Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea in Informatica

LO STANDARD XML-SIGNATURE IN AMBIENTE SANITARIO

Materia di Tesi: Basi di Dati e Sistemi Informativi

Relatore:
DANILO MONTESI

Presentata da:
ANDREA FELICE

Correlatore:
MANUELE TORRISI

I Sessione
Anno Accademico 2009-2010

Indice

Elenco delle figure

Capitolo 1

Introduzione

La firma digitale è divenuto ormai uno strumento molto diffuso in diversi campi soprattutto per la sua grande utilità nel caso in cui occorra sottoscrivere un documento mantenendo una garanzia di integrità dei dati del documento e di autenticità delle informazioni di chi lo ha sottoscritto. Inoltre un'interessante particolarità è che un documento sottoscritto con la firma digitale non può essere ripudiato in quanto ha assunto da anni validità legale come vedremo nel prossimo capitolo: la firma digitale di un documento informatico equivale alla tradizionale firma autografa apposta sui documenti cartacei.
In questo elaborato, iniziando ad introdurre sommariamente le proprietà principali della Firma Digitale, il quadro normativo e alcuni termini generali sull'argomento, si sposterà l'attenzione, a partire dal terzo capitolo, su uno dei tre formati di rappresentazione che ha assunto notevole rilevanza nella gestione elettronica dei flussi documentali in particolare in ambito bancario e sanitario: l'XML(eXtensible Markup Language).
Nel terzo capitolo si descriverà il formato XML-Signature, divenuto già da molti anni standard W3C[1]. In seguito si fornirà una panoramica di alcuni software per la firma e per la verifica di documenti elettronici.

[1] W3C è il World Wide Web Consortium cioè l'ente mondiale che dal 1994 definisce gli standard per il web.

Nel quinto capitolo, dopo essere passati a descrivere il procedimento di firma e di verifica di un file xml, si affronterà il caso di come questo formato può essere utilizzato in un ambiente sanitario dove ad esempio risulterebbe molto utile nella gestione di una cartella clinica di un paziente. Lo scopo principale dell'introduzione del digitale é il tentativo di eliminare o comunque ridurre al minimo il cartaceo. Si capisce che con questo nuovo modo di concepire il documento scritto, i vantaggi sarebbero notevoli sia per il cittadino ma in particolare per la pubblica amministrazione:

- prima di tutto una importante conseguenza che accomuna l'introduzione delle nuove tecnologie è il fatto di gestire in maniera più omogenea il ciclo di vita del documento dalla creazione passando per la protocollazione e per la trasmissione di esso fino alla conservazione.

- dal punto di vista strettamente pratico si ha un risparmio di spazio ma anche di costi relativi alla gestione del cartaceo in un luogo come può essere un ospedale.

- questa innovazione inoltre porta anche ad una maggiore efficienza,si pensi solo alla velocità di trasmissione dei documenti che potranno essere trasferiti in pochi minuti contro i tempi espressi in giorni del materiale cartaceo.

Dal punto di vista dei costi, il risparmio dell'eliminazione della carta si contrappone ai costi per la riorganizzazione e per la formazione del personale, che però possono essere facilmente affrontati visto i numerosi benefici che il digitale porterebbe.
Nel terzo capitolo si affronterà anche il problema delle firme multiple, tema molto importante in ambiente sanitario se si parla ad esempio di una cartella clinica di un paziente e quindi della possibilità di apporre più firme allo stesso documento ovviamente in tempi diversi e per ogni operazione eseguita sul paziente. Infatti una delle proprietà più interessanti e importanti della firma su XML riguarda la possibilità di firmare non solo tutto il documento (come accade con gli altri due formati), ma anche solo un sottoinsieme di

esso. Questo apre le porte anche alla realizzazione di applicazioni che permettano di far firmare parti differenti del documento da persone diverse (e quindi nel nostro problema da dottori differenti) in funzione delle specifiche responsabilità. Ad esempio si può quindi presentare il caso in cui in un referto medico, lo specialista potrà firmare solo le informazioni che competono alla sua specializzazione.

Capitolo 2

Firma Digitale

Abbiamo detto che la firma digitale è stata ormai equiparata alla firma autografa nella legislazione di molti paesi, tra cui l'Italia. Quindi la sua apposizione su un documento garantisce l'autenticazione, la non ripudiabilità e la piena validità legale. La firma digitale viene definita come un particolare tipo di firma elettronica qualificata basata su un sistema di chiavi asimmetriche a coppia. Più avanti verrà data una spiegazione più chiara mentre nell'Appendice A si possono trovare alcune definizioni di termini che useremo in questo documento.
In questo capitolo descriveremo in maniera più tecnica il concetto di firma digitale, richiamando e discutendo anche alcune definizioni importanti per l'argomento riportate sul sito del CNIPA [1] sotto forma di guida per l'utente [2]. Naturalmente quando faremo esempi o citazioni, riguarderanno l'ambito dove la firma per ora si è diffusa maggiormente cioè nella pubblica amministrazione.
Parleremo anche dei diversi tipi di formato che vengono usati per produrre file firmati digitalmente e quali sono gli strumenti che si devono utilizzare per avere la possibilità di firmare i documenti stessi.
Infine guarderemo anche l'aspetto giuridico e faremo una breve panoramica delle leggi che regolamentano e stabiliscono i modi di utilizzo ed il valore della firma digitale.

2.1 La firma nel dettaglio

Come viene descritto anche dalla guida definita dal CNIPA [2], affinchè la firma possa essere dichiarata valida ad equivalente a quella autografa, essa deve essere basata su un sistema a chiavi asimmetriche.

2.1.1 Crittografia asimmetrica

La crittografia asimmetrica è una sorta di miglioramento della tradizionale crittografia simmetrica per la quale viene utilizzata un'unica chiave sia per codificare che per decodificare i messaggi trasmessi. Infatti il problema della crittografia simmetrica è che la chiave deve essere conosciuta sia dal mittente che dal destinatario del messaggio, con la conseguenza di un possibile problema di sicurezza che può presentarsi nel momento in cui le due persone dovessero comunicarsi la chiave.
Nella crittografia asimmetrica, detta anche crittografia a chiave pubblica, esiste sempre una coppia di chiavi per ciascun utente, tra loro inseparabili: una privata e una pubblica. Le chiavi pubbliche vengono rese disponibili a tutti, mentre la chiave privata appartiene a un singolo individuo. Le chiavi pubbliche vengono trasferite agli utenti utilizzando come meccanismo di distribuzione un certificato. I certificati vengono firmati da un'autorità di certificazione (Certification Authority), che conferma la stretta e affidabile correlazione tra la chiave pubblica e i dati che identificano il titolare. Un destinatario che può accedere alla chiave pubblica della CA è in grado di stabilire se il certificato è stato firmato da un'autorità specifica. L'utilizzo della CA è attualmente il modo più utilizzato per pubblicare la propria chiave pubblica ed è anche considerato il più sicuro. Le chiavi pubbliche assumono solitamente la forma di certificati digitali; in questo modo l'utente che riceve il certificato del mittente, controlla l'autenticità attraverso la chiave pubblica in esso contenuta. Questo permette alle parti di comunicare in modo confidenziale senza un precendente scambio di informazioni segrete. Le Certification Authority sono caratteristiche di una infrastruttura a chiave

pubblica (Public Key Infrastructure),un sistema di certificazioni composto da entità fidate, che permettono l'uso della crittografia a chiave pubblica anche in ambienti con un alto numero di utenti e al destinatario di un documento elettronico qualificato la possibilità di recuperare la chiave pubblica del mittente. I certificati che vengono rilasciati dalla CA vengono chiamati public-key certificates: ad oggi il certificato standard è l' X.509 che al suo interno contiene non solo il nome e la chiave pubblica di un utente, ma molte altre informazioni:

- Versione;
- Numero seriale;
- Identificativo dell'algoritmo;
- Ente emettitore;
- Validità;
- Proprietario del certificato;
- Informazioni sulla chiave pubblica(algoritmo usato);
- Codice identificativo univoco dell'emittente (facoltativo);
- Codice identificativo univoco del proprietario (facoltativo);
- Algoritmo di firma del certificato;
- Firma del certificato.

Da notare che i certificati X.509 come molti altri tipi di certificati hanno un periodo di validità (che oscilla attorno ai 2-3 anni) al di fuori del quale vengono considerati non più validi. Possono inoltre essere revocati dalla CA per una serie di motivi come per esempio se all'utente è stata rubata la propria chiave privata e quindi ha richiesto un nuovo certificato con la conseguenza di vedersi revocare il vecchio. Allo scopo di gestire le revoche, le CA conservano e distribuiscono periodicamente un elenco di revoche di certificati

(Certificate Revocation List), al quale gli utenti di rete possono accedere per verificare la validità di un certificato. Sul sito del CNIPA [3] si può trovare anche una lista dei certificatori riconosciuti dalla legge italiana.
Tornando nello specifico alla crittografia, ciò che viene cifrato con la chiave pubblica può essere decifrato solo con la chiave privata corrispondente (operazione che può essere fatta solo dal proprietario della chiave): in questo modo non c'è più il problema di comunicare segretamente la chiave, in quanto risulta nota a tutti; per comunicare in modo sicuro con una persona basta cifrare il messaggio con la sua chiave pubblica. Nel processo di firma digitale la chiave privata viene utilizzata sia per creare una firma ad un documento che per decifrare messaggi come appena descritto, mentre la chiave pubblica serve sia per cifrare i messaggi che nel processo di verifica di un file firmato.

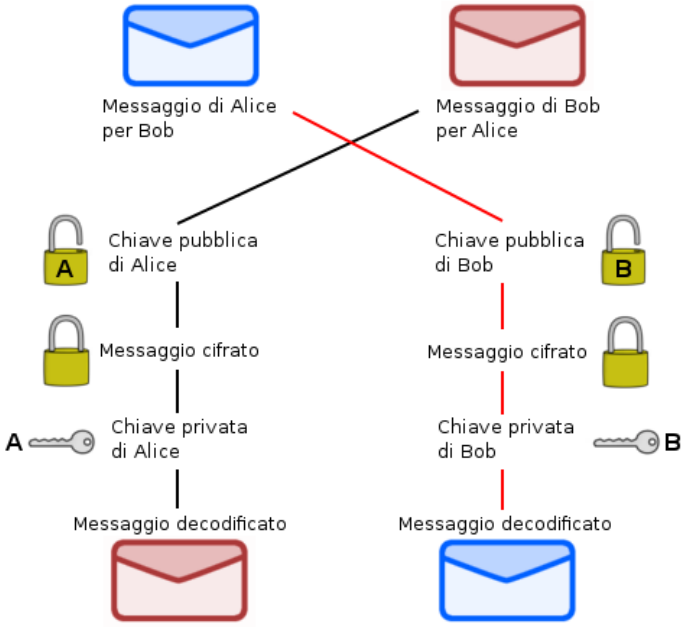

Figura 2.1: Esempio di uso della crittografia asimmetrica [4]

Ciò che avviene in pratica è che, posto che il mittente A e il destinatario B abbiano entrambi la loro coppia di chiavi, A firma il messaggio con la propria chiave privata e dopo essersi procurato la chiave pubblica di B (che

come abbiamo già detto viene distribuita a tutti coloro che devono scambiarsi messaggi con il proprietario di essa ad esempio attraverso l'autorità di certificazione descritta sopra) cifra lo stesso messaggio con questa chiave. In questo modo B ricevendo il messaggio, sarà il solo a poter decifrare il messaggio con la propria chiave privata garantendo quindi confidenzialità, e poi potrà verificare l'autenticità con la chiave pubblica di A. Va detto che non avviene esattamente così in quanto il mittente codifica e firma solamente il digest (impronta) del messaggio che puó venire inviato in allegato al messaggio vero e proprio,ma di questo parleremo più avanti.
Uno degli algoritmi più usati nella crittografia asimmetrica per generare la coppia chiave pubblica/privata è l' RSA (dal nome di tre ricercatori del MIT Rivest, Shamir e Adleman) creato intorno al 1978. Come tutti gli algoritmi asimmetrici, esso viene considerato sicuro in quanto le due chiavi che vengono create devono essere indipendenti l'una con l'altra quindi risulta quasi impossibile risalire ad una chiave privata avendo la chiave pubblica a disposizione. L'argoritmo è stato implementato utilizzando particolari proprietà dei numeri primi con qualche centinaia di cifre. In questo modo, può esistere la possibilità di decriptare un messaggio essendo a conoscenza della chiave pubblica, tuttavia il costo computazionale dell'operazione è così alto da fare di questo algoritmo un sistema decisamente affidabile.
Rimane inoltre da trattare come è possibile collocare l'apposizione della firma digitale nel tempo e ne parleremo nel prossimo paragrafo.

