Introdução à Criptografia

Cifras de Fluxo
RC4

Prof. Rodrigo Minetto
rminetto@dainf.ct.utfpr.edu.br
Universidade Tecnológica Federal do Paraná
Material compilado de: ...

Sumário

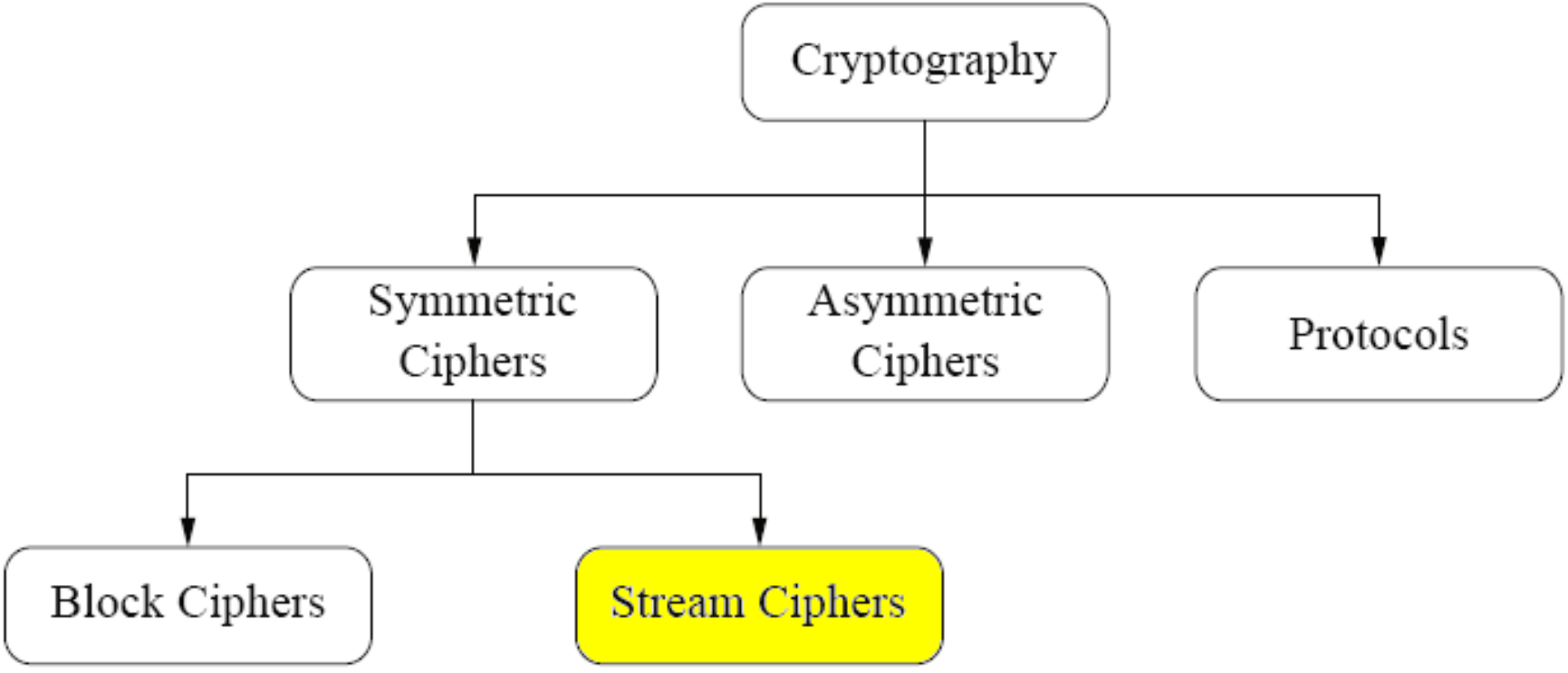
1 Introdução

2 Key Scheduling Algorithm (KSA)

3 Pseudo-Random Generation Algorithm (PRGA)

4 Wired Equivalent Privacy (WEP)

Algoritmos para criptografia



Introdução

O algoritmo de criptografia RC4, também conhecido como ARC4, foi desenvolvido em 1987 por Ronald Rivest, o mesmo criador do RSA e do MD5, para a empresa RSA Security. O RC4 é uma **cifra de fluxo** com tamanho de chave variado e orientado a byte.

