TRIARII

RESEARCH

Cryptography Workshop Lecture 1

Agenda

- Introduction
- Some cipher examples
- Stream cipher
- Block cipher

Introduction

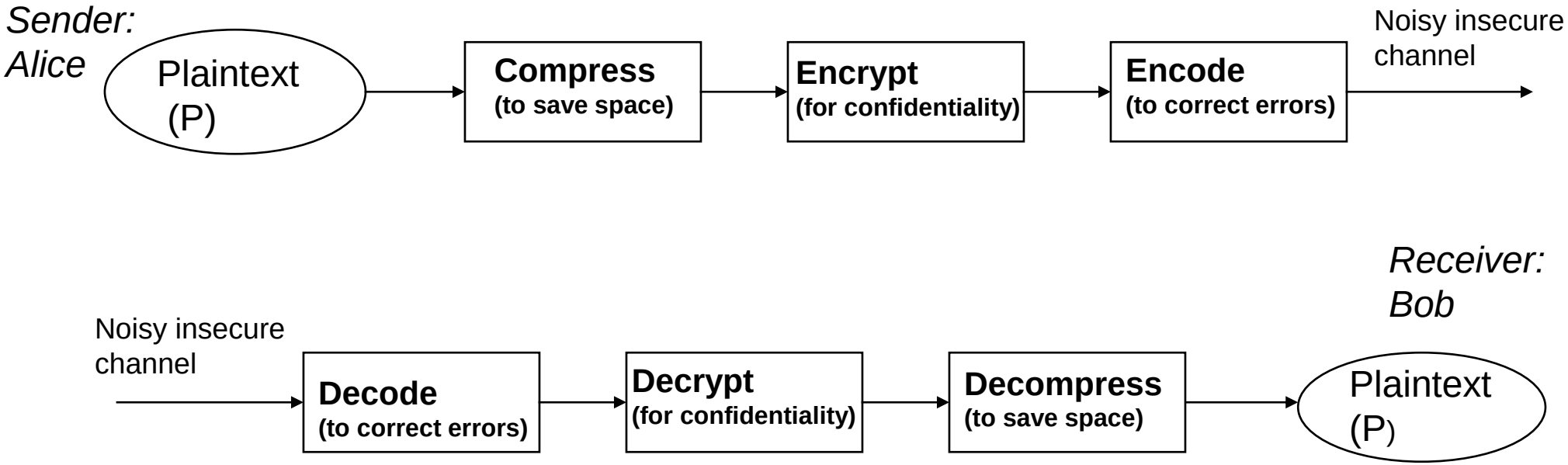
Introduction

- Past Cryptography- was considered as Art of solving and writing codes
- Modern Cryptography-A proven science Based on assumptions techniques(Digital data Security, Transactions and Distributed calculation).
- The goal: providing a reliable, authentic and secret communication
- Encryption- a mathematical scramble function
- Decryption- a mathematical unscramble function(matching the encryption)

Notation

- P-plaintext $p \in P$
- c-cipher text $c \in C$
- k_1- Encryption key
- k_2- Decryption key
- $k_1, k_2 \in K$
- $for\ every\ message\ m: D_{k_2}\left(E_{k_1}(\mathrm{P})\right) = \mathrm{P}$

Communication model



- Two parties: Alice and Bob
- Reliable communication line
- Shared encryption scheme: E,D, k_1, k_2
- Goal: Confidentially, Integrity, Authentication(CIA model)

Some Cipher Examples

Perfect cipher

- Message Space (m)- $\{0,1\}^n$
- Given a ciphertext C the probability (p_1) that $D_{k_2}(c) = P$ for any plaintext P is equal to the apriori probability (p_2) that P is the plaintext.
- $\Pr[Plaintext = P|C] = \Pr[plaintext = P]$ (p_1 = p_2).
- The probably is over the plaintext and the key spaces.

