Consiglio Nazionale delle Ricerche

Voto elettronico: uno strumento per la gestione di un seggio elettorale virtuale

S. Ruberti, M. Martinelli, L. Luconi Trombacchi

IIT TR-13/2002

Technical Report

Luglio 2002

Istituto di Informatica e Telematica

INDICE

INTRODUZIONE

L’obiettivo di questo documento è quello di illustrare le soluzioni tecniche adottate per la progettazione e la realizzazione di un sistema informatico sicuro per l’organizzazione e la gestione di un seggio elettorale virtuale.

Per sistema informatico si intende l'insieme degli strumenti informatici impiegati per il trattamento automatico delle informazioni di un'organizzazione, al fine di agevolare le funzioni del suo sistema informativo.

In particolare questo sistema prevede l’impiego di un modello di tipo client-server basato su form accessibili via Web tramite canale sicuro sfruttando il protocollo SSL e consentendo l’accesso ai dati mantenuti su un database SQL.

Questo tipo di soluzione tecnica risulta essere molto versatile consentendone l’impiego in tutti quei casi dove il corpo elettorale risulti distribuito geograficamente e quindi ottimizzando il più dispendioso metodo di votazione basato su seggi elettorali tradizionali.

i. Struttura del documento

Nel capitolo 1 vengono illustrati i metodi utilizzati dall’applicazione per garantire la sicurezza del sistema, con particolare riferimento al protocollo SSL.

Nel capitolo 2 vengono illustrate in modo approfondito, le principali caratteristiche del protocollo HTTP, del linguaggio di programmazione utilizzato PERL e del sistema di gestione della base di dati MySQL.

Nel capitolo 3 viene illustrata l’applicazione: l’analisi delle operazioni che possono essere eseguite dagli utenti del sistema, la progettazione e l’implementazione della base di dati e dell’interfaccia utente.

Il capitolo 4 presenta una panoramica d’uso dell’intera applicazione e fornisce, pertanto, una sorta di manuale utente per coloro che la utilizzeranno. In modo specifico sono descritte le operazioni a carico del presidente del seggio e quelle invece a carico dei singoli votanti.

1. SICUREZZA

Un tipo di applicazione come quella in oggetto, dove sono gestite informazioni riservate e personali, richiede una particolare attenzione il problema della sicurezza e tutti quei meccanismi necessari per garantirla. Trattandosi di un'applicazione accessibile tramite la rete Internet, occorre tenere sotto controllo sia la sicurezza dei dati, sia la sicurezza delle transazioni. Per quest'ultimo aspetto si devono garantire sia l'integrità che la riservatezza dei dati trattati.

Il problema può essere affrontato suddividendolo in più livelli logici:

- livello di applicazione;
- livello di comunicazione;
- livello di sistema operativo.

A seconda del livello considerato vengono implementati meccanismi diversi per garantire la sicurezza. In un qualsiasi sistema, sia esso manuale oppure automatico, il controllo dei dati previene eventi indesiderati, quali la perdita e/o l'incongruenza degli stessi e/o gli accessi non autorizzati. Un sistema di controllo è quello che prevede l’impiego di procedure di logging della attività, che costituiscono sicuramente un requisito fondamentale per il monitoraggio del sistema. Idealmente, una procedura di logging dovrebbe tenere traccia di tutte le transazioni che avvengono nel sistema e quando e da chi sono state generate. La procedura di controllo dovrebbe, insomma, rendere possibile la ricostruzione di ogni fase della vita di un record. Tale traccia dovrebbe permettere di individuare le responsabilità al fine di mantenere l'accuratezza e l'integrità dei dati.

Un altro aspetto importante per garantire la robustezza del sistema è l'impiego di procedure per la copia dei dati su dispositivi diversi di memorizzazione e il loro recupero in caso di malfunzionamenti e/o danneggiamento della base di dati.

La varietà dei tipi di attacchi di cui un sistema telematico può essere vittima, impone che venga posta molta attenzione nell'organizzazione di un buon piano di difesa. È buona norma, ad esempio, operare un costante aggiornamento del software e dei metodi di protezione, in modo da impedire ad un possibile “hacker” di sfruttare bugs noti o eventuali errori di programmazione.

1.1 Livello di applicazione

A questo livello occorre predisporre dei meccanismi in grado di controllare che l'accesso ai dati sia consentito solo agli utenti autorizzati e stabilire con esattezza le operazioni che essi sono abilitati a compiere. Ad ogni utente deve essere assegnato un insieme di diritti che stabiliscano i tipi di operazioni che esso può compiere sui dati.

1.2 Livello di comunicazione

I requisiti per garantire lo scambio di comunicazioni sicure possono essere riassunti nei punti seguenti:

- riservatezza dell'informazione: solo le persone autorizzate hanno accesso ai dati riservati;
- integrità dell'informazione: garantisce che durante la trasmissione delle informazioni, esse non sono state modificate, alterate o distrutte;
- garanzia di autenticità dell'interlocutore: deve essere previsto un meccanismo che associ univocamente ad ogni interlocutore una determinata azione.

Tutto questo può essere implementato mediante l'adozione: sia di un protocollo di comunicazione sicuro, che faccia uso di transazioni criptate, come SSL o TLS, come descritto nel paragrafo 1.6; sia tramite l'impiego di meccanismi in grado di associare la firma digitale ad un documento informatico, cioè implementare il meccanismo della firma elettronica di documenti informatici, come illustrato nel paragrafo 1.4.

1.3 Livello di sistema operativo

Uno dei metodi più utilizzati per rendere sicuro un server di rete è quello di disabilitare tutti i processi non indispensabili per l'erogazione del servizio e rendere sicuri quelli necessari. Alcuni processi, se mal configurati, possono consentire infatti accessi indesiderati da parte di utenti non autorizzati.

Tutti i servizi che non prevedono transazioni criptate, come per esempio il servizio TELNET, FTP (File Transfer Protocol), NNTP (Network News Transfer Protocol), RPC (Remote Procedure Calls), ecc., dovrebbero essere disattivati o comunque filtrati da firewall. Per transazioni in ambito LAN e/o WAN, è

consigliabile utilizzare protocolli che implementino meccanismi di cifratura dei dati (SSH, HTTPS, ecc.).

1.4 La firma elettronica

La firma elettronica ha lo scopo di certificare e garantire l'integrità e la provenienza dei dati. Per ottenere questo risultato occorre garantire che l'origine di questi sia autentica e che i dati non siano stati alterati dopo la firma.