2.1.2 Marcatura temporale

Di per se la sola firma digitale non contiene alcuna informazione circa l'esistenza del documento in un certo periodo del tempo. Premesso ciò, risulta evidente che potrebbero nascere dei problemi legati alla necessità di utilizzare un documento in un momento temporale molto posteriore a quello in cui il documento è stato firmato, quando il relativo certificato di validità risultava scaduto, revocato o sospeso. Quindi sarà necessario collocare nel tempo, l'esistenza della firma del documento in modo da poter dimostrare

che è stata prodotta in un momento in cui il suo certificato era ancora valido e che dopo tale istante non è stata apportata più alcuna modifica. Per far ciò basta viene utilizzato il servizio di timestamp attraverso il quale viene inserita una sorta di "etichetta elettronica" per dimostrare che il documento recante una data firma esisteva in un ben preciso momento. Come per l'uso dei certificati, anche per la marcatura è necessaria un'autorità fidata che crei questa etichetta (Time Stamp Authority) . Quindi per apporre una marcatura l'utente deve inoltrare una richiesta alla TSA seguendo delle direttive comuni e in questo modo ritornerà il messaggio con data e ora di quel preciso momento. Il timestamp che viene inviato come risposta all'utente deve contenere le seguenti informazioni:

- versione del protocollo usato (TSP) ;
- digest del documento (che era stato inviato dall'utente per la richiesta);
- numero seriale;
- policy con la quale è stato generato;
- timestamp token che rappresenta la data e l'ora di creazione;
- altri campi opzionali.

La marcatura temporale ha due estensioni: .tsr per la modalità detached (quindi solo marca separata dal file firmato) oppure .m7m per la modalità attached. Risulta molto utile selezionare il formato m7m nel caso in cui si stia marcando un file recante estensione .p7m che come vedremo nel prossimo paragrafo è il formato di un file firmato. Infatti l'apertura del documento marcato permetterebbe la contemporanea verifica del timestamp associato al documento stesso, ma anche la verifica della firma digitale associata. La necessità di una marcatura temporale di un documento elettronico sorge anche dal fatto che ne aumenterebbe la sua validità: infatti associando un timestamp ad un documento firmato ne verrebbe estesa la validità, oltre la naturale scadenza dei certificati di firma (solitamente attorno ai 2-3 anni),

a circa 20 anni e di conseguenza si avrebbe anche un'estensione del valore probatorio del documento.

2.2 Formati di firma

La legge italiana prevede l'utilizzo di tre formati per produrre firme digitali. Parleremo brevemente dei primi due di essi mentre nei capitoli successivi svilupperemo il formato di firma che a noi interessa maggiormente: il formato XML.
Il primo formato, presente dal lontano 1997 quando il regolamento recante criteri e modalità per la formazione, l'archiviazione e la trasmissione di documenti con strumenti informatici e telematici [5] stabilì per la prima volta in Italia la validità della firma digitale, è lo standard PKCS7, meglio noto come p7m, che è appunto l'estensione di un documento in questo formato. Uno dei primi software utilizzati per apporre e verificare firme è quello fornito direttamente dal CNIPA (FCMT [6]).
Il 16 febbraio 2006 il CNIPA ha siglato un protocollo di intesa con Adobe System, che riconosce il formato PDF come valido e legalmente riconosciuto per la firma digitale di documenti. Grazie a questo accordo,ora è possibile utilizzare il formato PDF con i relativi software: molto diffuso è, per ovvi motivi, sia Adobe Reader che Acrobat Professional (il primo gratuito). Grazie a questi programmi, dopo aver impostato i parametri opportuni ed installato i plugins necessari (queste procedure vengono ben definite nella guida alla firma digitale presente sul sito del CNIPA [2]), si ha la possibilità di firmare e verificare files come mostrano le immagini di esempio di un documento pdf in cui viene apposta una firma (2.2) e di un file già firmato di cui si effettua una verifica (2.3): da notare che da questa finestra che riguarda le proprietà della firma, sarà possibile andare a controllare se :

- il documento é stato modificato dopo la firma;
- il certificato del firmatario è garantito dalla CA;

- il certificato del firmatario è scaduto o revocato;
- il certificato del firmatario è stato sospeso.

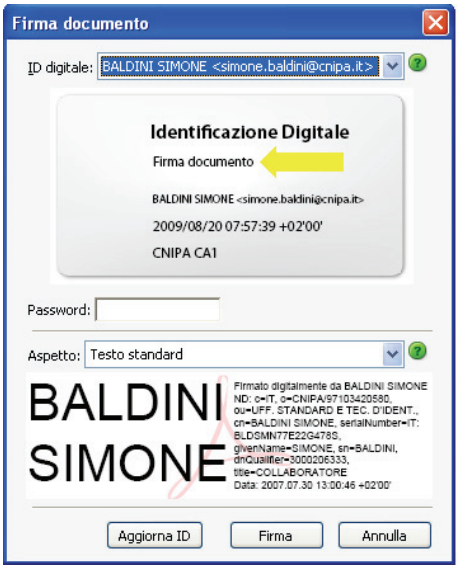

Figura 2.2: esempio di firma con Adobe Reader

Viene mostrata anche un'immagine di una verifica del certificato con il software fornito dal CNIPA, FCMT (2.4). Questo programma è ancora disponibile ma il CNIPA non ne fornisce più supporto o aggiornamento in quanto la sua attività di certificatore è terminata nel 2009.

In Italia gli strumenti hardware utilizzati per il processo di firma dei documenti con valore legale sono principalmente le smart card ma anche i token usb. Le smart card sono utilizzate solitamente per l'identificazione e la memorizzazione di informazioni personali riservate, grazie al loro grado di sicurezza molto alto. Tutte le smart card implementano dei sistemi di sicurezza in grado di impedire la copia o l'esportazione della chiave privata all'esterno della tessera. Tuttavia un ulteriore livello di sicurezza è il codice PIN (Personal Identification Number), senza il quale non è possibile utilizzare la scheda. Naturalmente per usare la smart card per i processi di firma digitale sono necessari anche un lettore di tessere e un opportuno software, in genere

2.2 Formati di firma

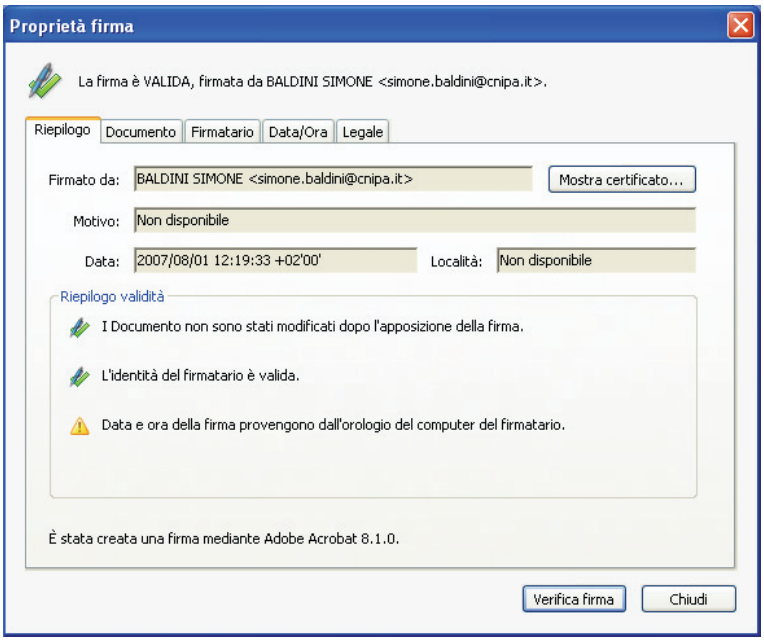

Figura 2.3: esempio di verifica con Adobe Reader

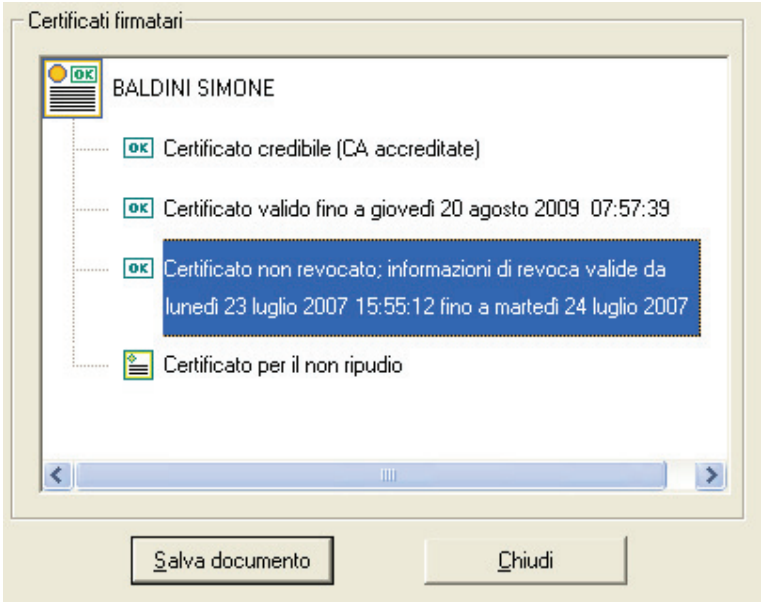

Figura 2.4: certificato visualizzato con FCMT

fornito insieme al lettore stesso in una sorta di "kit di firma" in modo che ci sia piena compatibilità tra tutti i componenti. Tutti questi strumenti, come scritto anche sulla guida alla firma digitale[2], hanno un costo complessivo che si aggira intorno ai 70-80 euro.

2.3 Valenza Giuridica della firma digitale

Ora ci occuperemo del valore legislativo della firma in Italia e di come esso è cambiato nel tempo. Come abbiamo già detto, l'equiparazione della firma digitale con firma autografa dal punto di vista legale è stata introdotta per la prima volta con il DPR 445/2000 nel quale viene scritto che "l'apposizione o l'associazione della firma digitale al documento informatico equivale alla sottoscrizione prevista per gli atti e documenti in forma scritta su supporto cartaceo" [7] e che nella pubblica amministrazione "l'uso della firma digitale integra e sostituisce ad ogni fine di legge l'apposizione di sigilli, punzoni, timbri, contrassegni e marchi comunque previsti" [8].
L'introduzione però della firma digitale in Italia è avvenuto con il DPR 513 del 10 novembre 1997. Altri decreti rilevanti ai fini del significato che la firma ha assunto negli anni, sono:

- DPCM 30 marzo 2009: l'ultimo in ordine temporale nel quale vengono definite le regole tecniche in materia di firme digitali (sostituisce il DPCM del 13 gennaio 2004);

- Delibera n.45 del 21 maggio 2009 nel quale si trovano le regole per il riconoscimento e la verifica del documento informatico;

- Circolare n.48 del 6 settembre 2005 del CNIPA che rappresenta la domanda di iscrizione all'elenco pubblico dei certificatori(il primo certificatore autorizzato a rilasciare dispositivi di firma per sottoscrivere documenti digitali è stato inserito in questo elenco nel 2000);

- Decreto legislativo 7 marzo 2005 n. 82 che in seguito è diventato il

documento fondamentale per il mondo digitale con il nome di codice dell'amministrazione digitale (CAD) nell'aprile 2006.

