Consiglio Nazionale delle Ricerche

Signed Web Forms

S. Minutoli, A. Marchetti, M. Tesconi

IIT TR-17/2005

Technical report

Settembre 2005

Istituto di Informatica e Telematica

Signed Web Forms

Salvatore Minutoli[1], Andrea Marchetti[1], Maurizio Tesconi[1]
[1]CNR, IIT Department, Via Moruzzi 1, I-56124, Pisa, Italy
{salvatore.minutoli, andrea.marchetti, maurizio.tesconi}@iit.cnr.it

keywords: Web Forms, Digital Signature, Web Service, Xml, WYSIWYS (What You See Is What You Sign)

Abstract

As more and more Web applications are available on the Internet, they are becoming a standard way also for many organizations and institutions to offer their services and/or improve the efficiency of office procedures. Some of these applications require the user to input some information, typically by filling out a form, and submit the data. In many cases the user is required to digitally sign the data submitted. The problem of the digital signature has been solved with appropriate algorithms based on the use of two different keys: the private key and the public key. The private key must be known only to its legitimate owner, certified by a Certification Authority, and must be protected from unauthorized access. This problem has been solved by means of smart-cards and USB-tokens. However when the user decides to sign a document displayed on the screen, the software actually uses his private key to sign an internal representation of the document. Thus, another problem arises: the user must be sure that the document actually signed is the same document he has been shown. Since few years the WYSIWYS (What You See Is What You Sign) technology has been suggested, so that users know exactly what they sign. We propose an architecture based on this technology. The signing module is embedded in a Web Service that must be invoked to obtain the digital signature of a given document. This Web Service shows the document to the user that decides whether to sign it or not. Finally, we have tested this architecture by implementing a prototype of a Form-based Web application.

Index

Introduzione

L'affermarsi di diverse tecnologie quali Internet, Web, XML, Java, Digital Signature, ha aperto nuove possibilità per lo sviluppo di applicazioni multi piattaforma, interoperabili, di utilizzo immediato e di facile installazione e manutenzione [Marchetti1]. Risultati questi che apparivano dei traguardi insperati fino a poco tempo fa.
Molte applicazioni che utilizzano queste tecnologie gestiscono gran parte dell'interazione man-machine tramite l'uso di pagine web organizzate come moduli (form). In seguito ci riferiremo a questo tipo di applicazioni con l'attributo "Web Based Forms".
Un requisito fondamentale per molte applicazioni di questo tipo è la possibilità di autenticare le informazioni che un utente inserisce. La firma digitale realizzata tramite PKI risolve questo aspetto insieme ad altri quali il non-ripudio e l'integrità dei dati inseriti. La firma digitale costituisce una valida alternativa alla firma manuale (handwritten signature) e la cosa è stata riconosciuta a livello legale con tutta una serie di normative emesse sia a livello nazionale che europeo [EC legislation].

Proprio il riconoscimento legale che la firma digitale ha ricevuto da queste normative mette però in risalto alcuni limiti della stessa rispetto alla firma manuale. Scopo di questo progetto è quello di definire meglio queste debolezze insite nella firma digitale, ma molto più spesso nelle sue applicazioni, e di proporre una soluzione.

Applicazioni Web basate su Form

Molte delle applicazioni che utilizziamo già oggi sono applicazioni web, applicazioni cioè che risiedono su web e interagiscono con l'utente tramite un browser.
Una tipica interazione con queste applicazioni prevede l'inserimento di alcuni dati da parte dell'utente tramite un modulo web. In genere si tratta di riempire delle pagine html contenenti Forms. In diversi casi è necessario che queste informazioni siano firmate dall'utente.
Per autenticare e garantire l'integrità dei dati inseriti la soluzione ad oggi più adeguata consiste nel firmare i dati inseriti con una chiave segreta dell'utente. Il calcolo della firma richiede l'esecuzione di alcuni programmi che in base alla chiave segreta dell'utente e del documento da firmare, generano la firma digitale del documento. Un requisito fondamentale delle applicazioni di firma digitale è che la chiave segreta di un utente sia nota solo all'utente stesso e non possa essere utilizzata da nessun'altra persona o software.
Normalmente le applicazioni web per ovviare alle differenze di capacità computazionale dei client e alle disomogeneità dei browser preferiscono spostare tutte le attività di calcolo sul lato server. In questo caso ciò comporterebbe il transito della chiave privata su internet ed in ogni caso la copia di questa ultima sulla piattaforma server. Questa politica non può quindi essere applicata in questo contesto.
Molti dei browser oggi sul mercato mettono a disposizione la funzionalità di firmare form html. Il limite di questa soluzione è che si perde completamente il controllo di quello che viene firmato. In altre parole l'utente si deve fidare del modulo di firma contenuto nel browser. Problemi sorgono anche da un punto di vista dello sviluppatore di applicazioni

web che si trova costretto ad accettare le impostazioni del browser senza poter intervenire direttamente sull'oggetto della firma[1].

Nella realizzazione di programmi per la firma digitale per tali applicazioni bisogna tenere conto di diverse problematiche che verranno discusse nei paragrafi successivi.

Sicurezza della chiave privata

La firma digitale viene implementata con la crittografia a chiave pubblica che prevede l'utilizzo, da parte dell'utente che vuole apporre la firma digitale, di una coppia di chiavi: la chiave pubblica e la chiave privata. Quando una persona genera una coppia di chiavi deve mantenere segreta quella privata e diffondere quella pubblica.
Quando delle informazioni sono firmate con una chiave privata solo la corrispondente chiave pubblica può essere usata per verificare che quelle informazioni siano state effettivamente firmate dal possessore della coppia di chiavi e che non siano state modificate.
Un altro aspetto importante riguarda l'autenticazione dell'autore di una firma**:** chi riceve un documento firmato deve essere sicuro che la chiave privata utilizzata appartenga effettivamente a chi sostiene di averlo firmato. Il problema dell'associazione tra chiave pubblica (e di conseguenza della relativa chiave privata) e autore della firma è basata sull'uso di infrastrutture PKI (Public Key Infrastructure). Una chiave privata viene riconosciuta appartenente ad una determinata persona solo se la relativa chiave pubblica è stata certificata da una Certification Authority ritenuta affidabile. La Certification Authority (CA) al momento della richiesta di un certificato per la chiave pubblica verifica l'identità del richiedente e si assicura che questo sia in possesso della corrispondente chiave privata. Se queste verifiche vanno a buon fine la CA rilascia un certificato contenente le generalità del possessore delle chiavi e la chiave pubblica, il tutto con la firma digitale della CA stessa.
Un'altra informazione contenuta nel certificato è il periodo di tempo in cui il certificato stesso è valido. In genere si consiglia di utilizzare un periodo di tempo di validità non troppo esteso. Infatti ogni volta che viene usata una chiave, viene generato un testo cifrato. Usando la stessa chiave più volte, si permette ad un eventuale avversario di reperire molti testi cifrati con la stessa chiave. Alcuni metodi di crittoanalisi usano i testi cifrati per decifrare i messaggi. La conseguenza è che più tempo si usa la stessa chiave, più testi cifrati (con una stessa chiave) ci sono in giro e più è facile decifrarli tramite la crittoanalisi.
Ovviamente chi riceve un documento firmato dovrà verificare, oltre alla validità della firma digitale, anche la validità del certificato ed il fatto che esso sia stato rilasciato da una CA di fiducia di chi esegue la verifica.
In base a queste specifiche per la firma digitale di un documento valgono le seguenti proprietà:

- Autenticità dell'autore della firma
- Integrità del documento
- Non ripudiabilità

[1] Un browser vede e firma i dati contenuti in un form web come una lista piatta di coppie nome,valore.

