Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

“Криптография и информационная безопасность”

Лабораторная работа 1

Проверил: Чербу Ольга

Выполнил: Чобану Артём

Кишинев 2022

**Алгоритм RC6**

**RC6** — симметричный блочный [криптографический алгоритм](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC), производный от алгоритма [RC5](https://ru.wikipedia.org/wiki/RC5). Был создан Роном Ривестом, Мэттом Робшау и Рэем Сиднеем для удовлетворения требований конкурса [Advanced Encryption Standard](https://ru.wikipedia.org/wiki/AES_(%D0%BA%D0%BE%D0%BD%D0%BA%D1%83%D1%80%D1%81)) (AES). Алгоритм был одним из пяти финалистов конкурса, был также представлен [NESSIE](https://ru.wikipedia.org/wiki/NESSIE) и [CRYPTREC](https://ru.wikipedia.org/wiki/CRYPTREC). Является [собственническим](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%9F%D0%9E) (проприетарным) алгоритмом, и запатентован RSA Security.

**Алгоритм:**

Так же, как и [RC5](https://ru.wikipedia.org/wiki/RC5), RC6 — полностью параметризированная семья алгоритмов шифрования. Для спецификации алгоритма с конкретными параметрами, принято обозначение RC6-w/r/b, где

w — длина [машинного слова](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D1%81%D0%BB%D0%BE%D0%B2%D0%BE" \o "Машинное слово) в [битах](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82" \o "Бит).

r — число раундов.

b — длина ключа в [битах](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82" \o "Бит). Возможные значения 0..255 бит.

Для того чтобы соответствовать требованиям [AES](https://ru.wikipedia.org/wiki/Advanced_Encryption_Standard), блочный шифр должен обращаться с 128-битовыми блоками. Так как [RC5](https://ru.wikipedia.org/wiki/RC5) — исключительно быстрый [блочный шифр](https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80" \o "Блочный шифр), расширение его, чтобы работать с 128-битовыми блоками привело бы к использованию двух 64-битовых рабочих регистров. Но архитектура и [языки программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F" \o ") ещё не поддерживают 64-битные операции, поэтому пришлось изменить проект так, чтобы использовать четыре 32-битных регистров вместо двух 64-битных.

1. **Расширение ключа**

а) Генерация констант: Так же, как и в [RC5](https://ru.wikipedia.org/wiki/RC5" \o "RC5), в RC6 генерируются две псевдослучайные величины, используя две математические константы:[экспонента](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0" \o "Экспонента) (e) и [золотое сечение](https://ru.wikipedia.org/wiki/%D0%97%D0%BE%D0%BB%D0%BE%D1%82%D0%BE%D0%B5_%D1%81%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5" \o "Золотое сечение) (f).

Qw = Odd((f-1) \* 2w)

Pw = Odd((e-2) \* 2w)

Где Odd - это [округление](https://ru.wikipedia.org/wiki/%D0%9E%D0%BA%D1%80%D1%83%D0%B3%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) до ближайшего нечетного целого.

При w = 32 бита (в шестнадцатеричном виде):

Q32 = 9E3779B9

P32 = B7E15163

{\displaystyle P\_{32}=B7E15163} b) Расширение ключа для RC6-w/r/b:

Таблица ключей для шифра RC6 также идентична таблице шифра [RC5](https://ru.wikipedia.org/wiki/RC5" \o "RC5). Отличие состоит в том, что большее количество слов из массива L получено из предоставленного пользователем ключа для использования в течение шифрования и расшифровки.

**Вход**: b-байтный ключ, заданный пользователем, предварительно преобразованный в массив из **c** слов **L** [0, … , c – 1]

**r** – количество раундов

**Выход**: w-битная таблица ключей S [0, … , 2r+3]

>>>, <<< - циклические сдвиги вправо и влево соответственно.

for (int s = 1; s <= v; s++)  
{  
 A = S[i] = S[i] + A + B <<< 3  
 B = L[j] = L[j] + A + B <<< A + B  
 i = (i + 1) mod (2 \* r + 4)  
 j = (j + 1) mod c;  
}

1. Ширфование

RC6 работает с четырьмя w-битными регистрами A, B, C и D, которые содержат входной исходный текст и выходной шифрованный текст в конце шифрования.

**Вход**: **r** – количество раундов

**w** – разрядные ключи для каждого раунда S [0, … , 2r+3]

**Выход**: шифрованный текст в A, B, C и D

B += S[0]  
D += S[1]  
  
int shift = log2 (32)  
  
for (int i = 1; i <= r; i++)  
{  
 var t = B \* (2 \* B + 1) <<< shift  
 var u = D \* (2 \* D + 1) <<< shift  
  
 A = A ^ t <<< u + S[2 \* i]  
 C = C ^ u <<< t + S[2 \* i + 1]  
  
 var temp = A;  
 A = B;  
 B = C;  
 C = D;  
 D = temp;  
}

A += S[2 \* r + 2]  
C += S[2 \* r + 3]

1. Расшифровка

**Вход**: **r** – количество раундов

**w** – разрядные ключи для каждого раунда S [0, … , 2r+3]

шифрованный текст в A, B, C и D

**Выход**: исходный текст в A, B, C и D

C -= S[2 \* r + 3]  
A -= S[2 \* r + 2]

for (var i = r; i >= 1; i--)  
{  
 var temp = D  
 D = C  
 C = B  
 B = A  
 A = temp  
  
 var u = (D \* (2 \* D + 1) <<< shift)  
 var t = (B \* (2 \* B + 1) <<< shift)  
  
 C = (C - S[2 \* i + 1] <<< t) ^ u  
 A = (A - S[2 \* i] <<< u) ^ t;  
}

D -= S[1]  
B -= S[0]