Robust WYSIWYS: A Method for Ensuring that What You See Is What You Sign

Audun Jøsang[1] **Bander AlFayyadh**[2]

[1]UniK, University Graduate Center, Kjeller, Norway
Email: josang@unik.no

[2]Information Security Institute, QUT, Australia
Email: b.alfayyadh@isi.qut.edu.au

Abstract

The security of digital signatures depends not only on the cryptographic strength of the digital signature algorithms used, but also on the integrity of the platform on which the digital signature application is running. Breach of platform integrity due to unintentional or intentional malfunctioning has the potential of wrongly imposing liability on, or wrongly taking liability away from signing parties. This problem is amplified by the fact that digital signatures may be generated on platforms that are not under the control of the signing party, and that there can be strong financial incentives for trying to manipulate the systems used for digital signatures. In practice it is extremely difficult to assess the integrity of a general purpose computing platform, so that digital signing on such platforms in principle is untrustworthy. This paper describes a method for robust WYSIWYS (What You See Is What You Sign) that ensures the integrity of digital documents and their digital signatures. This method can only be directly applied to documents written with traditional ASCII characters. For more advanced formatting a specific layout definition language must defined.

1 Introduction

The concept of "digital signature", first publicly described by Diffie and Hellman (1976) in their classic paper "New directions in Cryptography", (Diffie & Hellman 1976) suggests that it is a computer-based equivalent of physical written signatures. Several standards exist for digital signatures, such as the international standards ISO/IEC 9796 (ISO/IEC 2006*b*) and ISO/IEC 14888 (ISO/IEC 2006*a*), and the US Digital Signature Standard (DSS) (NIST 1994). It should be noted that these standards focus on cryptographic algorithms, and not on implementation aspects and visualisation of digital documents. The advantage of the standards is to specify methods and primitives which make digital signatures practical to implement and efficient to execute. However, they do not focus on the necessary integrity requirements of the implementations and underlying platform.

Although there are similarities between handwritten and digital signatures, there are also fundamental differences. The main similarity is that both types of signatures can provide evidence of authorship and authenticity of a document. The differences are due to the radically different nature of paper-based documents on the one hand, and electronic documents on the other. In paper-based transactions, a document consists of text printed as ink on a piece of paper, where the text represents the information and the paper represents the storage medium. In this way the information and the storage medium are inseparable. The validity of a paper-based document is authenticated by a signature written in ink on the same piece of paper. The signature serves as evidence of the signer's agreement to the text on the paper. A pen is the only instrument between the signer and the document for creating the hand written signature. No instrument, a part from optional optical glasses for improved vision, is required for verifying hand written signatures. Both pen and glasses can be considered reliable.

For digital signatures all of this changes. Documents are immaterial because the information is represented by logical bits that can be stored on, and copied to, any suitable electronic medium, and they only become meaningful to humans when represented through an analogue physical medium such as a computer screen or a printout. The validity of a digital document is authenticated by verifying that an immaterial digital signature logically matches the already immaterial document. Because a digital document in its immaterial form can not be observed directly by the signer, the digital signature can only serve as evidence of the signer's agreement to some analogue representation of the document, although it is usually assumed that it represents the signer's agreement to the immaterial electronic document itself. Highly complex instruments are now needed not only for viewing the document but also for producing the digital signature.

A desirable property of digital signature systems is to guarantee that "what you see is what you sign", abbreviated as WYSIWYS. The WYSIWYS property articulates that the bit representation of digital documents must be visualised consistently and as intended to the signer by the digital signature system. Any violation of the WYSIWYS property has the potential of wrongly imposing liability on, or wrongly taking liability away from people and organisations who apply digital signatures.

To our knowledge, no currently existing system for digital signatures is able to provide high WYSIWYS assurance *independently* of the integrity of the system used for displaying digital documents. In general, the property of providing WYSIWYS depends on the integrity of the digital signature system and platform. In practice

it is extremely difficult to assess the integrity of a general purpose computing platform, and thereby to ensure WYSIWYS (Alsaid & Mitchell 2005, Jøsang et al. 2002, Scheibelhoferm 2001, Kain et al. 2002, Spalka et al. 2001, Weber 1998).