One time pad

- Plaintext Space - $\{0,1\}^n$
- Key space - $\{0,1\}^n$
- The scheme is symmetric, key k_1 is true random.
- $E_{k_1}(P) = \mathrm{C} = \mathrm{P} \oplus k_1$
- $D_{k_1}(C) = \mathrm{C} \oplus k_1 = P$

Pros and Cons

- Plaintext Space - $\{0,1\}^n$
- Key space - $\{0,1\}^n$
- The scheme is symmetric, key k_1 is true random.
- $E_{k_1}(P) = \mathrm{C} = \mathrm{P} \oplus k_1$
- $D_{k_1}(C) = \mathrm{C} \oplus k_1 = P$

Manual Cipher

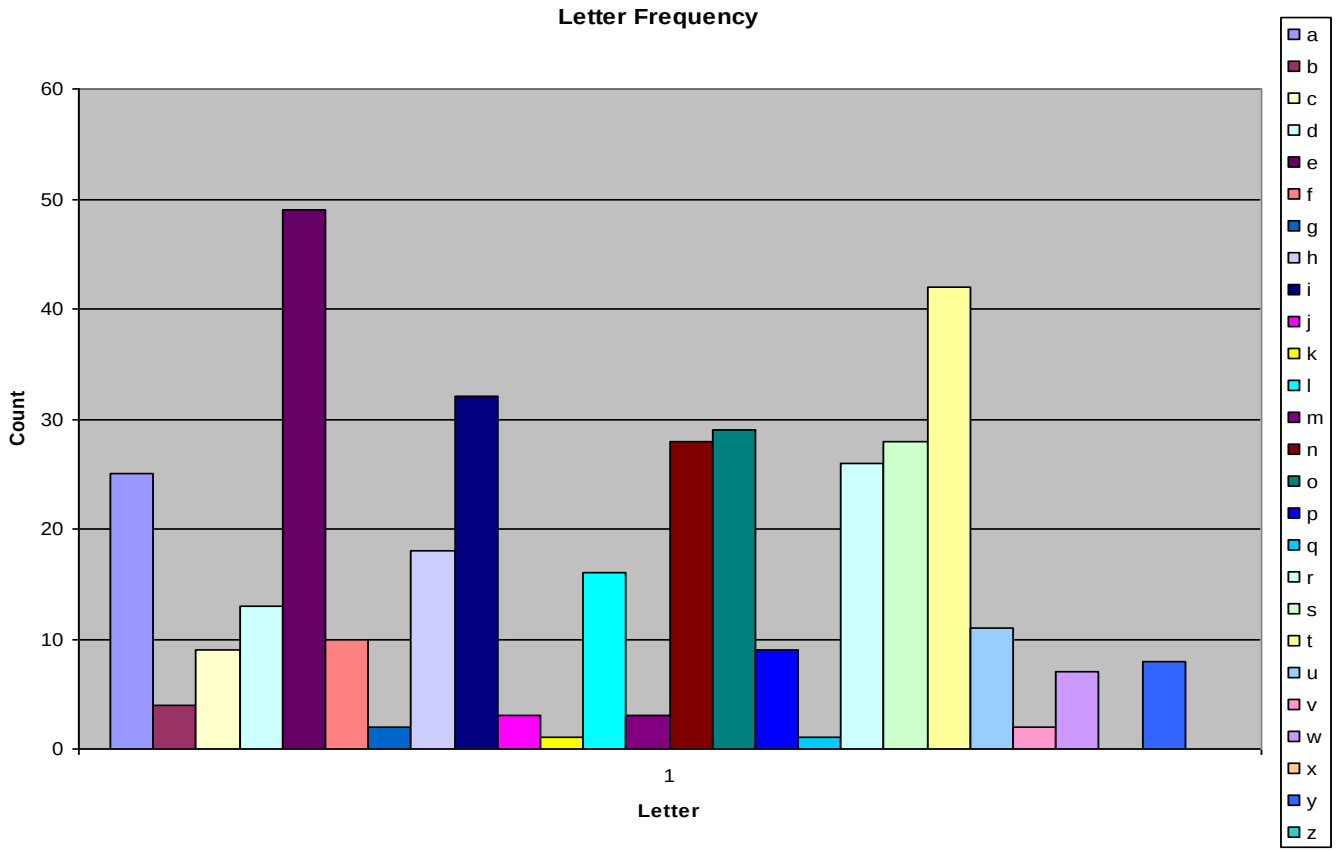
- Simple Transposition: Grilles, columnar transpositions.
- Monoalphabetic Substitution: Caesar, simple substitution
- Polyalphabetic substitution: Vigeniere

