Двойно шифриране

ЗАДАЧА

Стандартът за шифриране AES (Advanced Encryption Standard) прилага нов алгоритъм. Той работи с 3 блока от по 128 бита. Съобщение, представено чрез блок p, се шифрира от функцията E посредством ключ, представен с блок k, при което се получава блок c, представящ шифрираното съобщение:

$$c = E(p, k) .$$

Обратна на функцията за шифриране E е функцията за дешифриране D, такава че

$$D (E(p, k), k) = p , \quad E (D(c, k), k) = c .$$

При още по-новия метод *Double AES* (двойно шифриране) се използват последователно два независими блока за ключове k_1 и k_2 (първо k_1, след това k_2):

$$c_2 = E (E(p, k_1), k_2).$$

В настоящата задача е дадено още и цялото число s със следния смисъл: при работата на алгоритмите се използват само най-левите $4s$ бита от всеки ключ, т.е. останалите 128-$4s$ бита са нули.

Трябва да откриете двойката ключове, с която е било извършено шифрирането на известен брой съобщения чрез алгоритъма Double AES. Дадени са изходният текст p и съответният му, получен чрез Double AES, шифриран текст c_2, както и числото s, задаващо вида на шифриращите ключове.

Алгоритмите за шифроване и дешифриране на AES са дадени в библиотека.

Трябва да предадете разкритите ключове, а не програма за разкриване.

ВХОД

Дадени са 10 входни варианта за задачата в текстови файлове с имена `double1.in, ..., double10.in`. Всеки входен файл се състои от 3 реда. Първият от тях съдържа цялото число s, вторият – блока p, който подлежи на шифроване, а третият – блока c_2, получен от p чрез Double AES. Двата блока са записани като низове от по 32 шестнайсетични цифри ('0'..'9', 'A'..'F'). За всички входни данни задачата има решение.

ИЗХОД

Трябва да предадете 10 изходни файла, съответстващи по ред на дадените входни файлове. Всеки изходен файл трябва да се състои от 3 реда. Първият ред трябва да съдържа текста

```
#FILE double I
```

където `I` е номерът на съответния входен файл. Вторият ред трябва да съдържа блока-ключ k_1, а третият – блока-ключ k_2, такива че

$c_2 = E\ (\ E(p, k_1), k_2\ ).$

Двата блока трябва да са записани като низове от 32 на брой шестнадесетични цифри ('0'..'9', 'A'..'F'). Ако задачата има няколко решения, изведете само едно от тях.

Библиотеката предоставя процедури за превръщане на низовете в блокове и на блоковете в низове.

ПРИМЕР

Прилагаме пример за входен файл с номер 0:

`double0.in`

```
1
00112233445566778899AABBCCDDEEFF
6323B4A5BC16C479ED6D94F5B58FF0C2
```

A possible output file

```
#FILE double 0
A0000000000000000000000000000000
7000000000000000000000000000000
```

БИБЛИОТЕКА

FreePascal library (Linux: `aeslibp.p, aeslibp.ppu, aeslibp.o`;
Windows: `aeslibp.p, aeslibp.ppw, aeslibp.ow`):

```
type
  HexStr = String [ 32 ]; { only '0'..'9', 'A'..'F' }
  Block  = array [ 0..15 ] of Byte; { 128 bits }

procedure HexStrToBlock ( const hs: HexStr; var b: Block );
procedure BlockToHexStr ( const b: Block; var hs: HexStr );
procedure Encrypt ( const p, k: Block; var c: Block );
  { c = E(p,k) }
procedure Decrypt ( const c, k: Block; var p: Block );
  { p = D(c,k) }
```

Програмата `aestoolp.pas` илюстрира как да използвате библиотеката за FreePascal.

GNU C/C++ library (Linux и Windows: `aeslibc.h, aeslibc.o`):

```
typedef char HexStr[33]; /* '0'..'9', 'A'..'F', '\0'-terminated */
typedef unsigned char Block[16];  /* 128 bits */

void hexstr2block ( const HexStr hs, /* out-param */ Block b );
void block2hexstr ( const Block b, /* out-param */ HexStr hs );
void encrypt(const Block p, const Block k, /* out-param */ Block c);
  /* c = E(p,k) */
void decrypt(const Block c, const Block k, /* out-param */ Block p);
  /* p = D(c,k) */
```

Програмата `aestoolc.c` илюстрира как да използвате библиотеката за GNU C/C++.

ОГРАНИЧЕНИЯ

За числото s е в сила ограничението $1 \leq s \leq 5$.

ЗАБЕЛЕЖКА

Добрата програма може да открие за по-малко от 10 секунди търсените ключове за всеки вариант на входните данни.