Nel corso del tempo è stata modificata la modalità di disconoscere la firma: infatti mentre con il decreto del 2000 la firma poteva essere disconosciuta dal presunto sottoscrittore, con il decreto del 2002 n.10 viene scritto che

> "Il documento informatico, quando è sottoscritto con firma digitale o con un altro tipo di firma elettronica avanzata, e la firma è basata su di un certificato qualificato ed è generata mediante un dispositivo per la creazione di una firma sicura, fa inoltre piena prova, fino a querela di falso, della provenienza delle dichiarazioni da chi l'ha sottoscritto."

In questo modo il presunto sottoscrittore doveva intentare una querela di falso per vedere annullato il valore della firma. Con il decreto del 2005 viene ristabilita l'uguaglianza con la firma autografa; per disconoscere la firma è necessario dare una prova che sia stato usato il dispositivo di firma non dal proprietario. Rimane tuttavia un aspetto che viene modificato con il codice di amministrazione digitale del 2006 e cioè quello di chi può disconoscere la firma:

> "Il documento informatico, sottoscritto con firma digitale o con un altro tipo di firma elettronica qualificata, ha l'efficacia prevista dall'articolo 2702 del codice civile. L'utilizzo del dipositivo di firma si presume riconducibile al titolare, salvo che sia data prova contraria."
> Decreto Legislativo n. 82 7 marzo 2005

> "Il documento informatico, sottoscritto con firma digitale o con un altro tipo di firma elettronica qualificata, ha l'efficacia prevista dall'articolo 2702 del codice civile. L'utilizzo del dispositivo di firma si presume riconducibile al titolare, salvo che questi dia

la prova contraria."
Codice Amministrazione Digitale 2006

Dal primo decreto si nota che paradossalmente si può presentare il caso in cui chiunque può dare prova di un uso del dispositivo vietato dalle norme, e quindi di un annullamento degli effetti legali della firma digitale, anche se il presunto sottoscrittore ne riconosce la piena legittimità. Quindi con il successivo correttivo del 2006 la facoltà di disconoscere la firma viene giustamente riassegnata al solo titolare del dispositivo.
Pertanto, la firma digitale garantisce l'identificabilità dell'autore, l'integrità e l'immodificabilità del documento ed ha valore probatorio. Ci sono anche altre caratteristiche della firma digitale, che però non sono comuni anche a quella autografa: ad esempio ha un limite di tempo oltre il quale non viene più considerata, oppure non è possibile riconoscere un documento da una sua copia. Tuttavia le molte proprietà positive del digitale fanno pensare che l'avvento di questo documento possa sostituire completamente quello cartaceo, sebbene ci sia tuttora molto scetticismo da parte dell'utente, forse dovuto anche alla non completa conoscenza della materia, sulla sicurezza e sulla univoca riconducibilità della firma al titolare.

Capitolo 3

XML Signature

Inizieremo ora a parlare del formato di più recente utilizzo, considerato il più "potente" e completo rispetto agli altri precedenti, ma non ancora diffuso pienamente, forse per una sua maggiore complessità.

3.1 eXtensible Markup Language

Il formato XML è nato prevalentemente per produrre i documenti elettronici che devono poter essere scambiati senza dovere preventivamente stabilire un formato comune. I documenti XML sono ad esempio frequentemente utilizzati in applicazioni di tipo sanitario (referti medici) o finanziario. L'XML è un linguaggio di markup nato principalmente per fornire maggiore libertà nella definizione dei tag e come una sorta di integrazione di HTML. Esso é diventato standard W3C nel 1998 con la sua versione 1.0. Il suo successo è dovuto a molteplici motivi tra i quali possiamo sicuramente includere la sua espandibilità, la sua portabilià e anche la relativa semplicità rispetto ad altri linguaggi di markup.
Un documento XML è sostanzialmente un file di testo: per questo è facilmente trasferibile e modificabile su differenti piattaforme hardware e software, favorendo quindi l'interoperabilità nello scambio tra documenti, anche sottoscritti digitalmente. Inoltre sintatticamente è composto da una struttura

gerarchica che prevede l'elemento principale, o radice, e al suo interno altri elementi correttamente annidati l'uno nell'altro.

3.2 Sintassi XML per la firma digitale

Nel febbraio 2002 è stato definito uno standard per la firma digitale di documenti XML, chiamato XML-Signature[9]. Qualche mese più tardi in Italia l'AIPA[1] emana la circolare CR/40 sul "linguaggio XML come formato per la rappresentazione elettronica dei provvedimenti normativi" nella quale si dice che:

> La condivisione di un medesimo formalismo di marcatura dei testi normativi resi accessibili da organismi differenti, anche se dotati di sistemi informatici tecnologicamente eterogenei, consente di costruire un sistema di ricerca unitario, in grado di offrire funzionalità più efficaci ed un livello di precisione superiore a quello ottenibile con la semplice ricerca per parole. Inoltre, la marcatura dei provvedimenti normativi in base a regole definite consente di rappresentare informazioni relative anche a quelle specifiche parti del testo che contengono riferimenti ad altri provvedimenti e - soprattutto se attuata già a partire dalle fasi di drafting - rende possibile la realizzazione di sistemi informatici di supporto alle azioni di riordino normativo e di costruzione dei testi vigenti.

Dunque uno dei vantaggi maggiori, come detto anche dallo stesso W3C appena emesso lo standard, che ha portato all'adozione dell'XML per firmare documenti è che risulta possibile firmare anche una sola parte del documento invece che l'intero file. In questo modo sarà possibile aggiungere nuovi campi al file da parte di utenti diversi lasciando inalterati i tag precedentemente firmati. Si intuisce che questa flessibilità può risultare molto utile per un

[1] Autorità per l'informatica nella pubblica amministrazione poi divenuta nel 2003 il CNIPA

documento che debba essere modificato da più persone nel tempo. Una firma XML può essere fatta su più di un tipo di dato, ad esempio e' possibile firmare dati HTML, dati in formato JPG, dati in XML o dati appartenenti ad una sezione specifica di un file XML.

Ora passeremo a descrivere come è composto l'XML-Signature. Innanzitutto va detto che un documento XML dovrebbe essere formato da un tag detto radice che racchiude più tag al suo interno. Inoltre lo standard deve mantenere un certo rigore sintattico per avere la massima chiarezza e semplicità. I documenti che appartengono a queste regole sono chiamati documenti ben formati(well-formed).
Le principali regole per un documento XML ben formato sono le seguenti:

- Ci deve essere un solo elemento chiamato root che contenga tutti gli altri elementi,fatta eccezione per i commenti e per le dichiarazioni come quella sulla versione XML utilizzata.
- Ogni elemento deve avere un tag di apertura e di chiusura. Se è vuoto allora può essere nella forma abbreviata (/>).
- Gli elementi devono essere opportunamente nidificati, quindi i tag di chiusura devono seguire l'ordine inverso dei rispettivi tag di apertura.
- I nomi dei tag e degli attributi devono coincidere nei tag di apertura e chiusura tenendo conto anche di maiuscole e minuscole,in quanto XML è case sensitive.
- I valori degli attributi devono essere racchiusi tra singoli o doppi apici.

Per quanto riguarda il nostro caso specifico, la firma in XML avviene tramite una procedura che utilizza un algoritmo di crittografia a chiave asimmetrica. Spiegheremo in dettaglio questa procedura nel prossimo capitolo, insieme anche alla procedura di verifica della firma. Ora andremo ad elencare le componenti di un XML-Signature.

Osserviamo questa immagine (3.2):

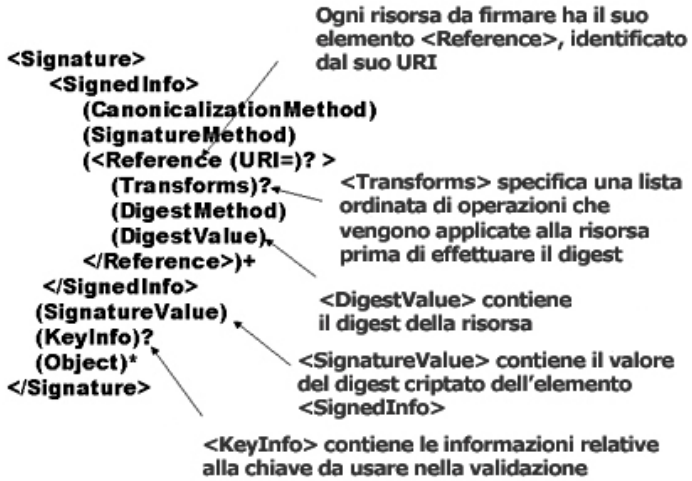

Figura 3.1: Componenti di XML-Signature[10]

Come già detto per la firma digitale in generale, non viene cifrato l'intero documento ma il suo digest. L'elemento Signature racchiude interamente la firma ed è principalmente composto dai tre tag <SignedInfo>, <SignatureValue> e <KeyInfo>. <SignedInfo> comprende informazioni su come la firma è stata generata: in particolare contiene la procedura usata per ottenere la forma canonica(CanonicalizationMethod), l'algoritmo vero e proprio che ha prodotto la firma(SignatureMethod) e informazioni sull'URI[2] che è stato firmato(Reference).
La forma canonica XML corrisponde alla rappresentazione della struttura del documento che risulta essere uguale a meno di differenze sintattiche come ad esempio la presenza di spazi bianchi o l'ordine degli attibuti, etc. L'operazione che appunto porta il documento a questa forma è detto XML Canonicalization. L'algoritmo per creare la firma e anche per la validazione è uno degli algoritmi che vengono utilizzati nella crittografia asimmetrica(ad

[2]Uniform Resource Identifier

esempio RSA descritto nel capitolo precedente). All'interno dell'elemento <Reference> vi sono informazioni che riguardano l'URI del dato da firmare ma anche il metodo che ne calcola il digest(DigestMethod), il valore del digest stesso(DigestValue) e il tag <Transforms> che è opzionale e che descrive quali operazione ha subito il file prima che venisse effettuato il digest,ad esempio cifratura di alcuni dati. L'elemento <SignatureValue> contiene il valore della firma digitale ottenuta secondo l'algoritmo indicato nell'elemento precedente. La struttura <KeyInfo> contiene informazioni sulla chiave pubblica di chi ha firmato il documento XML e si utilizza per decifrare la firma digitale XML nella fase di verifica dell'autenticità e dell'integrità del documento. Al suo interno può essere presente il valore della chiave pubblica secondo l'algoritmo usato(RSAKeyValue o DSAKeyValue). Questo elemento è opzionale in quanto il firmatario potrebbe non voler mostrare informazioni sulla propria chiave o questi dati potrebbero già essere noti e quindi non rappresentati esplicitamente. Ora passeremo a descrivere alcune regole per presentare le firme XML e quali algoritmi bisogna utilizzare, in base al tipo di firma; queste regole sono stabilite dalla specifica RFC[3] 3275 [11] ed emanate in Italia il 18 maggio 2006 [12]. La prima distinzione che viene fatta è tra le diverse modalità di firma: ne vengono definite 3, enveloped enveloping e detached. Nella prima la firma fa parte del documento,ne diviene un elemento, cioè l'oggetto firmato contiene la firma stessa; con la modalità enveloping invece è la firma che contiene tutti i dati dell'oggetto firmato. La modalità detached indica che ciò che viene firmato non è incluso nell'elemento firma, cioè il documento da firmare è completamente disgiunto da quello che contiene la firma digitale. In pratica una modalità si distingue dall'altra in base alla posizione dell'elemento <Signature> nel file, da come vediamo nei prossimi tre esempi.