La firma elettronica impiega meccanismi che fanno uso di algoritmi crittografici a ***chiavi pubbliche***, o anche detti a ***chiavi asimmetriche***, in quanto utilizzano chiavi diverse (*chiave privata e chiave pubblica*) per le operazioni di codifica e decodifica.

L'apposizione della firma elettronica ad un documento informatico è in grado di soddisfare i seguenti requisiti che sono propri della firma autografa:

- autenticità della firma, poiché la chiave privata utilizzata per produrla è nota solo al mittente e la funzione utilizzata è unidirezionale. Quindi solo il mittente può aver prodotto la firma del messaggio. Ciò impedisce al mittente di ripudiare la firma;
- la firma non è falsificabile, in quanto la funzione utilizzata per decifrare è nota a tutti, ma la chiave privata per la sua creazione è nota solo al mittente, quindi, chiunque voglia falsificare la firma del mittente dovrebbe scoprirne la chiave privata;
- la firma è non riutilizzabile su un altro documento poiché essa è immagine dello stesso documento firmato. Inoltre, il documento non può essere alterato perché, altrimenti, la firma dovrebbe cambiare di conseguenza;
- l'autenticità della firma, mediante l'utilizzo della chiave pubblica del mittente, può essere verificata anche da terze parti.

Nel processo di firma elettronica di un documento informatico che prevede l'impiego di algoritmi di crittografia asimmetrica, svolge un ruolo fondamentale l'ente certificatore (Certification Authority o CA), il quale ha il compito di identificare la persona fisica a cui sarà rilasciata la coppia di chiavi pubblica e privata, nonché un certificato contenente le informazioni necessarie per il processo di verifica della firma.

1.5 Certificati

Un certificato elettronico è un file contenente la chiave pubblica, i dati identificativi del suo titolare, e una o più firme di certificazione. Un certificato elettronico fornisce una prova inconfutabile di riconoscimento di una persona oppure di un servizio di rete.

Come accennato precedentemente le autorità di certificazione (Certification Authorities) validano le identità ed emettono i certificati. Un'azienda può creare una propria infrastruttura di certificazione oppure rivolgersi ad un fornitore di servizi esterno. I metodi usati per validare un'identità variano dipendentemente dalle politiche stabilite per ogni singola CA. In generale, prima di emettere un certificato, la CA deve utilizzare le proprie procedure di verifica per quel tipo di certificato, per assicurare che l'entità che sta per richiedere un certificato sia in realtà chi dichiara di essere.

I certificati aiutano a prevenire l'uso di chiavi pubbliche falsificate (assunzione di un'altra identità). Soltanto la chiave pubblica del certificato funzionerà con la corrispondente chiave privata posseduta dall'entità identificata dal certificato stesso.

In aggiunta alla chiave pubblica, un certificato contiene sempre il nome dell'entità che identifica, una data di scadenza, il nome della CA che ha emesso il certificato, un numero seriale e altre informazioni. È importante che i certificati includano sempre la firma digitale della CA emittente.

La firma digitale della CA permette al certificato di espletare la funzione di “referente” per gli utenti che riconoscono l’autenticità della CA, ma non conoscono l'entità identificata dal certificato.

Esistono cinque tipi di certificati usati comunemente:

- **Certificati SSL Client**. Usati per identificare i client da parte dei server attraverso il protocollo SSL (autenticazione client). Tipicamente l'identità del client si assume essere la stessa dell'identità di un essere umano, per esempio l’utente che accede ad un servizio. Questi certificati possono essere usati per firmare un modulo elettronico (form) di una pagina web.
- **Certificati SSL Server**. Usati per identificare i server da parte dei client attraverso il protocollo SSL (autenticazione server). L'autenticazione server

può essere usata con o senza autenticazione client. L'autenticazione server è un requisito della sessione SSL criptata (si veda il paragrafo 1.7).

- **Certificati S/MIME**. Usati per firmare e codificare e-mail. Come con i certificati SSL Client, l'identità del client si assume essere la stessa dell'identità di un essere umano. Un singolo certificato può essere usato sia come certificato S/MIME che come certificato SSL. I certificati S/MIME possono anche essere usati per firmare un modulo elettronico (form) di una pagina web.
- **Certificati per la firma di oggetti**. Usati per identificare i firmatari di codice Java, di codice JavaScript o di altri file firmati.
- **Certificati di CA**. Usati per identificare le Autorità di Certificazione. I software utilizzati sui client e sul server usano certificati di CA per determinare quali altri certificati possono essere fidati.

1.6 Il protocollo SSL

Secure Sockets Layer (SSL) è un protocollo che fornisce una comunicazione sicura, vitale per la transazioni riservate sulla rete. SSL è stato universalmente accettato come protocollo per la comunicazione su Internet, autenticata e criptata, tra client e server. SSL è stato sviluppato inizialmente da Netscape; il protocollo Transport Layer Security TLS è l'evoluzione del primo, come standard ratificato da IETF (Internet Engineering Task Force).

Sebbene il protocollo SSL abbia guadagnato notorietà come strumento per assicurare la trasmissione sicura per le transazioni HTTP, può essere usato per rendere sicure le transazioni di protocolli Internet intrinsecamente insicuri, come FTP, POP, IMAP, NNTP, ecc.

Il protocollo SSL fornisce meccanismi di privacy tra un'applicazione client e una server, usando un sistema di certificati server, certificati distribuiti da una Certification Authority e, opzionalmente, certificati client. L'uso del certificato di tipo server assicura all'utente la riservatezza nella transizione delle informazioni. In modo analogo, l’impiego del certificato di tipo client, garantisce al server l’autenticità dell’utente.

Il protocollo SSL stabilisce un canale sicuro tra client e server, cioè un canale che rispetta le seguenti tre proprietà:

- **Confidenzialità della transazione**: tutti i messaggi che usano SSL sono criptati usando una chiave segreta. L'informazione è per questo privata da e per la macchina client, così che eventuali intrusi non possano decifrare i messaggi intercettati.
- **Validazione della transazione**: l'affidabilità del canale è assicurata dall'uso di controlli di integrità sui messaggi. Ciò assicura che, nel caso in cui un messaggio venga alterato, questo non risulti più valido.
- **Autenticazione del server**: il canale è sempre autenticato dal lato server e, opzionalmente, dal lato client. Poiché il server spedisce il suo certificato di identità insieme ad ogni messaggio, il client può sempre essere sicuro che il messaggio provenga da una fonte fidata. Nel caso in cui anche il client invii la certificazione in risposta, il server può similarmente essere certo che i messaggi provengano da una fonte fidata.