Introdução

O **RC4** se tornou muito utilizado em diversas aplicações comerciais, tais como Internet Explorer, Netscape, Adobe Acrobat, dentre outros. Dentre os produtos que usam atualmente o RC4 pode-se citar os protocolos **SSL/TLS** (Secure Sockets Layer / Transport Layer Security), **WEP** (Wired Equivalent Privacy) e **WPA** (WiFi Protected Access).

Introdução

O algoritmo RC4 permaneceu em segredo até o ano de 1994, quando foi divulgado anônimamente em uma lista de discussão sobre criptografia. A partir deste momento começaram a surgir diversos ataques criptoanáliticos sobre o algoritmo e sobre protocolos criptográficos que o utilizam.

Introdução

O algoritmo RC4 é dividido em duas partes: **KSA** (Key Scheduling Algorithm) responsável por gerar uma permutação pseudo-aleatória do conteúdo de uma chave secreta; **PRGA** (Pseudo-Random Generation Algorithm) responsável pelo fluxo de números pseudo aleatórios.

Sumário

1 Introdução

2 Key Scheduling Algorithm (KSA)

3 Pseudo-Random Generation Algorithm (PRGA)

4 Wired Equivalent Privacy (WEP)

Key Scheduling Algorithm (KSA)

KSA: consiste em inicializar um vetor **S** de 256 bytes como uma **permutação** de todos os números de 8 bits (0 a 255). Essa permutação é condicionada a uma **chave** K utilizada no algoritmo.

Key Scheduling Algorithm (KSA)

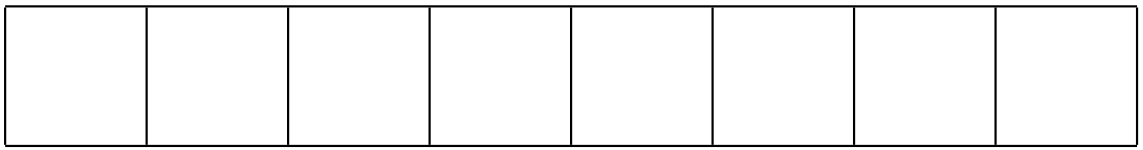
KSA $(K[0 \ldots M-1], \mathrm{S}[0 \ldots N-1])$

1. $j = 0$;
2. **Para** $i = \{0, \ldots N-1\}$ **faça**
3. $\quad \mathrm{S}[i] \leftarrow i$;
4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j + \mathrm{S}[i] + K[i \bmod M]) \bmod N$;
6. $\quad \mathrm{S}[i] \leftrightarrow \mathrm{S}[j]$;

Aplicações com RC4 usam $N = 256$.

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$ 

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

0	1	2	3	4	5	6	7

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

1. $j = 0$;
2. **Para** $i = \{0, \ldots N-1\}$ **faça**
3. $\quad$ S$[i] \leftarrow i$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

0	1	2	3	4	5	6	7

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j +$ S$[i] + K[i \bmod M]) \bmod N$;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$$S = \begin{array}{c} \text{i} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline \end{array} \end{array}$$

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j +$ S$[i] + K[i \bmod M]) \bmod N$;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$$S = \begin{array}{c} \text{i} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline \end{array} \end{array}$$

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (0 + S[0] + K[0 \bmod 3]) \bmod 8;$
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$$S = \begin{array}{c} \text{i} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline \end{array} \end{array}$$

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (0 + 0 + K[0])$ mod 8;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$$S = \begin{array}{c} \text{i} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline \end{array} \end{array}$$

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 3 \leftarrow (0 + 0 + 107) \bmod 8;$
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$$S = \begin{array}{cccccccc} \text{i} & & & \text{j} & & & & \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline \end{array}$$

KSA ($K[0 \ldots M-1]$, $S[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 3 \leftarrow (0 + 0 + 107) \bmod 8;$
6. $\quad S[0] \leftrightarrow S[3];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$S =$

	i			j			
3	1	2	0	4	5	6	7

KSA ($K[0 \ldots M-1]$, $S[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 3 \leftarrow (0 + 0 + 107) \bmod 8$;
6. $\quad S[0] \leftrightarrow S[3]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

	i						
3	1	2	0	4	5	6	7

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j +$ S$[i] + K[i \bmod M]) \bmod N$;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (101)