[3]Request For Comments: documenti che contengono specifiche tecniche standard definite dall' Internet Engineering Task Force (IETF) e riconosciute a livello mondiale.

```
<Signature Id="FirmaDetached" xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      ...
      <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>j4fdsu4325riwjerfow732ewjdp</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>...</SignatureValue>
      <KeyInfo>
        <KeyValue>
          <DSAKeyValue>
            <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
          </DSAKeyValue>
        </KeyValue>
      </KeyInfo>
</Signature>
```

Il primo rappresenta la firma detached e quindi mostriamo il file della firma che ha in <Reference> l'URI del documento che ha firmato:
Di seguito le modalità enveloped ed enveloping:

```
<paziente xmlns="http://www.medicalclinic.com/">
  <nome>Mario Rossi</nome>
  <codice>123456</codice>
  <visita data="30-10-2040">
       <esame>Analisi del sangue</esame>
       ...
  </visita>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
  </Signature>
  ...
</paziente>

/*****************************************************************/

<Signature>
  <SignedInfo>
    ...
   <Reference URI="#paziente1">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>...</DigestValue>
   </Reference>
```

```
  </SignedInfo>
  ...
  <Object id="paziente1">
    ...
  </Object>
</Signature>
```

Nella prima, come già detto, la firma è contenuta nel documento quindi l'elemento <Signature> è interno al file da firmare; nella enveloping invece <Signature> contiene l'oggetto da firmare. Delle tre tipologie, la firma enveloped risulta essere la meno utilizzata mentre le altre due tipologie sono altrettanto diffuse e importanti. La modalità detached abbiamo detto che si applica per avere il file da firmare e la firma separati; è possibile però anche avere entrambi nello stesso file con la condizione di trovarsi ad un pari livello di annidamento, cioè che i tag <Signature> e <Object>(dove sono contenuti i dati del documento da firmare) siano elementi "fratelli"(=sibling): in entrambi i metodi viene creato un nuovo file che potrà contenere o la sola firma o quest'ultima insieme a tutte le informazioni di ciò che si firma. C'è anche la possibilità di firmare più oggetti contemporaneamente associando in un unico file XML le diverse modalità di firma.
La delibera che ha permesso la definizione delle regole tecniche per la firma in XML descrive anche quali algoritmi usare durante le varie fasi di firma o verifica: l'algoritmo per creare il digest da firmare deve essere specificato in questo modo:

```
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
```

viene quindi utilizzata la funzione SHA-1 (Secure Hash Algorithm 1) per produrre il digest di lunghezza fissa di 160 bit a partire da un messaggio di lunghezza massima di 2^{64-1} bit. Per quanto riguarda l'algoritmo di firma, va applicato l'RSA-SHA1 all'elemento <SignedInfo> in questo modo

```
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
```

Degli algoritmi di canonicalizzazione le applicazioni di verifica devono saper gestire almeno il Canonical XML 1.0

```
<CanonicalizationMethod Algorithm="http://www.w3.org/tr/2001/REC-xml-c14n↩
    -20010315"/>
```

Le trasformazioni come abbiamo detto sono opzionali, ma é stato definito un insieme minimo che i programmi di verifica dovranno poi gestire: esso include operazioni come canonicalizzazione, encoding, decoding, XSLT, XPath, validazione XML-Schema

```
<Transforms>
  <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
</Transforms>
```

Vedremo che le trasformazioni XPath sono importanti nel caso in cui, nella modalità Enveloped, vengano applicate firme successive alla prima sugli stessi dati.

3.3 XAdES

Mentre XML-Signature è la struttura generale per firmare digitalmente documenti XML, XAdES (XML Advanced Electronic Signature) è una specializzazione di XML-Signature che definisce diversi profili per l'utilizzo della firma elettronica qualificata come definita dalla direttiva 1999/93/EC dall'Unione Europea. Un importante beneficio che XAdES ha portato è che i documenti firmati elettronicamente possono rimanere validi per lunghi periodi, anche se vengono distrutti gli algoritmi crittografici.
XAdES definisce sei profili che differiscono nel livello di protezione offerto. Ogni profilo include ed estende quello precedente:
XAdES-BES, formato base che soddisfa i requisiti legali minimi della Direttiva Europea sulla firma avanzata;
XAdES-T (timestamp), che appunto aggiunge il campo del timestamp per

proteggerla dal ripudio;
XAdES-C (completo), ingloba XAdES-BES e aggiunge tutti i dati (certificato e riferimento alla revocation list) per poter verificare la firma anche offline e nel futuro (ma non memorizza i dati reali);
XAdES-X (esteso), aggiunge i timestamps ai riferimenti introdotti da XAdES-C;
XAdES-X-L (lungo termine esteso), aggiunge i certificati e le liste reali di annullamento al documento firmato per permettere la verifica in futuro anche se la loro fonte originale non è più disponibile;
XAdES-A (archivistico), aggiunge la possibilità di poter inserire periodici timestamp(ad esempio ogni anno) per generare uno storico delle firme nel documento archiviato per impedire l'indebolimento della firma nel corso del periodo di immagazzinamento.
Il formato XAdES aggiunge elementi all' XML-Signature ed in particolare all'elemento <Object>

```
<ds:Signature>
...
<ds:Object>
        <QualifyingProperties>
                <SignedProperties>
                        <SignedSignatureProperties>
                          (SigningTime)?
                          (SigningCertificate)?
                          (SignaturePolicyIdentifier)?
                          (SignatureProductionPlace)?
                          (SignerRole)?
                        </SignedSignatureProperties>

                        <SignedDataObjectProperties>
                          (DataObjectFormat)*
                          (CommitmentTypeIndication)*
                          (AllDataObjectsTimeStamp)*
                          (IndividualDataObjectsTimeStamp)*
                        </SignedDataObjectProperties>
                </SignedProperties>

                <UnsignedProperties>
                        <UnsignedSignatureProperties>
```

```
                                    (CounterSignature)*
                                    (SignatureTimeStamp)+
                                </UnsignedSignatureProperties>
                            </UnsignedProperties>
                        </QualifyingProperties>
</ds:Object>
...
</ds:Signature>
```

Questa è la struttura principale del formato XAdES-T che si differenzia dal formato XAdES-BES solamente per l'aggiunta del tag (SignatureTimeStamp)+[4]. XAdES-T e XAdES-BES sono i due formati che, secondo la delibera del CNIPA [12] che fa riferimento alla specifica dell'European Telecommunications Standards Institute[14], le applicazioni di verifica di una firma devono gestire.
L'elemento <ds:Object> contiene un solo elemento <QualifyingProperties> che fa da contenitore a tutte le informazioni legate alla firma e quindi all'elemento <Signature>. Queste proprietà sono poi suddivise dagli elementi che devono essere firmati e da quelli che non vengono firmati da XML-Signature. Per ciò che deve essere firmato ci dovrà essere un reference in <ds:Signature> che si riferisce appunto all'elemento <SignedProperties>. Degli elementi da firmare <SigningCertificate> contiene un insieme ristretto di riferimenti ai certificati da utilizzare nella verifica della firma, mentre <SigningTime> è il momento in cui viene terminato il processo di firma. La struttura <SigningCertificate> se non è presente, allora il certificato deve essere presente nel tag <ds:KeyInfo> descritto in precedenza nell'elemento <ds:X509Data> ed un riferimento ad esso dovrà essere naturalmente in <ds:SignedInfo> in modo da essere incluso nel calcolo del valore della firma. In questo modo il certificato sarà garantito dalla firma stessa. Degli elementi che non devono essere firmati fa parte <CounterSignature> che rappresenta la controfirma quindi ci sarà un altro elemento <Signature> al suo interno con un <Reference> all'elemento <ds:SignatureValue> della firma originaria (<ds:Signature>).

[4] negli esempi che verranno proposti '?' indica zero o una occorrenza dell'elemento, '+' ne indica una o più mentre '*' ne denota zero o una o più di una.

Infine il timestamp contenuto in <SignatureTimeStamp> viene calcolato sul <ds:SignatureValue> ed è responsabilità del Time Stamp Provider fornire una marcatura della firma che quindi farà parte di uno degli attributi non firmati.
XaDES-BES è il minimo formato che soddisfa i requisiti di legge per le firme elettroniche, come definito nella direttiva europea sulle firme elettroniche. Esso prevede l'autenticazione di base e la garanzia di integrità. XaDES-T aggiunge il supporto delle marche temporali per evitare il ripudio della firma.

3.4 Firme Multiple

Un ulteriore vantaggio dell'uso di XML-Signature è la possibilità di apporre al documento firme multiple. In base alla delibera del 17 febbraio 2005[15] "una stessa busta crittografica può contenere più firme digitali". Ci sono però diversi modi di classificare le firme multiple che rischiano di creare non poca confusione. La principale definizione a cui verrebbe da pensare quando si parla di firme multiple è quella che viene chiamata firma multipla parallela e che consiste nella firma dei soli dati di partenza da parte di più utenti. In questo caso vi sarà un elemento <Signature> univoco per ogni firmatario e ognuno di questi conterrà nell'elemento <Reference> l'URI del documento firmato. Ovviamente le firme dovranno essere apposte in base alla modalità di imbustamento della prima. Se la modalità di partenza è enveloped anche le successive firme saranno apposte utilizzando la stessa modalità; è necessario però che le firme successive siano applicate agli stessi dati sui quali è stata calcolata la prima e in fase di verifica si deve comparare il documento originale (senza firma) con quello firmato. XMLDSig definisce un meccanismo per rimuovere l'elemento <Signature> dal documento: è necessario effettuare una trasformazione XPath al documento. Questa trasformazione, che dovrà essere gestita dalle applicazioni di firma e di verifica, viene posta nel tag opportuno all'interno dell'elemento <SignedInfo> e consiste nella rimozione di tutte le strutture di tipo <Signature>(quindi di tutte le firme

precedenti) presenti nel documento a partire dal nodo radice(escluso).Qui riportiamo l'esempio dell'operazione da effettuare:

```
<SignedInfo>
  <CanonicalizationMethod
          Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
  <SignatureMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
  <Reference URI="">
     <Transforms>
      <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
           <dsig-xpath:XPath Filter="subtract"
                        xmlns="http://www.w3.org/2002/06/xmldsig-filter2">
                  /descendant::Signature
           </dsig-xpath:XPath>
      </Transform>
      <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n↩
          -20010315" />
     </Transforms>
     <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
     <DigestValue >f/Rcq6wu9gORMioxAxaof7pZux8=</ds:DigestValue>
  </Reference>
</SignedInfo>
```

Se la modalità della prima firma è detached, anche le successive dovranno essere poste nella stessa modalità e l'elemento reference sarà uguale per ogni firma successiva. Invece nel caso in cui sia stata apposta una firma enveloping, le firme successive dovranno essere in modalità detached con il riferimento ai dati contenuti nel tag <Object>. Vediamone un esempio:

```
<Envelope>
  <Signature Id="Firma1" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
          <CanonicalizationMethod ........."/>
          <SignatureMethod ......."/>
          <Reference URI="#paziente"
                     Type="http://www.w3.org/2000/09xmldsig#object">
                  <DigestMethod ..."/>
                  <DigestValue >....</DigestValue>
          </Reference>
      </SignedInfo>
      <SignatureValue >....</SignatureValue>
```

```
        <KeyInfo> ..... </KeyInfo>
        <Object Id="paziente">
            <data> ... </data>
        </Object>
    </Signature>
    <Signature Id="Firma2" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod ........."/>
            <SignatureMethod ......."/>
            <Reference URI="#paziente">
                .....
            </Reference>
        </SignedInfo>
        .....
    </Signature>
</Envelope>
```

Il CNIPA definisce nella categoria delle firme multiple, quella a catena, che viene applicata al documento informatico; nel nostro caso di documenti XML si prenderebbe semplicemente tutto il file firmato, comprensivo sia di firma che di dati che sono stati firmati, e lo si firmerebbe nuovamente. Altro caso è rappresentato dalla controfirma(la firma applicata ad un'altra firma) che forse andrebbe chiamata solamente con questo nome invece che comprenderla nelle tipoligie delle firme multiple. Come abbiamo visto nel paragrafo 4 sui profili per l'utilizzo della firma elettronica qualificata, a partire da XAdES-BES la controfirma viene inclusa nell'elemento <CounterSignature>. In questo caso l'ordine delle firme applicate è importante quindi una firma dipende dalla precedente; più controfirme corrispondono a più elementi <CounterSignature>. Il contenuto di <CounterSignature> in una firma elettronica qualificata è di una firma (quindi un elemento <Signature>) del SignatureValue presente nella firma qualificata. Abbiamo detto nell'introduzione che l'apposizione di firme multiple risulta di notevole importanza in un contesto sanitario o di un ambulatorio medico nella gestione di documenti come una cartella clinica o nel caso di semplici firme da parte dei dottori ad un qualsiasi tipo di documento.

