PAdES Digital Signature v.1.0b

Linguaggio per la Firma Digitale e verifica di documenti PDF

Attività Progettuale

di

Linguaggi e Modelli Computazionali LM

Prof. Ing. Enrico Denti

Facoltà di Ingegneria Informatica,
Università degli Studi di Bologna

a.a. 2012/2013

Ongaro Francesco

Indice argomenti

- Obiettivi e strumenti
- Caratteristiche
- Il formato PAdES e la Signature Dictionary
- Architettura
- Grammatica
- Linguaggio, esempi
- Semantica
- Interfaccia grafica
- Conclusioni e sviluppi futuri
- Riferimenti

Obiettivi

- Realizzazione di un **linguaggio** semplificato per la definizione degli elementi necessari e opzionali per poter firmare digitalmente un documento pdf, e per verificare eventuali firme presenti.

- Sviluppo dell'**interprete** per il linguaggio creato, effettuando l'analisi sintattica e semantica.

- Progettazione di un'**applicazione grafica** usabile e intuitiva, che permetta agevolmente di:
  - **Apporre una firma digitale**, eventualmente personalizzata, ad un documento PDF, in accordo con il profilo di busta crittografica PAdES.
  - **Verificare** la/le **firma**/e presente/i in un documento.
  - Testare attraverso il **Parser** e il **Visitor** alcuni esempi di frasi del linguaggio, ottenendo un feedback relativamente ad errori sintattici o semantici.
  - Visualizzare graficamente l'**albero** rappresentativo delle frasi del linguaggio.
  - **Convertire files** immagine o documenti testuali in formato PDF.

Strumenti

- Tale Attività Progettuale si è in parte basata sulle:
  - "**Librerie per la Firma Digitale di documenti PDF**", realizzate durante l'Attività Progettuale di Sicurezza dell'Informazione LM, Prof. Rebecca Montanari, tutor Ing. Anna Riccioni. *[PAdES]*

- Inoltre sono state utilizzate le librerie:
  - **iTex** (*www.itextpdf.com*) per la manipolazione dei documenti PDF
  - **Bouncy Castle**(*www.bouncycastle.org*), librerie crittografiche
  - **Aspose.Words** (*www.aspose.com*), librerie commerciali per la conversione di documenti (usate solamente a titolo di prova).

- Ambiente di sviluppo: **Eclipse** SDK Juno v.4.2.2
- Linguaggio: **Java**
- Generazione del Parser: **JavaCC** v.1.5.0
- Generazione dell'APT: **JTB** v.1.4.0.2
- Interfaccia grafica: **WindowBuilder** v.1.5.1r42

Caratteristiche

- Il **linguaggio** e dunque **l'applicazione**, deve prevedere la possibilità di:
  - Usare un **keystore** (formato pkcs12) al cui interno vi sarà memorizzato il certificato digitale con le chiavi asimmetriche dell'utente.
  - Verificare lo **stato di revoca** del certificato prima di firmare.
  - **Convertire** il file da firmare qualora non fosse in formato pdf:
    - doc, docx, odp, rtf, txt → pdf (tramite le librerie commerciali Aspose.Words)
    - jpg, gif, png → pdf (tramite le librerie iText, anche se: *"Converting documents from one format to another is outside the scope of iText. And no: iText does not convert Word documents to PDF! … iText isn't a PDF viewer, iText can't convert PDF to an image, nor can iText be used to print a PDF … " - www.itextpdf.com/itext.php*)
  - Inserire la **risposta OCSP** fornita dall'OCSP Responder all'interno del PDF.
  - Inserire l'**intera CRL**, fornita dal CRL Distribution Pointer, all'interno del PDF.
  - **Marcare temporalmente** il documento in modo da fornire una data certa.
  - Personalizzare la **Signature Appearance**, ovvero l'aspetto grafico della firma digitale, specificandone la visibilità, il tipo, la dimensione, la posizione e l'eventuale immagine associata.
  - Specificare il **Certification Level**.
  - Specificare le **Signer Informations** relative al firmatario.
  - **Verificare** un documento PDF già firmato, estraendo i dettagli relativi allo **stato di revoca** del certificato, la **copertura** della firma e **l'integrità** di quest'ultima.