The difficulty of determining the integrity of a system from the user perspective can be seen as a security usability problem. The failure to provide the user with sufficient information to determine the system's security and integrity is a security usability vulnerability which can be exploited by attackers (Jøsang et al. 2007).

This problem is amplified by the fact that digital signatures may be generated on platforms that are not under the control of the signing party, and that there can be strong financial incentives for trying to manipulate the systems used for digital signatures.

In case there is insufficient evidence regarding the integrity of digital signature systems and platform, they can in principle not be trusted. This is a fundamental problem for the practical usage of digital signatures.

This paper describes a robust method for ensuring WYSIWYS. The method is based on using a personal portable signature platform which for example can be embedded in a mobile phone. The digital signature process takes place on a traditional computing platform in combination with the portable signature platform. In order to forge a digitally signed document by attacking the platform integrity, the attacker must compromise both the traditional computing platform and the portable signature platform. This provides a robust digital signature system because it is considered hard to simultaneously compromise both platforms. A limitation with our method is that it currently can only be applied to documents formatted with simple ASCII text, but this limitation can be reduced with a specific layout definition language.

The rest of the document is organised as follows. Sec. 2 describes the fundamental security problems of digital signatures. Sec.3 briefly reviews existing methods and proposals for providing high WYSIWYS assurance. Sec.4 introduces our method for robust WYSIWYS. Our method is currently only applicable to digital signatures on documents formatted with simple ASCII text. Sec.5 describes the possibility of defining a layout formatting language which would make our method applicable to documents with more advanced formatting. Our method is discussed in Sec.6, and Sec.7 concludes.

2 Security Problems with Digital Signatures

A fundamental aspect of digital documents is that displaying and digitally signing them are separate and unlinked processes. In addition, the same digital document is more often that not displayed differently by different applications on different systems. As a consequence it is difficult to determine what exactly has been signed, both from the signer's and the verifier's point of view.

Digital signature systems can cause two types of errors: (Arnellos et al. 2005, Jøsang et al. 2002):

- **False Positive Verification of Digital Signature**
  A digital document which is transformed and visualised differently, giving rise to different semantic interpretations, although only one digital signature applies.
- **False Negative Verification of Digital Signature**
  A bit-level alteration in the digital document which renders the digital signature invalid, but which does not affect the transformation, visualisation and semantic interpretation of the digital document.

The source of the problem is that digital signatures are applied at the bit level representation, whereas the semantic interpretation is based on high level transformation and analogue visualisation. This gives rise to *semantic level distance* (Arnellos et al. 2005) which is introduced by the complex systems and formats used for representing, transforming and ultimately visualising digital documents. Different document representation systems will be characterised by different semantic distances between the bit level representation and the visualisation, where for example MS-Word documents have a high semantic distance whereas bitmap images have a low semantic distance.

A digital document itself only represents a small part of all the elements needed to visually render the digital document. Additional elements range from the operating system and application software to font libraries, device drivers and even hardware and firmware.

In order to guarantee WYSIWYS on a digital signature system, the digital signature should be applied to all these elements. Unfortunately it would be impractical to follow this principle with current technology. Theoretically it would be possible to apply trusted computing technology (Mitchell ed.) in conjunction with digital signature technology to follow this principle. However, this would require the trusted computing system verification to cover the whole system, from hardware to software, which would not be practical.

In case of digitally signing XML documents, the principle of including as many elements as practical is followed, as described in Sec.3.1. However, it is relatively straightforward to manipulate other parts of a computer system in order to change the meaning of digital documents when signed. The following example was first described in (Jøsang et al. 2002).

In this scenario, Clark prepares a digital document with the following contents when displayed in Helvetica font:

On 10 February 2007, Clark borrowed from Alice the sum of ¥1000.