Vigeniere Cipher

6 Alphabet Direct Standard Example (Keyword: SYMBOL)

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ      PLAIN:  GET OUT NOW
--------------------------      KEY:    SYM BOL SYM
STUVWXYZABCDEFGHIJKLMNOPQR      CIPHER: YCF PIE FMI
          YZABCDEFGHIJKLMNOPQRSTUVWX
          MNOPQRSTUVWXYZABCDEFGHIJKL
          BCDEFGHIJKLMNOPQRSTUVWXYZA
          OPQRSTUVWXYZABCDEFGHIJKLMN
          LMNOPQRSTUVWXYZABCDEFGHIJK
```

Its all about statistics



Solving Vigeniere

- Determine Number of Alphabets
  - Repeated runs yield interval differences.
  - Number of alphabets is the gcd of these. (Kasiski)
- Statistics: Index of coincidence
- Determine Plaintext Alphabet
- Determine Ciphertext Alphabets

Stream cipher

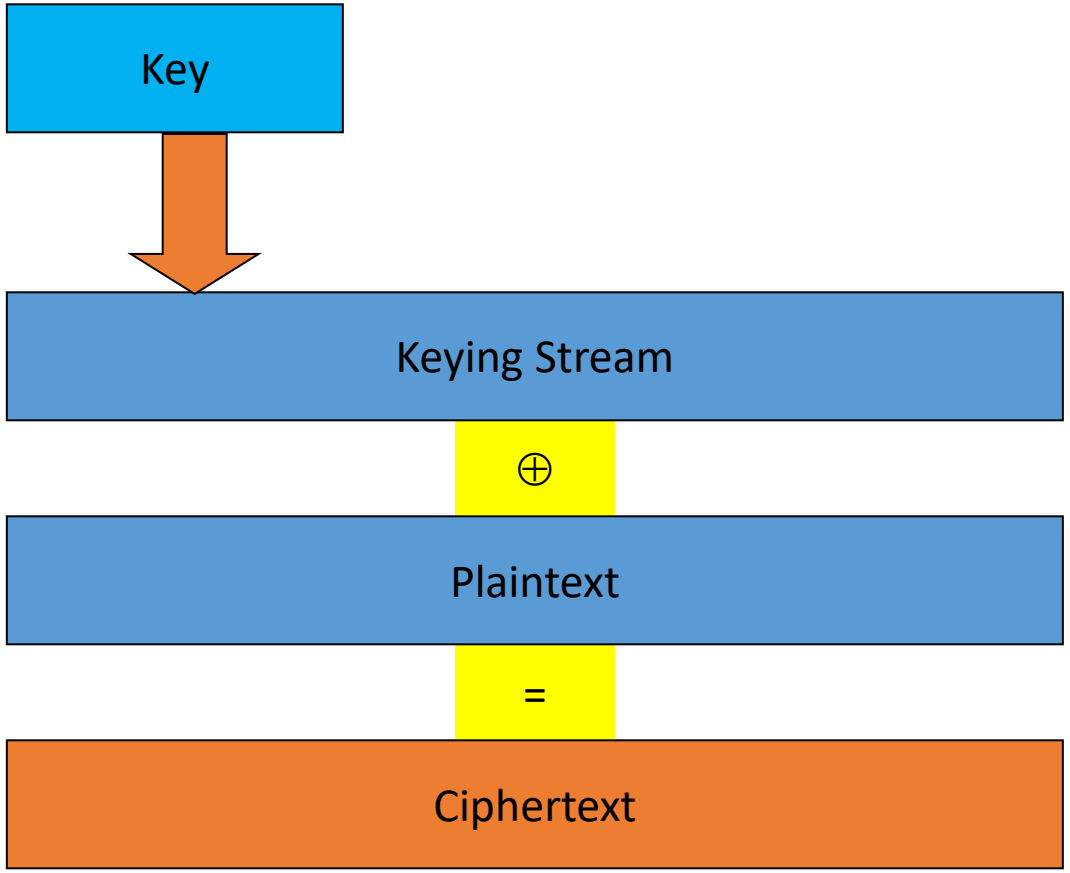
Stream Cipher properties

- Generalization of one-time pad
- A stream cipher takes an arbitrary text sequence of elements from the input (the plaintext) to the output(the cipher).