Per rispettare queste tre proprietà, le transazioni SSL sono divise in due differenti sezioni:

- **Procedura di handshake** nella quale vengono scambiati i dettagli del client e del server, e sono negoziate e concordate certe impostazioni per la sessione. Durante questa procedura il client ed il server verificheranno quali algoritmi crittografici e dimensioni delle chiavi saranno usati per la sessione. Il server fornirà anche un certificato a chiave pubblica con il quale il client verificherà l'autenticità del server. Al termine i partner della sessione genereranno separatamente un master secret usato per derivare le chiavi di codifica che saranno usate nella sessione seguente: il processo di passaggio dei record.
- **Processo di passaggio dei record** nel quale sono scambiate le informazioni sulla sessione in una forma criptata. Durante questo processo i messaggi criptati con la chiave master saranno spediti tra i computer client ed il server. Il protocollo SSL specifica un formato per questi messaggi, che include un controllo di integrità per assicurare che i messaggi non siano alterati durante la loro trasmissione.

2 SCELTE PROGETTUALI

Le scelte tecnologiche che caratterizzano questo progetto prevedono l'impiego di software *"open source"*, cioè distribuito secondo le norme stabilite dalla GNU GPL (GNU General Public License).

2.1 Accesso ai dati via WEB

Il modo più comune per pubblicare informazioni attraverso la rete è quello di utilizzare un server HTTP (HyperText Transfer Protocol) [RFC2616].

Le informazioni pubblicate in questo modo sono generalmente rivolte a tutti gli utenti della rete Internet e normalmente non è prevista alcuna fase di autenticazione.

L'accesso al servizio viene garantito dall'impiego di un apposito software che svolge le funzioni di HTTP server, che consente l'accesso ad un ramo particolare del file system del server dove risiedono i dati predisposti per la consultazione. L'utilizzo del servizio HTTP si compone di una serie di transazioni, ognuna delle quali si articola nelle seguenti fasi principali:

- apertura della connessione;
- invio da parte del client di una richiesta;
- risposta da parte del server;
- chiusura della connessione.

2.1.1 Common Gateway Interface

Un server HTTP non offre solo un servizio di semplice consultazione di documenti, ma permette anche l'utilizzo di programmi residenti sul server tramite i quali si possono compiere particolari azioni sul server stesso. Questi programmi sono collocati normalmente in un'area distinta da quella destinata ad ospitare i documenti, in modo tale che detti programmi possano solo essere eseguiti e non visualizzati. In questo contesto, tali programmi sono definiti gateway, e normalmente vengono chiamati programmi CGI, o più comunemente, cgi-bin. Tutti i programmi CGI, si trovano nella directory /cgi-bin/ la quale fa riferimento non alla directory /cgi-bin/ del filesystem del sistema ma, nel caso ad esempio del server Apache, alla directory definita dalla variabile ScriptAlias nel suo file di configurazione.

Quando un client invia una richiesta di accesso a una risorsa che costituisce un programma gateway, il server lo esegue e ne restituisce l'output al client: la costruzione dell'intestazione del messaggio di risposta è a carico del programma gateway.

2.2 Server web Apache + SSL

Apache è il server HTTP utilizzato per questo progetto. Come nella maggior parte dei sistemi *"open source"*, Apache è progettato in maniera modulare aperta ed estendibile, in modo da consentire l'aggiunta di moduli prodotti da terze parti. Una delle estensioni più importanti di cui Apache dispone, e che è risultata fondamentale per lo sviluppo del presente progetto è costituita dal modulo denominato Mod_SSL, che consente l'uso di connessioni protette sfruttando il protocollo SSL. Per poter usare Apache con l'estensione SSL, è necessario personalizzare opportunamente il file di configurazione del server: /etc/httpd/conf/httpd.conf.

```
DocumentRoot /var/www/html
ScriptAlias /cgi-bin/ /var/www/cgi-bin/
AddHandler cgi-script .cgi
<Directory /var/www/cgi-bin>
Options ExecCgi Indexes FollowSymlinks
AllowOverride AuthConfig FileInfo Indexes Limit Options
order deny,allow
allow from all
</Directory>
# Direttive di configurazione per l'attivazione del supporto SSL
SSLEngine on
SSLCertificateFile /etc/httpd/conf/ssl.crt/voto_elettronico.cert
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/voto_elettronico.key
```

Figura 2.1: Principali modifiche al file di configurazione di Apache.

2.2.1 OpenSSL

OpenSSL è un software *"open source"* costituito da un insieme di programmi e routine di libreria, che implementa i protocolli di rete SSL e TLS, e i relativi standard crittografici da essi richiesti [VMC02].

In particolare il programma openssl è uno strumento a linea di comando che permette di utilizzare le varie funzioni crittografiche della libreria *crypto* direttamente dalla shell dell'utente.

Può essere utilizzato per:

- creare le chiavi codificate secondo gli algoritmi RSA e DSA;
- creare certificati X.509 e le liste dei certificati sospesi e revocati;

- calcolare il Message Digest;
- codificare e decodificare con algoritmi di cifratura;
- testare il client e il server su una comunicazione SSL/TLS;
- gestire la posta codificata o firmata mediante lo standard S/MIME.

