Cifras de Fluxo

Gerência e Segurança de Redes

Objetivos de Aprendizagem

- Introduzir o conceito de cifras de fluxo
- Apresentar exemplos de implementação

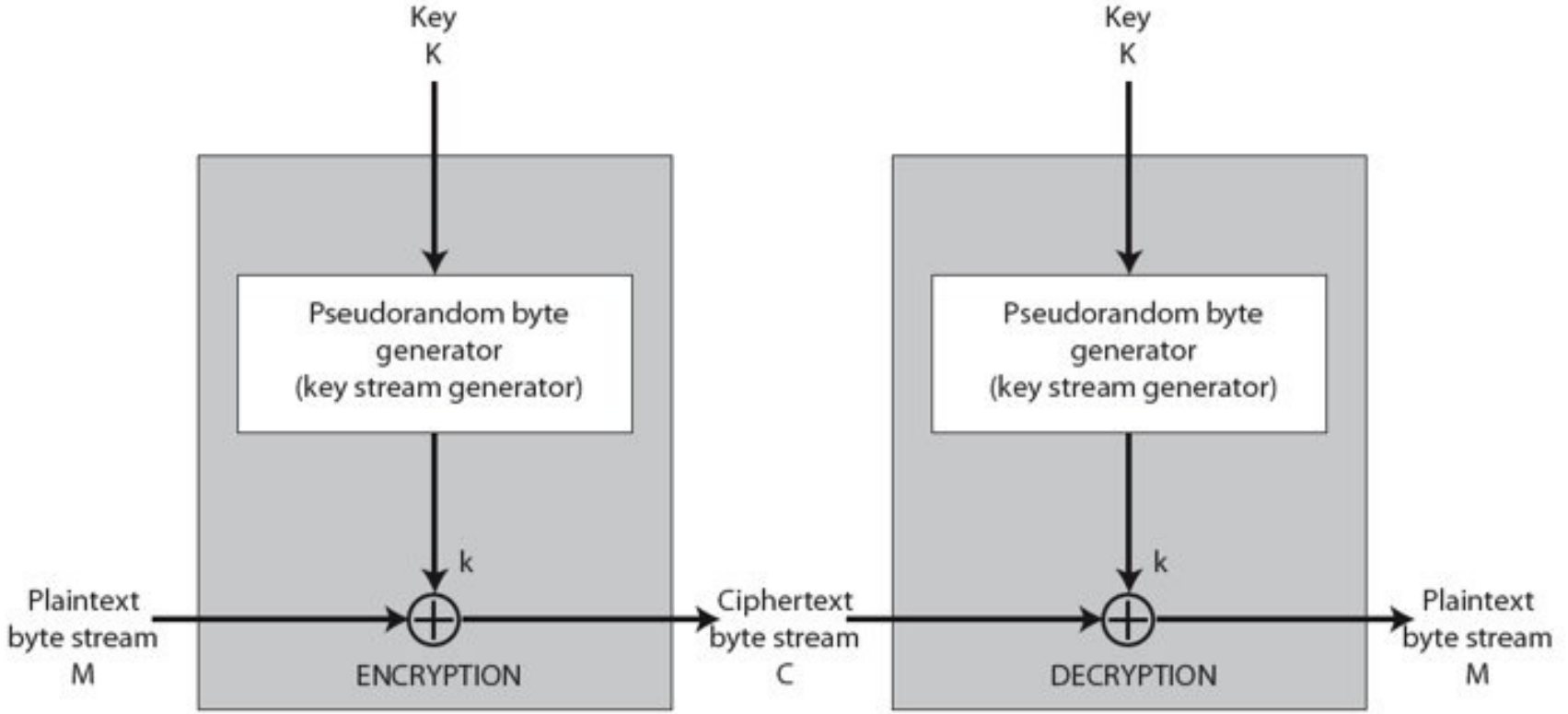
Agenda

- Estrutura
- Cifras de Bloco x Cifras de Fluxo
- RC4
- Aplicações
- Vulnerabilidades