$S =$

	i,j						
3	1	2	0	4	5	6	7

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 1 \leftarrow (3 + 1 + 101) \bmod 8;$
6. $\quad S[1] \leftrightarrow S[1];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

		i					
3	1	2	0	4	5	6	7

KSA ($K[0 \ldots M-1]$, $S[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j + S[i] + K[i \bmod M]) \bmod N$;
6. $\quad S[i] \leftrightarrow S[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (121)

$S =$

		i		j			
3	1	2	0	4	5	6	7

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 4 \leftarrow (1 + 2 + 121) \bmod 8;$
6. $\quad S[2] \leftrightarrow S[4];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (121)

$S =$

		i		j			
3	1	4	0	2	5	6	7

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 4 \leftarrow (1 + 2 + 121) \bmod 8;$
6. $\quad S[2] \leftrightarrow S[4];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

			i				
3	1	4	0	2	5	6	7

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j +$ S$[i] + K[i \bmod M]) \bmod N$;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$S =$

			i				j
3	1	4	0	2	5	6	7

KSA ($K[0 \ldots M-1]$, $S[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 7 \leftarrow (4 + 0 + 107) \bmod 8;$
6. $\quad S[3] \leftrightarrow S[7];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$S =$

			i				j
3	1	4	7	2	5	6	0

KSA ($K[0 \ldots M-1]$, $S[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 7 \leftarrow (4 + 0 + 107) \bmod 8;$
6. $\quad S[3] \leftrightarrow S[7];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

				i			
3	1	4	7	2	5	6	0

KSA ($K[0 \ldots M-1]$, $S[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j + S[i] + K[i \bmod M]) \bmod N$;
6. $\quad S[i] \leftrightarrow S[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (101)

$S =$

				i		j	
3	1	4	7	2	5	6	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 6 \leftarrow (7 + 2 + 101) \bmod 8;$
6. $\quad S[4] \leftrightarrow S[6];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (101)

$S =$

				i		j	
3	1	4	7	6	5	2	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 6 \leftarrow (7 + 2 + 101) \bmod 8;$
6. $\quad S[4] \leftrightarrow S[6];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

					i		
3	1	4	7	6	5	2	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j +$ S$[i] + K[i \bmod M]) \bmod N$;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (121)

$S =$

				j	i		
3	1	4	7	6	5	2	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 4 \leftarrow (6 + 5 + 121) \bmod 8;$
6. $\quad S[5] \leftrightarrow S[4];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (121)

$S =$

				j	i		
3	1	4	7	5	6	2	0

KSA ($K[0 \ldots M-1]$, $S[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 4 \leftarrow (6 + 5 + 121) \bmod 8;$
6. $\quad S[5] \leftrightarrow S[4];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

						i	
3	1	4	7	5	6	2	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j +$ S$[i] + K[i \bmod M]) \bmod N$;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$S =$

	j					i	
3	1	4	7	5	6	2	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 1 \leftarrow (4 + 2 + 107) \bmod 8;$
6. $\quad S[6] \leftrightarrow S[1];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (107)

$S =$

	j					i	
3	2	4	7	5	6	1	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 1 \leftarrow (4 + 2 + 107) \bmod 8;$
6. $\quad S[6] \leftrightarrow S[1];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key"

$S =$

							i
3	2	4	7	5	6	1	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j \leftarrow (j +$ S$[i] + K[i \bmod M]) \bmod N$;
6. $\quad$ S$[i] \leftrightarrow$ S$[j]$;

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (101)

$S =$

						j	i
3	2	4	7	5	6	1	0

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 6 \leftarrow (1 + 0 + 101) \bmod 8;$
6. $\quad S[7] \leftrightarrow S[6];$

Key Scheduling Algorithm (KSA)

RC4 simplificado ($N = 8$) e $K =$ "key" (101)

$S =$

						j	i
3	2	4	7	5	6	0	1

KSA ($K[0 \ldots M-1]$, S$[0 \ldots N-1]$)

4. **Para** $i = \{0, \ldots N-1\}$ **faça**
5. $\quad j = 6 \leftarrow (1 + 0 + 101) \bmod 8;$
6. $\quad S[7] \leftrightarrow S[6];$

Sumário

1 Introdução

2 Key Scheduling Algorithm (KSA)

3 Pseudo-Random Generation Algorithm (PRGA)

4 Wired Equivalent Privacy (WEP)

Pseudo-Random Generation Algorithm (PRGA)

PRGA: este algoritmo **produz um fluxo de bytes** com números pseudo aleatórios, que são utilizados para realizar a operação XOR com o fluxo de bytes do texto em claro (ou original), de tal forma a produzir o texto cifrado.

