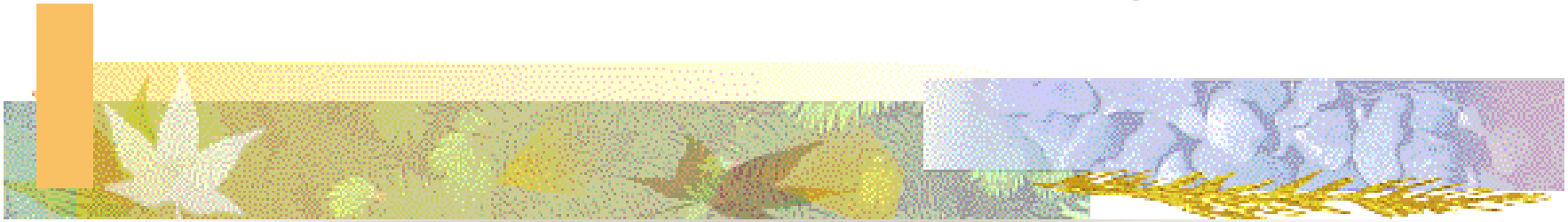
Introdução à Criptografia



Profa. Yeda

Aula 03 – Cifra de Mascaramento RC4

(Cap. 6 Stallings)

Cifra de Mascaramento (Stream)

- Processa elementos de entrada continuamente.
- Expansão de uma chave curta para uma seqüência de bits do tamanho da mensagem.
- *Stream cipher*:

$k_i = f(K, i)$ -- derivação de chave

$C_i = M_i \oplus k_i$ -- cifração

$M_i = C_i \oplus k_i$ -- decifração

- Na prática, agrupam-se os bits em blocos (por exemplo, em bytes).

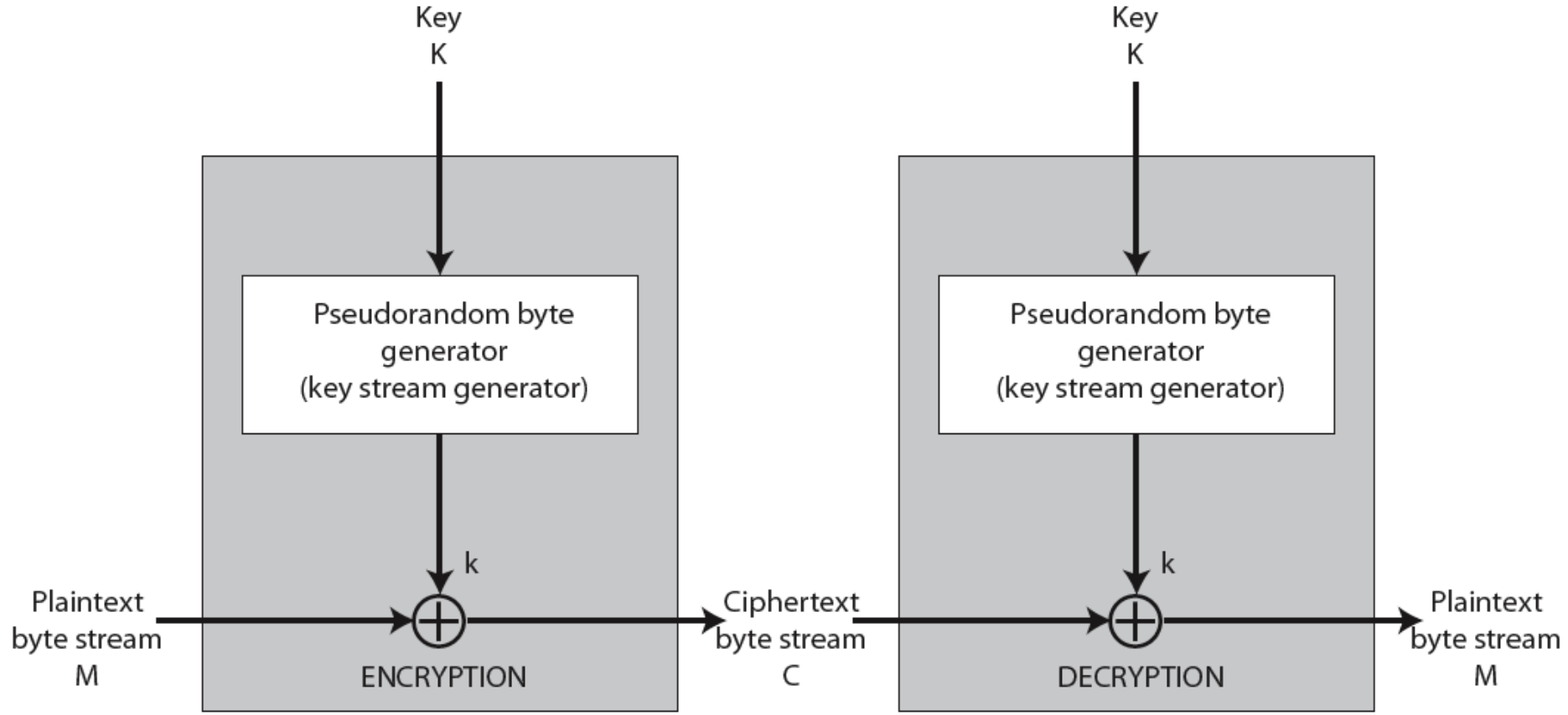
Cifra de Mascaramento (Stream)