2.2.1.1 Creazione di un certificato SSL-server

Per poter utilizzare il protocollo SSL in ambito Web, in modo da supportare transazioni sicure, è necessario richiedere ed utilizzare un certificato di tipo SSL-server. Per tale scopo è stato utilizzato il toolkit OpenSSL sopra descritto.
Come primo passo è necessario creare una chiave privata e una richiesta di certificato:

```
openssl req -new -config /usr/local/ssl/openssl.cnf > voto_elettronico.csr
```

OpenSSL ha un proprio file di configurazione /usr/local/openssl/openssl.cnf. Questo file può essere personalizzato in modo da fissare permanentemente alcuni parametri indispensabili per la generazione dei certificati, come per esempio le coordinate “geografiche” della macchina che ospita tale prodotto (Country Code, Nazione, Provincia, Organizzazione, ecc.). Tuttavia questa personalizzazione è opzionale: in tal modo, durante la fase di generazione di un certificato, tali coordinate saranno richieste un modo interattivo. L’opzione –config permette, comunque, di specificare il file di configurazione nel caso ciò si renda necessario.
Durante la generazione della chiave privata, viene obbligatamente richiesta una *passphrase* di almeno 4 caratteri, che eventualmente può in seguito essere eliminata tramite il comando:

```
openssl rsa -in privkey.pem -out voto_elettronico.key
```

A questo punto la richiesta deve essere “certificata” e ciò avviene tramite la sua firma con l’ausilio della chiave privata generata precedentemente:

```
openssl x509 -in voto_elettronico.csr -out voto_elettronico.cert
-req -signkey voto_elettronico.key -days 365
```

A questo punto la procedura è terminata: sono state generate la coppia di chiavi, pubblica e privata, ed il certificato SSL-server. La chiave privata del server (voto_elettronico.key) ed il certificato (voto_elettronico.cert) devono essere copiati o spostati nelle directory indicate nel file di configurazione httpd.conf. Nell'esempio di figura 2.1 è stata utilizzata la directory /etc/httpd/conf/ssl.crt/ per il

file (voto_elettronico.cert) mentre per il file (voto_elettronico.key) è stata specificata la directory /etc/httpd/conf/ssl.key/.

2.3 Perl

Il Perl (Practical Extraction and Report Language) [WCO00][SSP02] costituisce il linguaggio di programmazione utilizzato per la realizzazione del presente progetto. È un linguaggio di programmazione interpretato e ciò significa che il codice prodotto viene generato omettendo la fase di compilazione del/i file seguente/i per la generazione del file eseguibile in forma binaria. I programmi Perl sono degli script eseguiti dall'interprete Perl che per convenzione è collocato in /usr/bin/ o in /usr/local/bin/.

2.4 Sistema di Gestione di Basi di Dati

Un Sistema di Gestione di Basi di Dati (SGBD) è un sistema centralizzato o distribuito che consente di:

- definire schemi di dati;
- scegliere le strutture dati per la memorizzazione e l'accesso ai dati;
- memorizzare i dati rispettando i vincoli definiti nello schema;
- recuperare e modificare i dati interattivamente o tramite appositi programmi.

Un SGBD, per essere considerato tale, deve avere particolari caratteristiche. A nostro avviso, tra le più importanti e che permettono di comprendere il significato di un SGDB sono le seguenti:

- possibilità di gestione di grandi quantità di dati. Convenzionalmente si può affermare che un gruppo di informazioni è di grandi dimensioni quando queste non possono essere contenute tutte simultaneamente nella memoria centrale dell'elaboratore. In generale un SGBD non dovrebbe porre limiti alle dimensioni, tranne quelle imposte dai supporti fisici in cui devono essere memorizzate le informazioni;
- possibilità di condivisione dei dati: l'idea che sta alla base dei sistemi di gestione dei dati è quella di accentrare le informazioni in un sistema di amministrazione unico. In questo senso è poi necessario che questi dati siano condivisibili da diverse applicazioni e da diversi utenti;

- affidabilità e persistenza dei dati: si definiscono dati persistenti quando essi continuano a esistere anche dopo lo spegnimento della macchina con cui vengono elaborati; sono affidabili quando gli eventi per cui si possono produrre alterazioni accidentali sono estremamente limitati;
- controllo di accesso ai dati: dovendo trattare una grande mole di dati in modo condiviso, è indispensabile che esistano dei sistemi di controllo degli accessi, per evitare che determinate informazioni possano essere ottenute da chi non è autorizzato, oppure che vengano modificate da chi non ne è il responsabile.

Per una completa trattazione dell'argomento si veda [AGO97] [Alb01].

2.4.1 MySQL

L'SGDB utilizzato per questo progetto è il MySQL. Esso può essere considerato uno tra i più popolari server SQL disponibile per piattaforme Unix/Linux/Windows. È un database SQL multi-thread e multi-user con spiccate caratteristiche di velocità e robustezza. MySQL è distribuito dalla MySQL AB (Svedese) rispettando le condizioni di licenza d'uso della GNU GPL. Fornisce il supporto per la gestione delle transazioni ed è stato testato su piattaforme Unix differenti. Consente l'interfacciamento con un numero elevato di linguaggi di programmazione, per la gestione e la manipolazione dei dati. L'insieme completo delle API per MyQSL sono disponibili per i seguenti linguaggi di programmazione: C, C++, Eiffel, Java, Perl, PHP, Python e Tcl. Fornisce un completo supporto ODBC e dispone di un numero molto elevato di tool sviluppati da terze parti. MySQL si basa su un sistema di tipo client-server: il programma client interagisce con la base di dati attraverso la sottomissione di richieste al server.

Per questo progetto abbiamo utilizzato la libreria Perl DBI (Perl DataBase Interface), che consente di interfacciare gli script Perl in formato CGI direttamente con la base di dati. La libreria DBI offre una interfaccia comune per vari tipi di SQL server. Per permettere questa astrazione dal tipo di server SQL utilizzato, la libreria DBI è suddivisa in due parti distinte:

- la libreria vera e propria (cioè l'insieme delle funzioni e comandi per l'accesso al server SQL);
- un driver specifico per il server SQL utilizzato (DataBase Driver o DBD).

L’utilizzo della libreria DBI permette di cambiare il tipo di server SQL utilizzato, apportando, ove necessario, semplici modifiche al codice sviluppato.
MySQL prevede la definizione di un utente amministratore del servizio che può anche non coincidere con l’amministratore del sistema, cioè l’utente ***root***. Generalmente è consuetudine utilizzare l’utente ***mysql*** [AWCLD02].

3 DESCRIZIONE DEL PROGETTO

Il progetto nasce dall'esigenza di realizzare un sistema automatizzato per la gestione di un seggio elettorale a votazione palese mediante il supporto delle tecnologie informatiche.

Questa applicazione è stata realizzata impiegando le apparecchiature già in uso alla Registration Authority, con l'obbiettivo di realizzare un primo prototipo funzionante di seggio elettorale virtuale, sufficientemente flessibile e in grado di adattarsi facilmente a sviluppi futuri.

3.1 Descrizione della realtà

La descrizione del flusso informativo dei dati è abbastanza semplice. È stata prevista una fase di costituzione di un seggio elettorale ed una successiva fase, che coincide con l'apertura del seggio, che prevede la registrazione delle preferenze espresse da parte dei votanti ai quesiti proposti. Il tutto si conclude con lo spoglio delle schede elettorali elettroniche.