Queste proprietà si basano ovviamente sul fatto che la chiave privata sia a conoscenza esclusivamente del legittimo proprietario, da chi cioè ha richiesto il rilascio del relativo certificato.
Il punto critico di questa implementazione della firma digitale risiede nella segretezza della chiave privata. Chiunque riuscisse a venire in possesso di tale chiave potrebbe firmare qualsiasi documento a nome del legittimo proprietario della chiave stessa. Per questo motivo si è cercato di sfruttare le tecnologie hw/sw che si rendevano via via disponibili per rendere più sicura la chiave privata.
Quando inizialmente furono disponibili i primi software per la firma digitale la chiave privata doveva essere memorizzata su un hard-disk – soluzione poco sicura specialmente se il computer è accessibile da più persone – o su un floppy-disk da inserire solo al momento di effettuare una firma. Una prima forma di protezione delle chiavi consisteva nel poter accedere alle chiavi solo mediante una password. Tuttavia rimane il problema che una volta che un programma ha accesso alla chiave privata, l'utente non ha la certezza di quello che può essere firmato con quella chiave, se il programma stesso memorizzi la chiave per utilizzarla successivamente con altri documenti o se un altro programma in esecuzione in quel momento non riesca ad accedere alla memoria del programma di firma per leggere la chiave. Per ovviare a questi inconvenienti si è cercato quindi di isolare la chiave segreta in un sistema autonomo, cioè con capacità di calcolo autonome, che si collega al sistema da cui riceve il documento da firmare tramite una connessione sicura e restituisce la firma digitale del documento evitando di rendere direttamente accessibile la chiave privata [Scheibelhofer1].
Oggi esistono sul mercato dispositivi (smart card, token USB) dotati di microchip che oltre a poter contenere la chiave segreta dispongono del supporto hw/fw/sw necessario per calcolare la firma digitale di documenti o di campi hash che gli vengono passati. Si collegano ai PC tramite appositi lettori ed eseguono la firma solo se abilitati dal possessore del dispositivo tramite l'inserimento di un PIN o di atri sistemi di autenticazione basati su valori biometrici.
Questi dispositivi effettuano autonomamente le operazioni di crittografia a chiave pubblica, sono previsti meccanismi di protezione della modifica hardware ("tamper proof"), sono previsti meccanismi di protezione della chiave privata: la chiave privata non lascia mai la smart card, né sono disponibili funzioni per estrarla.
La smart card è simile, per forma e dimensioni, ad una tradizionale carta di credito (vedi figura 1)

figura 1

A differenza di questa ultima, incorpora un processore in grado di memorizzare dati ed informazioni ed effettuare calcoli crittografici; ha il compito di conservare in modo

protetto le chiavi private e generare al proprio interno le firme digitali. L'accesso a tali operazioni è regolato da un codice di sicurezza riservato e personale (PIN).
La smart card si collega con il computer mediante un apposito lettore ed il relativo software di interfaccia (vedi figura 2).

figura 2

Come detto in precedenza la smart card espleta più funzioni tra cui:

- custodire la chiave privata necessaria per apporre la firma digitale con valore legale ed eseguire i calcoli crittografici connessi con questa operazione;
- custodire la ulteriore chiave privata usata per decifrare documenti ed eseguire i calcoli crittografici connessi con questa operazione.

Questa seconda funzione può essere utilizzata per poter ricevere documenti riservati da canali di comunicazione non sicuri (es. rete Internet).
La smart card utilizzata per la firma digitale soddisfa una serie di caratteristiche di sicurezza tra le quali vi è la capacità di resistere, qualora cada in mani estranee, ai tentativi di estrarre da essa le chiavi private in essa custodite.
Comunque è sempre necessario utilizzarla con estrema attenzione, in quanto un suo uso imprudente ne vanifica le caratteristiche di sicurezza. Inoltre va considerato che anche lo strumento più sicuro è poco efficace se non viene adoperato correttamente: "la sicurezza è un processo, non un prodotto".
La smart card è lo strumento che consente l'attestazione dell'identità elettronica dell'utente in termini di firma digitale.

Si ribadisce che il sistema di firma digitale genera firme dell'utente titolare proprio attraverso tale carta elettronica, per cui se un malintenzionato riuscisse ad operare con tale smart card, questi emetterebbe firme digitali con valore legale al posto del legittimo titolare. Tale evento viene tuttavia scongiurato per il fatto che la smart card può essere adoperata solo fornendo un codice di sicurezza riservato e personale detto PIN . Il PIN di

conseguenza deve essere conosciuto solo dal legittimo titolare. Questo è un aspetto da tenere in considerazione anche perché in alcuni casi le smart card personali, ed il relativo PIN, vengono affidati a terzi (segretaria, commercialista…), o perché smart card incustodite sono soggette a possibili sottrazioni (anche temporanee). Anche il PIN può essere sottratto: fogliettini, magari custoditi insieme alle smart card, durante la digitazione attraverso software o hardware di "snooping".

Dispositivi di questo tipo soddisfano i requisiti imposti dalla legge Italiana secondo la quale "la generazione della coppia di chiavi deve essere effettuata mediante apparati e procedure che assicurino, in rapporto allo stato delle conoscenze scientifiche e tecnologiche, l'unicità e la robustezza della coppia generata, nonché la segretezza della chiave privata".

Rimane il problema che al momento della digitazione del PIN e della conferma, l'utente deve decidere se firmare o no un documento in base a quello che vede sullo schermo, senza avere un controllo effettivo su cosa verrà poi inviato alla smart card.

Un problema da affrontare al momento del progetto di una applicazione Web è che al momento questi dispositivi sono poco diffusi e fino a quando non verranno considerati parte integrante della struttura di base di un personal computer alla stessa stregua di altri dispositivi di I/O difficilmente potranno essere proposti per applicazioni Web basate su Form.