Cifra de Fluxo

- CF típica criptografa um *byte* por vez
- Comumente aplicada em canais de comunicação ou aplicações cliente/servidor
- A estrutura básica consiste em:
  - Chave de entrada para o gerador de bits pseudo-aleatório (K)
  - Fluxo de bits (k), saída do gerador de bits
  - Função XOR
  - Texto claro
  - Texto cifrado

Estrutura


Key
K
Pseudorandom byte generator (key stream generator)
k
Plaintext byte stream M
ENCRYPTION
Ciphertext byte stream C
Key
K
Pseudorandom byte generator (key stream generator)
k
DECRYPTION
Plaintext byte stream M

Exemplo

Criptografia

	1	1	0	0	1	1	0	0	Texto claro
	0	1	1	0	1	1	0	0	Fluxo de chave
XOR	1	0	1	0	0	0	0	0	Texto cifrado

	1	0	1	0	0	0	0	0	Texto cifrado
	0	1	1	0	1	1	0	0	Fluxo de chave
XOR	1	1	0	0	1	1	0	0	Texto claro

Decriptografia

Gerador de Números Pseudo-Aleatórios

Fluxo de chave

Um fluxo pseudo-aleatório é aquele que é imprevisível sem conhecimento da chave de entrada

Gerador de Números Aleatórios

One Time Pad

Um fluxo **genuinamente aleatório** é aquele que é totalmente imprevisível. Mais seguro quando comparado a um fluxo pseudo-aleatório

Por que não utilizar um gerador genuinamente aleatório?

Cifras de Fluxo

- O gerador de números pseudo-aleatório utiliza funções determinísticas e periódicas para geração do fluxo
- Quanto maior o período, maior a dificuldade de criptoanálise
- Quanto mais próximo do fluxo genuinamente aleatório, maior a dificuldade de criptoanálise
- Chave K de pelo menos 128 bits
- Segurança da CF é comparável às CB

Vantagens

- Implementação simplificada compara às CB
- Execução rápida
  - DES, 9Mbps
  - 3DES, 3Mbps
  - RC4, 45Mbps
- Reutilização de chaves
- Aplicação
  - Canais de comunicação
  - Cliente/Servidor
  - Criptografia em unidades de armazenamento

Cifras de Fluxo Típicas

- Cifra de Vernam
- Salsa20
- RC4

Cifra de Vernam

- Uma das primeiras CF, proposta por Gilbert Vernam em 1917
- Utiliza chave secreta do mesmo tamanho dos dados originais
- Realiza a operação XOR bit a bit entre o fluxo de chave e os dados
- A cifra de Vernam é considerada inquebrável se a chave for utilizada apenas uma vez e for verdadeiramente aleatória
- Generalização do *One Time Pad*

Salsa20

- O Salsa20 é um Stream Cipher projetado por Daniel Bernstein em 2005
- Conhecido por sua segurança e eficiência, sendo utilizado em diversas aplicações, como criptografia de disco e comunicações seguras
- O Salsa20 é considerado um algoritmo criptográfico bastante seguro
- Utiliza chaves de 128, 192 e 256 bits