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

PRGA $(S[0 \ldots N-1])$

1. $i = (i + 1) \bmod N$;
2. $j = (j + S[i]) \bmod N$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod N]$;

Aplicações com RC4 usam $N = 256$.

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = (i + 1) \bmod N$;
2. $j = (j + S[i]) \bmod N$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod N]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = (i + 1) \bmod \mathbf{8}$;
2. $j = (j + S[i]) \bmod \mathbf{8}$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod \mathbf{8}]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = (\mathbf{0} + 1) \bmod 8$;
2. $j = (j + S[i]) \bmod 8$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i						
3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = \mathbf{1}$;
2. $j = (j + S[i]) \bmod 8$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i						
3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = (\mathbf{0} + S[\mathbf{1}]) \bmod 8$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i						
3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = (0 + \mathbf{2}) \bmod 8$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i		j				
3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = \mathbf{2}$;
3. $S[i] \leftrightarrow S[j]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i		j				
3	2	4	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = 2$;
3. $S[\mathbf{1}] \leftrightarrow S[\mathbf{2}]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i	j					
3	4	2	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = 2$;
3. $S[1] \leftrightarrow S[2]$;
4. **Retorne** $S[(S[i] + S[j]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i		j				
3	4	2	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = 2$;
3. $S[1] \leftrightarrow S[2]$;
4. **Retorne** $S[(S[\mathbf{1}] + S[\mathbf{2}]) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i		j				
3	4	2	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = 2$;
3. $S[1] \leftrightarrow S[2]$;
4. **Retorne** $S[(\mathbf{4} + \mathbf{2}) \bmod 8]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

$S =$

	i		j				
3	4	2	7	5	6	0	1

PRGA $(S[0 \ldots N-1])$

1. $i = 1$;
2. $j = 2$;
3. $S[1] \leftrightarrow S[2]$;
4. **Retorne** $S[\mathbf{6}]$;

Pseudo-Random Generation Algorithm (PRGA)

Inicialização: $i = 0, j = 0$;