- Create a secret key(“seed”).
- Generate keying stream by stretching the key.
- Combine the stream with the Key plaintext to produce the ciphertext(usually XOR why?) like a one-time pad

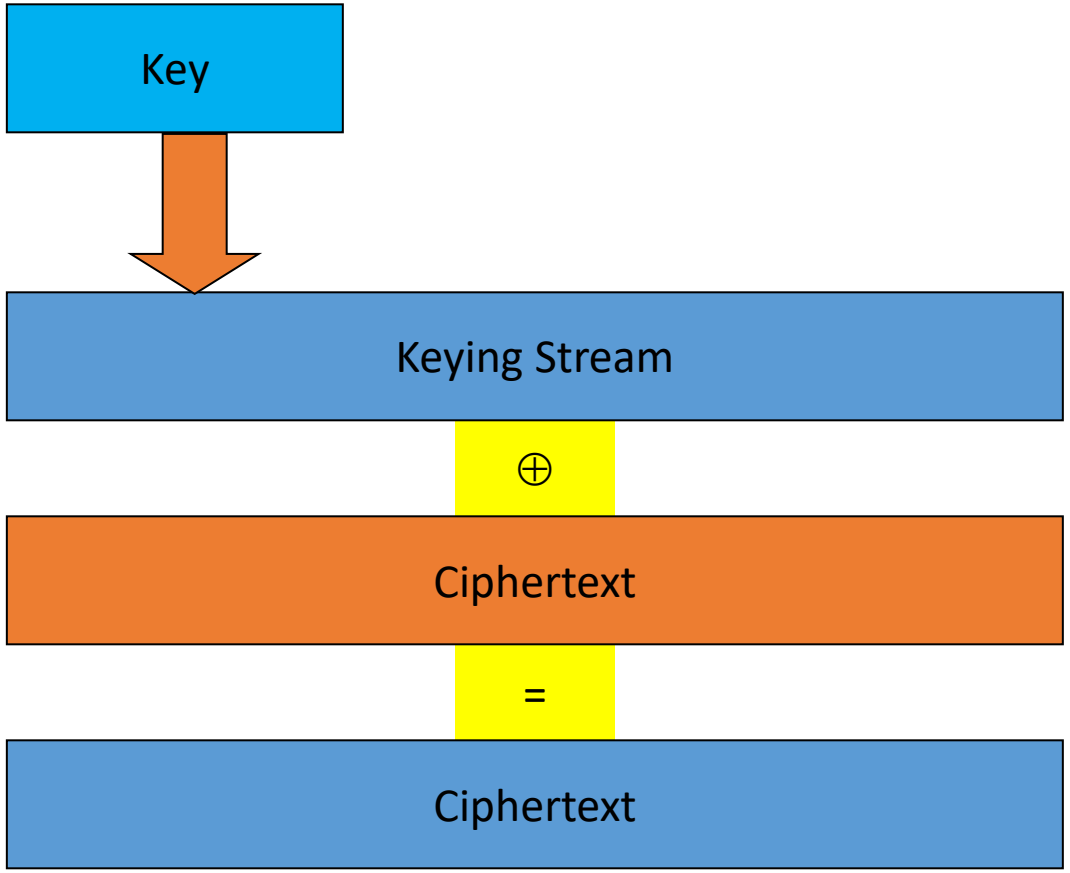
Stream Cipher Structure

- Randomness of stream key completely destroys statistically properties in message
- Must never reuse stream key
  - otherwise, can recover messages

Stream cipher encryption



Stream cipher decryption



Real stream cipher

- German enigma
- LFSR(Linear Feedback Shift Register)
- A5- Cellular GSM encryption
- RC-4/5

Stream cipher Terminology

- Stream cipher is synchronous if the key stream does not depended on the plaintext.
- Otherwise, cipher is asynchronous.

RC-4

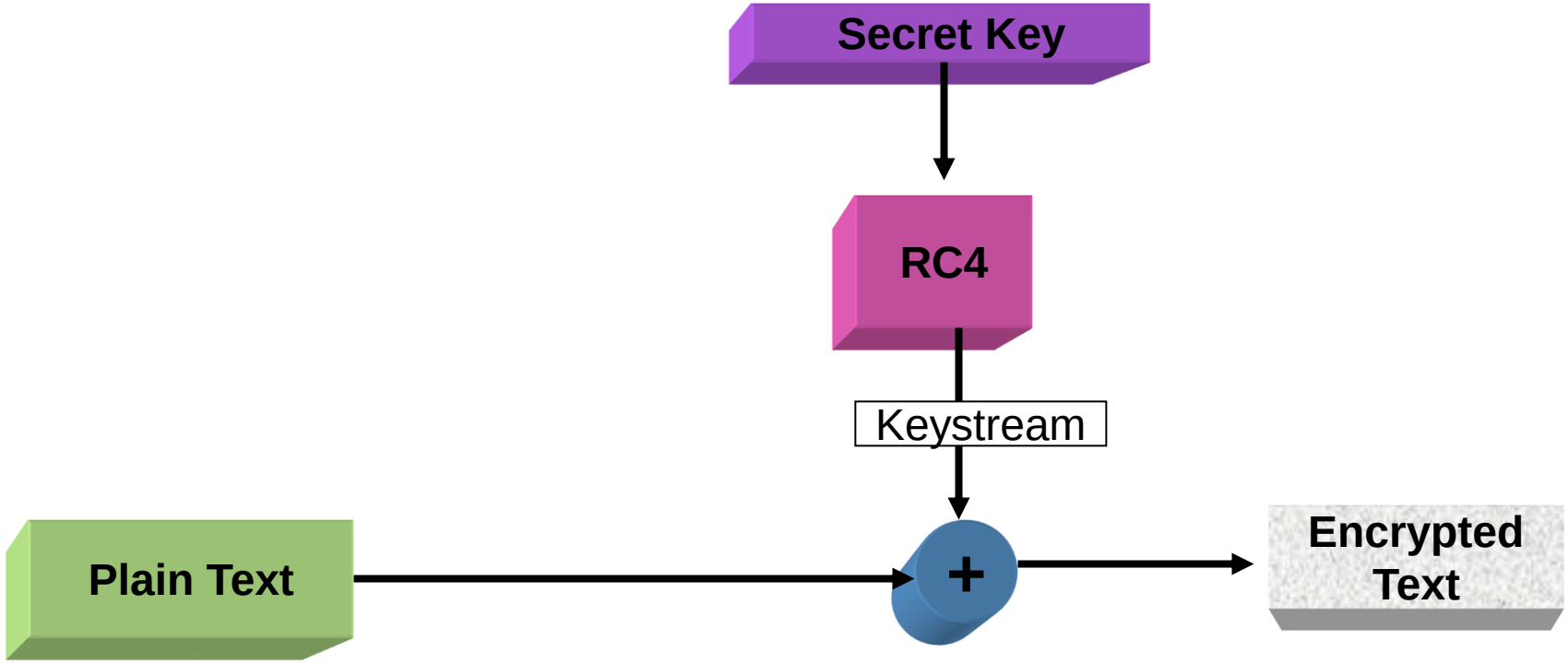
RC-4

- A symmetric key encryption algorithm invented by Ron Rivest