- A transformação opera em mensagens de qualquer tamanho.
- O tamanho do texto cifrado é o mesmo do texto claro.

$$E: \{0, 1\}^m \times \{0, 1\}^* \rightarrow \{0, 1\}^*$$

- A cifração de cada bit da mensagem altera o estado interno do algoritmo (memória).

Cifra de Mascaramento



RC2 / RC4 / RC5

- Foram desenvolvidos por Ronald Rivest e mantido como segredo da RSA Data Security.
- Os algoritmos foram revelados publicamente através de uma publicação anônima na USENET.
- A implementação do RC2 vendida pela RSA permite o uso de chaves de tamanho variável entre 1 e 2048 bits.
- A chave criptográfica do RC4 pode variar de tamanho entre 1 e 2048 bits.
- O RC5 permite o uso de um tamanho de chave qualquer definida pelo usuário

RC4

- “Rivest Cipher #4”.
- Algoritmo proprietário não divulgado – reconstituído por engenharia reversa pública.
- Padrão em SSL (Secure Socket Layer).
- Chave de tamanho variável:
  - múltiplo de 8 bits,
  - tamanho máximo de 2048 bits (256 bytes)
- Tabela S representando o estado interno do algoritmo
  - inicializada com a chave e
  - atualizada durante a operação.

RC4: Inicialização

- Tabela preenchida com valores seqüenciais:

$$S[i] \leftarrow i\ , \qquad i = 0, 1, \ldots, 255.$$

- As células são permutadas em pares:

$$S[i] \leftrightarrow S[p]\ , \qquad i = 0, 1, \ldots, 255.$$

- O índice p que determina o par S[p] de cada célula S[i] é escolhido em função:
  - histórico de inicialização,
  - conteúdo corrente da tabela e bytes da chave K.

RC4: Inicialização

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

Estado interno inicial

Permuta o vetor de estado interno

RC4: Geração do Fluxo

- As células da tabela são permutadas em pares:
- S[i] $\leftrightarrow$ S[p]
  - Os índices permutados i e p são escolhidos em função
    - posição do byte da mensagem a ser cifrado,
    - histórico de operação do algoritmo e
    - conteúdo corrente da tabela.
- Um byte de mascaramento é calculado a partir das células permutadas e da posição do byte da mensagem a ser cifrado.
  - z $\leftarrow$ (S[i] + S[p]) mod 256.

RC4: Geração do Fluxo

```
i = j = 0
for each message byte Mk
    i = (i + 1)mod 256
    j = (j + S[i])mod 256
    swap(S[i], S[j])
    t = (S[i] + S[j])mod 256
    Ck = Mk XOR S[t]
```