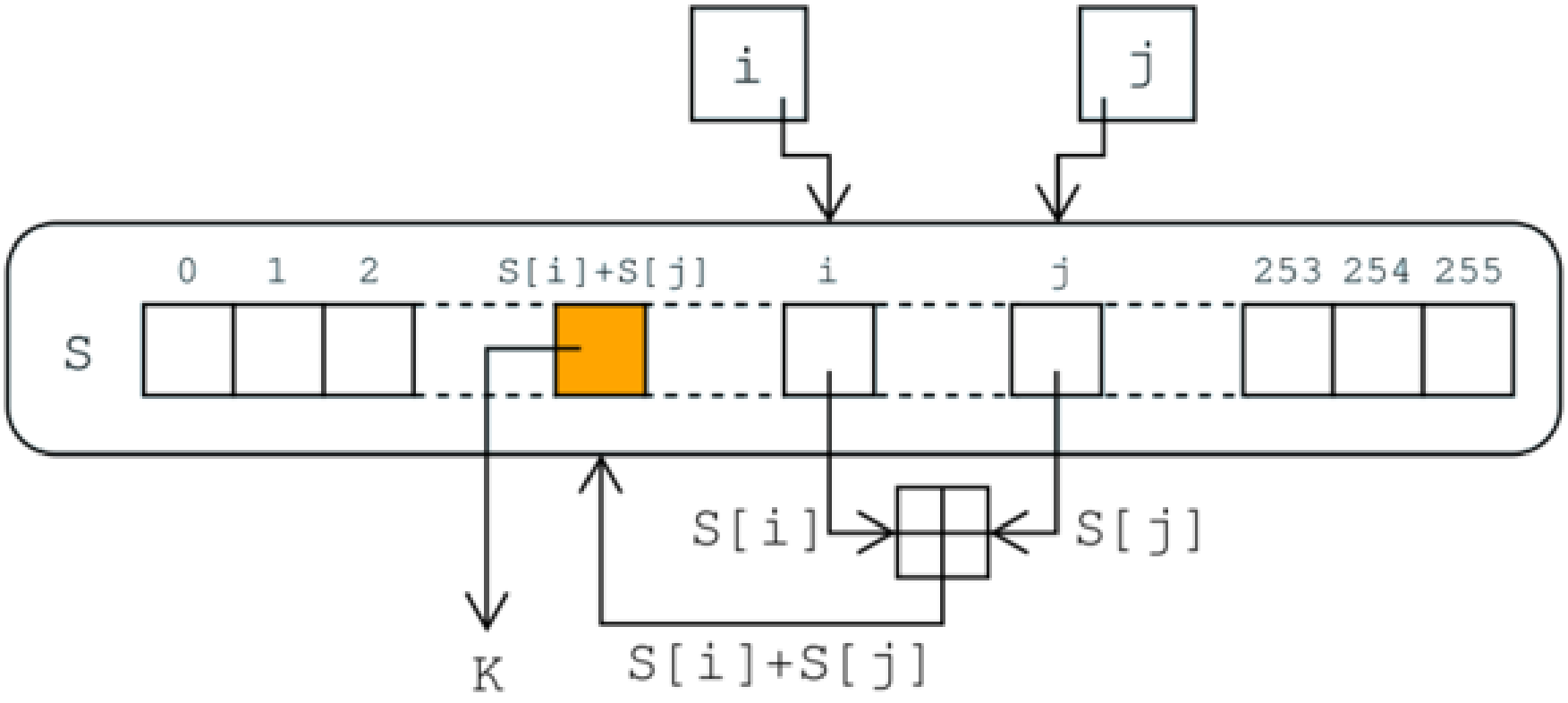
$S =$

	i		j				
3	4	2	7	5	6	**0**	1

PRGA ($S[0 \ldots N-1]$)

1. $i = 1$;
2. $j = 2$;
3. $S[1] \leftrightarrow S[2]$;
4. **Retorne** $S[6]$;

Pseudo-Random Generation Algorithm (PRGA)



Sumário

1 Introdução

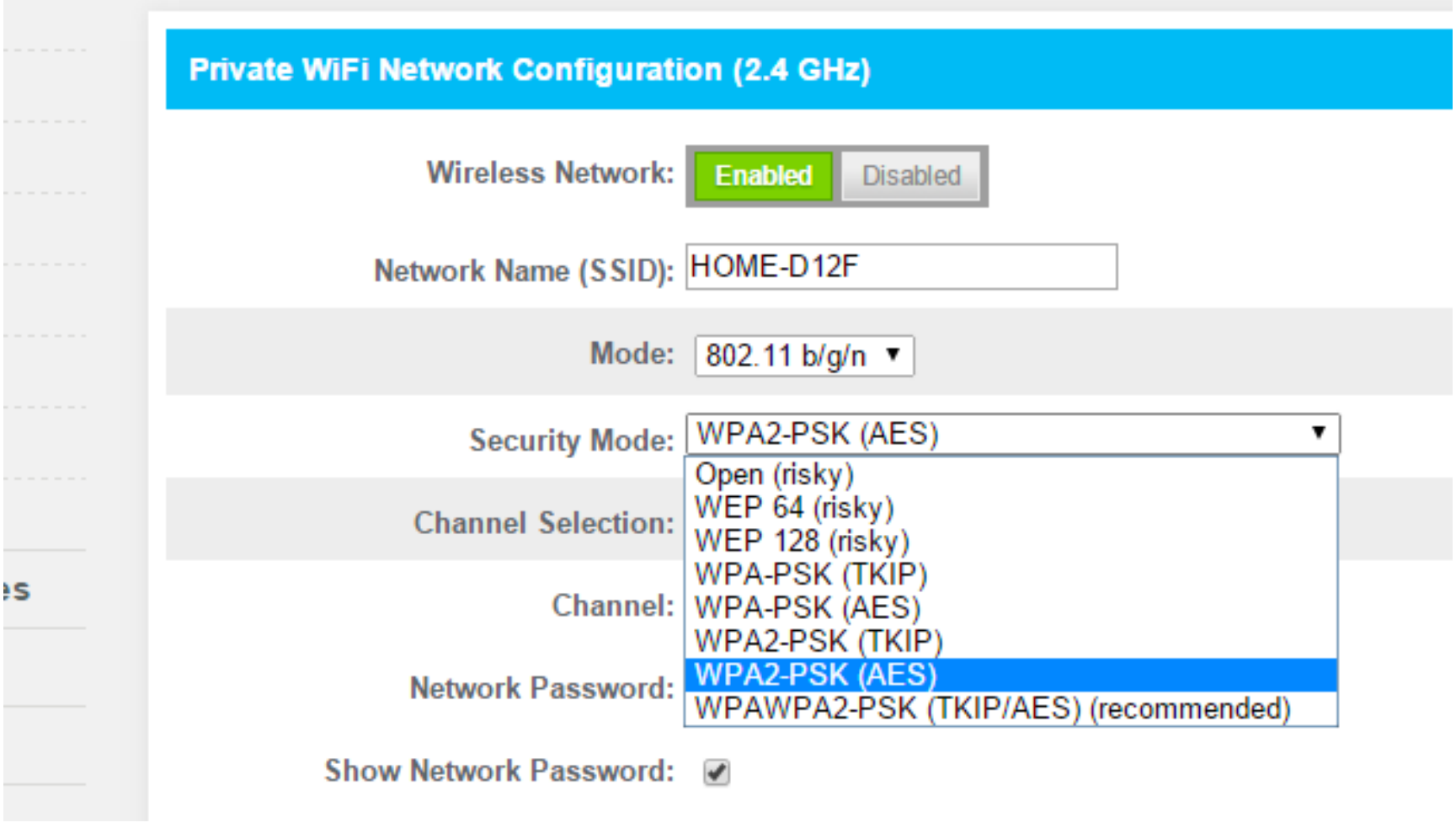
2 Key Scheduling Algorithm (KSA)