3.2 Descrizione dei dati

Per definire l'ambiemte su cui si vuole operare, occorre innanzi tutto specificare le entità che si vogliono rappresentare e le relazioni che le coinvolgono.

Una scheda elettorale è costituita da un numero finito di quesiti per i quali è necessario che ogni singolo votante esprima il proprio voto. Una scheda elettorale può contenere anche un solo quesito.

Per ogni quesito è prevista una descrizione sommaria, la formulazione del quesito su cui i votanti dovranno esprimere la propria preferenza, la data di inizio e la data di fine votazione. Inoltre per ogni quesito, in fase di scrutinio, è necessario conoscere quali e quante sono state le preferenze espresse dal gruppo dei votanti.

Per ogni votante è previsto l'inserimento del nome e cognome, l'indirizzo di posta elettronica, un codice utente (PIN) e la relativa password necessari per l'autenticazione sul sistema.

3.3 Descrizione delle operazioni

Le operazioni che possono essere effettuate dagli utenti dell'applicazione sui dati sono:

- definizione di un nuovo account per un votante da parte del presidente del seggio elettorale;
- visualizzazione dell'elenco dei votanti accreditati al voto;
- modifica dei dati relativi all'identificazione dei votanti;
- attivazione di una tornata elettorale, con conseguente definizione di un numero finito di quesiti;
- procedura di voto da parte dei singoli votanti;
- spoglio delle schede elettorali virtuali.

3.4 Progetto della base di dati

Abbiamo provveduto alla definizione e formalizzazione di una base di dati capace di gestire le informazioni relative ad un seggio elettorale virtuale. Il database è stato strutturato secondo il seguente schema:

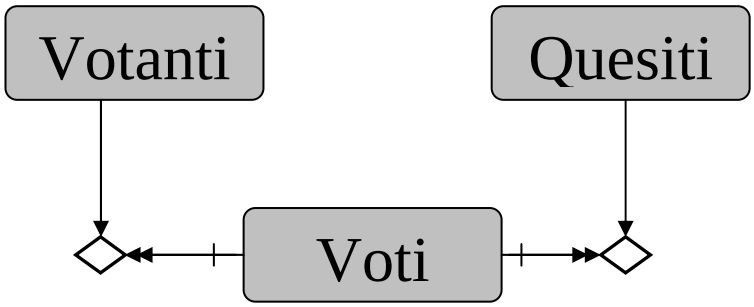

Figura 3.1: Schema entità-relazioni della base di dati

La rappresentazione grafica della struttura della base di dati evidenzia la semplicità architetturale. Come si può notare la base di dati è costituita solo da tre tabelle: VOTANTI, QUESITI e VOTI.

La tabella VOTANTI contiene l'elenco delle persone che costituiscono il corpo elettorale. In questa tabella ogni individuo accreditato al voto viene identificato da un codice univoco che verrà utilizzato per poter associare il votante al/i quesito/i su cui viene chiamato al voto. Gli altri dati contenuti in questa tabella sono il nome e cognome del votante, il codice di accesso (PIN) e la password che gli consentiranno di poter esprimere il proprio voto accedendo alla sezione riservata al

votante sul sito WEB. La registrazione dei nominativi relativi al singoli accreditati al voto è a cura del presidente del seggio elettorale.

La definizione della tabella votanti è la seguente:

Field	Type	Null	Key	Default	Extra
id	mediumint(8) unsigned		PRI	NULL	auto_increment
nome	varchar(64)		MUL		
email	varchar(64)	YES		NULL	
username	varchar(32)		MUL		
pin	varchar(32)				

Tabella 3.1: Tabella votanti

La tabella QUESITI contiene le informazioni inerenti gli argomenti oggetto delle votazioni. Ogni quesito è identificato da un codice univoco che verrà utilizzato per poter associare il quesito al voto espresso dal votante, da un attributo che contiene il quesito vero e proprio su cui viene richiesto il voto, da una sua descrizione esplicativa e dal periodo (date di inizio e fine).

La definizione della tabella quesiti è la seguente:

Field	Type	Null	Key	Default	Extra
id	mediumint(8) unsigned		PRI	NULL	auto_increment
quesito	varchar(255)		MUL		
descrizione	text				
data_inizio	date		MUL	0000-00-00	
data_fine	date		MUL	0000-00-00	

Tabella 3.2: Tabella quesiti

La tabella VOTI tiene traccia delle singole preferenze espresse dai votanti per i quesiti per cui viene richiesto di votare. Ogni voto è identificato da un codice univoco, dai codici identificativi rispettivamente del singolo votante e del quesito per cui è stato richiesto il voto, dal voto ed infine dalla data in cui questo è stato espresso. La definizione della tabella voti è la seguente:

Field	Type	Null	Key	Default	Extra
id	mediumint(8) unsigned		PRI	NULL	auto_increment
idquesito	mediumint(8) unsigned		MUL	0	
idvotante	mediumint(8) unsigned		MUL	0	
voto	enum('SI','NO','ASTENUTO')		MUL	SI	
datavoto	timestamp(14)	YES		NULL	

Tabella 3.3: Tabella voti

Questo tipo di struttura consente di effettuare in maniera efficiente ricerche all'interno della base di dati come per esempio:

- la gestione dello scrutinio delle schede elettorali;
- la possibilità di ottenere per ogni quesito, le preferenze espresse da ogni singolo votante;
- le preferenze espresse dai singoli votanti e la relativa data di votazione, nel caso di voto palese.

3.5 Interfaccia grafica di tipo client-server

L'interfaccia grafica utilizzata per la gestione delle transazioni di tipo client-server utilizza un server WEB APACHE [LL99], di pubblico dominio, opportunamente configurato in modo da supportare transazioni sulla rete Internet in modalità sicura basandosi sul protocollo SSL.

