Sicurezza dei Sistemi Informatici

Esercitazioni OpenSSL

Esercitazione 2

Funzioni hash

Obiettivo

- Scambio file su applicazione client-server.

- Client e server condividono una chiave simmetrica.

- Il client vuole inviare al server un file
  - garantendo la confidenzialità
  - garantendo l’integrità

Funzioni client/server

- Allocazione contesto

```
char* alg="sha1";
const EVP_MD* md = EVP_get_digestbyname(alg);
OpenSSL_add_all_digests();
EVP_MD_CTX* mdctx;
mdctx = malloc(sizeof(EVP_MD_CTX));
```

- Preparazione contesto

```
EVP_MD_CTX_init(mdctx);
EVP_DigestInit(mdctx, md);
```

- Deallocazione contesto

```
EVP_MD_CTX_cleanup(mdctx);
free(mdctx);
```

Operazioni client/server

- Allocazione contesto
- Preparazione contesto
- Calcolo message digest

```
EVP_DigestUpdate(mdctx,&in[pos],k);
EVP_DigestFinal_ex(mdctx,out,&lout);
```

- Deallocazione contesto

Operazioni client/server

- Calcolo message digest

  EVP_DigestUpdate(mdctx,&in[pos],k);
  EVP_DigestFinal_ex(mdctx,out,&lout);

  - mdctx: contesto
  - in: buffer con il testo di cui calcolare il digest
  - pos: posizione corrente nel buffer *in*
  - k: numero di byte di *in* da processare
  - out: buffer per contenere il digest
  - lout: dimensione in byte del digest prodotto

- La dimensione in byte del buffer *out* è calcolabile come EVP_MD_SIZE(const EVP_MD* md).

- L'input della funzione hash non può essere troppo grande. Il digest si calcola *k* byte per volta, invocando più volte la funzione EVP_DigestUpdate().

Esercizio

- Si consideri il cifrario DES in modalità ECB e l'algoritmo di hash SHA-1.

- Fare riferimento all'esercitazione 1 per il recupero della chiave simmetrica su client e server.

- Il client legge un file, e invia il suo contenuto *m* al server, secondo lo schema **E(m, h(m))**.

- Il server decifra quanto ricevuto dal client, ne verifica l'integrità, e lo salva su file locale.

- Utilizzare la funzione di libreria *strncmp()* per confrontare due digest.