Encryption Standard (AES)
- Message to be encrypted broken into blocks of k bits each
  - *e.g.*, $k = 64$
- Each block encrypted independently
- *Encryption process*: one-to-one function used to map the k bit block of plaintext to a k bit block of ciphertext

Example

- $k = 3$; mapping in table used for encryption
- Plaintext: 010110001111
- Ciphertext:
  - 101000111001

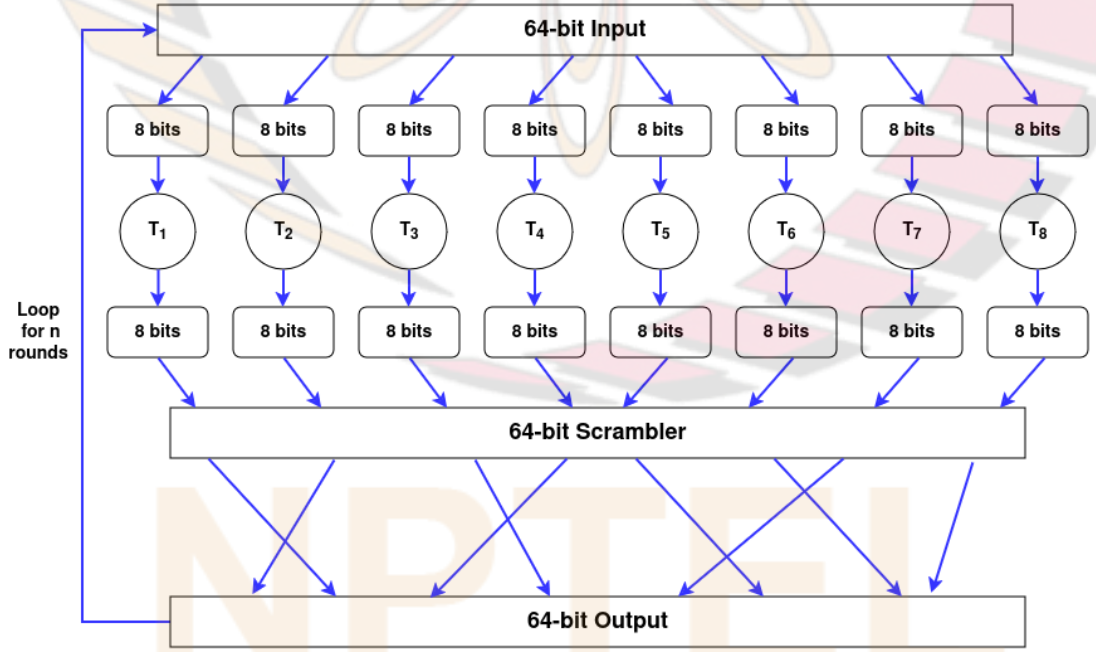
Input	Output	Input	Output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

Key in Block Cipher

- Key in a block cipher:
  - the one-to-one mapping used to generate ciphertext from plaintext
- Consider a k bit block cipher
- Number of possible mappings:
  - $(2^k)!$
- To defend against brute force attack that tries out all possible keys, k must be large
- Typical values: $k = 64$ or larger
  - $2^{64} \approx 1.8 \times 10^{19}$
- To implement a k bit block cipher using a look-up table:
  - 2^k rows required in the table; infeasible
- Alternative approach:
  - a one-to-one function that simulates a randomly permuted look-up table used

Example

- 64 bit input (plaintext) divided into chunks of 8 bits each
- Each chunk passed through a look-up table
- Outputs of the 8 chunks combined and scrambled (permuted) to produce a 64 bit intermediate output
  - ❑ this is input to the function
- n such rounds performed
- Key for this cipher:
  - ❑ the 8 look-up tables (assuming that the scrambling function is publicly known)



Objective in Design of Block Cipher

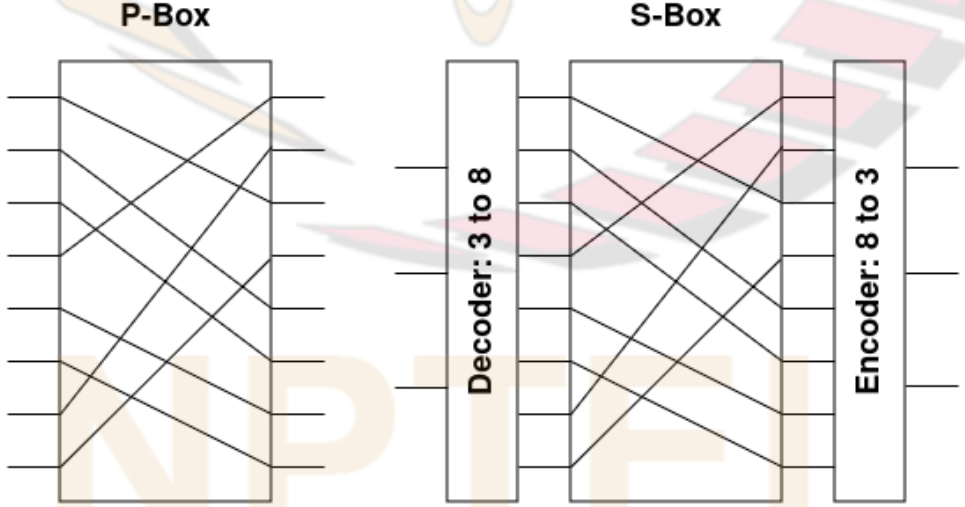
- To take a reasonable-length key (e.g., 64 bits long) and generate a one-to-one mapping that looks completely random to someone who doesn't know key
- For a block cipher of k bits, "completely random" means the following:
  - ❑ for first input of k bits, output selected from among the 2^k possible outputs uniformly at random
  - ❑ for next input of k bits, output selected uniformly at random from the remaining $2^k - 1$ possible outputs
  - ❑ and so on
- Advantage of completely random mapping:
  - ❑ if the input were mapped to output in predictable ways, may be possible for intruder to guess parts of plaintext by looking at ciphertext
  - ❑ e.g., if 10^{th} bit of output always changes when 5^{th} bit of input changes, then intruder who observes two ciphertext blocks that differ in 10^{th} bit would be able to find out that 5^{th} bit of input may have changed
  - ❑ e.g., if bits $17, \ldots 24$ of output are functions of only bits $57, \ldots, 64$ of input, then intruder who observes two ciphertext blocks with identical bits $17, \ldots 24$ could infer that corresponding plaintext blocks have the same bits in places $57, \ldots, 64$
- Hence, ideally, even if a single bit in input changes, new output should be a random string chosen independently of the old output

