Si vuole implementare uno script perl **cryptocat** che simuli il comando linux **`netcat`** o **nc** (per leggere/scrivere dati utilizzando connessioni *tcp/ip*, di seguito specificato) e che fornisca in aggiunta le funzionalità opportune per *cifrare/decifrare* i dati trasmessi (sfruttando a tal fine il comando **`openssl enc`** di seguito specificato).

Con il comando **`netcat`** è possibile connettersi ad un *host in remoto* (o localmente su *localhost*) e collegare *stdin* e *stdout* al flusso dati spedito o ricevuto da remoto. In particolare, il comando `netcat` o `nc` può essere usato:

- in modalità server con le seguente sintassi:

  `netcat –l porta`

  `–l` è l'opzione che attiva la modalità *listen*, cioè mette il sever "in ascolto"

  `porta` è un intero che specifica il *numero di porta* su cui il server è in ascolto

- in modaità client con le seguente sintassi:

  `netcat <hostname> porta`

  `hostname` è *l'indirizzo IP* del server a cui connettersi

  `porta` è un intero che specifica il *numero di porta* su cui connettersi

In modalità "listen" `netcat` rimane in ascolto su di una porta fino a che qualche client da remoto ci si connette (il client potrebbe essere anche in locale). A questo punto realizzerà un collegamento tra client e server.
In modalità client, invece, `netcat` si connette ad un server remoto e potrà inviargli dei dati.

Il più semplice e primitivo uso di `netcat` è quello di un semplicissima situazione ***server-client.*** Ad esempio, creo sul pc1 (che ha come indirizzo ip `192.168.0.1`) un server in ascolto sulla porta `3333` digitando il comando:

```
netcat -l 3333
```

Sul pc2 invece eseguo il comando:

```
netcat 192.168.0.1 3333
```

con il quale mi connetto tramite `netcat` al pc1 sulla sua porta `3333`

A questo punto tutto quello che scrivo nella console del pc1 verrà trasferito al pc2 e viceversa, dato che lo standard input di uno è connesso allo standard output dell'altro.

Con il comando **`openssl enc`** è possibile cifrare e decifrare blocchi di dati con un nutrito insieme di algoritmi crittografici. In particolare, il comando `openssl enc` può essere usato con la seguente sintassi:

```
openssl enc [-algoritmo] [-e] [-d] [-k chiave] [-in file] [-out file]
```

`-algoritmo` serve ad indicare l'algoritmo da usare per svolgere le operazioni di crifratura simmetrica (la lista completa si degli algoritmi a disposizione si visualizza con *openssl enc -h*)

`-e` indica che l'operazione da svolgere è l' *encryption* (naturalmente esclude l'opzione `-d`)

`-d` indica che l'operazione da svolgere è la *decryption* (naturalmente esclude l'opzione `-e`)

`-k` indica la chiave per le operazioni di crittografia simmetrica

`-in file` è il file in input da cifrare

`-out file` è il file di output prodotto

`-base64` serve per avere l'output in base64

Lo script **cryptocat** dovrà essere invocato nel seguente modo:

```
cryptocat [opzioni][hostname] porta
```

- `opzioni` è un parametro che comprende un elenco di opzioni tra cui:
  - `–l`, se presente, attiva la modalità *listen*, cioè la modalità server
  - `-k` indica la chiave per le operazioni di crittografia
  - `–a algoritmo`, se presente, specifica l'algoritmo per le operazioni di cifratura/decifratura altrimenti si assume l'utilizzo di quello di default
- `hostname` un parametro opzionale che indica – quando cryptocat è invocato come client- *l'indirizzo IP* del server a cui connettersi
- `porta` è un intero che specifica il *numero di porta* su cui connettersi lato client o su cui restare in ascolto lato server

Lo script <u>se attivato in modalità client</u> (non è presente t'opzione `-l`):

1. legge da *stdin* e cifra il flusso di dati letto mediante l'utilizzo del comando `openssl enc` (il quale verrà invocato con i parametri eventualmente specificati, ad esempio la `chiave`, l'`algoritmo`)
2. invia il flusso cifrato prodotto (in `base64`) al server `hostname` sulla porta `porta` utilizzando a tal fine il comando `netcat`

Lo script <u>se attivato in modalità server</u> (è presente t'opzione `-l`):

1. invoca il comando `netcat` in modalità *listen,* quindi rimane in ascolto sulla porta `porta` fino a che non riceve il flusso di dati proveniente da un client
2. legge i dati in arrivo dal client e produce su *stdout* decifrando il flusso di dati mediante l'utilizzo del comando `openssl enc` (il quale verrà invocato con i parametri eventualmente indicati, ad esempio la `chiave`, l'`algoritmo`).

Suggerimento : è possible invocare in cascata i comandi linux *cat*, openssl *enc* e *netcat* per (1) fornire un flusso di dati da (2) cifrare e (3) inviare al server remoto; e viceversa si possono concatenare i comandi *netcat* ed openssl *enc* per (1) mettersi in ascolto ed in attesa di dati provenienti da un client e (2) decifrali

Si lanci lo script **cryptocat** in entrambe le modalità (su due shell diverse) analizzando tramite il software **wireshark** il traffico di dati tra server e client.