Kryptografia

Laboratorium - lista nr 2, 23 III

Zadanie 1 (10 pkt) Schemat RC4 składa się z dwóch algorytmów. Algorytm KSA(N, T), jest zależny od klucza K, inicjalizuje tablicę S. Natomiast algorytm PRGA(N) generuje losowe bajty z zainicjalizowanej tablicy S.

Algorithm 1: $\mathsf{KSA}_k(\mathsf{N},\mathsf{T})$ – operacja $k[i]$ zwraca i-ty BAJT klucza. L jest długością klucza w bajtach.

```
1  for i from 0 to N − 1 do
2  |  S[i] := i
3  end
4  j := 0;
5  for i from 0 to T do
6  |  j := (j + S[i] + k[i mod L]) mod N;
7  |  swap(S[i mod N], S[j mod N]);
8  end
```

Algorithm 2: $\mathsf{PRGA}_S(\mathsf{N})$

```
1  i := 0;
2  j := 0;
3  while GeneratingOutput do
4  |  i := (i + 1) mod N;
5  |  j := (j + S[i]) mod N;
6  |  swap(S[i], S[j]);
7  |  K := S[(S[i] + S[j]) mod N];
8  |  outputK
9  end
```

Algorithm 3: $\mathsf{KSA\text{-}RS}_k(\mathsf{N},\mathsf{T})$ – operacja $k[i]$ zwraca i-ty BIT klucza. L jest długością klucza w bitach.

```
1  for i from 0 to N − 1 do
2  |  S[i] := i
3  end
4  for r from 0 to T do
5  |  Top = array();
6  |  Bottom = array();;
7  |  for i from 0 to N do
8  |  |  if key[rN + i mod L] == 0 then
9  |  |  |  Top.push(i)
10 |  |  else
11 |  |  |  Bottom.push(i)
12 |  |  end
13 |  end
14 |  foreach Top as i ⇒ v do
15 |  |  newS[i] := S[v]
16 |  end
17 |  foreach Bottom as i ⇒ v do
18 |  |  newS[Top.size + i] := S[v]
19 |  end
20 |  S := newS;
21 end
```

Oryginalne RC4 = RC4(N) = RC4(N, T) polega na wykonaniu:

1. $S :=$ $\mathsf{KSA}_k(\mathsf{N},\mathsf{T})$
2. $outputStream \leftarrow$ $\mathsf{PRGA}_S(\mathsf{N})$

RC4-RS(N, T) polega na wykonaniu:

1. $S :=$ $\mathsf{KSA\text{-}RS}_k(\mathsf{N},\mathsf{T})$
2. $outputStream \leftarrow$ $\mathsf{PRGA}_S(\mathsf{N})$

Dla RC4 $N = 256 = T$. Funkcja RC4-drop[D] powoduje, że pierwsze D bajtów wyjścia algorytmu PRGA jest opuszczane.

Twoim zadaniem jest zaimplementowanie powyższych algorytmów, a następnie sprawdzenie "jakości" losowości generowanych bitów w następujących konfiguracjach:

1. RC4(16, 16)
2. RC4(16, 16)-drop[48]
3. RC4(16, 64)
4. RC4-RS(16, 64)

5. RC4-RS(16, 92)

Powtórz eksperymenty dla kluczy długości $8, 16, 24, 32, 40$ i 64 bitów. Za każdym razem generuj ciągi długości 128 bitów.

Do testów wykorzystaj oprogramowanie do testy statystyczne (`http://csrc.nist.gov/groups/ST/toolkit/rng/index.html`)