OpenSSL API

http://www.fi.infn.it/info/ssl.ps.gz

Cos'è OpenSSL

- Implementazione *open-source* di Secure Socket Layer e Transport Layer Security
- Attualmente le API relative ad OpenSSL sono composte da un insieme di 214 *funzioni (SSL calls) C*

Strutture dati principali

- SSL_METHOD
- SSL_CHIPER
- SSL_CTX
- SSL_SESSION
- SSL

Funzioni Principali

- `int SSL_library_init(void);`
- `SSL_METHOD *SSLv3_method(void);`
- `SSL_CTX *SSL_CTX_new(SSL_METHOD *method);`
- `int SSL_CTX_use_certificate_file(SSL_CTX *ctx, const char *file, int type);`
- `SSL *SSL_new(SSL_CTX *ctx);`
- `int SSL_accept(SSL *ssl);`
- `int SSL_set_fd(SSL *ssl, int fd);`
- `int SSL_connect(SSL *ssl);`
- `int SSL_read(SSL *ssl, void *buf, int n);`
- `int SSL_write(SSL *ssl, const void *buf, int n);`
- `int SSL_shutdown(SSL *ssl);`
- `void SSL_free(SSL *ssl):`

Esempio: SSL server (1)

```
SSL_CTX*    ctx;
SSL*        ssl;
SSL_METHOD  *meth;
int main(int argc, char **argv) {
   ...
   int connfd;
   SSL_load_error_strings();
   SSL_library_init();
   meth = SSLv3_server_method();
   ctx = SSL_CTX_new(meth);
   SSL_CTX_use_certificate_file(ctx, CERTF, SSL_FILETYPE_PEM);
   SSL_CTX_use_PrivateKey_file(ctx, KEYF, SSL_FILETYPE_PEM);
   if (!SSL_CTX_check_private_key(ctx)) {
     fprintf(stderr,"Private key does not match the certificate
   public key\n");
     exit(1);
   }
   socket(…); listen(…);
   connfd = accept(…);
```

Esempio: SSL server (2)

```
        ssl = SSL_new(ctx);
        SSL_set_fd(ssl,connfd); /* connfd è un socket TCP */
        err = SSL_accept(ssl);
        fprintf(stderr,"Started SSL connection using %s\n",
SSL_get_cipher (ssl));
        err = SSL_read (ssl, buf, sizeof(buf) - 1);
        buf[err] = '\0';
        printf("%s\n",buf);
        err = SSL_write (ssl, "I hear you", strlen("I hear
you"));
        SSL_free (ssl);
        close(connfd);
        SSL_CTX_free (ctx);
        return 0;
}
```

Esempio: SSL client

```
SSL_CTX*     ctx;
SSL*         ssl;
SSL_METHOD   *meth;
X509*        server_cert;
main() {
        ...
        meth = SSLv3_client_method();
        ...
        err = SSL_connect(ssl);
        server_cert = SSL_get_peer_certificate (ssl);
        err = SSL_write(ssl, "Hello World", strlen("Hello World"));
        err = SSL_read (ssl, buf, sizeof(buf) - 1);
        buf[err]='\0';
        printf("%s\n",buf);
        SSL_shutdown (ssl);
        close (socket);
        SSL_free (ssl);
        SSL_CTX_free (ctx);
        return 0;
}
```

Conclusioni

Si possono scrivere applicazioni tramite le API OpenSSL basate sui socket TCP ma che in più includono meccanismi di autenticazione, non-repudiation e confidenzialità, questo permette di avere applicazioni che forniscono servizi sicuri