Sicurezza dei Sistemi Informatici

Esercitazioni OpenSSL

Generazione chiavi Diffie-Hellman

Roberta Daidone

roberta.daidone@iet.unipi.it

Obiettivo

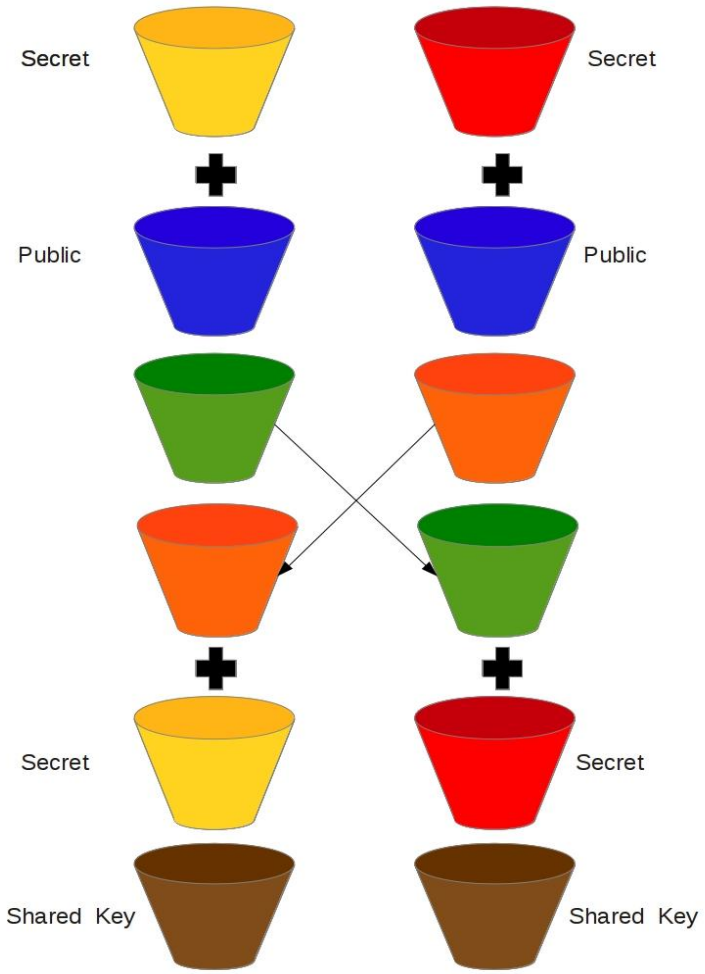

Figura ispirata da “Codici e Segreti ”di Simon Singh

- Stabilire un canale sicuro fra client e server grazie a Diffie-Hellman.
- Il server genera e controlla i paramentri p e g condivisi col client.
- Sia il client che il server generano una coppia di parametri (uno pubblico e uno privato) e si scambiano il parametro pubblico.
- Client e server generano **autonomamente** la chiave simmetrica condivisa

Funzioni client/server

- #include <openssl/dh.h>
- #include <openssl/bn.h>

- Allocazione della struttura dati DH:
  DH* dh = malloc(sizeof(DH));

- Campi salienti della struttura DH:

```
struct
        {
        BIGNUM *p;              // numero primo condiviso
        BIGNUM *g;              // generatore condiviso
        BIGNUM *priv_key;       // privato (DH value x)
        BIGNUM *pub_key;        // pubblico D(H value g^x)
        // ...
        };
DH
```

Funzioni client/server

- Gestione dei BIGNUM facenti parte della struttura DH:
  - Creazione di un BIGNUM
    BIGNUM* bn = BN_new();
  - Calcolo della lunghezza in byte di un BIGNUM
    len = BN_num_bytes(bn);
  - Distruzione di un BIGNUM
    BN_free(bn);
- Conversioni utili per inviare/ricevere BIGNUM tramite socket:
  - Conversione da BIGNUM a binario
    BN_bn2bin(bn, buf);
  - Conversione da binario a BIGNUM
    bn = BN_bin2bn(buf, len, bn);

Funzioni client/server

- Generare una coppia di chiavi (parametri DH)
  DH_generate_key(dh);

  Se torna 1 (successo):
  dh->priv_key contiene il valore privato x,
  dh->pub_key contiene il valore pubblico g^x,

- Generazione della chiave simmatrica Diffie-Hellman
  len = DH_compute_key(secret, other_pub_key, dh);

  secret torna l'unsigned char* con il segreto condiviso
  other_pub_key è il BIGNUM* contenente la chiave pubblica del client nel caso del server e del server nel caso del client
  len è la dimensione in byte di secret, vale -1 in caso di errore.

- Deallocazione della struttura dati DH:
  DH_free(dh);

Funzioni server

- Generazione dei parametri Diffie-Hellman

  dh = DH_generate_parameters(length, DH_GENERATOR_5, NULL, NULL);

  2° parametro: generatore g. I valori consigliati sono ***DH_GENERATOR_2*** e ***DH_GENERATOR_5***.
  3° parametro: puntatore alla funzione di callback che dà un feedback di avanzamento della generazione dei parametri.
  4° parametro: argomenti passati alla funzione di callback.

- Verifica di validità dei parametri generati

  DH_check(dh, codes);

  codes è di tipo int* e, se la funzione ritorna con errore, può essere usato per controllare i flag.
  La funzione ritorna 1 se la verifica ha successo, 0 altrimenti.

Funzione client

- Creazione della struttura DH che verrà riempita coi paramentri generati dal server.

```
dh = DH_new();
```

- La precedente è l'unica funzione che riguarda esclusivamente il client. È necessario creare una struttura DH perché chi riceve i parametri (generati dal server) deve avere una struttura pronta per collocarli.

Esercizio

- Si consideri la generazione di un canale sicuro Diffie-Hellman.

- Il server:
  - Genera i paramentri Diffie-Hellman e li invia al client
  - Genera dh->priv_key e dh->pub_key
  - Invia dh->pub_key al client

- Il client:
  - Genera una struttura DH e vi colloca i parametri del server
  - Genera dh->priv_key e dh->pub_key
  - Invia dh->pub_key al server

- Entrambi:
  - Generano la chiave simmetrica Diffie-Hellman

- Il client usa la chiave per cifrare un file, che viene inviato al server che lo decifra.