Salsa20

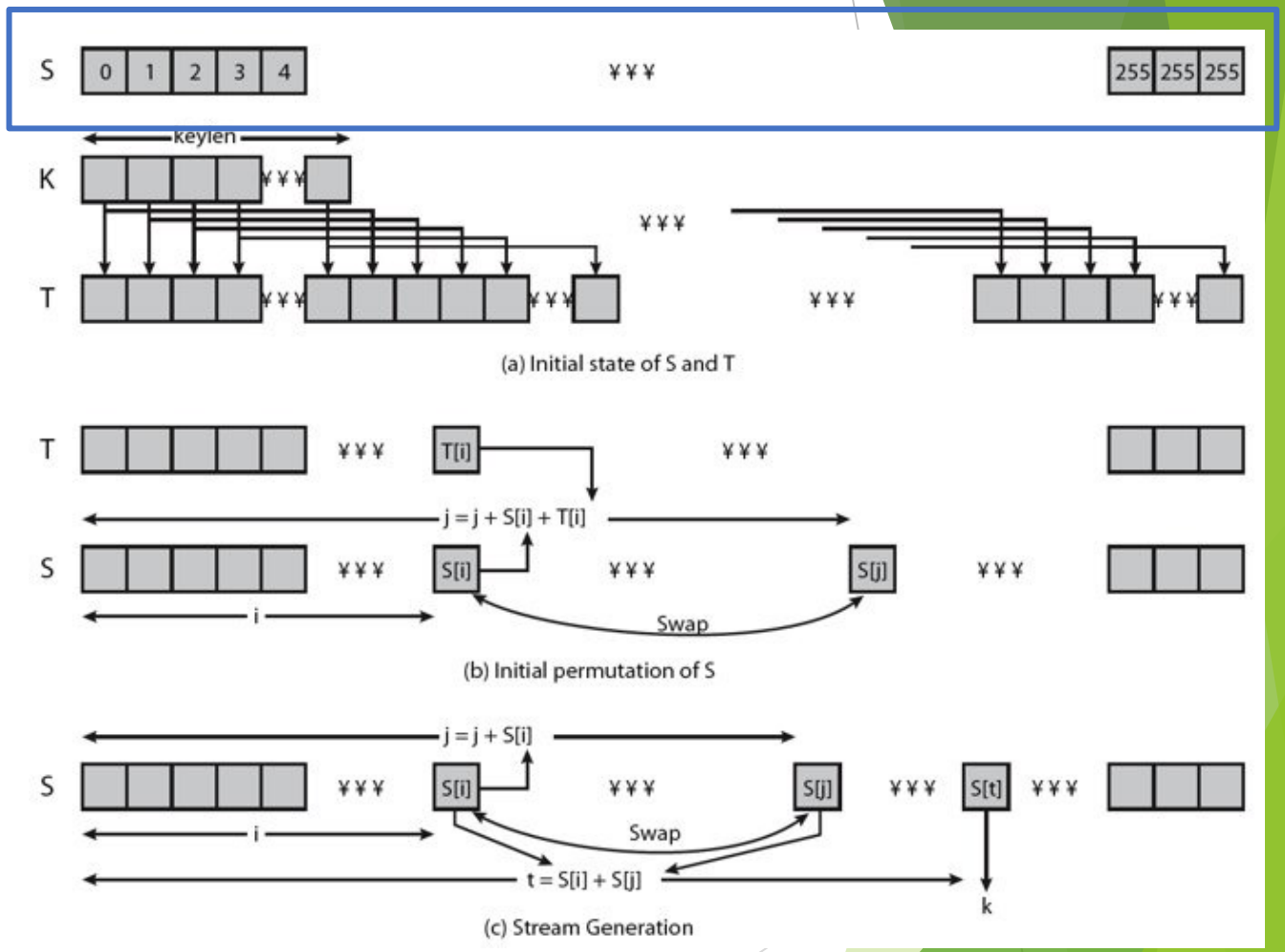
- ***Number used Once*** (*nonce*) é um número aleatório utilizado uma única vez em um algoritmo criptográfico
- Salsa20 utiliza um *nonce* de 64 bits como entrada junto com uma chave secreta (128, 192 ou 256 bits)
- Chave secreta + *nonce* = Fluxo de chaves
- Fluxo de chaves $\oplus$ mensagem = Texto Cifrado (Criptografia)

RC4

- Rivest Cipher 4, criado por Ron Rivest para a RSA (1994)
- Tamanho de chave variável
- Fundamenta-se na permutação aleatória
- Baixa demanda computacional 8 a 16 operações de processador por byte de saída
- Aplicações
  - TLS 1.0
  - WEP (*Wired Equivalent Private*)
  - WPA (*WiFi Protectec Access*)

