UNIVERSITÀ DEGLI STUDI DI ROMA
"TOR VERGATA"

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

Progetto per il corso di Sicurezza dei Sistemi Informatici

LICARUS LICENSE SERVER

Professore:
Prof. GIANLUIGI ME

Studente:
STEFANO PERNA

Professore:
Prof. GIUSEPPE ITALIANNO

ANNO ACCADEMICO 2007-2008

Prefazione

La seguente relazione è stata prodotta come parte integrante del progetto svolto nell'ambito del corso di Sicurezza dei Sistemi Informatici.

Il progetto di riferimento è il numero 17: *License Server*.

La traccia integrale del progetto è riportata di seguito:

Si realizzi una applicazione client/server in cui il client ad intervalli di tempo stabiliti/ad ogni esecuzione verifichi la propria idoneità all´esecuzione. Lo schema di funzionamento prevede che a valle della mutua autenticazione tra client e server la verifica della licenza del client salvata sul server. Nel caso in cui la licenza risultasse scaduta deve essere impedito al client di procedere con l´esecuzione.

In questo documento verranno riportati gli elementi d'analisi del progetto e l'implementazione dello stesso.

Indice

Prefazione **iii**

Indice **iv**

1 Introduzione **1**

2 Le specifiche del progetto **4**
2.1 Ambito del progetto 4
2.2 Caratteristiche del software 4
2.3 Schema di funzionamento 6
2.4 Vantaggi e svantaggi 7

3 Il sistema nel dettaglio **9**
3.1 Tecnologie e Scelte implementative 9
3.1.1 Il sistema e l'ambiente d'esecuzione 9
3.1.2 L'infrastruttura PKI 10
3.1.3 Il sistema di cifratura della comunicazione 10
3.1.4 Le applicazioni *client* e *server* 10
3.2 Analisi e disegno del sistema 11
3.2.1 Il *client* 11
3.2.2 Il *server* 17

4 Sperimentazione del sistema **24**
4.1 L'avvio del server e le sue funzioni principali 24
4.1.1 Gestione delle licenze mediante LicarusLicenseManager . 26
4.2 L'avvio del client e le sue funzioni principali 27
4.2.1 Impostazione dei parametri di connessione 29

4.2.2 Riciesta di una nuova licenza 30
4.3 Verifica della licenza 30

5 Conclusioni **34**

A L'algoritmo di RSA **36**

B Codice Sorgente **38**
B.1 License.cs 38
B.2 LicenseController.cs 39
B.3 LicenseUtil.cs 41
B.4 LicenseVerifierForm.cs 42
B.5 MainForm.cs 43

Elenco delle figure **46**

Bibliografia **48**

Capitolo 1

Introduzione

La pirateria del Software è oggi uno dei problemi più difficili da affrontare nel mondo dell'informatica.

Definiamo la pirateria software come *la copia o la diffusione non autorizzata di software coperto da copyright* .

Tale definizione può essere meglio compresa individuando le metodologie di pirateria oggi diffuse. In particolare è definibile pirateria del software copiare, scaricare, condividere, vendere o installare senza alcuna autorizzazione del software coperto da diritti d'autore.

Possiamo quindi facilmente immaginare l'entità della pirateria del software, che oggi, insieme alla diffusione illegale della musica e dei film, rappresenta una delle principali fonti di perdita economica per le aziende produttrici del software stesso.

Nel documento del *Fourth Annual BSA and IDC Global Software Piracy Study* vengono riportati i dati ufficiali della diffusione della pirateria del software nelle varie regioni del mondo e dei numeri ad essi associati.

Si stima che nel 2006 il 35% del software installato sui computer sia stato ottenuto illegalmente, con un incremento del 15% rispetto all'anno precedente ed una perdita economica stimata intorno ai 40 bilioni di $.

É interessante inoltre notare che l'Unione Europea si attesta al primo posto nelle regioni dove si registra il maggior danno economico. In Italia sono state stimate perdite per oltre 1400 milioni di $.