3.5 XML Key Management Specification

Sebbene non sia una parte di XMLDSIG, XKMS ne è strettamente legato. XML Key Management Specification è un protocollo sviluppato dal W3C che descrive la distribuzione e la registrazione delle chiavi pubbliche; utilizza il quadro di servizi web per rendere più facile per gli sviluppatori garantire la comunicazione tra applicazioni utilizzando un'infrastruttura a chiave pubblica, per ricevere informazioni aggiornate sulla chiave per la cifratura o autenticazione. Come XML-Signature e anche XML-Encryption[16], esso fa parte degli standard XML Security che forniscono un insieme di protocolli che puntano a soddisfare i requisiti sulla sicurezza e creano le basi ad altre tecnologie basate su XML (come i Web Services) per garantirla. XKMS, progettato per integrare e migliorare XML-Signature e XML-Encryption e non per competere con essi, assume che il web service che si occupa dell'elaborazione del codice XML risieda in un ambiente in cui sia le chiavi che i certificati vengano gestiti in modo sicuro. XKMS fornirà uno standard di definizioni basate su XML che consentiranno di usare servizi remoti di crittografia, creazione e gestione di firme digitali e chiavi offerti da terze parti fidate. Quindi questo standard definisce dei protocolli di gestione per la creazione di una coppia di chiavi (pubblica e privata), collegamento (binding) di queste con identità e la rappresentazione di questa coppia di chiavi in diversi formati. Le specifiche prevedono una serie di tag per l'interrogazione dei servizi esterni di gestione chiavi e di validazione firme e altri tag da usare per l'invio delle risposte. Vengono ad esempio utilizzate le informazioni contenute nel tag <KeyInfo> definito in XMLDSIG attraverso il XKISS (XML Key Information Service Specification) che è appunto un servizio di informazione della coppia di chiavi. Quindi XKMS supporta l'uso di XMLDSIG per mantenere l'integrità e l'autenticità dei messaggi; definisce oltre ad essi, altri processi di autenticazione, supporto per le informazioni sulla propria chiave e funzionalità per la sicurezza.

Capitolo 4

Panoramica dispositivi di firma digitale

Abbiamo già accennato che per generare firme digitali bisogna essere dotati di un dispositivo sicuro per la generazione di esse (una smartcard o un token USB), eventualmente un lettore di smartcard e un software in grado di interagire con il dispositivo. I costi del kit varia da certificatore a certificatore ma in genere non si va oltre i 70-80 euro. Anche il certificato ha un costo intorno ai 10 euro e soprattutto una scadenza, e deve quindi essere rinnovato in genere ogni 2-3 anni (in base alla sua durata che dipende dal certificatore). Nel caso in cui un azienda voglia dotare del kit un notevole numero di dipendenti (come può avvenire nel caso di un ospedale), ogni certificatore fornirà vantaggiose condizioni economiche per forniture di particolare rilievo.
In genere i dispositivi di firma forniti sono per la maggior parte freeware, o meglio molte aziende forniscono delle versioni gratuite e fruibili ai privati dei loro software di firma, tenendo però la versione più completa (solitamente quella che ha la possibilità di apporre marcature temporali) a pagamento. Sono stati presi in considerazione alcuni sistemi tra i più diffusi. Va detto che, come accade in molti campi nell'informatica, anche lo sviluppo di software e hardware per la firma digitale ha ampia diffusione ed è ampiamente supportato su sistemi windows. Infatti quasi tutti i programmi di firma vengono

distribuiti per windows, mentre per alcuni viene offerta anche la versione per sistemi linux e mac.

4.1 Freeware

4.1.1 DigitalSign

Fornito dall'azienda Comped [17], un'azienda specializzata in prodotti informatici, nella versione 3.0 è disponibile in diverse edizioni:

- Reader: solamente un visualizzatore di documenti che verifica firme digitali e marche temporali. Viene fornito gratuitamente.
- Lite: permette di firmare documenti con la propria smartcard ma manca di alcuni servizi come la marcatura temporale, la cifratura e decifratura, la possibilità di apporre firme multiple. Viene anche'esso fornito gratuitamente per un utilizzo privato.

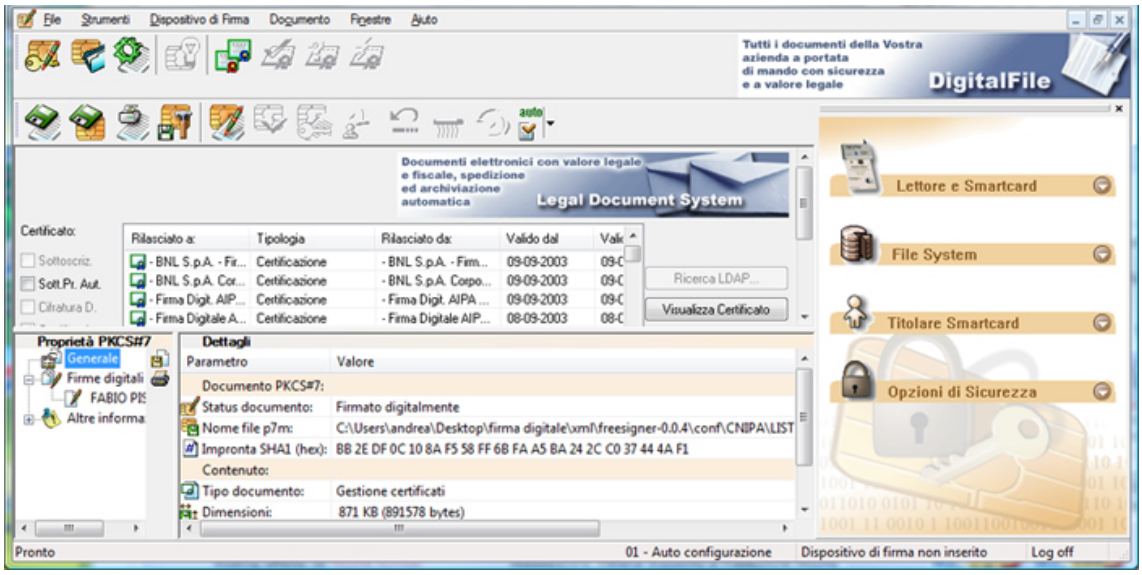

Figura 4.1: DigitalSign 3.0 Lite

- Professional: è l'edizione completa che fornisce tutte le funzionalità per un intero processo di firma o di verifica di un documento. Utile la registrazione delle attività, che fornisce il dettaglio delle operazioni eseguite dall'applicazione con anche la data e l'ora dell'esecuzione.

Nella sua versione Professional è certamente un software completo, ma anche l'edizione gratuita fornisce la possibilità di svolgere le operazioni di firma e verifica in modo chiaro grazie anche alla visualizzazione dei dettagli del file aperto come il valore dell'hash su cui è stata o verrà calcolata la firma. Al momento dell'apertura di un documento firmato digitalmente, vengono verificate subito tutte le firme apposte su di esso e le eventuali marche temporali che ne prolungano la validità. Nella sezione dei dettagli vengono visualizzati anche messaggi di warning nel caso in cui il documento contenga elementi che potrebbero metterne in discussione la sua validità. Naturalmente è presente anche la possibilità di apporre firme multiple o di controfirmare un documento già firmato. Il software risulta compatibile con le smartcard fornite dai principali certificatori, o almeno dai certificatori definiti nella guida alla firma digitale[2] sul sito del CNIPA; viene inoltre adottato da Telecom Italia che fornisce il sistema dal suo sito insieme al kit di firma. Forse può risultare un po' complicata e dispersiva l'interfaccia utente per chi è alle prime armi ma tutto questo è dovuto alle sue molteplici funzioni. Inoltre è disponibile solo per sistemi Windows ed è necessario anche Internet Explorer per le sue operazioni di verifica dei certificati o di altre operazioni come l'apertura di documenti HTML e XML. In particolare un file in formato XML viene visualizzato con Internet Explorer se il documento non supera una certa dimensione, mentre nel caso contrario viene utilizzato il viewer integrato (SicurView) che si occupa anche dell'apertura di pdf o di immagini.

4.1.2 Dike

Dike è un caso in cui il certificatore ha creato anche il software di firma. InfoCert SpA [13], tra i primi enti certificatori qualificati in Italia dopo il passaggio di consegna da parte di Infocamere, oltre che rilasciare certificati

ad un costo di 7 euro, fornisce un software gratuito più completo rispetto alla versione equivalente fornita da Comped, ed uno per il quale è necessario acquistare la licenza al prezzo di 220 euro.Inoltre il sito del gestore fornisce marche temporali al costo di 30 euro per un pacchetto da 100. Dike è uno dei programmi gratuiti più diffusi e completi disponibili; la sua versione free è già sufficiente a compiere tutte le operazioni necessarie ad un privato: firma, verifica e marcatura temporale di uno o più file. L'interfaccia utente è forse tra le più semplici ed intuitive, inoltre sembra che sia il più utilizzato anche perchè disponibile per tutte le piattaforme. Infatti fino a qualche anno fa le versioni che venivano fornite per sistemi diversi da quelli windows non erano sufficientemente supportate e davano problemi di compatibilità con molti lettori e smartcard; negli ultimi tempi invece sono stati portati avanti i progetti anche per ambienti come GNU/Linux (per linux si è arrivati alla versione 4.2.4 mentre su windows alla 4.3.0) ed ora almeno i lettori e le smartcard per cui la stessa Infocert fornisce i driver sono pienamente compatibili con il programma fornito. Il software DikePro per il quale è necessario l'acquisto di una licenza, possiede alcune funzionalità in più che possono tornare utili in particolare ad aziende che hanno la necessità di firmare numerosi documenti contemporaneamente: infatti fornisce la possibilità di digitare solamente una volta il pin di firma anche per firmare più file, oppure consente la firma detached quindi la creazione della firma in un file diverso dall'originale che deve essere firmato. Sebbene ad utenti privati possa sembrare eccessivo e superfluo l'acquisto della licenza per DikePro, per aziende potrebbe essere invece un notevole risparmio di tempo l'apposizione di firme o di marche temporali a centinaia di file contemporaneamente. Insieme a Dike, Infocert fornisce un software freeware separato per la gestione delle smartcard, DikeUtil: esso permette di attivare o verificare smartcard, scegliere il lettore da utilizzare per le operazioni elettroniche e anche di verificare i certificati o di rinnovare il proprio. Tornando a Dike, al momento dell'apertura di un file, non viene mostrata alcuna informazione su di esso come avveniva invece con DigitalSign: è necessario selezionare l'operazione che si vuole eseguire.

4.1 Freeware

In seguito alla firma e alla verifica appaiono schermate simili a queste figure.

