Self-Contained Digitally Signed Documents

Approaching “What You See Is What You Sign”

Håkan Söderström
Söderström Programvaruverkstad AB
Stockholm, Sweden
hs -at- soderstrom.se

Abstract—The “what you see is what you sign” challenge has been part of digital signatures since the very start. Digital signatures apply to the bit level. Users see a higher level, so how can they know what they sign? A sample of real-life applications indicates that the issue is still open. We propose a method for improved assurance based on simple tenets. The document to be signed is a well-defined visual impression. Exactly that visual impression is signed. After signing all parties have a copy of the signed document, including its signatures. PDF makes it possible to store signatures and metadata in the document. The method is being implemented in an e-government web platform for a major Swedish city.

Keywords: digital signature, electronic signature, what you see is what you sign, forensics, e-government, PDF

1. Introduction

Digital signatures have been legally binding for over 10 years in many countries. As for Europe, an EU directive in 1999 conferred legal status to electronic signatures [2]. By 2003 it had been followed by national legislation in the member states. A similar development took place world-wide. These days digital signatures are part of an established technology, enabling e-government and many commercial uses. Millions of documents are signed digitally every year.

The problem known as *what you see is what you sign* (WYSIWYS) has haunted digital signatures from the start. A duality in the signing situation creates the problem. The signature is created on the bit level but users never see the bits, they act on visual impressions on a higher level. Given this discrepancy, how can users be sure that what they see accurately reflects the bits they sign? The problem cannot be avoided because it is inherent in the digital process.

This paper presents a method for improving the assurance that “what you see is what you sign”. It is founded on two simple principles that are a matter of course in the world of paper documents.

- The signed document is a specific visual impression. All of it is considered signed, nothing else is considered signed.
- After signing the parties take identical copies of the signed document, including its signatures. They are free to manage their copies in any way they see fit.

It is rather surprising that few signing environments support these principles. Few offer the signer access to their own signatures.

The security of digital signatures has its origin in public key cryptographic mechanisms. This is an extremely relevant research topic, but not further discussed in this paper.

The work reported here has been carried out in Sweden. Legislation applying to digital signatures may vary between countries, even within the European Union. In spite of differences, we believe there is enough commonality to make our discussion relevant to an international readership.

Legal texts in the EU use the term *electronic* signatures. We prefer *digital* because electronics is only one of several possible signature carriers.

2. Previous Work

A decade or so has passed since the EU and other countries granted digital signatures legally binding status. One of many important trends during this time is industrialization.

- Digital signatures have developed from theory to industry. A supporting infrastructure has developed on a global scale.
- Viruses and Trojans were a source of concern from the beginning [1]. Unfortunately, malware has also developed into an industry.

A Master's thesis from 2001 describes the components of a signature solution [3]. The basic principles are still valid. The main difference is that what used to be science is now an everyday reality experienced by the citizens of many countries. For instance, XML DSIG used in the thesis just mentioned has turned into an international standard [10].

Industrialization has its own side effects. Commercial pressure may compel organizations to adopt inadequate solutions [11].

The response to increasingly sophisticated malware is, among other things, smart cards where the private key never leaves the card, and dedicated card reader hardware. The track record of supposedly secure hardware was somewhat shaded by discouraging early tests [4].

Signature stripping has been identified as a specific risk for digital signatures [5]. In contrast to a paper document, a digital signature is more or less tacked onto the signed document. It may be possible to remove the signature and replace it with some other valid signature.

The center of the “what you see is what you sign” problem is the transformation from bit level to visual impression. XML on the bit level with

XSLT for a trusted transformation has been suggested as a solution [3][8]. The problem has been stated as a cumulative syntactic and semantic distance between the signer and the party requiring a signature [7]. It has been analyzed for several common formats, like XML, ASN.1, PDF, HTML [6].

A widely cited proposal for trustworthy signatures is [9]. The document to be signed is shown on a PC screen, photographed by a mobile phone, decoded by OCR software in the phone, sent back and compared to the original. The method seems unwieldy even if security level is high.