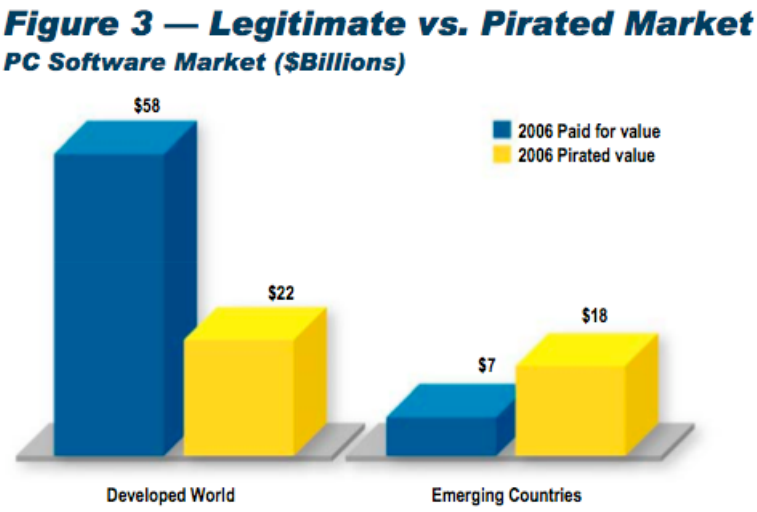

Figura 1.1: Alcuni dettagli del documento del BSA.

Table 2 — Ranking by 2006 PC Software Piracy Losses
Countries with $200 Million or More

COUNTRY	$M
United States	$7,289
China	$5,429
France	$2,676
Russia	$2,197
Japan	$1,781
United Kingdom	$1,670
Germany	$1,642
Italy	$1,403
India	$1,275
Brazil	$1,148
Spain	$865
Canada	$784
Mexico	$748
Australia	$515

COUNTRY	$M
Poland	$484
South Korea	$440
Thailand	$421
Netherlands	$419
Indonesia	$350
Ukraine	$337
Switzerland	$324
Turkey	$314
Sweden	$313
Venezuela	$307
Argentina	$303
Malaysia	$289
South Africa	$225
Belgium	$222

Figura 1.2: Alcuni dettagli del documento del BSA.

Uno dei problemi più importanti leagati alla pirateria resta quindi quello di trovare procedure o tecniche di protezione del software in grado di arginare il fenomeno, riducendone l'impatto economico.

Il progetto realizzato ha l'obiettivo di creare un sistema di gestione delle licenze software in grado di riconoscere se il software in esecuzione su un determinato computer possa essere eseguito o meno.

L'architettura presentata è quella di un sistema client/server, in cui il client (ovvero il software in esecuzione sul computer dell'utente) ad ogni esecuzioni comunichi con un server (della software house) per verificare la sua idoneità all'esecuzione.

Il sistema è stato progettato utilizzando un'infrastruttura a chiave pubblica e privata.

A tale proposito sono state realizzate un'applicazione Server, chiamata Licarus License Manager e delle librerie destinate all'applicazione client. Si potranno realizzare così diverse applicazioni client semplicemente integrando le librerie client di Licarus.

Capitolo 2

Le specifiche del progetto

2.1 Ambito del progetto

Il progetto si occuperà di realizzare un software client/server in grado di gestire le licenze di ogni client attraverso la mutua autenticazione di entrambi. Il *client*, in particolare, dovrà comunicare ad ogni esecuzione la propria licenza al *server*, il quale ne verificherà l'autenticità e la validità e autorizzerà il *client* all'esecuzione.

La buona riuscita del progetto risulterà nel creare un sistema in grado di resistere a violazioni o modificazioni delle licenze, ed in grado di resistere ad altre manipolazioni del sistema client da parte dell'utente.

2.2 Caratteristiche del software

L'architettura del sistema sarà basata su architetture di rete client/server per consentire la validazione delle licenze in uso.

Il sistema dovrà consentire l'esecuzione dell'applicazione client solo previa autenticazione e validazione della licenza ad esso asseganta. La comunicazione con il server dovrà avvenire mediante connesione remota. Il server dovrà essere quindi in grado di verificare le licenze inviate dai client, per evitare che licenze scadute o non autorizzate possano essere usate per l'esecuzione. Il server dovrà inoltre consentire l'autenticazione e la validazione simultanea di più client contemporaneamente.