- A proprietary cipher owned by RSA, kept secret
- Code released anonymously in Cyberpunks mailing list in 1994
- Later posted sci.crypt newsgroup
- Variable key size, byte-oriented stream cipher
- Normally uses 64 bit and 128-bit key sizes.

RC-4

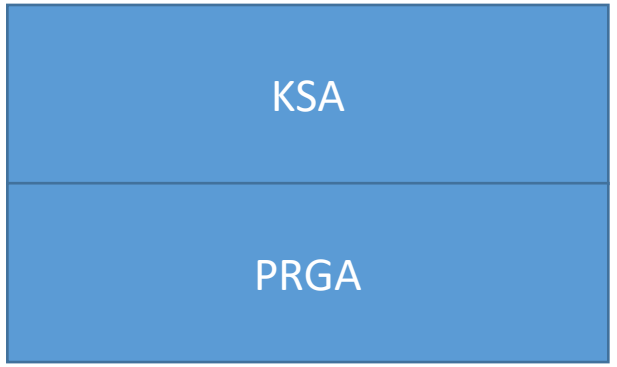
- Used in
  - SSL/TLS (Secure socket, transport layer security) between web browsers and servers,
  - IEEE 802.11 wirelss LAN std: WEP (Wired Equivalent Privacy), WPA (WiFi Protocol Access) protocol

RC-4 Block Diagram



RC-4 under the Hood

- Consists of 2 parts:
  - Key Scheduling Algorithm (KSA)
  - Pseudo-Random Generation Algorithm (PRGA)
- KSA
  - Generate State array
  - PRGA on the KSA
  - Generate keystream
  - XOR keystream with the data to generated encrypted stream



RC-4 KSA

- Use the secret key to initialize and permutation of state vector S, done in two steps
- Use 8-bit index pointers i and j

1