3 Pseudo-Random Generation Algorithm (PRGA)

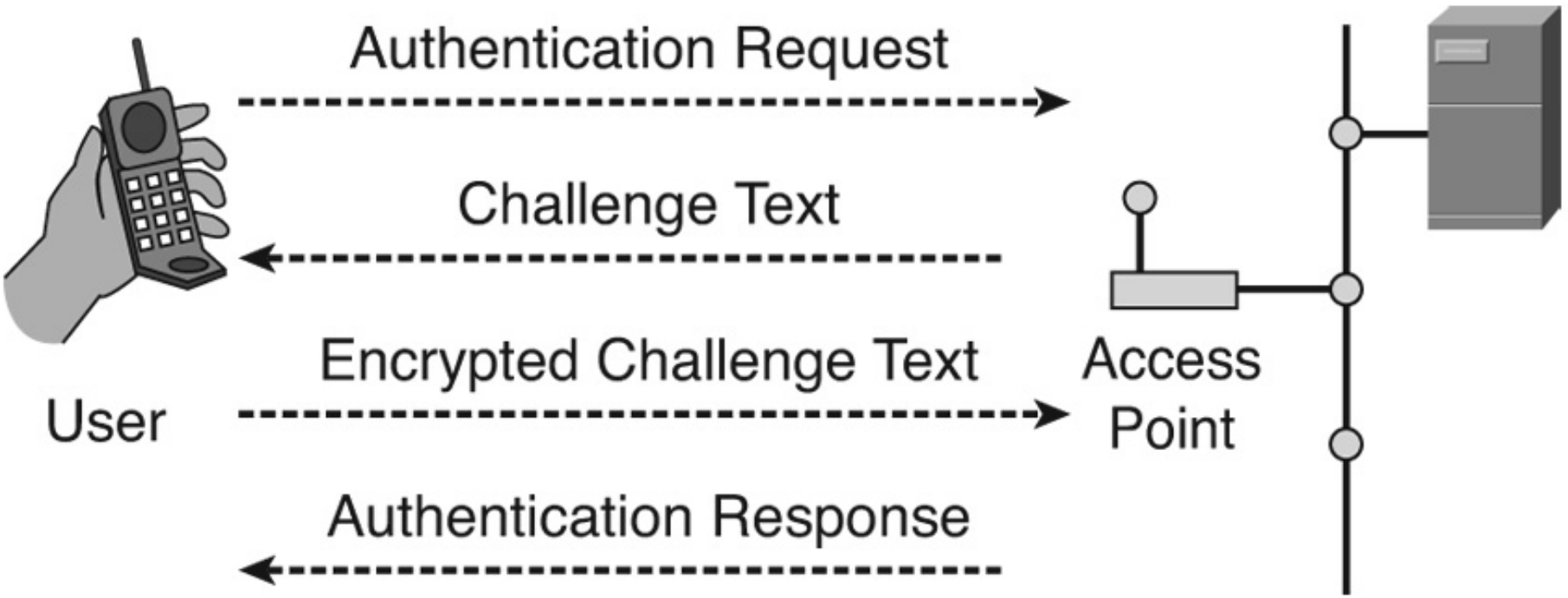
4 Wired Equivalent Privacy (WEP)

