LEONARDO

PATENT BROCHURE

Multi-Level Token Polymorphic SW License (MuToPS-L)

Area tecnologica principale → Elaborazione Automatica Dati

Keyword → token | criptografia | memoria | generazione casuale | polimorphic engine

Sistema e metodo di protezione di dati informativi, in particolare di verifica e concessione dell'autorizzazione all'accesso a tali dati informativi.

Figura 1 - Esempio di token utilizzabile

CARATTERISTICHE TECNICHE

Sistema di protezione di "applicativi software" versatile e modulare con protezione a più livelli che utilizza un dispositivi HW e un SW.

Come dispositivo HW può essere usato un qualsiasi "Token" PKCS11 compliant (in figura un esempio) in grado di memorizza dati su un "chip" interno con due aree: una accessibile a chiunque e una accessibile tramite "Login" (non "estraibile/leggibile" da un potenziale "hacker"). Alcuni tipi di "Token" si autodistruggono quando c'è un tentativo di manomissione.

Il SW include due componenti:

- un "Initializer" per inizializzare la licenza SW al primo accesso, che rimane in carico alla produttore del SW da proteggere
- un "Enforcer" da includere all'interno del SW (sotto forma di DLL) per proteggere e gestire le limitazioni d'uso.

La protezione attraverso SW avviene attraverso i seguenti passaggi:

1. Combinazione di "dati invariabili nel tempo" (e.g. identificativo del SW, nome dell'azienda, …) con "dati dipendenti dal tempo" (l'ora, la licenza, …) per ottenere un "Payload" costituito dai vari frammenti che lo compongono ("useful data")
2. Scelta di un "Messaggio (P)" abbastanza lungo da consentire un elevato numero di possibili modi di scrittura del "Payload" all'interno di "P" ("mutazioni")

3. Selezione, tramite una "polymorphic engine", di una delle possibili mutazioni descritta da uno "schema" **S** che permette di scrivere i vari frammenti del Payload all'interno messaggio **P** riempiendo ("padding") gli "spazi vuoti (gaps)" degli "useful data" con dei "token-generated random data". L'engine è progettato in modo da evolvere in modo imprevedibile, ma garantire che la scelta di **S** e quindi del messaggio **P** risultante non si ripeta mai nelle sue successive evoluzioni e tra token diversi, di modo che essi risultino sempre unici in ogni istante
4. Criptazione del messaggio **P** attraverso una "token-generated random key" e un "crypto-mechanism" (e.g. AES, RSA, DES, ecc.); il risultato è un "cypher-text" **C**. Operazione analoga, utilizzando una chiave diversa, viene fatta sullo schema **S** generandone una versione criptata. **C** ed **S** vengono memorizzati sull'area privata (accessibile con "login") del Token
5. L'insieme delle "Keys" e dei "cypher-texts" memorizzati, formano il primo "sigillo" (o "**Seal**")
6. Un codice a elevata complessità ("PIN") è generato automaticamente, fornito alla persona che ha sviluppato il SW e archiviato per consentire l'accesso all'area private del Token in caso di necessità di riprogrammazione dei token. Tale PIN è generato con una lunghezza molto grande (>128 caratteri) di modo da prevenire eventuali tentativi di brute forcing; nell'eventualità dopo 3 tentativi errati il token viene infatti bloccato. Gli utilizzatori del SW non saranno a conoscenza di tale PIN e non avranno quindi modo di accedere a tale area.
   A questo punto, il SW protetto (all'interno del quale viene integrato l'**Enforcer**) può essere liberamente installato sull' HW del Cliente. Durante l'esecuzione, l'**Enforcer** entra in funzione
7. Quando lo **User** inserisce la sua password, verifica che l'HW del token fornito sia del tipo adeguato (modello, versione, ecc.)
8. Se il Token è del tipo atteso, l'**Enforcer** inizia la sessione di accesso all'area privata usando il PIN
9. A questo punto, controlla che siano presenti **C** ed **S**
10. Dopodiché decripta **C** e, in maniera automatica e trasparente per lo User, ne controlla la "consistenza" fra le parti del "payload" (posizione e valore di tutti i dati). Se si trova un solo dato alterato (valore e/o posizione diversi o, per i dati variabili qualsiasi anomalia rispetto alla legge di controllo impostata), il controllo dà esito negativo e il SW stesso genera un messaggio di "warning" seguito dalla chiusura del SW
11. L'**Enforcer** scarta il sigillo usato (i.e. padding scheme, keys, random data, payload context-variable data) ed ottiene dati nuovi attraverso il "polymorphic engine" e ricostruisce un nuovo **Seal** (attraverso i passaggi 1-5)
12. Dopo un certo tempo (variabile) oppure dopo un evento specifico che lo sviluppatore ritiene chiave per il funzionamento del proprio SW (e.g. compilazione, salvataggio, stampa, avvio di un modulo specifico, convalida, ecc.), il **Seal** è nuovamente decriptato e la sua evoluzione controllata continuamente in termini di consistenza dei valori fissi e variabili e della loro posizione
13. I passaggi 9-11 sono ripetuti in background finché il SW da proteggere è attivo.

In questo modo si stabiliscono i seguenti livelli di protezione:

a) Protezione fisica in quanto è necessario l'accesso al Token. Se questo non viene inserito o viene rimosso, il SW rispettivamente non si avvia o si blocca.
b) Il Seal di dati criptati nell'area privata non è accessibile allo User.
c) Il meccanismo criptografico è implementato via HW dal Token e non manomissibile

d) Se anche un hacker riuscisse ad accedere all'area privata, dovrebbe decodificare sia **S** che **P** ed essere a conoscenza del meccanismo di correlazione tra i due e del "*polymorphic engine*" per ingannare i controlli successivi
e) La decodifica e lettura di un singolo **Seal** non consente di soddisfare il criterio per operare la licenza d'uso.

INNOVAZIONE/VANTAGGI

- **Profitti dai propri prodotti SW:**
  Questa soluzione consente di realizzare profitti attraverso la vendita dei prodotti SW in tutta sicurezza.
- **Controllo personalizzato delle licenze:**
  Le licenze possono essere personalizzate sulla base delle proprie esigenze (e.g. Customer Basis).
  Inoltre, per come è pensato, si può gestire un SW modulare imponendo limitazioni d'accesso personalizzate ai singoli Utenti tramite la licenza immagazzinata sul token assegnato ad ogni utente.
- **Estensibilità ad altri ambiti (non solo SW):**
  Il sistema può essere applicato anche nell'ambito delle comunicazioni sicure / scambio dati.

CAMPI DI APPLICAZIONE

Informatica, protezione delle informazioni e dei software applicativi

INFORMAZIONI BREVETTUALI

Data di priorità – 28/05/2012

Codice di priorità – IT TO2012A000462

Codici IPC - H04L9/08 - H04L9/32 - G06F21/10 - G06F21/62 - G09C1/00

Depositi nazionali attivi

EPO – EP2670080B1; data deposito 28/05/2013; data concessione 31/01/2018
Estensione in: Italia - Germania – Francia - Regno Unito – Spagna

USA - US9246684; data deposito 24/05/2013; data concessione 26/01/2016

Giappone - JP06184751; data deposito 28/05/2013; data concessione 4/08/2017

Cina - CN103559454; data deposito 28/05/2013; data concessione 17/04/2018

Leonardo internal code

LDO-A490