RC4

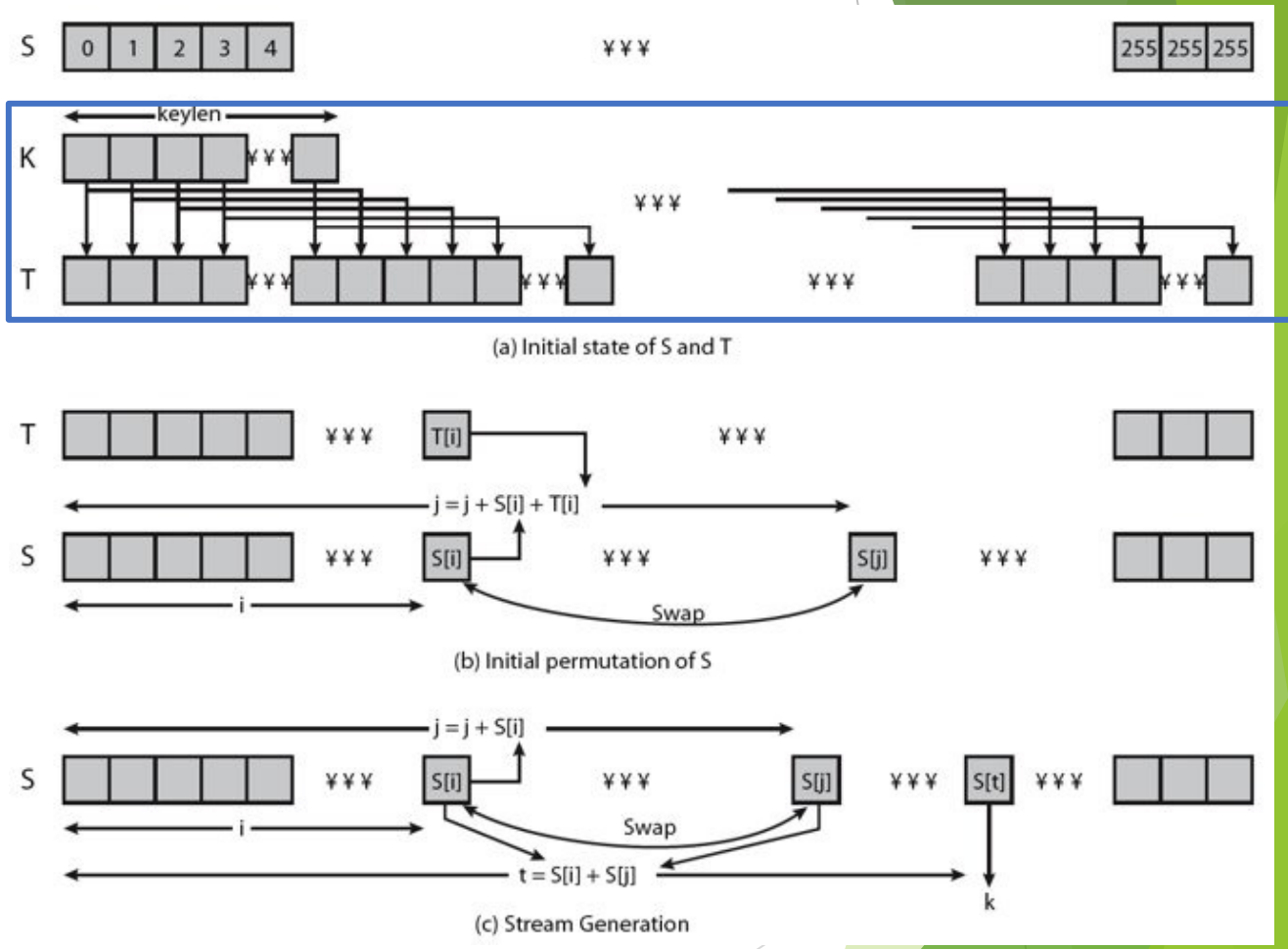
- Chave de tamanho variável
  - 1 a 256 Bytes (8 a 2048 bits)
- A chave inicializa o vetor de estados S de 256 bytes
- S[0]=0, … , S[255]=255