3. Current WYSIWYS Practice

This section contains samples of how “what you see is what you sign” is doing in practice. The assorted snapshots presented here are current as of 2014.

There are indications that the computer security industry is aware of the “what you see is what you sign” issue and sees it as a market with some potential. A range of commercial products are marketed under the abbreviated catchphrase “see what you sign”. One category of such products is card readers with a screen that may be used by applications [12]. The idea is that applications display text to be signed on the screen of the card reader.

As for the public sector, the “what you see is what you sign” issue is conspicuously absent from governmental instructions and directives.

The following scenarios assume a traditional personal computer with a smart card reader. Bringing in mobile devices is very relevant, but would shift the focus towards security concerns rather than “what you see is what you sign”. The messages in the scenarios are authentic, translated from Swedish.

The scenarios assume the user has previously been authenticated by a smart card and then proceeds to initiate a transaction that requires signing by a smart card.

3.1 Signing Scenario: Banking

In Swedish online banking a user typically goes through the following interactions on a pc.

- Specify the transaction, for instance a number of payments, by filling out fields in one or more web pages
- After indicating willingness to sign, the signing software takes over the screen and presents a message similar to: “`Transaction: payment. From account: xxxxx. Total amount: yyyy. Number of payments: zz. Reference number: xxxxxx. Please enter PIN code.`”

- Enter PIN code on the card reader. Return to a web page where the transaction is visible.

The user sees a condensed version of the transaction to be signed. It contains a reference number that makes the transaction traceable if the user remembers to make a note or take a screen dump. It is reasonable to say that users see what they sign. However, the user is not afforded a copy of the signed document. The main security concern is that the pc may be compromised. The text to be signed is processed locally by software running on the pc. It could possibly be affected by malware.

3.2 Signing Scenario: Tax Declaration

This example is concerned with the National Tax Agency of Sweden. The agency handles a large volume of on-line statements every year. The scenario describes signing a business VAT statement, potentially involving considerable value.

- Enter basic information, mostly amounts, typically in a handful of fields on a web page
- After indicating willingness to sign, the data entered is shown again, this time read-only.
- A click later the signing software takes over the screen and presents the message: “`You sign the information you previously checked and chose to sign and submit to the Tax Agency.`” This is the text to be signed.
- Enter PIN code on the card reader. Return to a web page containing a summary of the statement called “Receipt”. Besides the summary there is a receipt number.

In this scenario the user signs a text that is unrelated to the purpose and content of the VAT statement. Curiously the text says “You sign” and not “I sign”. The risk of malware altering the signed text is of no consequence since it is unrelated to the transaction.

A scenario like this is unthinkable in the world of paper documents. No one in his right mind would sign an agreement saying “You sign unspecified information found elsewhere”.

3.3 Signing Scenario: A Welfare Application

The authority involved in this scenario is the National Social Insurance Agency of Sweden. Like the tax agency it handles a large volume of on-line applications every year. This particular example is concerned with applying for an allowance to cover for parental leave.

- Enter basic information on one or more web pages. If this is a first-time application (the birth of a baby) it can be quite a lot.
- After indicating willingness to sign, the data entered is shown as a read-only summary.

- One more click and the signing software takes over the screen with the message: “`Sign your request!`” This is the text to be signed.
- Enter PIN code on the card reader. Return to a web page containing, among other things, a receipt number and a link to a PDF document that may be downloaded. The PDF contains the information entered previously, but some items are curiously excluded.

The user signs a text that has nothing to do with the transaction.

4. Analysis and Propositions

The real world scenarios in the previous section leave an unmistakable conclusion: The “what you see is what you sign” issue is far from settled.

From a security point of view all scenarios follow current best practice by using smart cards and dedicated card readers. As for “what you see is what you sign” there is a sharp difference. The banking example creates a summary text that links the signature to the transaction. The public sector scenarios first let users see something and then make them sign something else. The signed texts do not contain a single bit of unique information nor do they communicate the purpose of the signature. None of the examples link the signature to a single well-defined visual impression.

The scenarios have another aspect in common: The fate of the signature is obscure. The user creates a signature, but has no access to it. From the user's point of view it disappears into the unknown.