Molte soluzioni iniziali tuttora presenti prevedono l'uso di applet firmate [Marchetti1] [Secude] [RSA] che vengono scaricate dal server centrale o di plug-in [Secure] pre-installati che accedono alla chiave privata dell'utente tramite l'inserimento di un PIN e firmano in locale il contenuto del form quindi spediscono indietro la firma digitale insieme al contenuto del modulo. Queste soluzioni sono più deboli della precedente in quanto il software di firma non viene scelto dall'utente e in quanto il canale di comunicazione con la chiave segreta non è sicuro.

L'uso della chiave privata da parte del software di firma avviene sulla fiducia (TRUST) che il software non faccia un uso improprio della chiave stessa (copia della chiave per eseguirvi altre azioni ignote al detentore della chiave). In particolare la chiave segreta utilizzata per la firma digitale non deve essere copiata.
Normalmente quando ci si ritrova di fronte ad un programma che richiede la nostra chiave segreta per firmare, abbiamo solo due alternative: fidarsi (TRUST) del programma o rinunciare alla firma. Sebbene il discorso di fiducia non si può abbandonare del tutto, possiamo comunque cercare di restringere il più possibile l'estensione di questa fiducia.

Cosa si firma

Con la firma analogica o manuale l'autore della firma sa cosa firmare in base a quello che vede [Scheibelhofer1], [Scheibelhofer2], [Scheibelhofer3], [Josang]. L'azione del firmare consiste nell'apporre un proprio segno univoco su di un documento, da un punto di vista

legale corrisponde ad accettare (commit) il contenuto visivo del documento. L'uso della doppia copia del documento, scambiato tra i contraenti di un contratto così come la deposizione di un atto firmato presso un notaio, serve a impedire tentativi di modifica del documento. Ma tutto si basa sulle capacità di riconoscimento visive umane. Con un documento elettronico le cose cambiano. Risulta difficile stabilire cosa si deve firmare. Se ci si deve limitare al solo contenuto di un documento o anche alla sua struttura o se si devono considerare anche le informazioni di formattazione (grassetto, corsivo, indentazione) a cui si attribuiscono significati semantici (un numero intercalato con barre e posto in basso a sinistra rappresenta la data di emissione del documento). Un documento elettronico è essenzialmente una sequenza di bit logici [Josang] e la sua visualizzazione, dipende dal supporto hw/sw necessario. Ogni tentativo di autenticazione non può prescindere dalla sicurezza di tale supporto.

Il secondo problema, quello di cosa effettivamente si firma quando si utilizza la chiave privata, ha ricevuto minor attenzione perché forse risulta di difficile definizione. Esiste comunque in bibliografia tutta una serie di pubblicazioni così come anche dei prodotti software che vanno sotto il nome di sistemi WYSIWYS, "What You See Is What You Sign". La frase riassume il problema affrontato. L'utente che firma manualmente un documento cartaceo esprime un consenso in base a quello che vede. Analogamente, si vuole consentire che l'utente che appone la propria firma digitale su di un documento elettronico esprima il consenso su quello che vede e solo su quello.

Alcune proposte software come [Utimaco] suggeriscono che la firma dell'utente avvenga direttamente sull'immagine del documento. L'assenso dell'utente avviene visionando una immagine raster (GIF, JPEG) ottenuta a partire dal documento elettronico e quello che viene firmato è proprio l'immagine. Questa soluzione garantisce in una certa misura che l'oggetto firmato non contenga dei dati non visibili. Ha però lo svantaggio che il prodotto firmato è una immagine che risulta non più elaborabile e quindi non adatta a tutte quelle applicazioni che richiedono delle catene di firme come i document workflow.
Scheibelhofer [Scheibelhofer1] propone di separare, tramite l'uso di documenti XML il contenuto del documento dalla sua rappresentazione, la quale dipenderà sia dalle caratteristiche della stazione client (PC, mobile device, …) sia dai requisiti dell'utente (accessibilità). In questo modo l'utente firma il solo contenuto del documento e un minimo di struttura, mentre il rendering del documento su cui si esprime il consenso avviene tramite l'uso di stylesheet XSLT. Rimane comunque il problema che anche lo stylesheet deve essere firmato e deve viaggiare insieme al documento.

Implementazioni non corrette

In questi ultimi anni sono state riscontrate diverse vulnerabilità in alcuni software di firma digitale di enti certificatori italiani. Mostriamo alcuni esempi.

Un problema riguarda la possibilità di creare un certificato falso con qualsiasi nome, nell'esempio che segue [InterLex2] è stato usato Arsène Lupin (il celebre ladro inventato da Maurice Leblanc nel 1905), e di far verificare una firma apposta con la corrispondente chiave privata al software Firma&Cifra.

In fase di verifica di una firma, secondo la normativa Italiana, dopo aver accertato l'integrità del documento, il software deve verificare che il certificato del sottoscrittore sia effettivamente stato firmato da un certificatore iscritto nell'elenco dei certificatori accreditati. A questo fine deve fare riferimento ad un'area protetta in cui sono contenuti tutti, e solo, i certificati "root" dei certificatori accreditati, depositati presso l'AIPA.
Il "baco" del software Firma&Cifra si verifica proprio a questo livello: se nella struttura PKCS#7 non si include soltanto il certificato del sottoscrittore, ma anche il certificato root utilizzato per firmare il certificato del sottoscrittore, il software non va a cercare quello ufficiale, ma si "accontenta" di quello fornito e decreta che la firma è autentica.
Quindi, per far credere ad un utente di Firma&Cifra che un documento è stato firmato da qualcun altro è sufficiente:

1. Generare un falso certificato Root inserendovi una denominazione uguale a quella di uno dei certificatori dell'elenco pubblico (nel nostro esempio PosteCom)
2. Usare questo certificato per firmare un falso certificato di firma digitale intestato al soggetto che si vuole impersonare (nell'esempio Arsène Lupin)
3. Usare il falso certificato di Arsène Lupin per firmare il documento
4. Aggiungere al documento firmato il falso certificato di Postecom.