I principali requisiti software possono essere riassunti come segue:

1. Adottare un'architettura Client/Server.
2. Consentire l'esecuzione del Client solo dopo la verifica della propria licenza.
3. Permettere la validazione della licenza mediante connessione remota.
4. Consentire la connesione simultanea di più client con il server.
5. Resistere ad evetuali manipolazioni dei file di licenza.
6. Resistere ad eventuali manipolazioni locali al client da parte dell'utente.

Dall'analisi delle generiche caratteristiche sopra citate possiamo iniziare a delineare alcune delle peculiarità specifiche che il sistema dovrà possedere.

La comunicazione tra client e server dovrà essere effettuata mediante connessione remota, quindi estremamente suscettibile ad attacchi di tipo Man in the middle. La comunicazione tra le parti dovrà quindi essere sicura ed il sistema di autenticazione e validazione delle licenze dovrà resistere ad attacchi di questo tipo. La comunicazione avverrà in modo cifrato.

Per comunicare la validità della licenza il sistema sarà realizzato mediante l'uso di una infrastruttura a chiave pubblica (PKI). Il server custodirà la chiave privata, mentre il client la chiave pubblica. Ad ogni richiesta di validazione della licenza il server firmerà il file di licenza e comunicherà la firma al client, il quale potrà verificarne la correttezza mediante la propria chiave pubblica.

Il sistema di cifratura della comunicazione dovrà consentire lo scambio di mesaggi in maniera sicura adottando un algoritmo di crittografia sicuro e ben testato.

Per prevenire l'uso di una stessa licenza su client diversi le licenze software dovranno essere inoltre generate in base ad un meccanismo di associazione tra licenza e computer in uso, identificando e legando la licenza stessa all'Hardware Fingerprint[1]del client.

[2]

[2]L'Hardware Fingerprint è la firma del PC, o meglio, il codice identificativo specifico di ogni macchina che consente al software di lavorare su un determinato PC ma non su un altro che non disponga di una licenza propria, regolarmente concessa.

Le librerie client dovranno essere in grado di verificare inoltre che la licenza in uso sia realmente associata all'hardware in uso, e che non sia stata manipolata.

2.3 Schema di funzionamento

Lo schema di funzionamento base del sistema è riportato nella figura 2.1.

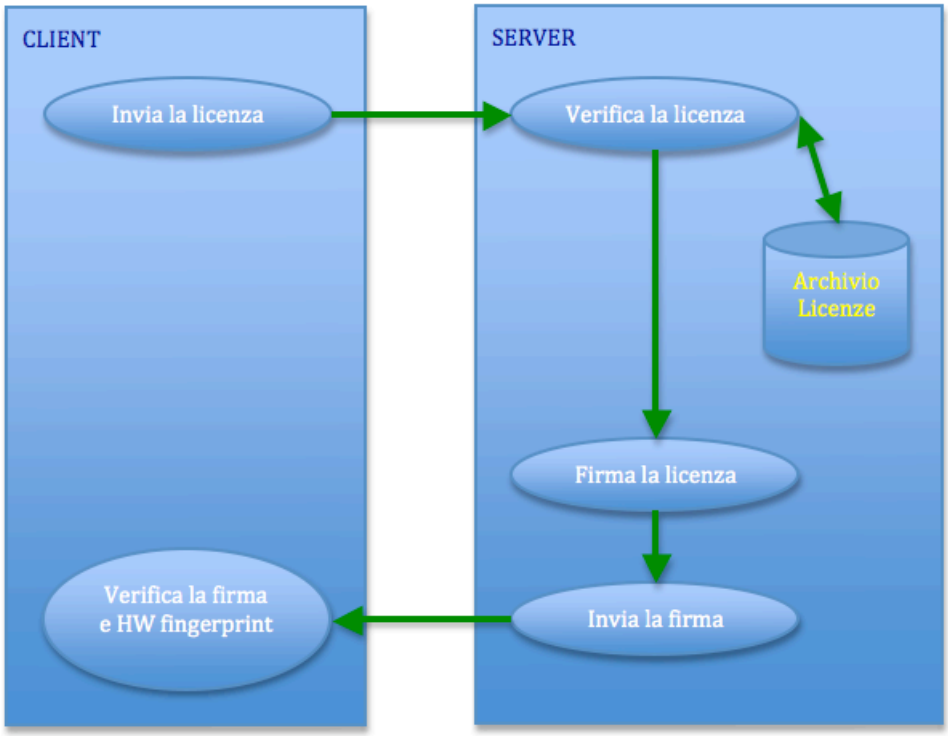

Figura 2.1: Schema di funzionamento.