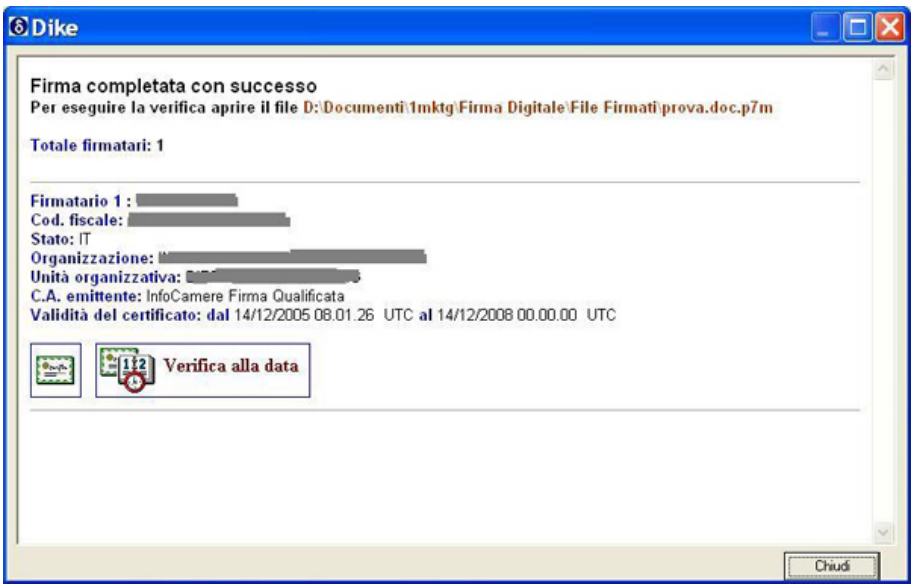

Figura 4.2: Firma di un file con Dike

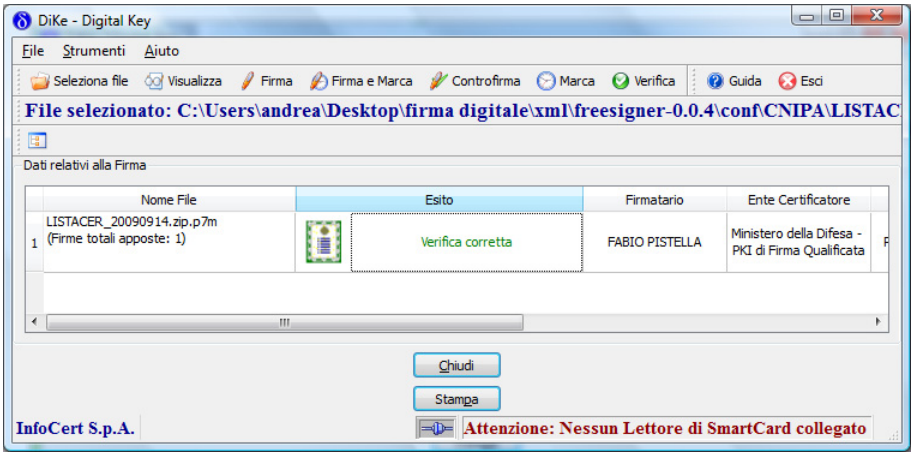

Figura 4.3: Verifica di un file con Dike

InfoCert dopo l'emanazione delle "regole tecniche per la definizione del

profilo di busta crittografica per la firma digitale in linguaggio XML" ha distribuito anche un programma che verifica le firme XML, verificatoreXML appunto, disponibile solamente per windows. Esso consente di verificare un qualunque documento XML posto in una delle tre modalità descritte dallo standard XML-Signature [9].

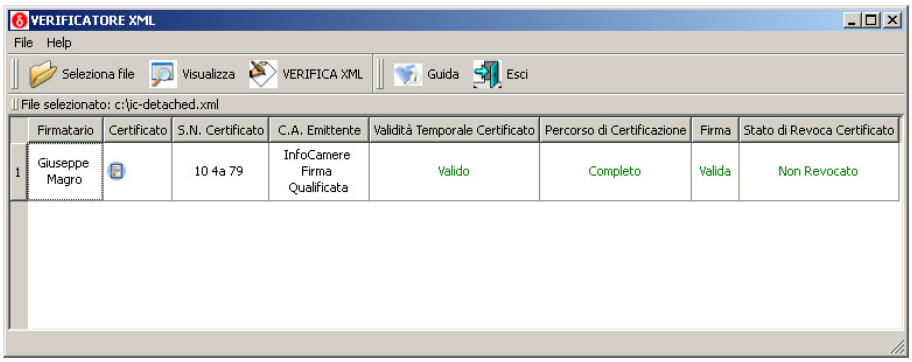

Figura 4.4: Verifica di un documento XML

4.2 Open Source

Sono molti i progetti open source sviluppati nel corso degli ultimi anni. Tuttavia la maggior parte di questi sono rimasti solamente progetti in fase di sviluppo a causa di mancanza di supporto.

4.2.1 Freesigner

Freesigner è uno di questi, fornito anch'esso da InfoCert in collaborazione con il comune di Trento e il dipartimento di Ingegneria dell'Informazione dell'università di Padova, sulla carta permette la firma e la verifica di file. Si basa su un secondo progetto chiamato j4sign, sempre del comune di Trento. Questo progetto e' un tentativo di realizzare un software libero, con sorgenti aperti che affianchi il software proprietario sviluppato dai certificatori inclusi nell'elenco pubblico. Attualmente è in fase di sviluppo ma non sembra per

adesso che da parte di Infocert ci sia la voglia di supportarlo veramente. A conferma di questo vi è il fatto che negli ultimi tempi, come abbiamo già detto, ci sia stato un netto miglioramento del software Dike nella compatibilità con l'ambiente GNU/Linux. Infatti dovrebbe essere compito dell'ente certificatore fornire supporto per le varie tipologie di smartcard in modo che il mondo del software libero riesca ad utilizzarle. Il software è disponibile per windows e linux ed utilizza OpenSc, un set di librerie e utilities che mettono a disposizione driver per diverse marche di smartcard. La configurazione del lettore, la firma e la verifica sono le tre operazioni possibili con questo software.

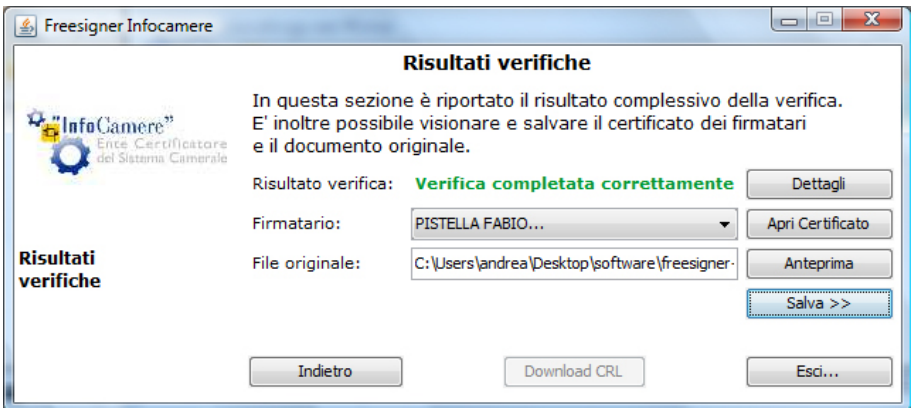

Figura 4.5: Verifica documento

4.2.2 OpenSignature

OpenSignature [18] è l'evoluzione di Firma, anch'esso tentativo di fornire un programma libero con il supporto al maggior numero di smartcard. Iniziato nel 2002, ha raggiunto un buon livello di usabilità e supporta le smartcard di InfoCert, Poste ed anche quelle per la CNS[1]. Può essere installato su windows o linux e necessita anch'esso delle librerie OpenSc. Qualche

[1]La Carta Nazionale dei Servizi è una smart card utilizzata per accedere ai servizi online della pubblica amministrazione

anno fa è stato distribuito anche il software per firmare un documento pdf, OpenSignPdf.

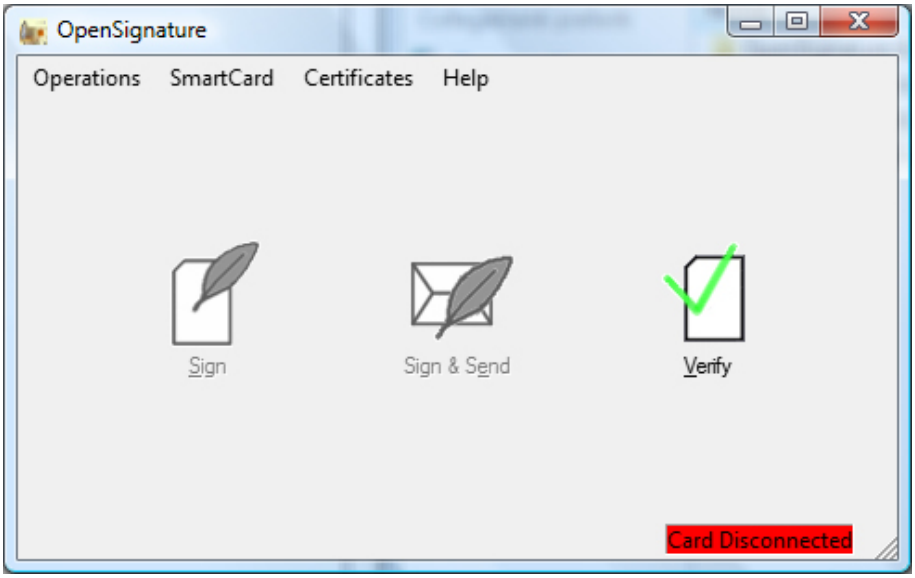

Figura 4.6: Software OpenSignature

4.2.3 Javasign

Un altro progetto con licenza GPL è javasign, che come dice il nome stesso è stato scritto in java per cercare di renderlo più portabile. Javasign supporta tutte le carte compatibili con le specifiche definite per la Carta Nazionale dei Servizi ed è stato testato con le carte fornite da InfoCert; è fornito per ambienti windows e linux ma naturalmente molte smartcard hanno problemi con linux, ad eccezione di quelle emesse da InfoCert. Ha bisogno delle librerie PCSCLite che permettono di usare PC/SC con linux che è il protocollo standard per i lettori di carte. Il software permette anche di generare licenze Copyzero X [19], calcolando automaticamente il digest SHA-1 relativo alle opere licenziate. Offre anche la possibilità di effettuare l'operazione di marcatura temporale. L'immagine sotto presenta un esempio

di verifica di un documento con le relative informazioni sull'esito del processo eseguito.

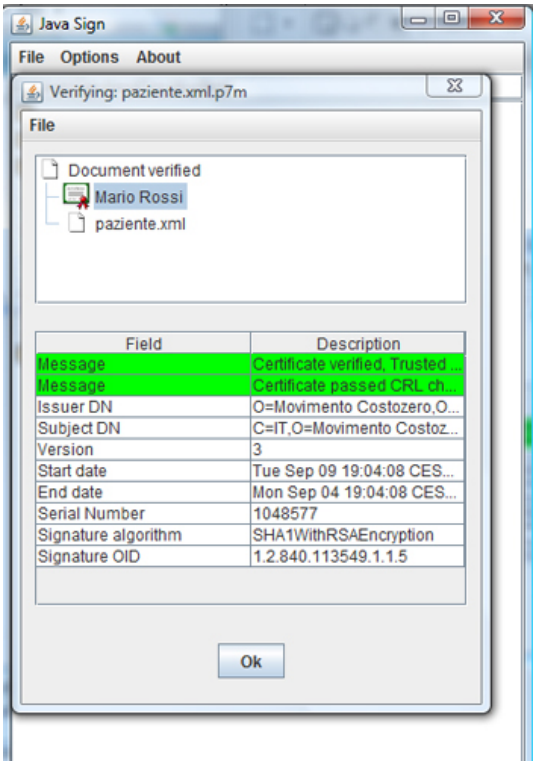

Figura 4.7: Javasign

Capitolo 5

XML-Signature Healthcare

Abbiamo già detto nei capitoli precedenti che una delle proprietà più interessanti e importanti della firma su XML riguarda la possibilità di firmare non solo tutto il documento (come accade per gli altri due formati), ma anche solo un sottoinsieme di esso. Questo apre le porte anche alla realizzazione di applicazioni che permettano di far firmare parti differenti del documento da persone diverse (e quindi nel nostro problema da dottori differenti) in funzione delle specifiche responsabilità. Nell'assistenza sanitaria le molteplici applicazioni devono preservare la privacy e la sicurezza delle informazioni trasmesse; inoltre diverso personale deve poter avere privilegi di accesso differenti alle informazioni del paziente. L'utilizzo di questi standard XML-Security e di queste tecnologie, mira come abbiamo già detto alla riduzione del materiale cartaceo, al tentativo di diminuire le liste d'attesa, al miglioramento in generale di servizi sanitari e funzioni amministrative con nuovi metodi più veloci e affidabili.
XMLDSIG attualmente è ampiamente supportato da moltissimi Framework di sviluppo e linguaggi di programmazione. In questo capitolo descriveremo la procedura di firma e di verifica e verrà posto un caso di studio di firma utilizzando java e le API XML che riguardano XML-Security rilasciate dal progetto Apache XML Security. Nell'Appendice B si trovano alcuni script di esempio in .NET e PHP per firmare o verificare un documento XML.