Il risultato della verifica eseguita da "Firma&Cifra 1.0" sul documento è mostrato in figura 3

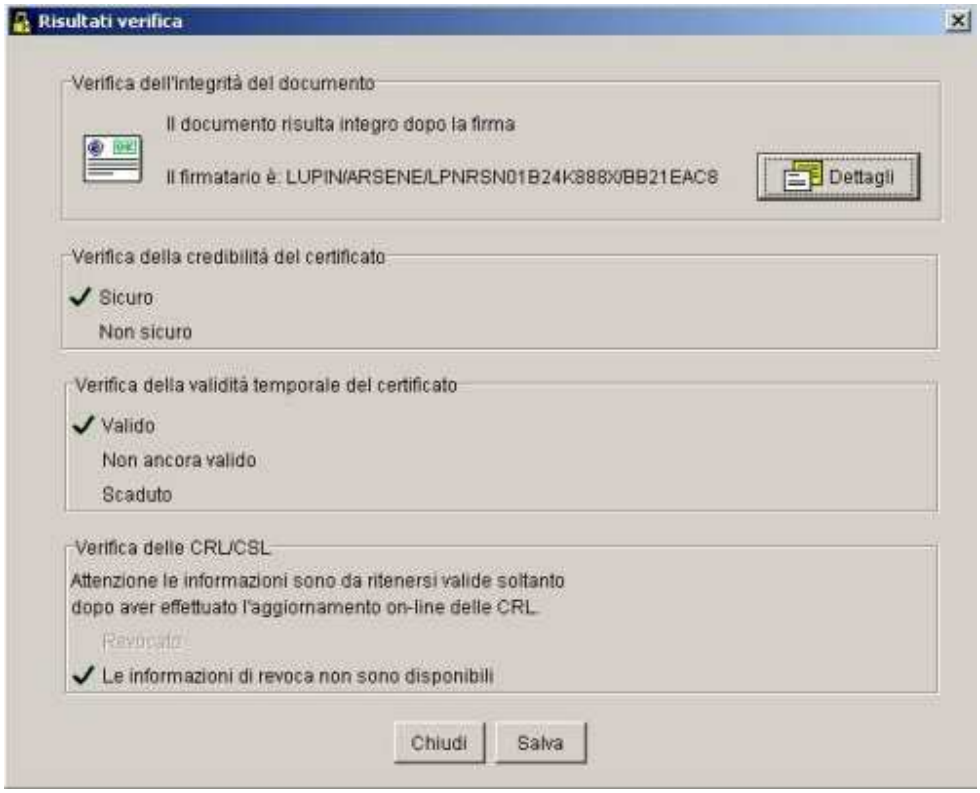


figura 3

In figura 4 si vede la finestra che mostra i dati generali del certificato: è stato rilasciato da POSTECOM al signor LUPIN ARSENE di Parigi... dopo averlo "identificato con certezza", come prescrivono le norme.

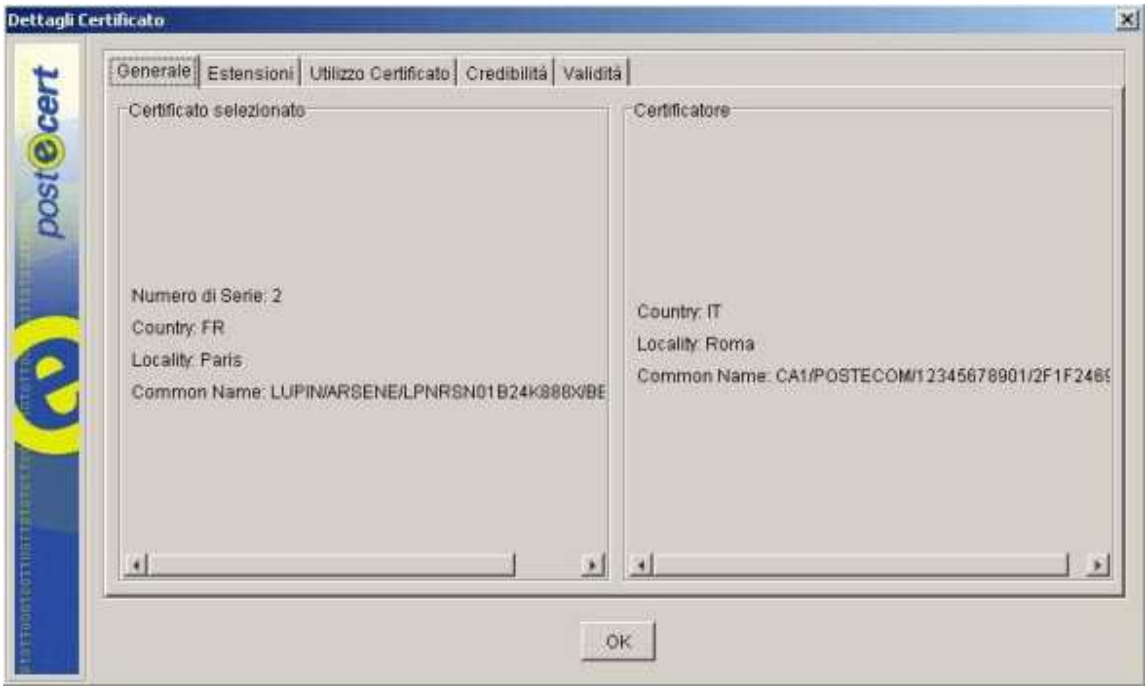


figura 4

La figura 5 mostra cosa si vede controllando il documento con un altro software (in questo caso DigitalSign):

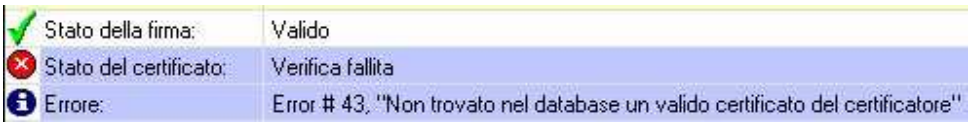


figura 5

Nell'esempio che segue mostreremo un caso in cui si evidenzia la necessità di affrontare in modo rigoroso il problema della corrispondenza tra ciò che viene visualizzato, su video o su carta, e ciò che viene effettivamente firmato. Si tratta di un caso in cui viene creato un documento viene creato e, dopo averlo visionato, firmato. Successivamente il documento stesso sarà verificato con successo, ma il contenuto apparirà diverso da come era stato scritto [InterLex1].

Tale comportamento capita se si firma, un documento scritto in MS Word contenente campi dinamici. Nel caso che illustreremo è stato utilizzato il software Dike di InfoCamere, tuttavia è molto probabile che lo stesso problema si possa presentare con altri software di firma digitale. Nei documenti Word è possibile inserire campi "dinamici", cioè campi che possono essere configurati per aggiornarsi automaticamente all'apertura del documento e il cui valore è impostato in apposite variabili all'interno di Word o del sistema operativo del computer in uso. Esempi di campi sono la data, l'ora, il nome del documento, l'autore. Se si usano alcuni di questi campi in un documento Word, il destinatario verificherà correttamente la firma, ma il contenuto dei campi potrà cambiare in funzione delle variabili d'ambiente presenti sul suo computer. In questo modo è possibile far firmare a qualcuno un documento Word con qualsiasi strumento di firma digitale e fare sì che in seguito il testo venga modificato in un modo arbitrario, senza che lo strumento di verifica della firma noti alcunché di anomalo. Ad esempio un campo "{DataCorrente}" fa sì che il documento, ogni volta che viene aperto, mostri nell'intestazione la data del giorno attuale: da un certo punto di vista (es. dell'utente che vede il documento sullo schermo o che lo stampa) dunque il documento è stato modificato, nel senso che aprendolo evidentemente non appare più uguale a com'era ad esempio ieri; ma da un altro (sequenza di bit in memoria) non è stato modificato, in quanto ovviamente i bit del file sono uguali a com'erano ieri!

