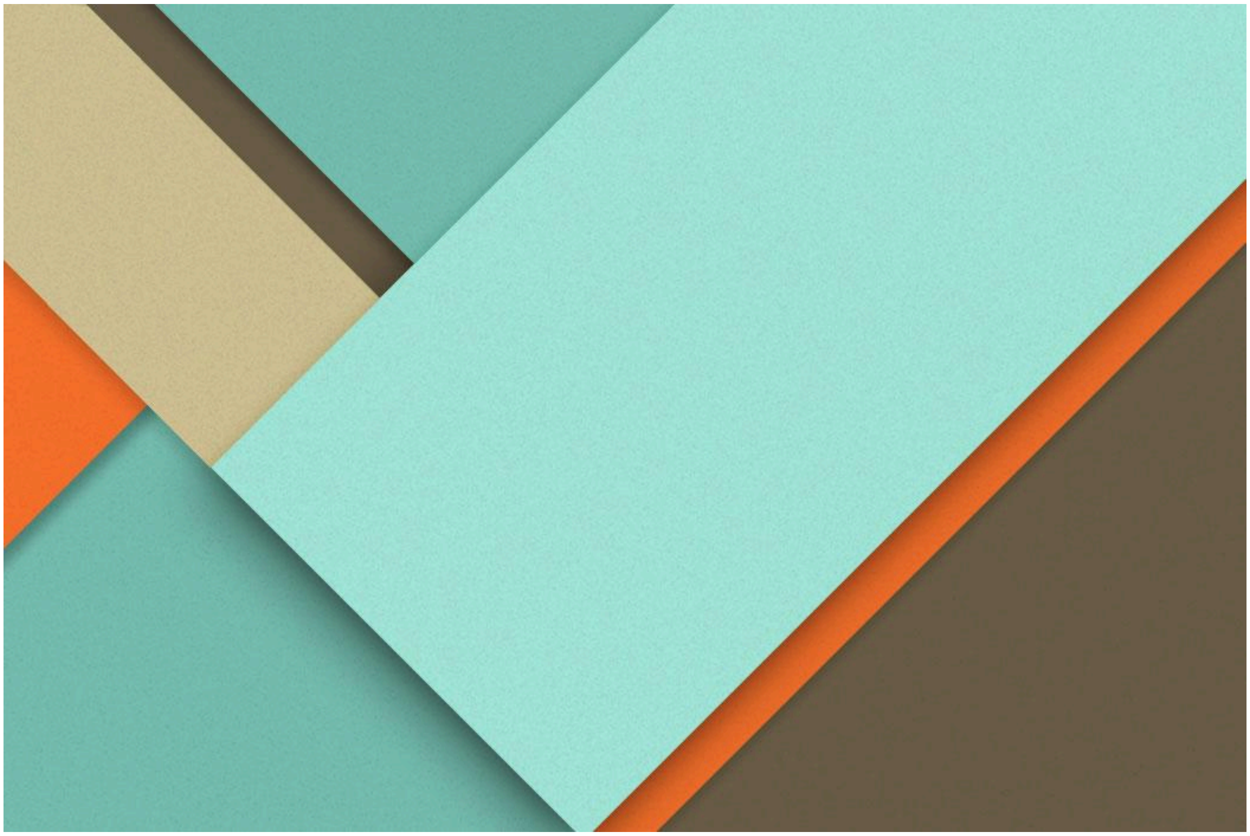


RC4 Key Generation Algorithm

01.11.2023

—

Ayat AbuAllan
191060

RC4

RC4 is a variable-length key stream cipher algorithm that encrypts data on a byte-by-byte basis or in larger units as needed. It utilizes a pseudorandom bit generator as its key input, which generates an unpredictable stream of 8-bit numbers. This stream, known as the key-stream, is combined with the plaintext stream cipher one byte at a time using the XOR (exclusive OR) operation.

RC4 Stream Cipher - How It Works:

Key Initialization:

1. RC4 uses a variable-length key, which can be 1 to 256 bytes.
2. The algorithm initializes a state vector S of 256 bytes, numbered from 0 to 255.
3. It creates a temporary vector T.

Key-Scheduling Algorithm:

1. If the key is exactly 256 bytes long, it's assigned to T. Otherwise, for a shorter key, T is filled with repeated portions of the key.
2. Using T, the algorithm performs an initial permutation of the state vector S.
3. This permutation is accomplished by iterating through S and swapping its elements in a specific pattern based on values from T.

Pseudo-Random Generation (Stream Generation):

1. Once the state vector S is initialized, it's used to generate a pseudo-random key-stream.
2. In a continuous loop, two indices i and j are maintained.
3. Elements in S are swapped repeatedly in a systematic way.
4. The current configuration of S is used to generate a pseudo-random byte k for the key-stream.
5. This key-stream can be used for both encryption and decryption.

Three different tests to evaluate the security weaknesses of the RC4 stream cipher:

- ### Test 1 - Probability of the Second Byte Being Zero:
  - This test generates 2 bytes 10,000 times.
  - It calculates the probability of the second byte being zero.
  - The expected result is that the probability is approximately 2/256, which is equivalent to 1/128. The test checks if this condition is met.
  - If the probability is less than 2/256, the test is considered successful because it indicates that the second byte's value is not predictable.