La struttura del sistema client-server impiegato è la seguente:

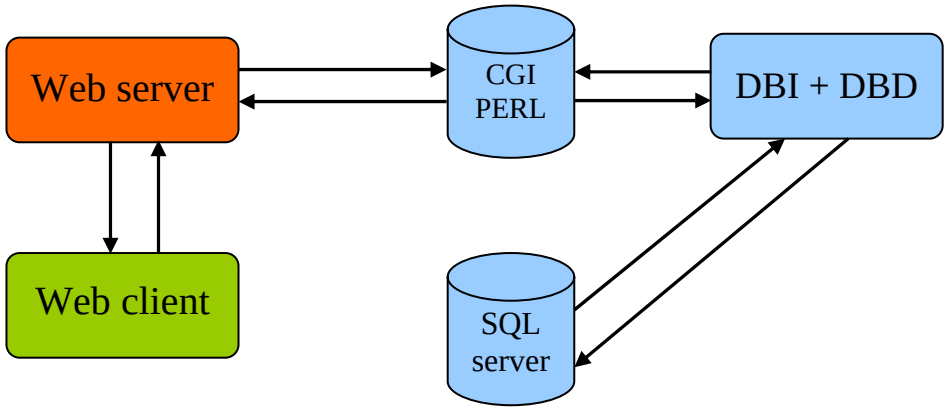

Figura 2: Struttura del modello client-server

Al fine di rendere il seggio elettorale virtuale flessibile, facilmente gestibile ed estendibile in funzione anche di sviluppi futuri, sono state previste due sezioni principali: un riservata al presidente del seggio e una riservata al corpo elettorale.

3.5.1 Sezione riservata la presidente del seggio

Al fine di rendere più semplice ed agevole la realizzazione delle pagine dedicate alla votazione, è stata approntata una sezione riservata al presidente del seggio elettorale all'interno della quale è stato previsto un form on-line che permette di specificare, di volta in volta, i quesiti sui quali i votanti sono chiamati alle "urne elettroniche" ed il periodo durante il quale sarà possibile esprimere il voto. La fase di creazione dei quesiti elettorali si conclude con la registrazione nel database MySQL dei questi elettorali. Questo sistema è interamente automatico e produce,

dinamicamente, le pagine Web (e quindi i relativi form on-line) dedicati alla votazione in oggetto. Tale soluzione è volta a semplificare la gestione dell'intero sistema ed esclude, in maniera totale, l'accesso alla macchina secondo i protocolli e le applicazioni tradizionali (ftp, telnet, ssh). Effettuata la fase di predisposizione dei quesiti, sarà compito del presidente del seggio informare i membri accreditati al voto (ad esempio tramite una apposita mailing-list) dell'apertura del seggio elettorale virtuale.

3.5.2 Sezione rivolta al corpo elettorale

Questa sezione è accessibile via Web ai membri accreditati al voto, tramite un codice identificativo univoco (PIN) ed una password che identificano il votante. PIN e password vengono generati in maniera casuale dal presidente del seggio elettorale, tramite un'apposita procedura. PIN e password potranno essere notificati al votante tramite posta di superficie, oppure tramite posta elettronica certificata. Le politiche legate alla gestione del tempo di vita dei PIN assegnati a ciascun membro del corpo elettorale sono a discrezione del presidente del seggio elettorale, il quale potrà decidere se generarli ad ogni votazione, oppure assegnargli una validità temporale.

Le pagine Web dedicate alla votazione, che sono generate dinamicamente sulla base dei dati inseriti dal presidente del seggio elettorale, sono accessibili in modalità sicura via protocollo SSL, in modo da proteggere il PIN e la password associati a ciascun votante e sono corredate anche di apposite istruzioni per effettuare la votazione.

4 MANUALE UTENTE

4.1 Installazione

Uno dei maggiori vantaggi di questa applicazione è che non prevede l'installazione di software aggiuntivo sulle macchine di tipo client, in quanto l'interfaccia utente, costituisce un qualsiasi programma di accesso per la navigazione Internet che supporti JavaScript. L'applicazione è stata sviluppata ponendo particolare attenzione alla portabilità del codice, volta a consentirne l'uso dai più comuni browser a disposizione degli utenti Internet. È stata testata impiegando personal computer equipaggiati con sistema operativo Windows 2000 e Linux RedHat 7.3. I browser utilizzati sono stati: Netscape Communicator (sia nelle versioni Windows che Linux) e Microsoft Internet Explorer, versioni 5.0 e successive. L'aspetto delle pagine Web prodotte, differisce in maniera irrilevante a seconda del tipo di browser e del sistema operativo usati. Il codice JavaScript, che controlla la correttezza dei valori dei dati inseriti dall'utente nei moduli delle pagine Web, non contiene funzioni specifiche introdotte da Netscape o Microsoft incompatibili con gli altri browser.

4.1.1 Configurazione del server

Sulla macchina destinata ad ospitare l'applicazione e la relativa base di dati che si intende utilizzare, devono essere installati e correttamente configurati, i seguenti programmi:

- un Web server che preveda il supporto del protocollo SSL;
- l'interprete del linguaggio Perl necessario per poter eseguire gli script CGI;
- il sistema di gestione di basi di dati MySQL;
- le librerie Perl DBI/DBD per l'interfacciamento tra il Perl e la base di dati MySQL;
- la libreria CGI per facilitare la realizzazione di script CGI in linguaggio Perl.

4.1.2 Configurazione dell'applicazione

In questo paragrafo viene mostrata la configurazione dell'applicazione e la procedura per la creazione della base di dati. L'applicazione prevede l'impiego di alcuni documenti HTML e i corrispondenti script cgi-bin. Questi ultimi fanno

riferimento ad un file di configurazione che contiene le definizioni e inizializzazioni necessarie per il database utilizzato.

```
#!/usr/local/bin/perl
#
# SQL Information
#
$SQLTYPE = "mysql";
$DB_NAME = "VOTO_ELETTRONICO";
$DB_USERNAME = "VOTODB";
$DB_PASSWORD = "some_password";
$DB_SERVER = "www.foo.it";
1;
```

Figura 4.1: dbm.ph

Le variabili inizializzate in questo file specificano il tipo di database utilizzato, il nome della base di dati, la username, la relativa password ed il nome del server che ospita la base di dati. Queste informazioni sono indispensabili per poter utilizzare la libreria Perl DBI/DBD. Come si può notare, la base di dati può essere ospitata anche su un server distinto rispetto a quello che ospita il servizio Web. In questo caso, sia il server Web che la base di dati sono ospitate, per semplicità, sullo stesso server.