We would like to think that anything requiring a signature is a bilateral agreement. The banking case is clearly so. The bank commits to perform a transaction if the user signs the order. The public sector transactions are less clear in this respect. These agencies seem to consider the signature as a one-way obligation. The signature is treated, more or less, as an authentication.

Let us compare the scenarios with the traditional world of paper documents. A number of practices are taken for granted.

- A signature applies not only to the semantic contents of the document. The visual impression, the layout, is included and cannot be modified. This is the definition of “what you see is what you sign”.
- If anything is signed, each party takes a copy of the signed document to protect it from being modified by some other party. Each party is in possession of the signatures of all parties.

The outcome of the above analysis leads us to propose a number of tenets for implementing digital signatures.

1. Anything requiring a signature is a bilateral agreement
2. The agreement is a specific visual impression
3. The digital signature signs the visual impression

4. Each party gets a copy of the signed agreement including the signatures

5. Implementation

This section describes a digital signature implementation of an e-government application for a city or municipality. Citizens are offered to fill out and sign forms over the web. A completed form starts off a BPMN process when submitted [13]. At the time of writing the implementation is in pilot trial in the city of Malmö, Sweden.

The implementation is all open source, a natural choice for the public sector.

The following is an outline of the user experience,

- The user enters the city's web portal and is authenticated by means of a a smart card. The user now has access to “My Pages” where personal information is stored. Assume that the user picks a form, perhaps a day care application, and begins filling it out.
- The input is validated as the user types. The user is informed of missing pieces or any invalid input. The user may save a partially filled-out form on “My Pages” and pick it up later.
- After the user has completed the form it is available as a PDF link. The user may download and examine it.
- In order to formally submit the form, the user is offered to sign it digitally using the same smart card as for the authentication. A new PDF is generated where the signature is visible as an added page. The signed PDF is freely available to the user.
- A form may be signed by any number of signatories.
- The user is kept informed about the application as it progresses through the BPMN machinery.

5.1 Behind the Scenes

The preferred format for filled-out forms in this application is XML. It is converted to DocBook and then to PDF/A-1a, the target format [14][15].

Every form is assigned a short and readable document number [18]. A checksum over the document is taken with the SHA-256 algorithm.

The text to be signed by the user contains the document number and the checksum. The user has full access to the document before and after signing and may verify the checksum on many independent Internet sites.

5.2 Self-Containedness

The generated PDF document contains all information pertaining to the filled-out and signed form. Except for audit logging no information is stored outside the document. The document is easily available to all parties having

a legitimate interest in the matter. In addition to the visual impression, the following data is stored in the document.

- RDF metadata as required by the PDF/A standard [16]. It may be extracted even by non-PDF-capable software.
- The original XML form data is stored in machine-readable format
- Additional metadata to achieve traceability
- Any number of signatures

Technically the extra items are stored in page dictionaries.

5.3 The Portable Document Format (PDF)

PDF is a family of related standards. The basic idea behind PDF is to be able to reproduce the visual impression of a document on many platforms. The PDF/A format was chosen because it is intended for long-term preservation. All resources needed by the document, notably font definitions, must be embedded in the document. PDF/A-1 is an ISO standard [17].

5.4 Signature Stripping

We noted earlier that signature stripping is a risk specific to digital signatures. Various precautions prevent this from being a serious problem. The user must be authenticated to get access to the document to sign. The right to sign is only granted to the authenticated user.

The person who is to sign the document should be mentioned in the document text. It is not possible to modify the document without modifying the checksum. We also recommend the receiving agency to validate and sign incoming documents. These factors combine to a strong protection against signature stripping.

5.5 Aging

Incoming signed forms should be validated. The result of the validation should be added to the document contents. After the addition, the document should be signed by the receiving party.

Certificates expire sooner or later. The digital signatures of archived documents become invalid after some time. It is important to retain a trustworthy record of the initial validation.

5.6 PKI Quirk

There is a peculiar Swedish quirk to this story. One of the national major smart card providers does not publish their root certificate except in return for a commercial agreement. This means that the general public cannot validate their own signatures, or repudiate a signature falsely alleged to be made by them.