Wired Equivalent Privacy (WEP)

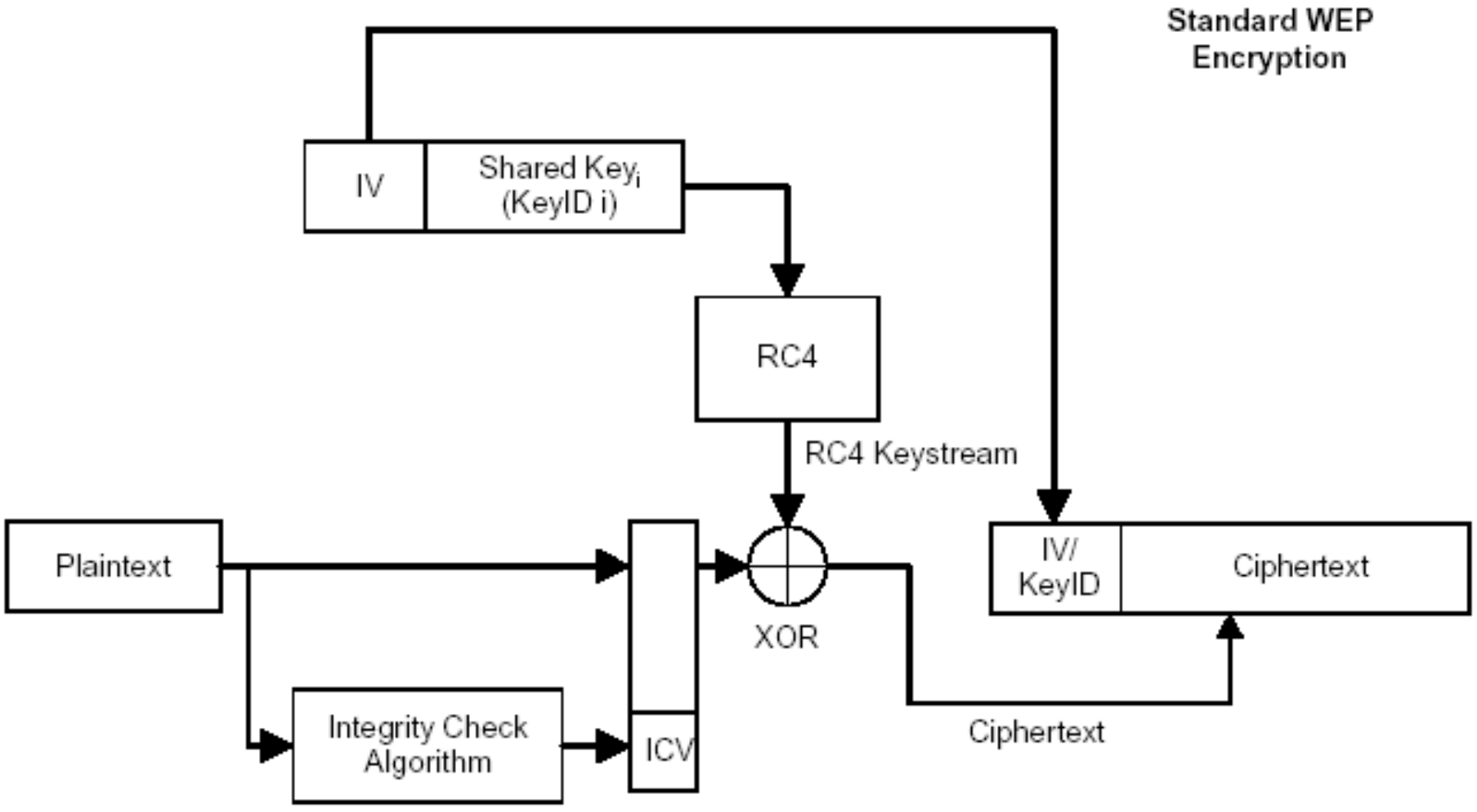
WEP × WAP × WAP2



Wired Equivalent Privacy (WEP) - Prot. 802.11



Ciframento WEP



Wired Equivalent Privacy (WEP)

Detalhes:

- Um valor novo para o IV deve ser gerado a cada nova mensagem!
- Chaves WEP: 64 bits (40 bits) ou 128 (104 bits):
  - Cuidado: 24 bits são do IV (não produzem segurança).
- O padrão não determina como produzir o IV (alguns aparelhos começam em 0 e incrementam o valor a cada iteração).