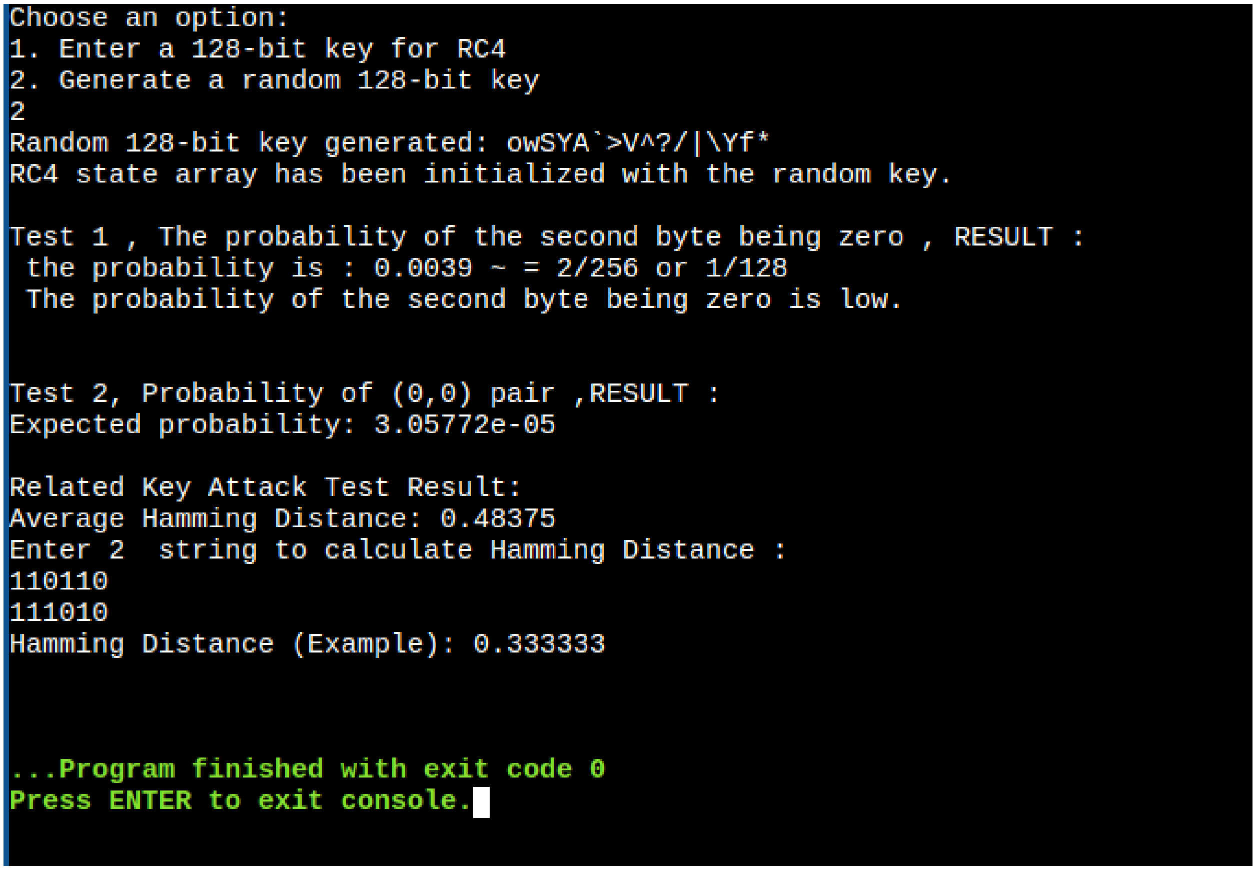
- ### Test 2 - Probability of (0,0) Pair Occurrence:
  - In this test, 1,000,000 bytes are generated.
  - The test calculates the probability of the (0,0) pair occurring, where both consecutive bytes are zero.
  - The expected result is based on the probability of the first byte being zero (2/256) and the second byte being zero (1/256), which is 2/(256^2) + 1/(256^3).
  - The test checks if the observed probability matches the expected probability.

- ### Related Key Attack Test:
  - In this test, the code demonstrates a related key attack.
  - It starts with an original key and generates a new key with a single bit difference from the original.
  - The RC4 state array is initialized with the original key, and the first byte is generated.
  - Then, the RC4 state array is initialized with the modified key (one bit flipped), and the first byte is generated again.
  - The Hamming distance between the binary representations of these two bytes is calculated, and this process is repeated 1,000 times.
  - The average Hamming distance is computed.
  - A lower expected threshold (0.5) is defined for a related key attack. If the average Hamming distance is below this threshold, it indicates that a related key attack is possible.

Here is the Link of the code “ Drive-link ”:

https://drive.google.com/file/d/1y2IY3Bth_qkj0LNhFjcVZ1n93SRCUZN2/view?usp=sharing

The tests results are presented in the screenshot below :



```
Choose an option:
1. Enter a 128-bit key for RC4
2. Generate a random 128-bit key
2
Random 128-bit key generated: owSYA`>V^?/|\Yf*
RC4 state array has been initialized with the random key.

Test 1 , The probability of the second byte being zero , RESULT :
 the probability is : 0.0039 ~ = 2/256 or 1/128
 The probability of the second byte being zero is low.


Test 2, Probability of (0,0) pair ,RESULT :
Expected probability: 3.05772e-05

Related Key Attack Test Result:
Average Hamming Distance: 0.48375
Enter 2  string to calculate Hamming Distance :
110110
111010
Hamming Distance (Example): 0.333333



...Program finished with exit code 0
Press ENTER to exit console.
```

RC4 is considered **weak** and is **NOT** recommended for use in modern cryptographic systems due to these and other vulnerabilities. The tests are conducted to showcase these weaknesses in practice.