Vediamo adesso cosa accade se si firma un file con campi dinamici. L'utente che crea il documento vede sullo schermo il valore dei campi dinamici, ad esempio vede "20/01/2005" e non un'indicazione del tipo "{DataCorrente}". Al momento della firma comunque il file conterrà "{DataCorrente}", quindi sarà questa la sequenza di caratteri che contribuirà a determinare l'hash e la firma digitale. Se il giorno dopo qualcuno tenterà di verificare la correttezza della firma, il software di verifica constaterà che la firma digitale corrisponde effettivamente al file. Al momento della visualizzazione però l'utente vedrà sul video "21/01/2005", cioè l'interpretazione attuale di "{DataCorrente}".

Nelle figure seguenti viene mostrata una sequenza di quanto descritto prima.

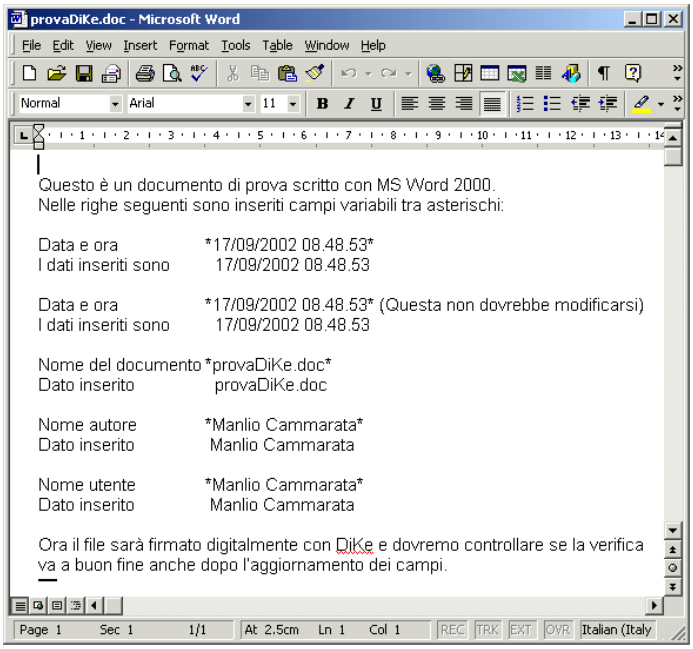


Figura 6. Questo è il documento originale preparato con Word 2000

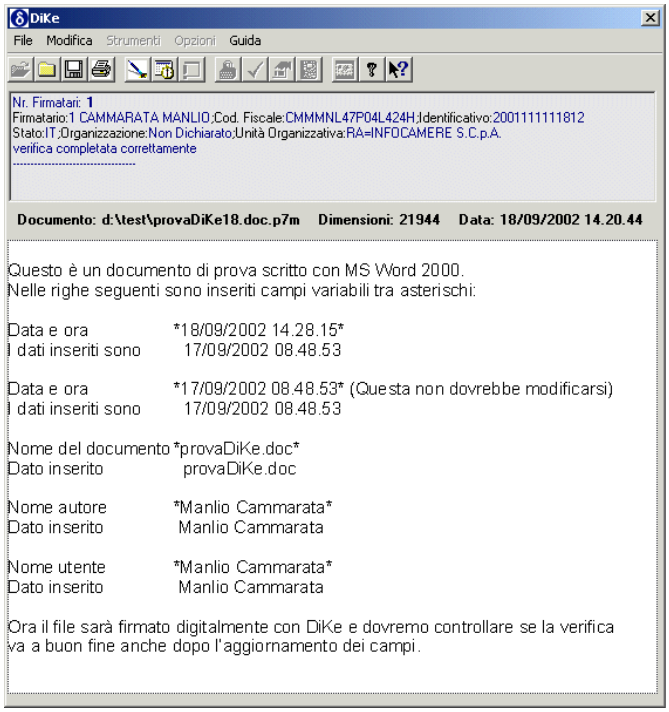


Figura 7. DiKe: sono cambiate la data e l'ora, ma il software non se ne accorge

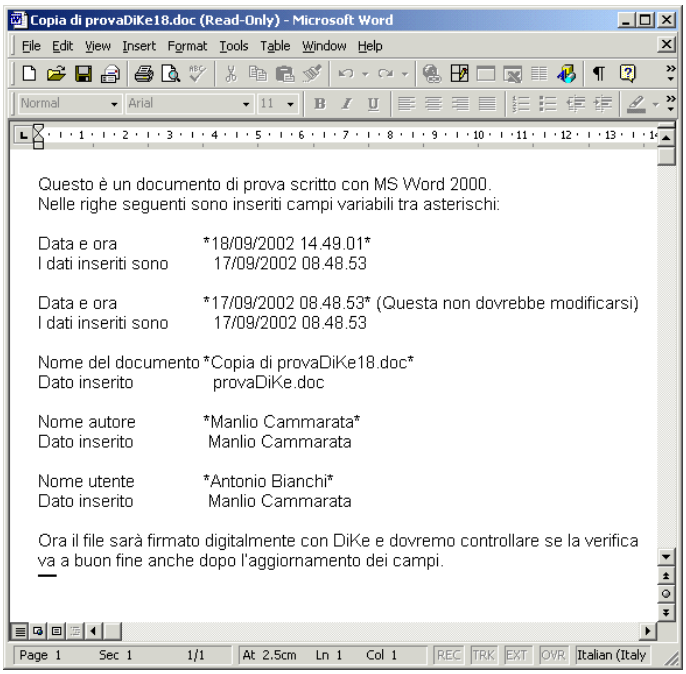


Figura 8. In Word sono cambiati la data, l'ora, il titolo del file e anche il nome dell'utente

Questo apparente paradosso scombussola i principi del diritto su cui si poggia l'istituto della firma digitale, i quali prevedono che il documento sia rappresentazione immutabile di atti, dati o fatti giuridicamente rilevanti; laddove l'immutabilità si riferisce ovviamente al contenuto informativo del documento ("quello che vedo sullo schermo") e non alla sua struttura fisica ("i bit registrati sul disco"). L'equivoco semantico porta ad un dilemma assai spinoso: firmare digitalmente un documento contenente campi variabili, come lo sono i file di Word, è come chiedere al notaio di autenticare un documento scritto a matita, ossia assurdo.