Per poter creare il database prima è necessario registrare il suo amministratore tra gli utenti del sistema MySQL. Per fare ciò si utilizza il comando:

```
shell> mysql --user=root mysql
mysql> INSERT INTO user (Host,User,Password)
VALUES('www.foo.it','VOTODB',PASSWORD('some_password'));
mysql> FLUSH PRIVILEGES;
```

Figura 4.2: Comandi SQL per aggiungere un utente

Per creare la base di dati e le principali tabelle si può operare utilizzando la shell dei comandi del database MySQL, specificando in sequenza i vari comandi SQL, oppure si può utilizzare uno script shell Unix opportunamente predisposto.

Una volta creata la base di dati è indispensabile specificare quali azioni l'utente VOTODB può compiere sulla base di dati:

```
shell> mysql --user=root mysql
mysql> INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update_priv)
VALUES
('www.foo.it','VOTO_ELETTRONICO','VOTODB','Y','Y','Y');
mysql> FLUSH PRIVILEGES;
```

Figura 4.3: Comandi SQL per definire le autorizzazioni sulla base di dati

Con questi comandi viene specificato che l'utente VOTODB può operare sulla base di dati VOTO_ELETTRONICO solo dalla macchina www.foo.it ed è autorizzato soltanto per operazioni di ricerca, inserimento e aggiornamento.

4.2 Autenticazione

L'accesso all'applicazione da parte del presidente del seggio oppure dai singoli votanti, avviene specificando tramite il browser a disposizione dell'utente, la URL relativa al sito Web ospitante l'applicazione e utilizzando il protocollo https. Per es.: `https://www.foo.it`

A questo punto all'utente vengono mostrate le informazioni sul certificato del sito WEB inviate dal server. L'utente può decidere se accettare il certificato oppure rifiutarlo. Accettando il certificato l'applicazione utilizzerà il protocollo sicuro, verificabile dalla presenza sul browser dell'icona raffigurante un lucchetto chiuso. Successivamente all'utente viene presentata la pagina Web tramite la quale è possibile inserire il PIN e la password per l'autenticazione.

4.3 Gestione del seggio elettorale virtuale

L'apertura del seggio elettorale virtuale prevede alcuni passi fondamentali per la predisposizione dei dati necessari al corretto svolgimento della tornata elettorale.

4.3.1 Definizione e gestione degli elettori

L'apertura del seggio è demandata ad una figura che si può identificare come il presidente del seggio, il quale ha il compito di stabilire e, successivamente, inserire l'elenco degli accreditati al voto nella base di dati. Per facilitare la gestione dei dati relativi agli elettori, è stata predisposta una sezione denominata "Gestione elettori", tramite la quale è possibile effettuare le seguenti azioni:

- Inserimento/modifica nuovo elettore

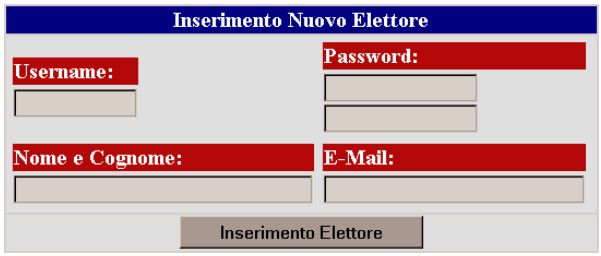

Figura 4.4: Form per la creazione/modifica di un elettore

- Visualizzazione degli elettori definiti nella base di dati

Username	Nome	E-Mail
maurizio	Maurizio Martinelli	maurizio.martinelli@iit.cnr.it
stefano	Stefano Ruberti	stefano.ruberti@iit.cnr.it

Figura 4.5: Esempio di elenco account elettori attivati sulla base di dati

4.3.2 Inserimento dei quesiti elettorali

La procedura di generazione dei quesiti elettorali consiste nell'accesso alla sezione riservata al presidente del seggio elettorale, protetta da un codice utente e una password. Una volta all'interno di questa sezione, il presidente del seggio deve stabilire la data di inizio, la data di fine votazione e il numero di quesiti che intende predisporre.

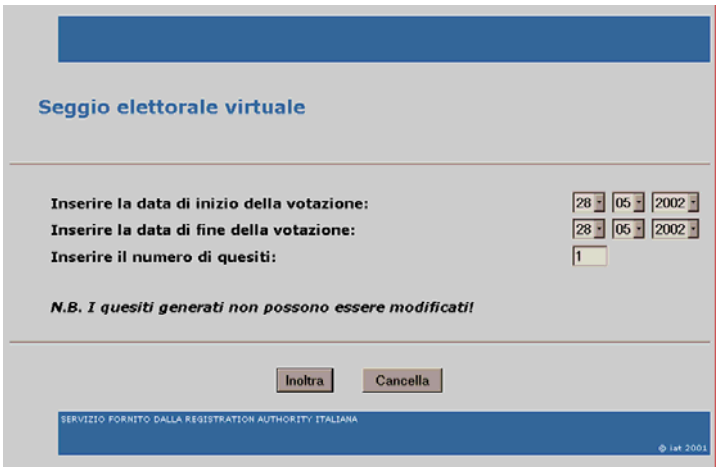

Figura 4.6: Pagina principale relativa alla definizione dei quesiti

Successivamente il sistema propone un modulo elettronico, generato dinamicamente in base al numero di quesiti. Tramite questo modulo, il presidente del seggio ha la possibilità di specificare, per ogni quesito, la relativa descrizione e l'oggetto su cui esprimere il voto. L'oggetto del quesito può avere una lunghezza massima di 255 caratteri (controllata dinamicamente tramite apposito controllo javascript), mentre per la descrizione non è stato inserito alcun vincolo. Una volta compilato il modulo in tutte le sue parti, il presidente del seggio è un grado di registrare nella base di dati i quesiti inerenti la votazione.

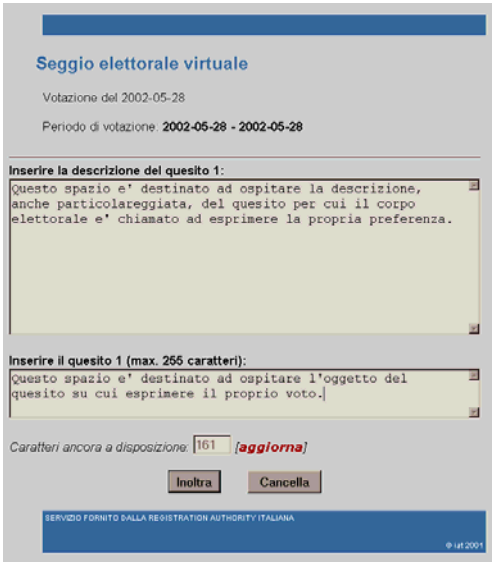

Figura 4.7: Modulo per la predisposizione dei quesiti

4.3.3 Procedura di voto

La procedura di voto è riservata a tutti gli accreditati al voto, cioè tutti coloro che hanno ottenuto un codice di accesso e una password per la votazione. Sarà cura del presidente del seggio invitare gli elettori alle urne, comunicando l'intervallo di tempo per la votazione e la URL che ospiterà il seggio virtuale. Ogni singolo votante, per poter esprimere la propria preferenza, deve effettuare l'autenticazione al sistema referenziando la URL specificata nell'invito al voto e indicando il proprio PIN la e password.