```
for i = 0 to 255 do
   S[i] = i;
   T[i] = K[i mod(|K|)]);
```

2

```
j = 0;
for i = 0 to 255 do
  j = (j+S[i]+T[i])(mod 256)
swap (S[i], S[j])
```

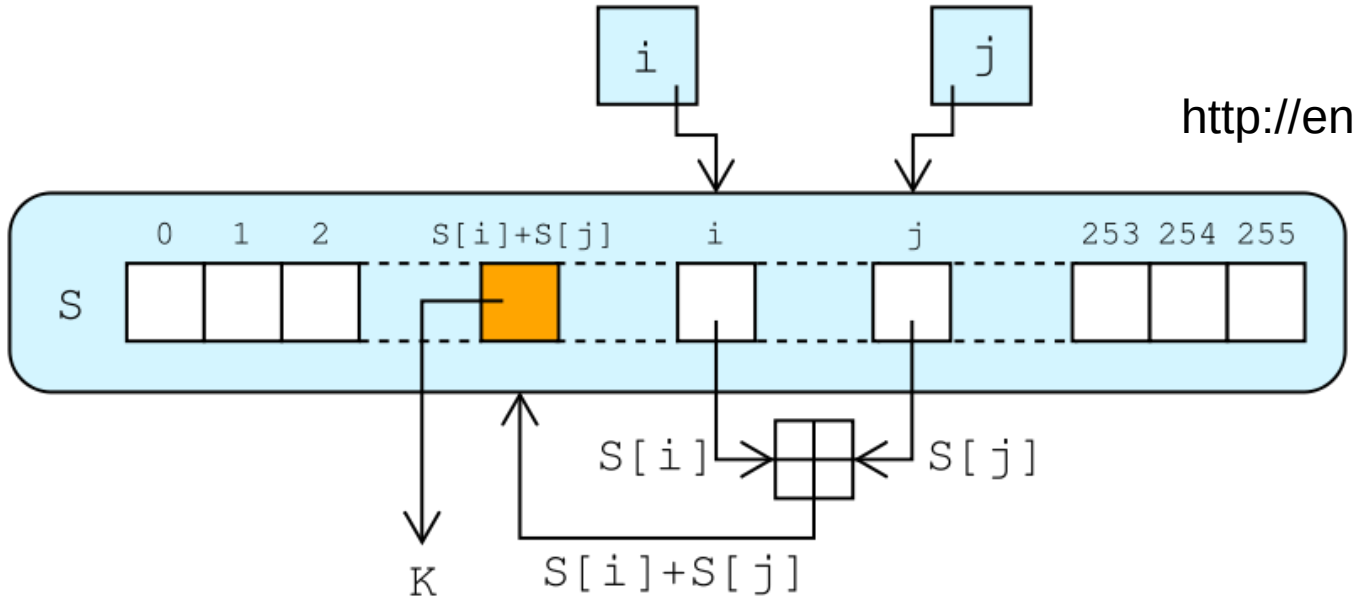
The PRGA

- Generate key stream k , one by one
- XOR S[k] with next byte of message to encrypt/decrypt