Proprio per questo motivo dal punto di vista della legge Italiana la firma digitale su documenti in formato Word (e altri formati che contengono campi dinamici, macro, etc.) non ha valore legale.

Il problema comunque non è affatto da sottovalutare, visto che Microsoft Word è uno dei programmi di videoscrittura più diffusi e che la maggior parte dei suoi utenti ha ben poca dimestichezza con i vari formati di memorizzazione, al punto che molti non conoscono neppure la differenza tra un file di word e un file di testo, ritenendo, in buona fede, che per tale si intenda un qualsiasi documento realizzato con un programma di videoscrittura.

Peraltro, non appare inutile sottolineare che anche l'uso di campi dinamici è molto diffuso, perché consente, in breve tempo, di compilare modelli e documenti in genere, proprio grazie agli automatismi che il software permette di utilizzare. Inoltre tali informazioni modificate possono cambiare nel tempo. E' infatti sufficiente cambiare le

impostazioni in Word per avere gli stessi campi modificati, la prossima volta che si apre il documento.

Software open source o proprietario

Non è mai stata chiara la ragione della scelta, operata dai certificatori, di produrre software diversi, anziché sviluppare una sinergia che conducesse, in minor tempo peraltro, alla necessaria interoperabilità richiesta dalle norme in materia. Scopo perfettamente lecito, visto che si tratta, per la maggior parte, di Società che perseguono fini di lucro, ma ciò non toglie che, in presenza di software freeware e opensource, sarebbe stato decisamente più opportuno, visto l'interesse in gioco, optare per una scelta che consentisse, quantomeno, di avere la disponibilità del codice sorgente di tali programmi, così da consentire all'utente finale di controllare, verificare, testare ed eventualmente segnalare i difetti del software prescelto.

Il vantaggio di utilizzare un software opensource risiede proprio nella maggiore sicurezza che la condivisione dei sorgenti garantisce, come ha da tempo dimostrato, anche ai più scettici, il sistema operativo Linux.

La scelta di sviluppare diversi software, oltre ad essere poco condivisibile, dal punto di vista dell'obiettivo comune di rendere compatibili i sistemi, si è rivelata critica per la sicurezza dell'intero impianto della firma digitale e rischia di trasformare in paura e in sfiducia il già evidente timore degli utenti.

What you see is what you sign

L'idea del WYSIWYS si basa sul fatto che l'applicazione presenti esattamente tutto il contenuto che l'utente deve firmare.
Nel flusso documentale tradizionale (cartaceo) i documenti di un certo rilievo vengono regolarmente firmati prima di essere spediti o archiviati. Nell'ufficio elettronico questo non avviene: i documenti circolano senza essere stati firmati. La necessità di trovare un equivalente pratico alla firma ha costituito per molto tempo un ostacolo alla piena realizzazione di un vero "ufficio senza carta". A livello governativo le problematiche legali relative alla firma digitale sono state per lo più risolte, da entrambi i lati dell'Atlantico.
Una delle sfide principali con cui deve confrontarsi una soluzione di firma digitale è il fatto che ciò che si vede a video non è che una rappresentazione istantanea del documento che si sta per firmare; peggio ancora, il documento potrebbe contenere oggetti dinamici invisibili o macro, che potrebbero determinare una variazione del contenuto del documento stesso in qualsiasi momento.

Lo stesso rischio si verifica quando un documento firmato viene verificato: ciò che si vede sul video è esattamente identico a ciò che vedeva il sottoscrittore quando ha apposto la firma? Con l'avvento delle normative nazionali sulla firma digitale, che le hanno conferito valore legale, è di fondamentale importanza garantire che vi sia un rapporto non ambiguo tra il contenuto originale firmato e ciò che viene visto dal destinatario, o utilizzato per essere elaborato.
Il visualizzatore sicuro è un componente che provvede a convertire e visualizzare i documenti sotto forma di immagini statiche, e disabilita e impedisce qualsiasi modifica automatica ai documenti da parte di un eventuale contenuto attivo o macro. Pertanto il visualizzatore sicuro deve essere progettato e certificato affinché agisca in modo tale da garantire che non vi siano differenze tra ciò che si vede e ciò che viene firmato. Il risultato è analogo a un foglio di carta firmato: una rappresentazione statica firmata del documento. Ciò consente a tutte le parti in causa di verificare il documento visualizzato e utilizzare il visualizzatore sicuro per esaminare l'esatto contenuto del documento così come lo ha determinato l'autore.
L'obiettivo di ottenere WYSIWYS non è cosi' facile realizzare nel mondo del digitale per parecchi motivi:

- I dati e la presentazione potrebbero essere separati (per esempio stylesheets)
- I dati potrebbero essere dinamici (per esempio dhtml, script e macro)

La tecnologia WYSIWYS (What You See Is What You Sign) [DigitalSign] conferisce all'operazione di firma il più elevato livello di sicurezza oggi raggiungibile. Va evidenziato che per la prima volta nella storia il sottoscrittore non ha alcuna possibilità di controllo fisico diretto sul documento che firma, ma solo attraverso gli strumenti informatici di firma digitale. I documenti informatici sono sequenze di "bit", incomprensibili all'utente e interpretabili solo attraverso il software di visualizzazione. La stessa operazione di firma è una procedura informatica (applicata ad una "impronta" del documento), anch'essa al di fuori del controllo del sottoscrittore, che associa al documento informatico una sequenza di bit, la firma digitale, e impegna il soggetto titolare. Risulta facile comprendere come sia importante disporre di un proprio software di firma digitale che garantisca un ragionevole livello di sicurezza e che ciò che firmiamo rappresenti proprio ciò che ci è stato presentato a video.

Soluzione proposta e obiettivi

L'affermazione su vasta scala della firma digitale garantirebbe il superamento dei vincoli spazio-temporali che esistono con la firma analogica, ovvero la possibilità di firmare contemporaneamente più documenti che possono risiedere in posti differenti. Le ragioni della diffidenza alla firma digitale sono sia di natura legale (bisognerebbe capire tutte le implicazioni del diritto che sono dietro all'azione di firma di un documento) che di natura psicologica, ognuno di noi si fida, a volte "ciecamente", di ciò che vede. L'idea è quella di sviluppare un modulo software che funzioni come la metafora di un occhio elettronico personale "personal electronic eye" capace di interpretare e rendere in un modo consono all'utente un documento elettronico proveniente da una qualsiasi applicazione Web e di cui possiamo fidarci come se fosse appunto un nostro senso. Infatti è assolutamente

necessario che il documento visualizzato coincida esattamente con il documento a cui verrà applicata la firma.