RC4

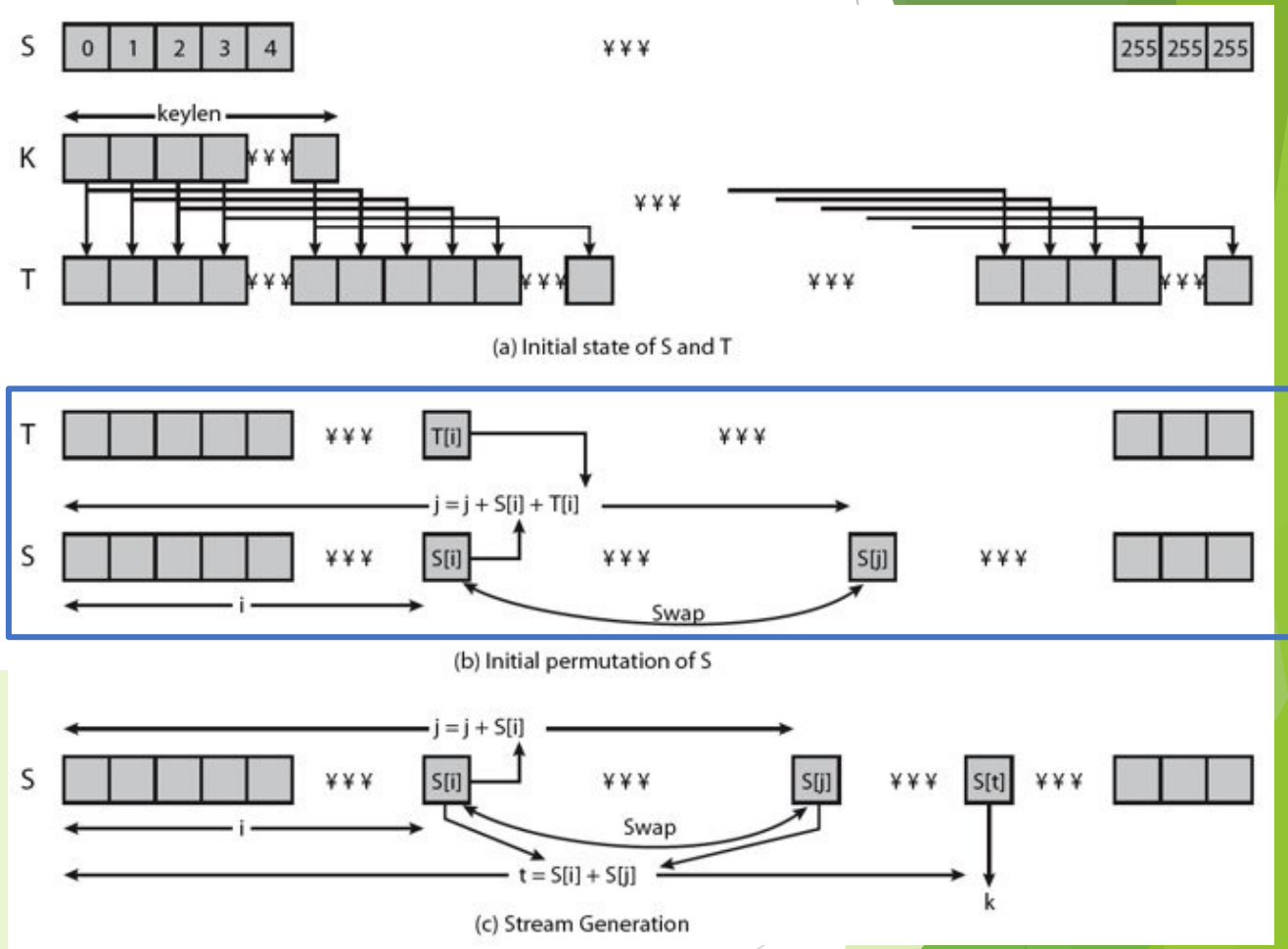
- Um vetor temporário T
- Se K tiver 256 bytes, T = S
- Caso contrário K é replicado em T até que o tamanho de T seja o mesmo de S
- K é utilizada apenas para inicializar S

```
for i = 0 to 255 do
  S[i] = i;
  T[i] = K[i mod keylen];
```