Objective in Design of Block Cipher (contd.)

- Recall: one way to generate a completely random one-to-one mapping is to use a lookup table (key) of size $k2^k$ bits
  - infeasible when $k = 64$ or larger
- Another possible approach: we define the mapping in terms of a set of linear equations:
  - Let x_j denote j'th bit of plaintext block and y_i denote i'th bit of ciphertext block, where $i, j \in \{1, \ldots, k\}$
  - Consider the block cipher defined by the k linear equations:
    - $y_i = \sum_{j=1}^{k} a_{ij}\, x_j$, where $a_{ij} \in \{0,1\}$
  - Key:
    - $a_{ij}, i, j \in \{1, \ldots, k\}$
  - So key is k^2 bits in length
  - Disadvantage of this cipher:
    - Vulnerable to cryptanalysis by attacker who is aware of structure of encryption algorithm and has somehow found out some plaintext-ciphertext pairs; he/ she can compute/ guess some elements of key
- Hence, instead, modern block ciphers are "*product ciphers*"
  - approximate a completely random mapping, use a small key (e.g., 64 bits long) and robust to cryptanalysis

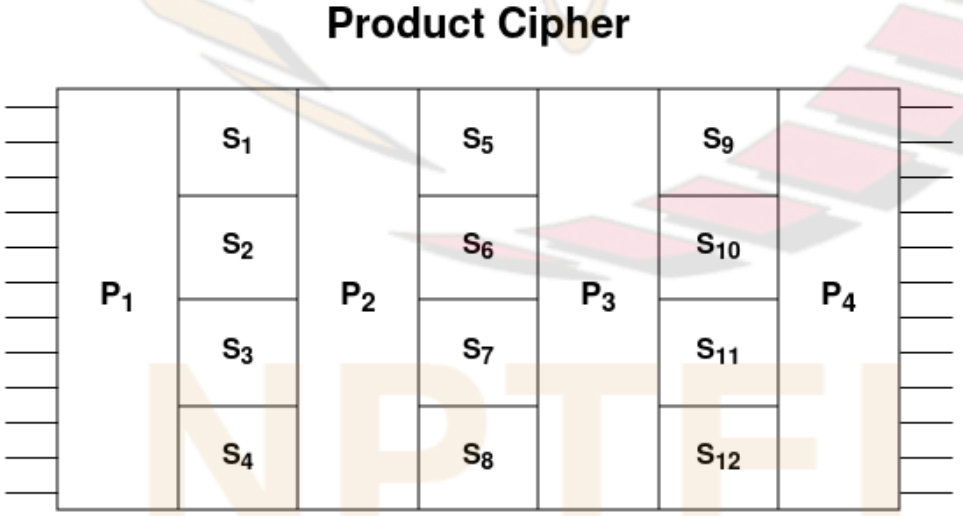
Product Cipher

- Building blocks of product cipher:
  - P-box (implements permutation)
  - S-box (implements substitution, i.e., look-up table)
- Fig. below shows hardware implementations of P-box and S-box
- No. of bits required to store k bit P-box:
  - $\approx k \log_2 k$
  - e.g., for $k = 64$, 384 bits required
- No. of bits required to store m bit S-box:
  - $m2^m$
  - e.g., too large (1.18×10^{21}) for $m = 64$, but small (2048) for $m = 8$



Product Cipher (contd.)

- Series of S-boxes and P-boxes cascaded to get a product cipher
- E.g.: in fig.:
  - 12 bits are permuted by the P-box P_1
  - next, they are broken into four groups of 3 bits, each of which is independently passed through a look-up table (S_1, S_2, S_3, S_4)
  - and so on
- By using sufficient number of stages in a product cipher:
  - the output can be made a highly complicated function of the input
  - approximates a completely random one-to-one function
  - in particular, each input bit affects all output bits
  - also, product cipher uses a small key (e.g., 64 bits long)



Product Cipher

- The example block cipher that we studied earlier (see fig. below) is a product cipher
- Several block ciphers used in the Internet are product ciphers
  - e.g., Data Encryption Standard (DES), 3DES, Advanced Encryption Standard (AES)