```
i = j = 0;
While (more_byte_to_encrypt)
    i = (i + 1) (mod 256);
    j = (j + S[i]) (mod 256);
    swap(S[i], S[j]);
    k = (S[i] + S[j]) (mod 256);
    C_i = M_i XOR S[k];
```

RC4 Lookup Stage

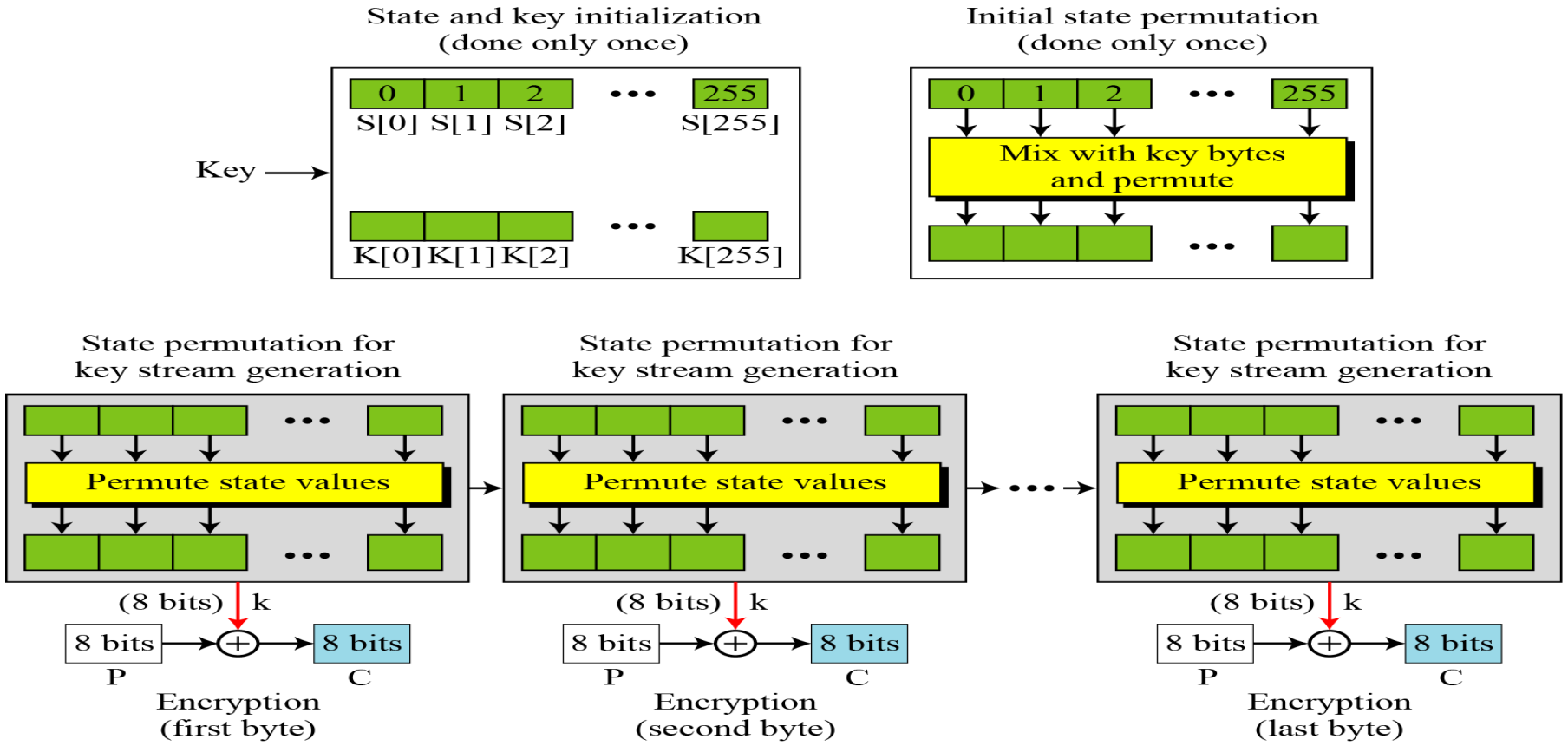
- The output byte is selected by looking up the values of S[i] and S[j], adding them together modulo 256, and then looking up the sum in S
- S [S[i] + S[j]] is used as a byte of the key stream, K



http://en.wikipedia.org/wiki/File:RC4.svg

```
i = j = 0;
While (more_byte_to_encrypt)
      i = (i + 1) (mod 256);
      j = (j + S[i]) (mod 256);
      swap(S[i], S[j]);
      k = (S[i] + S[j]) (mod 256);
      C_i = M_i XOR S[k];
```

Overall Operation of RC4



Decryption using RC4

- Use the same secret key as during the encryption phase.
- Generate keystream by running the KSA and PRGA.
- XOR keystream with the encrypted text to generate the plain text.
- Logic is simple :

(A xor B) xor B = A

A = Plain Text or Data

B = KeyStream

Block Cipher

Block Cipher

Block Cipher

- Encrypt a block of input to a block of output
- Break the data to the encryption block size if needed padding is being used.
- Most symmetric key systems block size is 64
- AES block size is 128
- Different modes for encrypting plaintext longer than a block

Block cipher mode

- Electronic code book(ECB)
- Cipher block chaining(CBC)
- Cipher feed back(CFB)
- Output feed back(OFB)
- Counter(CTR)

Electronic code book(ECB)

- Cipher block chaining(CBC)

Cipher block chaining(CBC)

- Cipher feed back(CFB)

Cipher feed back(CFB)

- Output feed back(OFB)

Output feed back(OFB)

- Counter(CTR)

Counter(CTR)

- Counter(CTR)