Le applicazioni che devono far firmare i loro documenti, comunicano con questo modulo, di cui devono essere a conoscenza, e gli passano il documento da firmare. Il modulo visualizza il contenuto del documento e, se ottiene il consenso dall'utente, restituisce la firma digitale del documento eseguita con una chiave segreta dell'utente a cui il modulo deve poter accedere.
Per garantire l'interazione tra il modulo di firma e le applicazioni web si impone la scelta di una serie di tecnologie standard da adottare sia dal punto di vista dei protocolli di comunicazione che dal punto di vista del formato dei dati. La nostra proposta si basa principalmente sull'uso di tecnologie raccomandate dal W3C:

- UNICODE come la codifica del set di caratteri utilizzato nei documenti
- XML come sintassi dei documenti
- XML-SIGNATURE per la semantica della firma digitale
- SOAP, WSDL, UDDI ovvero le tecnologie che stanno dietro i web services per l'interfaccia con cui interagire con il modulo software.

Specifica dei requisiti del web server e dell' utente

Web server

Un web server per poter usufruire del nostro servizio deve:

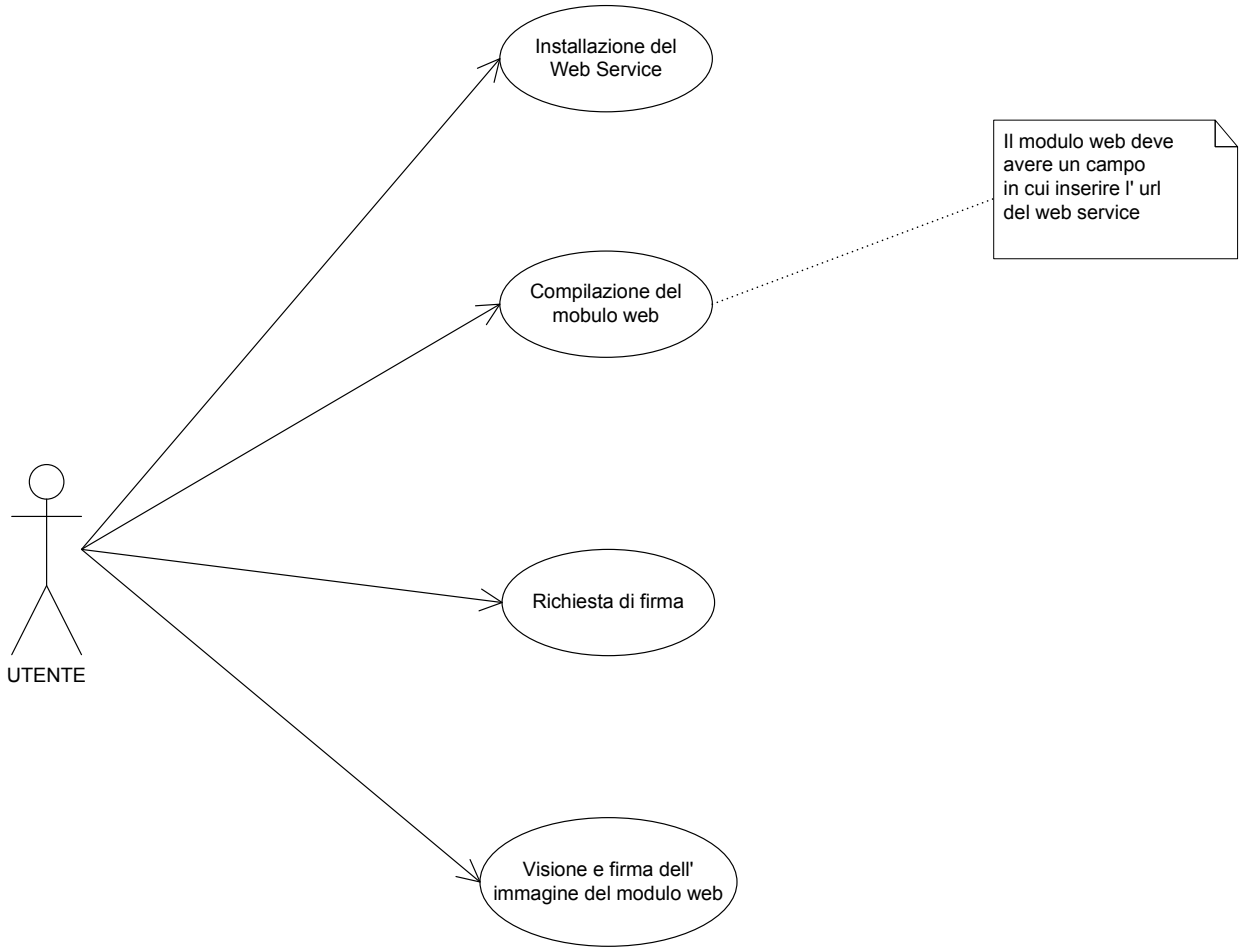
- Essere in grado di recuperare l'URL del Web Service di firma di ogni utente.
- Prevedere di dialogare con questo web service per ottenere la firma digitale, in formato XML-Signature, dei dati inseriti dall'utente. Può ad esempio possedere una applicazione server (servlet), la quale processa i valori contenuti nel modulo ricevuto producendo il file XML relativo, creando un'immagine in formato jpg o gif del modulo compilato e invocando il servizio di firma offerto dal web service presente sul lato client.