RC4

- T é utilizado para causar permutações em S, conforme abaixo

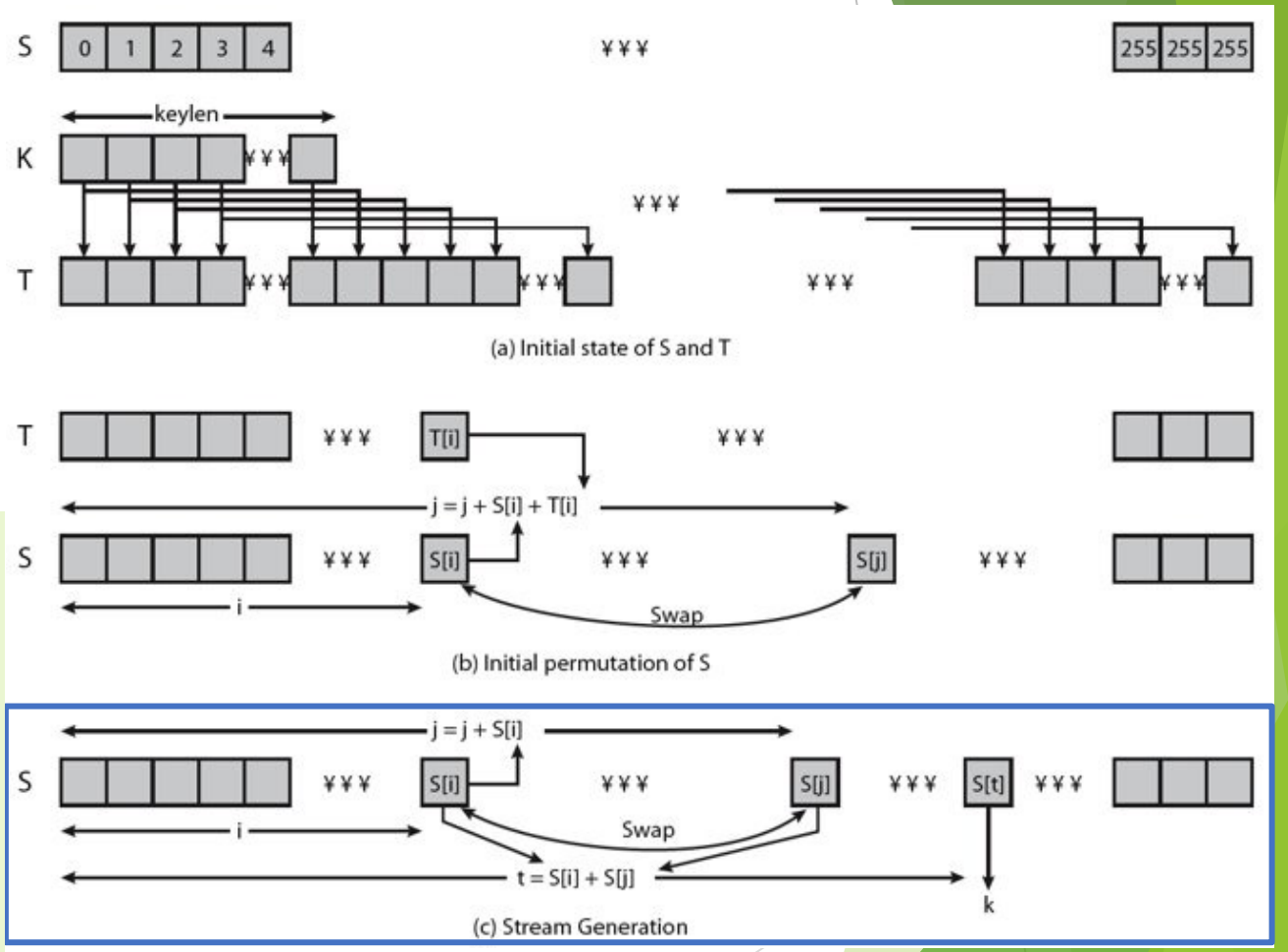



```
j = 0
for i = 0 to 255 do
  j = (j+S[i]+T[i]) mod 256
  Swap (S[i],S[j]);
```