Wired Equivalent Privacy (WEP)

decriframento wep

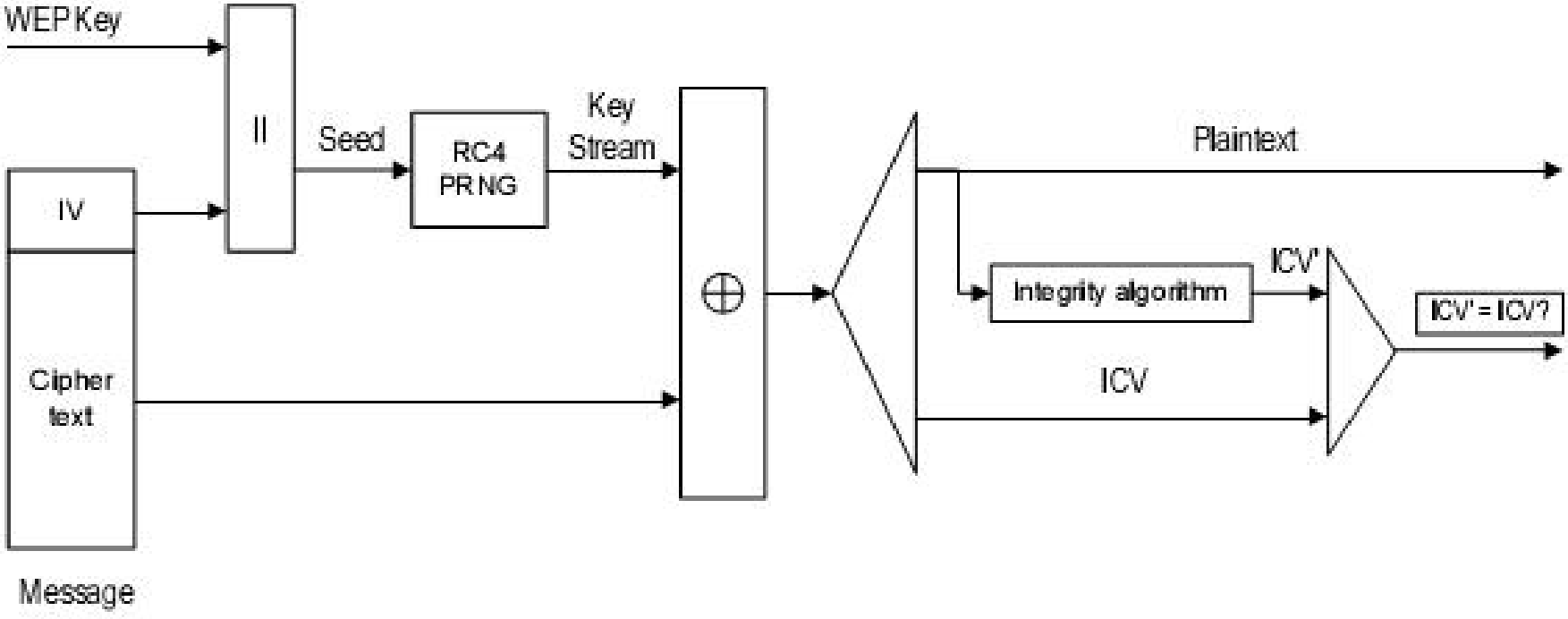


Figure 8-3—WEP decapsulation block diagram

Wired Equivalent Privacy (WEP)

Resumo:

- Ataque: Autenticação.
- Ataque: Colisão de IV (24 bits = 17 milhões de valores $\approx$ 7 horas).
- WPA (802.11i): utiliza RC4 mas em um protocolo conhecido como TKIP (Temporal Key Integrity Protocol).
- WEP **não é seguro**.

Wired Equivalent Privacy (WEP)

WEP × WAP × WAP2

	WEP	WPA	WPA2
Cipher	RC4	RC4	AES
Key Size	40 bits	128 bits	128 bits
Key Life	24 bit IV	48 bit IV	48 bit IV
Packet Key	Concatenated	Mixing Function	Not Needed
Data Integrity	CRC - 32	Michael	CCM
Replay Attack	None	IV Sequence	IV Sequence
Key Management	None	EAP - Based	EAP - Based