Doppia cifratura

Il problema

L'Advanced Encryption Standard (AES) contiene un nuovo algoritmo forte per la cifratura. Funziona con tre *blocchi* di 128 bit. Dato un blocco p contenente il messaggio (in chiaro) e un blocco k contenente la chiave, la funzione di cifratura dell'AES E restituisce un blocco c contenente il testo cifrato:

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

L'inversa della funzione E di cifratura dell'AES è la funzione di decifratura D, che soddisfa ovviamente

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

Nel *doppio AES* vengono usate in sequenza due chiavi indipendenti k_1 e k_2, applicando prima k_1 e poi k_2:

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

In questo problema, vi viene anche dato un intero s: solo i $4s$ bit più a sinistra delle chiavi sono rilevanti, mentre tutti gli altri bit (cioè i 128 meno $4s$ bit più a destra) saranno zero.

Dovete scoprire la coppia di chiavi di cifratura di alcuni messaggi cifrati usando il doppio AES. Vi vengono dati sia il messaggio in chiaro p che il corrispondente messaggio doppiamente cifrato c_2, oltre alla struttura delle chiavi di cifratura espressa mediante l'intero s.

Gli algoritmi di cifratura e decifratura sono disponibili in una libreria. **Dovete sottoporre le chiavi che avete scoperto, non il programma che le trova.**

Dati in input

Vi vengono date dieci istanze del problema in altrettanti file di testo di nome `double1.in`, ..., `double10.in`. Ognuno di questi file contiene tre righe. La prima riga contiene l'intero s, la seconda riga il blocco contenente il messaggio in chiaro p, e la terza riga il blocco contenente il messaggio cifrato c_2, ottenuto a partire da p mediante cifratura con doppio AES. Entrambi i blocchi sono scritti usando strighe di 32 cifre esadecimali ('0' ... '9', 'A' ... 'F'). La libreria fornisce una routine che converte stringhe in blocchi. Tutti i file di input sono risolvibili.

Dati in output

Dovete sottoporre dieci file di output corrispondenti ai dieci file di input. Ogni file di output contiene tre righe. La prima riga contiene il testo

```
#FILE double I
```

dove `I` è il numero del corrispondente file di input. La seconda riga contiene il blocco con la chiave k_1 e la terza riga il blocco con la chiave k_2, che devono soddisfare

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

Entrambi i blocchi devono essere scritti come stringhe di 32 cifre esadecimali ('0' ...'9', 'A' ...'F'). La libreria fornisce una routine che converte blocchi in stringhe. Se ci sono soluzioni multiple, dovete comunque sottoporne una sola.

Esempio

A titolo di esempio, usiamo il file di input numero 0.

```
double0.in                                  Un possibile file di output

1                                           #FILE double 0
00112233445566778899AABBCCDDEEFF            A00000000000000000000000000000000
6323B4A5BC16C479ED6D94F5B58FF0C2            70000000000000000000000000000000
```

Libreria

Libreria per FreePascal (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) }
```

Il programma `aestoolp.pas` illustra come usare la libreria per il FreePascal.

Libreria per lo GNU C/C++ (Linux e 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, /* risultato */ Block b );
void block2hexstr ( const Block b, /* risultato */ HexStr hs );
void encrypt ( const Block p, const Block k, /* risultato */ Block c );
  /* c = E(p,k) */
void decrypt ( const Block c, const Block k, /* risultato */ Block p );
  /* p = D(c,k) */
```

Il programma `aestoolc.c` illustra come usare la libreria per lo GNU C/C++.

Vincoli

Per quanto riguarda il numero s di cifre esadecimali rilevanti in una chiave vale che $1 \leq s \leq 5$.

SUGGERIMENTO: Un buon programma può scoprire chiavi in meno di 10 secondi su qualunque file di input.