6. Conclusions

The “what you see is what you sign” issue has been part of digital signatures from the very start. A quick overview of important current applications shows that the issue is far from settled. Important public sector applications let users see something and then deliberately make them sign something else.

For the implementation of an e-government application we propose the restoration of some deeply rooted traditions from the world of paper documents. As a starting point we suggest that anything requiring a signature is treated as a bilateral agreement. Many current applications tend to view a signature as a single-sided commitment, similar to authentication.

We propose the use of PDF for packaging everything pertaining to a signed agreement into a single container document. The main purpose of the PDF document is to define a visual impression. It may be examined before signing and it looks the same after signing, except for the signature. There is little doubt about what has been signed. The parties keep identical copies of the signed document, including signatures. The PDF/A-1a format is chosen for its long-term preservation properties and because metadata and signatures may be stored as part of a document.

Acknowledgements

The experience reported here derives from an active e-government project. Many thanks go to the colleagues in the Motrice project for inspiration and advice in writing this paper.

References

[1] A. Weber, “See what you sign, Secure implementations of digital signatures,” in Proceedings of the International Conference on Intelligence and Services in Networks, 1998, pp. 509-520.

[2] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures.

[3] K. Scheibelhofer, “Signing XML documents and the concept of “what you see is what you sign”. Institute for Applied Information Processing and Communications, Graz University of Technology, 2001.

[4] A. Spalka, A. Cremers, H. Langweg, “The fairy tale of 'what you see is what you sign' – Trojan horse attacks on software for digital signatures,” in Proceedings of the IFIP WG, vol. 9, no. 11.7, pp. 75-86, 2001.

[5] A. McCullagh, W. Caelli, P. Little, “Signature stripping: A digital dilemma”, The Journal of Information, Law and Technology (JILT), 2001. http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2001_1/mccullagh/ accessed Jan 2014.

[6] A. Jøsang, D. Povey, A. Ho, "What you see is not always what you sign," in Proceedings of the Australian Unix User Group Symposium, AUUG2002, Melbourne, 4-6 September 2002.

[7] A. Arnellos, D. Lekkas, T. Spyrou, J. Darzentas, “A framework for the analysis of the reliability of digital signatures for secure e-commerce”. The

electronic Journal for Emerging Tools & Applications (eJETA.org), vol. 1, issue 4, Dec 2005.

[8] W. Kubbilun, S. Gajek, M. Psarros, J. Schwenk, “Trustworthy verification and visualization of multiple XML-signatures,” in Communications and multimedia security, pp. 311-320, Springer-Verlag, Heidelberg, 2005.

[9] A. Jøsang, B. AlFayyadh, “Robust WYSIWYS: a method for ensuring that what you see is what you sign,” in Proceedings of the sixth Australasian conference on Information security, vol. 81, pp. 53-58, Australian Computer Society, Inc., 2008.

[10] World Wide Web Consortium (W3C), “XML signature syntax and processing (Second Edition), http://www.w3.org/TR/xmldsig-core/.

[11] A. Jøsang, “Trust extortion on the Internet,” in Security and Trust Management, Springer Berlin Heidelberg, pp. 6-21, 2012.

[12] VASCO Data Security International, Inc, “DIGIPASS 870”, www.vasco.com, accessed 2014-02-04.

[13] Object Management Group: “Business Process Model and Notation (BPMN), Version 2.0,” http://www.omg.org/spec/BPMN/2.0.

[14] N. Walsh, “DocBook 5: The Definitive Guide,” O'Reilly 2010.

[15] The Apache XML Graphics Project, “Apache FOP, a Java-based XSL-FO (formatting objects) processor”, http://xmlgraphics.apache.org/fop/

[16] World Wide Web Consortium, “Resource Description Framework (RDF),”

[17] ISO, “ISO 19005-1:2005, Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1),”

[18] D. Crockford, “Base32 Encoding,” http://www.crockford.com/wrmg/base32.html, accessed February 2014.