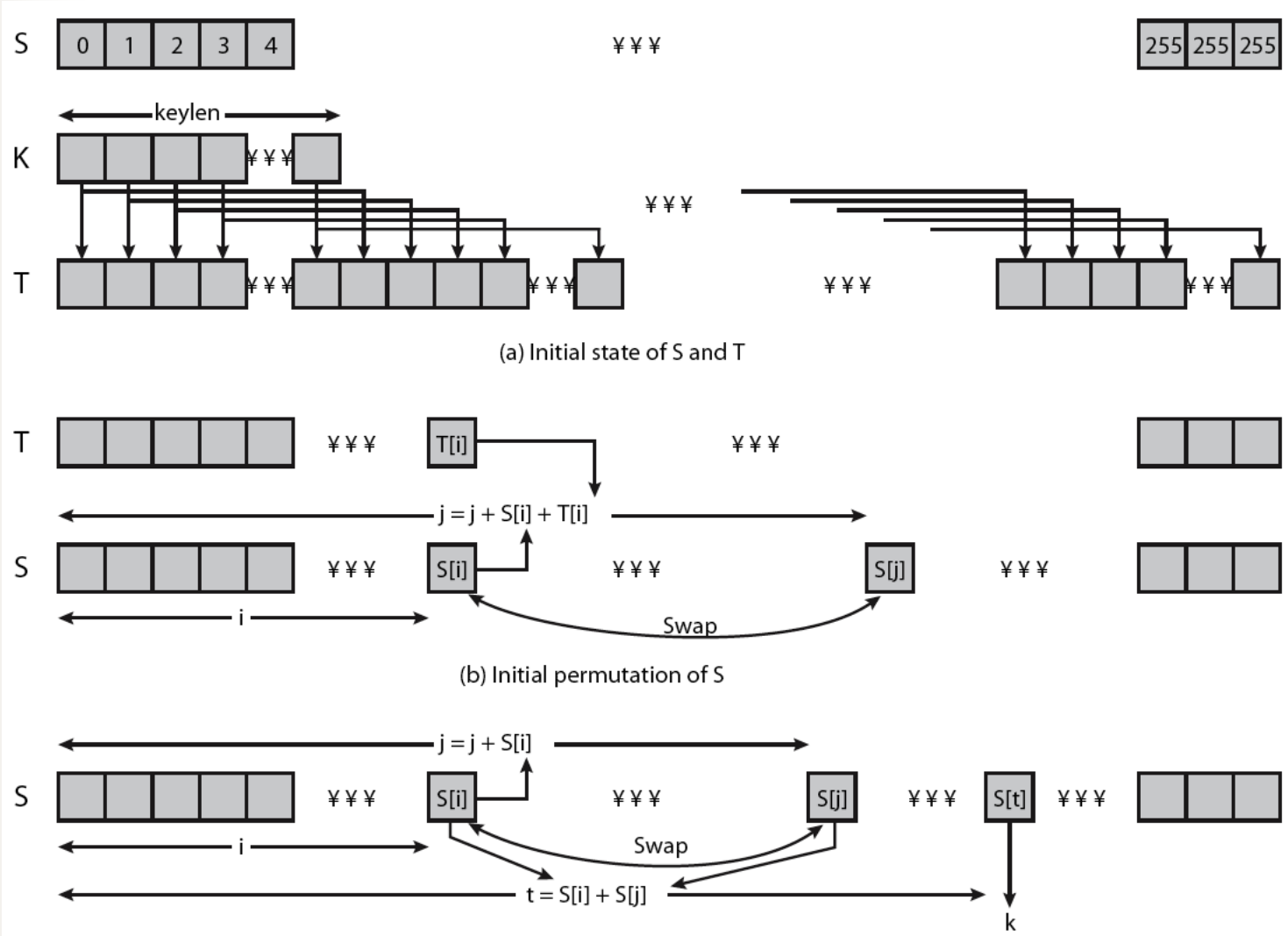
Índice de permutação

Nova permutação

Byte de mascaramento

Criptografia

RC4: Geração do Fluxo



Vulnerabilidade do Sistema

- A segurança de qualquer cifra de mascaramento baseia-se na hipótese de que a **chave de fluxo é utilizada uma única vez**.

- Motivo: a diferença entre mensagens cifradas é igual à diferença entre as mensagens claras correspondentes.

Vulnerabilidade do Sistema

$$\left.\begin{aligned} C_i &= M_i \oplus k_i \\ C_i^* &= M_i^* \oplus k_i \end{aligned}\right\} \Rightarrow M_i^* \oplus M_i = C_i^* \oplus C_i$$

- Conseqüentemente, se uma mensagem clara M for acidentalmente comprometida, qualquer outra mensagem M* pode ser recuperada pela relação:

$$M^* = M \oplus C^* \oplus C$$

Conseqüências

- Embora este exemplo utilize o RC4, *qualquer* cifra de mascaramento sofreria o mesmo problema.

- A vulnerabilidade não está no algoritmo, mas em sua utilização imprópria.

Outras cifras de mascaramento

- A5 (GSM → quebrável em tempo real).
- Helix (confidencialidade + integridade → recentemente enfraquecido por um ataque de esforço 288 passos que recupera chaves de até 256 bits).
- Resultados recentes indicam ser extremamente difícil projetar uma cifra de mascaramento realmente sólida.

Cifra de Bloco vs Mascaramento

- Cifra de blocos processam mensagens em blocos, cada qual é então cifrada/decifrada
  - Como uma substituição sobre vários caracteres
  - 64-bits ou mais
- Cifra de mascaramento processa mensagens bit a bit (ou byte) por vez quando cifra/decifra
- Muitos cifradores atuais são de blocos
- Com ampla faixa de aplicações

ATIVIDADE 02 – RC4

- Implementar o algoritmo de criptografia RC4
  - Preparar o algoritmo para receber uma sequência de texto cifrado em Hexadecimal e uma chave no formato string (sequência de caracteres)
  - Preparar o algoritmo para entregar na saída o texto plano (string).
  - Na próxima aula será entregue um arquivo txt com o texto cifrado em hexadecimal e a chave utilizada. Você deverá decifrar com seu código.