5.1 Creazione della firma XML

Ora passeremo a descrivere una procedura per creare una firma XML a partire da un qualunque tipo di file:

- Innanzitutto si decide quale URI firmare (“paziente.xml”, “paziente.xml#pressione” ...) e quindi anche che modalità di imbustamento utilizzare.

- Calcolare il digest dell'oggetto che andrà a far parte dell'elemento <Reference> nei campi DigestMethod e DigestValue con aggiunta delle eventuali trasformazioni eseguite in precedenza sul documento.

- Inserire nel tag <SignedInfo> gli elementi Reference, uno per ogni risorsa da firmare, a cui si aggiunge anche il CanonicalizationMethod e il SignatureMethod.

- Firmare effettivamente il documento, calcolando il digest dell'elemento <SignedInfo> appena creato secondo il SignatureMethod usato in modo da ottenere così il SignatureValue.

- Aggiungere se necessario le informazioni sulle chiavi e sui certificati usati dentro a <KeyInfo>.

- Infine creare la Signature che racchiude tutti questi elementi.

5.2 Verifica della firma XML

Per verificare una firma, principalmente il ricevente deve verificare che tutti i Reference generino lo stesso digest per la stesso documento, utilizzando gli URI per localizzare la risorsa e le informazioni sull'algoritmo di creazione. Se il documento non ha subito modifiche successive alla firma il digest generato sarà uguale a quello riportato in <Reference>. Più nel dettaglio questa procedura si compone di due fasi:

- Verificare la firma applicata all'elemento <SignedInfo>. Per fare ciò, bisogna calcolare il digest dell'elemento <SignedInfo> (usando l'algoritmo specificato nell'elemento <SignatureMethod>) ottenendo le informazioni riguardanti le chiavi crittografiche, ed usare la chiave pubblica per verificare se il valore dell'elemento <SignatureValue> è corretto.

- Per ogni elemento Reference all'interno di <SignedInfo> è necessario applicare al documento richiamato dall'URI le eventuali trasformazioni presenti e il DigestMethod per calcolarne l'hash ed infine paragonarlo con i valori contenuti nei rispettivi tag <DigestValue>.

5.3 Java XML Digital Signature

Sun Microsystems fornisce uno standard di Java API[20] per firmare e verificare documenti XML e binari, definito JSR 105. Questo standard è contenuto nel Java Web Services Developer Pack e a partire dalla Java SE versione 6. Vengono utilizzati numerosi package come il javax.xml.crypto.dsig che include le interfacce che rappresentano gli elementi fondamentali definiti nella specifica W3C su XMLDSIG. Alcune delle interfacce importanti in questo pacchetto sono XMLSignature, SignedInfo, CanonicalizationMethod, SignatureMethod, Reference e DigestMethod, appunto le strutture di cui abbiamo parlato in questo documento. La classe XMLSignatureFactory prevista dal pacchetto javax.xml.crypto.dsig rappresenta il costruttore dell'oggetto Signature.
La classe XMLSignature rappresenta l'elemento Signature definito nello standard W3C. I metodi sign() e validate() di questa classe servono rispettivamente a proteggere e convalidare i dati. SignatureMethod rappresenta l'algoritmo di firma usato dalle operazioni sign() e validate(). La classe SignedInfo può avere più di un Reference, quindi più di un dato oggetto di firma.
I packages javax.xml.crypto.dsig.dom e javax.xml.crypto.dom contengono le classi specifiche per DOM [21] usato per navigare e modificare un documento XML.

5.3.1 Esempio di ricovero di un paziente

Verranno qui riportati frammenti di codice sulla creazione di una firma detached e una firma enveloped di un documento XML.

```
<?xml version="1.0" encoding="utf-8"?>
<PatientRecord>
    <Name>Andrea Felice</Name>
    <user>123456</user>
    <Visit date="10-08-2020">
    <BloodPressure id="BloodPressureReading">
    <MaximumPressure>80</MaximumPressure>
    <MinimumPressure>120</MinimumPressure>
    </BloodPressure>
    <Diagnosis>.....</Diagnosis>
    </Visit>
</PatientRecord>
```

Il documento XML simula una cartella clinica di un paziente in un ospedale. All'interno del campo <Visit> appare ciò che viene fatto al paziente in quella giornata. Nel caso di ulteriori esami in un'altra giornata da parte dello stesso paziente basterà aggiungere un altro tag Visit naturalmente con l'opportuna data.

Nel primo caso di firma detached mostriamo come è possibile firmare un solo campo della cartella clinica, come può essere il caso di una misurazione della pressione da parte di un'infermiera mentre nel secondo esempio la firma enveloped viene effettuata sull'intero documento (caso di una firma di accettazione da parte di un dottore) .

```
/*Creo l'istanza factory per poi inserire la firma e il documento*/
XMLSignatureFactory sig = XMLSignatureFactory.getInstance("DOM");
DigestMethod digestMethod = sig.newDigestMethod
    ("http://www.w3.org/2000/09/xmldsig#sha1", null);
/* suppongo che la prima operazione sia la misurazione della pressione
   quindi il dato trovato viene firmato da un'infermiera */
Reference ref = sig.newReference("#BloodPressureReading",digestMethod);
ArrayList refList = new ArrayList();
refList.add(ref);
```

```
CanonicalizationMethod cm =  sig.newCanonicalizationMethod(
    "http://www.w3.org/2001/10/xml-exc-c14n#",null);
SignatureMethod sm = sig.newSignatureMethod(
    "http://www.w3.org/2000/09/xmldsig#rsa-sha1",null);
/* per creare il prossimo elemento devo passare l'algoritmo di firma(sm),
   di canonicalizzazione(cm) e la lista dei reference come da standard */
SignedInfo signedInfo = sig.newSignedInfo(cm,sm,refList);
DOMSignContext signContext = null;
/* il metodo DOMSignContext accetta la chiave privata e il nodo nel
   quale viene poi inserita la signature quindi nel nostro caso
   securityHeader potrebbe essere l'elemento "PatientRecord" */
signContext = new DOMSignContext(privKey,securityHeader);
signContext.setURIDereferencer(new URIResolverImpl());
/* le tre righe seguenti riguardano l'elemento KeyInfo che nel capitolo
   3 abbiamo detto essere opzionale  */
KeyInfoFactory keyFactory = KeyInfoFactory.getInstance();
DOMStructure domKeyInfo = new DOMStructure(tokenReference);
KeyInfo keyInfo =
    keyFactory.newKeyInfo(Collections.singletonList(domKeyInfo));
/* qui viene creata la signature XML */
XMLSignature signature = sig.newXMLSignature(signedInfo,keyInfo);
/* questa ultima operazione utilizza il DOMSignContext per la firma  */
signature.sign(signContext);
```

Firma Enveloped

```
/* simile procedura con la differenza che si deve escludere l'elemento
   signature dal calcolo della firma */
XMLSignatureFactory sig = XMLSignatureFactory.getInstance("DOM");
DigestMethod digestMethod =
  sig.newDigestMethod("http://www.w3.org/2000/09/xmldsig#sha1", null);
C14NMethodParameterSpec spec = null;
CanonicalizationMethod cm = sig.newCanonicalizationMethod(
  "http://www.w3.org/2001/10/xml-exc-c14n#",spec);
SignatureMethod sm = sig.newSignatureMethod(
  "http://www.w3.org/2000/09/xmldsig#rsa-sha1",null);
ArrayList transformList = new ArrayList();
TransformParameterSpec transformSpec = null;
Transform envTransform = sig.newTransform(
  "http://www.w3.org/2001/10/xml-exc-c14n#",transformSpec);
Transform exc14nTransform = sig.newTransform(
  "http://www.w3.org/2000/09/xmldsig#enveloped-signature",transformSpec);
transformList.add(envTransform);
transformList.add(exc14nTransform);
/* quando non viene specificato l'elemento da firmare, con ""
   si intende la root del documento */
```

```
Reference ref = sig.newReference(
   "",digestMethod,transformList,null,null);
ArrayList refList = new ArrayList();
refList.add(ref);
SignedInfo signedInfo =sig.newSignedInfo(cm,sm,refList);
/* .... qui prosegue come per il caso della firma detached */
....
```

Conclusioni

La firma digitale descritta in questo documento è utile in sistemi statici amministrati centralmente e basati sull'identità (ad esempio aziende, pubbliche amministrazioni). Per riassumere brevemente, l'utente (pubblico o privato) che vuole dare valore giuridico alle proprie dichiarazioni elettroniche deve dotarsi di un opportuno kit per l'attività di firma: questa operazione consta nell'uso di un algoritmo crittografico eseguito da una macchina in seguito all'immissione di una chiave privata. Un'altra chiave è utilizzata per la verifica ed è pubblica e viene affiancata da un certificato fornito da una terza parte fidata (ente centificatore) necessario per il valore legale della firma.
Il fenomeno "firma digitale" è stato prima di tutto una sfida culturale, tuttora non è facile fare accettare a tutti questa rivoluzione. Da una ricerca dell'Osservatorio Ict di Sanità della School of management del Politecnico di Milano in collaborazione con l'Ict Institute del Politecnico e presentata a Bologna alla fiera Exposanità che si è svolta tra il 26 e il 29 maggio, è stato stimato che entro i prossimi tre anni in Italia, una struttura sanitaria su tre (precisamente il 32%) aumenterà la spesa sull'innovazione informatica di oltre il 20% rispetto al triennio scorso, a dispetto della crisi generale di questo periodo. Per quanto riguarda i servizi digitali per il cittadino è emersa la necessità di un'uniformità nell'erogazione dei servizi a livello sovraziendale e nazionale, con il forte ruolo di presidio che ogni regione dovrebbe avere. Negli ultimi anni l'utilizzo nella sanità di tecnologie come Wi-Fi,GPRS, Umts, RFId [1], ha

[1] Radio Frequency IDentification è una tecnologia per l'identificazione automatica di oggetti o persone ad esempio attraverso un codice a barre

portato all'adozione di numerose applicazioni per medici o infermieri come palmari e tablet connessi in wi-fi al sistema informativo ospedaliero. Secondo una ricerca americana [22] si stima che il mercato della tecnologia legata alla comunicazione sanitaria via smartphone, tablet e altri dispositivi avrà un'incremento del 25% all'anno fino al 2014, raggiungendo un giro d'affari di 4,6 miliardi di dollari solamente negli Stati Uniti. Cellulari e dispositivi wireless si sono diffusi nel settore sanitario per la loro maggiore funzionalità, il basso costo, l'alta affidabilità e la facilità di utilizzo. Tuttavia, in queste applicazioni, la privacy e la sicurezza delle informazioni trasmesse devono essere preservate. Proprio in questo contesto la crittografia e l'XML-Signature si pongono come soluzioni per mantenere la privacy dei pazienti. Soprattutto in questo ambiente, la riservatezza, l'autenticazione, l'integrità e il non ripudio dei dati sanitari sono essenziali per l'assistenza sanitaria mobile. Inoltre il servizio sanitario definisce livelli di specializzazioni diversi, come medico, infermiera, laboratorio, farmacia, amministratore e ogni livello dovrebbe avere accesso a differenti dati del sistema informativo centrale. La diffusione di tutte queste tecnologie descritte è tuttavia condizionata anche da processi decisionali delle pubbliche amministrazioni, e da una certa avversione al cambiamento riscontrabile spesso ad alti livelli ma anche nel personale della struttura.

Appendice A

Definizioni

In questa Appendice si trovano alcune definizioni utilizzate nel documento

Firma elettronica : L'insieme dei dati in forma elettronica, allegati oppure connessi tramite associazione logica ad altri dati elettronici, utilizzati come metodo di identificazione informatica.