Il client effettua la connessione remota al server inviandogli la propria licenza cifrata.

Il server, ottenuta e decifrata la licenza, verificherà, attraverso il proprio archivio, che questa sia valida.

Se tali verifiche daranno buon esito, il server firmerà con la propria chiave privata la licenza e invierà tale firma al client in modo cifrato.

Il client, ricevendo la firma, potrà verificare che la licenza in suo possesso sia valida e che l'Hardware Fingerprint associato alla licenza sia lo stesso di quello del computer dove il client è in esecuzione. Se tali verifiche risulteranno corrette allora l'applicazione client avrà l'autorizzazione per essere eseguita.

Lo schema di funzionamento è quindi molto semplice: prevede un'infrastruttura PKI per verificare le licenze mediante firma digitale delle licenze stesse e l'implementazione di un algoritmo di crittografia per consentire una comunicazione cifrata.

2.4 Vantaggi e svantaggi

I vantaggi di questo sistema di licensing sono molteplici.

Le licenze in uso potranno essere costantemente monitorate e si potrà controllare che non siano state duplicate o manipolate, rendendo l'applicazione abbastanza sicura ad attacchi deboli. Questo è garantito anche dalla presenza di un'infratruttura PKI. La comunicazione cifrata garantirà inoltre una doppia protezione al sistema di validazione basato su firma della licenza.

Un possibile attacco al sistema di validazione potrebbe consistere nel tentare di scoprire la chiave privata del License Manager. Scoprire tale chiave è comunque un'impresa ardua e molto onerosa, quindi in grado di proteggere il sistema dalla maggior parte dei maleintenzionati. Possiamo inoltre considerarla computazionalmente sicura se associata ad una generazione di nuove coppie di chiavi nell'arco del tempo.

L'applicazione risulta inoltre protetta da una licenza associata univocamente all'hardware del computer in uso. Anche ciò garantisce una buona protezione contro attacchi di basso livello. É anche vero però che tale protezione potrebbe essere ingannata cambiando gli ID del proprio hardware con quelli presenti in una licenza valida. Questo tipo di attacco è però molto macchinoso per la maggior parte degli utenti che non si cimenterebbe in un'avventura del genere, rendendo questo tipo di protezione forte contro la maggior parte degli attacchi.

La modifica del file di licenza è protetta dalla cifratua della licenza stessa.

Un altro possibile attacco potrebbe consistere nel clonare il server e cambiare le chiavi pubblica e privata del sistema PKI. In questo caso, reinderizzando

l'applicazione client su un server clonato e cambiando la coppia di chiavi pubblica e privata, il maleintenzionato potrebbe firmare il file di licenza indipendentemente dall'applicazione server originale. Comunicando in modo cifrato tale attacco verrebbe vanificato a meno di non conoscere il segreto necessario per la cifratura e decifratura della comunicazione.

L'unico attacco efficace risulterebbe quindi nella decompilazione dell'applicazione e nella ricerca dei punti deboli dove poter modificare il codice per far risultare l'autenticazione sempre valida. Questo è uno degli attacchi più diffusi ad applicazioni software ed uno dei più difficili da frenare, in quanto le procedure attualmente adottate non forniscono misure preventive che funzionino efficaciemente o impediscano l'attacco. Difficilmente sono inoltre definibili computazionalmente sicure tali procedure di protezione, poichè in genere la buona riuscita di questo tipo di attacco sarà determinata solo dalla preparazione tecnica dell'attaccante. Il più semplice sistema di protezione che si potrebbe adottatare in questo caso è l'offuscamento e lo scrambling del codice sorgente, sebbene il sistema risulterebe in ogni caso violabile[3].

[3]Un esempio di come tali tecniche possano essere agirate è stato l'insuccesso del sistema CSS adottato per la protezione della copia dei DVD.