Utente

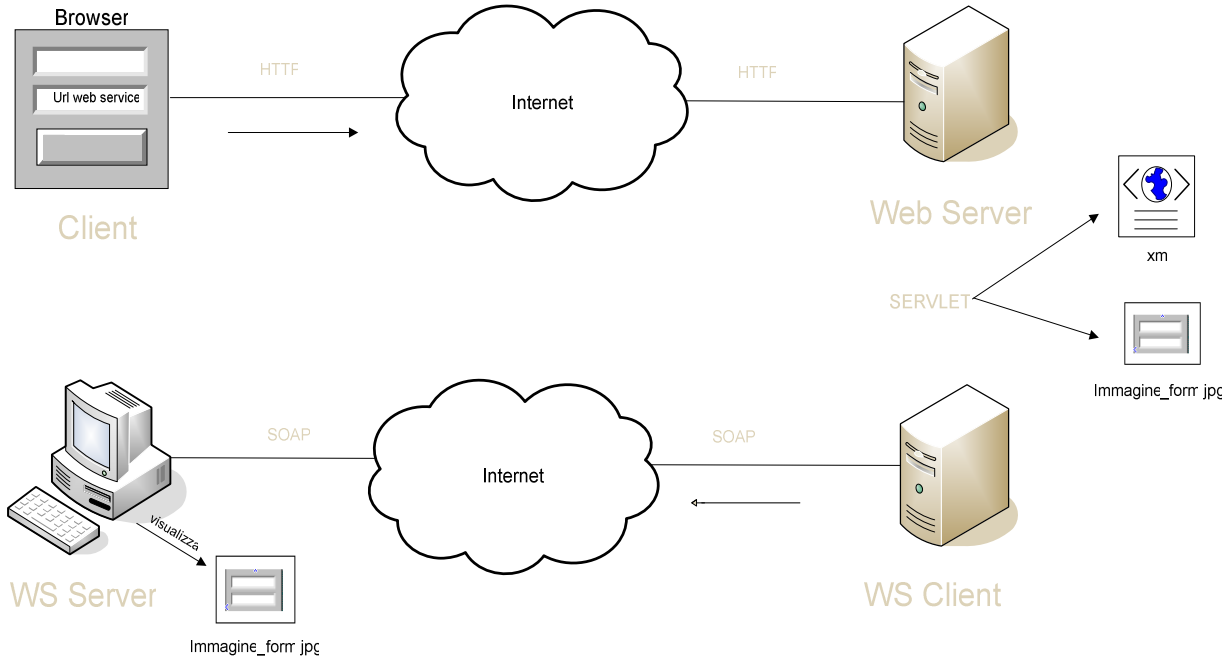
L' utente per usufruire del servizio deve:

- Possedere una coppia di chiavi (chiave pubblica e chiave privata)
- Installare in locale sulla sua macchina il web service che fornisce come servizio la firma digitale.

Nel seguente use case diagram sono rappresentate le operazioni che ogni utente deve fare per usufruire del nostro sistema:



Un web server per entrare a far parte del servizio deve possedere un'applicazione server (una servlet) che gli permetta di elaborare i moduli web compilati dagli utenti producendo il relativo file xml e una loro immagine in formato gif o jpg.
Il cliente, per poter usufruire del servizio, deve installare il web-service in locale sulla propria macchina oltre che naturalmente possedere una coppia di chiavi (chiave pubblica e chiave privata).
Dopo aver installato il web-service e ottenuta la coppia di chiavi può connettersi al web server e richiedere una pagina web.
Una volta fatta visione della pagina e riempito i vari campi la invia al server fornendogli anche l'url del Web Service presente sulla sua macchina, che gestisce il servizio di firma.
Il server elabora la pagina e attraverso la servlet invoca il servizio di firma del Web Service.
Il cliente a questo punto vede apparire sul suo monitor una immagine della pagina web che precedentemente aveva riempito.
Fatta visione può decidere se apporvi o meno la firma.
Una visione schematica del funzionamento del sistema è mostrato in figura.



Ricadute

Il superamento di tutti i vincoli spazio-temporali legati alla firma analogica, la possibilità da un punto di vista giuridico di firmare un documento, aprirebbe al possibilità di automatizzare tutta una serie di attività umane. Questo garantirebbe ad esempio la possibile affermazione di molti workflow documentali.

Possibili sviluppi

Questo occhio elettronico deve basarsi sempre sulle capacità visive umane. In pratica funziona come un trasduttore che crea un bridge tra il mondo digitale e quello analogico. Questa caratteristica di traduzione richiede comunque sempre la presenza sincrona della persona. Il passo successivo consiste nel delegare un agente software della possibilità di apporre la firma digitale di una persona. In pratica l'agente deve essere in grado di comprendere il contenuto del documento e decidere di apporre la firma in base a direttive precedentemente impostate.

Bibliografia

- [Marchetti1] Marchetti, S. Minutoli, P.Lazzareschi, and M.Martinelli. A System for Managing Documents in a Step by Step Process. In Proc. XML World Euro Edition, 26-28 March 2001, Amsterdam-Holland.
- [Scheibelhofer1] Signing XML Documents and the Concept of "What You See Is What You Sign" Karl Scheibelhofer Master's Thesis in Telematics January 2001 http://www.iaik.tu-graz.ac.at/teaching/11_diplomarbeiten/archive/scheibelhofer.pdf
- [Scheibelhofer2] What You See Is What You Sign - Trustworthy Display of XML Documents for Signing and Verification", Karl Scheibelhofer Proceedings of CMS 2001 Conference, Darmstadt, Germany, 21-22 May 2001 pp. 3-13, Kluwer Academic Publishers, ISBN 0-7923-7365-0
- [Scheibelhofer3] "What You See Is What You Sign", Karl Scheibelhofer, Proceedings of the RSA Conference 2001 Europe (Online), 15-18 October 2001, Amsterdam, Netherlands.
- [Scheibelhofer4] "eGovernment-modules for creating and verifying digital signatures", Karl Scheibelhofer , The Twelfth International World Wide Web Conference, 20-24 May 2003, Budapest, Hungary
- [Boyer] John Boyer DBLP http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/b/Boyer:John.html
- [Rundgren] Anders Rundgren http://w1.181.telia.com/~u18116613/
- [Josang] What You See is Not Always What You Sign, Audun Jøsang, Dean Povey, Anthony Ho http://security.dstc.edu.au/papers/JPH2002-AUUG.pdf
- [Secude] SECUDE DevSuite http://www.secude.com/
- [Secure] Secure Form Signing http://www.cyber.ust.hk/projects/eForm/index.html
- [Rsa] RSA E-Sign http://www.rsasecurity.com/products/bsafe/datasheets/dsesign.html
- [Utimaco] Utimaco Safeware Sign&Crypt Office WYSIWYS What You See Is What You Sign Digital Transaction Security - Marketing http://www.utimaco.de/content_pdf/wysiwys.pdf
- [AIDA] Advanced Interactive Digital Administrations (AIDA) project http://aida.infonova.at/aida.htm
- [Berbecaru] A Framework for Secure Digital Administration, Diana Berbecaru Antonio Lioy Marius Marian Proceedings of EuroWeb 2001 Conference, Pisa, Italy, 18-20 December, 2001 http://security.polito.it/diana/Articles/reprintEuroWeb2001cr_v4.pdf
- [EC legislation] EC legislation and preparatory documents with relation to e-commerce and electronic-signature http://europa.eu.int/information_society/topics/ebusiness/ecommerce/8epolicy_elaw/law_ecommerce/legal/1signatures/index_en.htm
- [InterLex1] InterLex: Diritto Tecnologia Informazione – "La firma è sicura, il documento no", http://www.interlex.it/docdigit/gelpi2.htm
- [InterLex2] InterLex: Diritto Tecnologia Informazione – "Il certificato di Arsène Lupin", http://www.interlex.it/docdigit/arsene.htm
- [DigitalSign] DigitalSign, http://www.partnerdata.it/applicazioni-firma.htm