Firma elettronica qualificata : La firma elettronica ottenuta attraverso una procedura informatica che garantisce la connessione univoca al firmatario, creata con mezzi sui quali il firmatario può conservare un controllo esclusivo e collegata ai dati ai quali si riferisce in modo da consentire di rilevare se i dati stessi siano stati successivamente modificati, che sia basata su un certificato qualificato e realizzata mediante un dispositivo sicuro per la creazione della firma.

Firma digitale : Particolare tipo di firma elettronica qualificata basata su un sistema di chiavi asimmetriche a coppia, una pubblica e una privata, che consente al titolare tramite la chiave privata e al destinatario tramite la chiave pubblica, rispettivamente, di rendere manifesta e di verificare la provenienza e l'integrità di un documento informatico o di un insieme di documenti informatici.

Chiave privata : L'elemento della coppia di chiavi asimmetriche, utilizzato dal soggetto titolare, mediante il quale si appone la firma digitale sul documento informatico.

Chiave pubblica : La chiave della coppia utilizzata da chiunque esegua la verifica di una firma digitale, cioè l'elemento della coppia di chiavi asimmetriche destinato ad essere reso pubblico, con il quale si verifica la firma digitale apposta sul documento informatico dal titolare delle chiavi asimmetriche.

Certificato digitale : documento elettronico dotato di una firma digitale, che permette di confermare l'associazione tra una chiave pubblica e l'identità del soggetto che sostiene di esserne il legittimo proprietario.

Appendice B

Script di esempio

Questa funzione riprende le API JAVA verificando l'autenticità di un documento firmato con le procedure del capitolo 5.

```
public boolean validate(Element signature){
   DOMValidateContext validationContext = new DOMValidateContext(new ↩
       KeySelectorImpl(), signature);
   XMLSignatureFactory signatureFactory = XMLSignatureFactory.getInstance(↩
       "DOM");
   XMLSignature signature =  signatureFactory.unmarshalXMLSignature(↩
       validationContext);
   validationContext.setURIDereferencer(new URIResolverImpl());
   boolean validMessage  = signature.validate(validationContext);
      if(validMessage){
          System.out.println("Signature Validation passed");
      }else{
          System.out.println("Signature Validation Failed");
      }
 return validMessage;
}
```

Il .NET Framework dispone di un insieme di classi per supportare XMLDSIG che sono contenute nel namespace System.Security.Cryptography.Xml (in unione con System.Security.Cryptography per la crittografia) dell'assembly System.Security. Viene mostrato un esempio che restituisce un file firmato prelevandolo tramite upload.

```
// Carico il documento XML
XmlDocument doc = new XmlDocument();
doc.Load(xmlFile.PostedFile.InputStream);

// Elimino il nodo di dichiarazione xml (<?xml..
foreach (XmlNode n in doc.ChildNodes)
  if (n.NodeType == XmlNodeType.XmlDeclaration)
    doc.RemoveChild(n);

// Creo l'oggetto per firmare
SignedXml xs = new SignedXml();

// Creo un object con la lista degli XmlNode da firmare
// in questo caso il DocumentElement
DataObject data = new DataObject();
data.Data = doc.ChildNodes;
data.Id = "MyDocId";

// Lo aggiungo alla lista di oggetti da firmare
xs.AddObject(data);

// Indico che gli algoritmi RSA devono memorizzare
// le chiavi sulla macchina
RSACryptoServiceProvider.UseMachineKeyStore = true;

// Creo un algoritmo RSA per criptare il digest
RSA key = new RSACryptoServiceProvider();

// Indico di usare questo algoritmo
xs.SigningKey = key;

// Creo una KeyInfo sulla base dell'algoritmo
// per infomare il tipo di criptazione utilizzata
// e mostrare la chiave pubblica
KeyInfo ki = new KeyInfo();
RSAKeyValue rsa = new RSAKeyValue(key);
ki.AddClause(rsa);
xs.KeyInfo = ki;
```

B Script di esempio

```
// Aggiungo la reference al mio oggetto
// con lo stesso Uri dell'Id
Reference r = new Reference();
r.Uri = "#MyDocId";
xs.AddReference(r);

// Calcolo la firma
xs.ComputeSignature();

// Mando in output l'xml risultante
Response.Clear();
Response.ContentType = "text/xml";
Response.Write(xs.GetXml().OuterXml);
Response.End();
```

Funzione in PHP che effettua la firma di un documento[23].

```
<?
.....
function sign($xmldoc, $type , $keyInfoArr =null)
  {

    $root = $xmldoc->document_element();

    $rootDs = $xmldoc->create_element (  'Signature' );
    $rootDs->set_attribute("xmlns", "http://www.w3.org/2000/09/xmldsig#");

    $root->append_child($rootDs);

    $SignedInfo = $xmldoc->create_element ( 'SignedInfo' );
    $rootDs->append_child($SignedInfo);

    $SignatureValue  = $xmldoc->create_element ( 'SignatureValue' );
    $rootDs->append_child($SignatureValue);

    $KeyInfo = $xmldoc->create_element ( 'KeyInfo' );
    $rootDs->append_child($KeyInfo);
        $KeyName = $xmldoc->create_element ( 'KeyName' );
        $KeyInfo ->append_child($KeyName);

    if (  $keyInfoArr != null  && is_array($keyInfoArr)  )
    {
        foreach( $keyInfoArr as $keyInfoEl => $keyInfoE2 ){
              $el = '';
          if ($keyInfoEl  == XMLSEC_X509DATA )
            $el = XMLSEC_X509DATA;
          if ($el == '') continue;

              $el2 = '';
          if ($keyInfoE2  == XMLSEC_X509CERTIFICATE )
            $el2 = XMLSEC_X509CERTIFICATE;
          if ($el2 == '') continue;

           $elDom = $xmldoc->create_element ( $el );
           $el2Dom = $xmldoc->create_element ( $el2 );

           $elDom ->append_child($el2Dom );
           $KeyInfo ->append_child($elDom);
        }
    }
```

```
$CanonicalizationMethod = $xmldoc->create_element (
     'CanonicalizationMethod' );
$CanonicalizationMethod -> set_attribute(
     "Algorithm", "http://www.w3.org/TR/2001/REC-xml-c14n-20010315");
$SignedInfo -> append_child($CanonicalizationMethod);

$SignatureMethod  = $xmldoc->create_element ( 'SignatureMethod' );

$algorithm = '';
if ($type == XMLSEC_DSA_SHA1 )
    $algorithm = 'http://www.w3.org/2000/09/xmldsig#'.XMLSEC_DSA_SHA1;
if ($type == XMLSEC_RSA_SHA1 )
    $algorithm = 'http://www.w3.org/2000/09/xmldsig#'.XMLSEC_RSA_SHA1;
if ( $algorithm == ''){
       $this->errorMsg = "undefinit algorithm type:  ".$type;
        return false;
}

$SignatureMethod  -> set_attribute("Algorithm", $algorithm );
$SignedInfo -> append_child($SignatureMethod );

$Reference  = $xmldoc->create_element ( 'Reference' );
$Reference  -> set_attribute("URI", "");
$SignedInfo -> append_child($Reference );


$Transforms = $xmldoc->create_element ( 'Transforms' );
$Reference -> append_child($Transforms);

$Transform = $xmldoc->create_element ( 'Transform' );
$Transform  -> set_attribute("Algorithm",
          'http://www.w3.org/2000/09/xmldsig#enveloped-signature' );
$Transforms-> append_child($Transform);

$DigestMethod  = $xmldoc->create_element ( 'DigestMethod' );

$DigestMethod  -> set_attribute("Algorithm",
          'http://www.w3.org/2000/09/xmldsig#sha1' );
$Reference -> append_child($DigestMethod );

$DigestValue  = $xmldoc->create_element ( 'DigestValue' );
$Reference -> append_child($DigestValue );

if ( !$this->setPath() ) return false;

$tmplName = $this->saveXml( $xmldoc , "in" );
if( !$tmplName)  return false;
```

```
    if( isset($this->temp_path))
          $outputName = tempnam($this->temp_path   , "out");
    else {
         $this->errorMsg = "can't define temp path";
         return false;
    }

    $keyfile = $this->xmlkeyfilename;
    $cmd = "xmlsec1 sign  --output $outputName  --keys-file
                                          $keyfile $tmplName 2>&1";
    $this->cmd = $cmd ; // for debugging


    $this->exec();
    $outDocument = file_get_contents($outputName);

    if ( $this->unlink and !unlink($outputName)){
         $this->errorMsg = "can't unlink data file ".$outputName;
         return false;
     }

    if ( $this->unlink and !unlink($tmplName)){
         $this->errorMsg = "can't unlink data file ".$tmplName;
         return false;
     }

     if ( $outDocument == '' ){
         $this->errorMsg = "the error in the crypto conversion ";
         $this->errorMsg .= "\n ".$read;
         return false;
     }

    return $outDocument;
  }
.....
?>
```

Bibliografia

[1] Centro Nazionale per l'informatica nella Pubblica Amministrazione
http://www.cnipa.gov.it

[2] Vedi "Guida alla firma digitale"
http://www.cnipa.gov.it/site/it-IT/Attività/Firma_ digitale

[3] Lista dei Certificatori Accreditati
http://www.cnipa.gov.it/site/it-IT/Attività/Firma_ digitale/Certificatori_ accreditati/

[4] Immagine presa da: *http://it.wikipedia.org/wiki/Crittografia_ asimmetrica*

[5] D.P.R. n.513 del 10 novembre 1997: criteri e modalità per l'archiviazione
e la trasmissione di documenti con strumenti informatici e telematici

[6] Software FCMT
http://www.cnipa.gov.it/site/it-it/Attività/Firma_ digitale/Il_ Certificatore/Software/

[7] D.P.R. 445/2000 art. 23 comma 2

[8] D.P.R. 445/2000 art. 25 comma 1 e comma 2

[9] Anche chiamato XMLDSIG http://www.w3.org/TR/xmldsig-core/

[10] Dal sito *http://www.tecnes.com/tecnologie/Firma+digitale+XML.html*

[11] (Extensible Markup Language) XML-Signature Syntax and Processing
http://www.ietf.org/rfc/rfc3275.txt

[12] "Regole tecniche per la definizione del profilo di busta crittografica per la firma digitale in linguaggio XML" Deliberazione n.34/2006 del 18 maggio 2006

[13] Vedi sito di InfoCert: *https://www.firma.infocert.it/*

[14] "XML Advanced Electronic Signatures"
Specifica ETSI TS 101 903 V1.2.2 (2004-04)

[15] "Regole per il riconoscimento e la verifica del documento informatico"
Deliberazione n.4/2005 del 17 febbraio 2005

[16] Specifica che permette di criptare un documento XML o parti di esso
http://www.w3.org/TR/xmlenc-core/

[17] Vedi sito Comped: *http://www.comped.it/*

[18] Sito del progetto OpenSignature: *http://opensignature.sourceforge.net/*

[19] Informazioni sul movimento copyzero e sulle licenze copyzero X:
http://www.costozero.org/wai/licenza.html

[20] Le API di XML-Signature si possono trovare qui:
http://java.sun.com/webservices/docs/2.0/xmldsig/api/

[21] Document Object Model: *http://www.w3.org/DOM/*

[22] Articolo sullo sviluppo di tecnologie mobili nellà sanità
http://www.ihealthbeat.org/articles/2010/3/22/report-mobile-health-market-expected-to-reach-4-6b-by-2014.aspx

[23] L'esempio usa le librerie XMLSec ed è stato preso dal sito
http://www.phpclasses.org/

[24] Jonathan Katz *Digital Signatures (Advances in Information Security)*
edizione Springer 3 giugno 2010

[25] Stephen Paine, Ben Hammond, Mohan Atreya, Paul Starret, Stephen Wu *Digital Signatures* Osborne/McGraw-Hill (February 8, 2002)

[26] Donald E. Eastlake, Kitty Niles *Secure XML: The New Syntax for Signatures and Encryption* Pearson Education; 1st edition (July 19, 2002)