RC4

- A geração de fluxo se dá percorrendo cada elemento S[i] e realizando a permutação

```
i, j = 0
while (true)
  i = (i + 1) mod 256;
  j = (j+S[i]) mod 256
  Swap (S[i],S[j]);
  t = (S[i], S[j]) mode 256;
  k = S[t];
```



Vulnerabilidades Clássicas

- WEP
  - Confidencialidade em redes sem fio 802.11
  - Vulnerabilidade na geração de chaves do RC4 na implementação do WEP
- BEAST
  - Browser Exploit Against SSL/TLS
  - Vulnerabilidade existente nas cifras de bloco no modo CBC nas suítes SSLv3 e TLS 1.0
  - O alvo desse ataque é comprometer a confiencialidade das conexões HTTPS

WEP

- *Wired Equivalent Privacy*
- Implementado em 1997 em diversos APs como primeiro método popular de proteção de redes sem fio
- Baseia-se no cifra de fluxo RC4
- Versões de 64 ou 128 bits
  - WEP 64 bits -> chave de 40 bits
  - WEP 128 bits - > chave de 104 bits
- A principal falha consiste na implementação dos Initialization Vectors (IVs) reutilizando-os
- A **não** reutilização de chaves é um princípio básico a ser seguido em segurança da informação

WEP

- Tamanhos de chave relativamente pequenos facilitando a quebra
  - Quebra de chaves de 40 bits em minutos
- Chave simétrica e única durante toda a comunicação
- Vulnerável a ataques de repetição

BEAST

- Condições para um ataque bem sucedido
  - Versão vulnerável da TLS (v3 ou inferior) ou TLS (1.0 ou inferior)
  - Atacante deve ser capaz de escutar o tráfego entre cliente e servidor (man-in-the-middle)
  - Atacante deve ser capaz de injetar tráfego conhecido na origem (Applet ou Javascript, por exemplo)

RC4 como solução ao BEAST

- Explora uma fragilidade nas cifras de bloco utilizando o modo CBC relacionada com o vetor de inicialização
- Uma das recomendações de mitigação consiste em evitar o uso de CB e utilizar CF
- A única CF especificada na TLS 1.0 é exatamente o RC4
- O uso do RC4 em substituição às cifras de bloco de modo CBC
- Entretanto, a partir de 2013, diversas publicações apresentaram fragilidades no RC4

Vulnerabilidades RC4

- Viés estatístico nos primeiros bytes do fluxo da cifra, permitindo atacantes "prever" o fluxo
- Mecanismos de inicialização pobres
- Correlação entre os bytes da chave e o fluxo
- Em função das vulnerabilidades acima o RC4 foi descontinuado nas versões posteriores a TLS 1.0

Referências

1. Killing RC4: The Long Goodbye (cloudflare.com)
2. imperva_Hacker_Intelligence_Initiative_No21_Mar2015_v1c.pdf