Figura 4.8: Autenticazione del votante

Una volta superata con successo la fase di autenticazione, ad ogni votante viene presentata una form contenente i quesiti su cui esprimere la preferenza. Per ogni quesito, è possibile esprimere il voto a favore, il voto contrario oppure

l’astensione. Non è prevista la possibilità di esprimere voto nullo. Il sistema propone all’elettore tutti e soli i quesiti non ancora scaduti per cui egli non ha già espresso la propria preferenza. Ciò implica che ogni elettore può, per ogni tornata elettorale, esprimere la propria preferenza ad un sottoinsieme di quesiti in tempi diversi, fermo restando il fatto che vengano rispettati i periodi di apertura e chiusura dei seggi.

La figura seguente rappresenta un esempio di scheda elettorale virtuale:

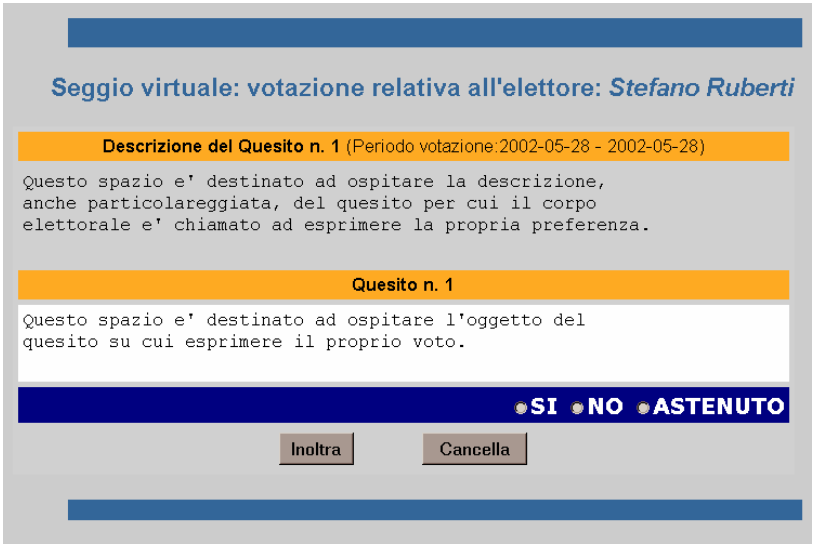

Figura 4.9: Scheda elettorale virtuale

4.3.4 Procedura di scrutinio

La procedura di scrutinio consiste in una interrogazione al database delle votazioni da parte del presidente del seggio. I risultati degli scrutini potranno essere pubblicati sul sito Web in un’apposita sezione di pubblico accesso, oppure accessibile soltanto agli accreditati al voto.

5 SVILUPPI FUTURI

Sono previsti, in futuro, miglioramenti ed estensioni all'implementazione corrente, che facciano uso delle tecnologie attualmente all'avanguardia e che permettano un uso più flessibile e sicuro dello strumento descritto.

I principali dei quali sono i seguenti:

- Realizzazione di procedure che consentano all'utente un'autenticazione "forte" basata su algoritmi a chiavi pubbliche e private;
- Studio e progettazione di un meccanismo che permetta ai votanti di esprimere il proprio voto anche in maniera riservata;
- Implementazione di un meccanismo che consenta di associare univocamente un quesito ad una specifica tornata elettorale;
- Predisposizione di un'interfaccia di ricerca ad hoc per la consultazione dell'archivio elettorale virtuale;
- Realizzazione di un'apposita sezione, accessibile tramite autenticazione a tutti i votanti, contenente lo scrutinio finale della votazione.

6 CONCLUSIONI

La soluzione tecnica illustrata, ha la peculiarità di predisporre un seggio elettorale virtuale sfruttando del software di pubblico dominio. È importante sottolineare che lo sforzo maggiore è stato quello di integrare i vari sotto-sistemi (http server, SQL server, SSL, ecc.) e sviluppare tutte le varie procedure per armonizzarne il funzionamento.

7 APPENDICE A

Esempio richiesta di certificato:

```
shell> openssl req -new > voto_elettronico.csr
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
..............................++++++
.....................................++++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IT
State or Province Name (full name) [Some-State]:Italy
Locality Name (eg, city) []:Pisa
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CNR
Organizational Unit Name (eg, section) []:IIT
Common Name (eg, your name or your server's hostname) []:www.foo.it
Email Address []:webmaster@foo.it

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

8 BIBLIOGRAFIA

[AGO97] A. Albano, G. Ghelli, R. Orsini. *Basi di Dati Relazionali e a Oggetti*. Zanichelli, 1997.

[Alb01] A. Albano. *Costruire Sistemi per Basi di Dati*, Addison-Wesley, 2001.

[RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. *Hypertext Transfer Protocol -- HTTP/1.1*, June 1999.

[LL99] B. Laurie, P. Laurie. *Apache: The Definitive Guide, 2nd Edition*, O'Reilly & Associates, February 1999 (http://www.apache.org).

[VMC02] J. Viega, M. Messier, P. Chandra. *Network Security with OpenSSL*, O'Reilly & Associates, June 2002 (disponibile alla URL: http://www.openssl.org).

[WCO00] L. Wall, T. Christiansen, J. Orwant. *Programming Perl, 3rd Edition*, O'Reilly & Associates, July 2000.

[SSP02] S. Spainhour, E. Siever, N. Patwardhan. *Perl in a Nutshell, 2nd Edition*, O'Reilly & Associates, June 2002.

[AWCLD02] D. Axmark, M. M. Widenius, J. Cole, A. Lentz, P. DuBois. *MySQL Reference Manual for version 4.0.2*, MySQL AB 2002 (http://www.mysql.com).