Clark then creates a font type which is similar to Helvetica in style, but for which the glyphs for "¥" and "$" are interchanged. Clark calls this new font type "Helvetica", but we call it "Helvetica′" here in order to distinguish it from the original one. Utilities such as *TrueType Font Namer* (UniTech-MyTools 2007)) exist for computing platforms such as MS-Windows to change font type names, so that in practice any font type can have any name. Clark substitutes the original Helvetica font type with Helvetica′ on his computer, with the result that the document looks like:

On 10 February 2007, Clark borrowed from Alice the sum of $1000.

Clark then borrows $1000 from Alice and digitally signs the document. Alice is satisfied by visual inspection of the document and verification of the digital signature on Clark's computer. Alice copies the digitally signed document to her electronic storage medium as evidence for Clark's debt to her. When Alice tries to prove her case and displays the digitally signed document in a court room the font Helvetica′ is replaced with Helvetica, with the result that the evidence indicates a debt of ¥1000 instead of $1000.

In this scenario Alice has no way of verifying that the

font type has been manipulated before signing, because Clark's new font carries the name she expects to see. The court is equally unable to find indications of malicious manipulation of the document.

This simple example illustrates the fundamental challenge in achieving the WYSIWYS goal in digital signature systems.

3 Existing Proposals for WYSIWYS Assurance

3.1 XMLDSig

The extensible markup language (XML) can be used to represent digital documents in the form of tagged elements that can be composed in a tree based structure. Digital signatures can then be applied to each element of an XML document separately. Like XML itself this method of digitally signing XML documents is specified by W3C in the standard known as XMLDSig (Bartel et al. 2002).

A basic XML document does not carry information about how to display the data. The correct display of an XML document e.g. in a browser is expressed with XML based languages specifically defined for graphical layout of documents, such as the original HTML (hypertext markup language), or by linking in external formatting documents expressed in XML based languages such as CSS (Cascading style sheets) or the XSL (extensible stylesheet language) family[1]. XSL can be used to alter the format of XML data, either into HTML or other formats that are suitable for a browser to display. When applying this type of document transformation, the visualisation of an XML document depends not only on the XML document itself, but also on the external formatting documents.

In order to digitally sign an XML document, the signature program must also sign any such external document, i.e. any document referred to from within the XML document itself, as well as other formatting documents indirectly referred to. This principle is expressed in the XMLDSig standard as follows (Bartel et al. 2002):

> *Just as a user should only sign what he or she "sees," persons and automated mechanism that trust the validity of a transformed document on the basis of a valid signature should operate over the data that was transformed (including canonicalization) and signed, not the original pre-transformed data. This recommendation applies to transforms specified within the signature as well as those included as part of the document itself. For instance, if an XML document includes an embedded style sheet [XSLT] it is the transformed document that should be represented to the user and signed. To meet this recommendation where a document references an external style sheet, the content of that external resource should also be signed as via a signature Reference otherwise the content of that external content might change which alters the resulting document without invalidating the signature.*

The transformation and graphical rendering of documents in this fashion obviously becomes extremely complex, and complexity has always been the enemy of security. This fact is recognised in W3C's XMLDSig standard as follows (Bartel et al. 2002):

> *Some applications might operate over the original or intermediary data but should be extremely careful about potential weaknesses introduced between the original and transformed data. This is a trust decision about the character and meaning of the transforms that an application needs to make with caution. Consider a canonicalization algorithm that normalizes character case (lower to upper) or character composition ('e and accent' to 'accented-e'). An adversary could introduce changes that are normalized and consequently inconsequential to signature validity but material to a DOM processor. For instance, by changing the case of a character one might influence the result of an XPath selection. A serious risk is introduced if that change is normalized for signature validation but the processor operates over the original data and returns a different result than intended.*

The XMLDSig standard is being used for practical applications on the Internet, and also allows flexible multi-signing of XML documents (Kubbilun et al. 2005). However, the reservation expressed in the XMLDSig standard itself indicates that this is not a technology that can provide robust WYSIWYS.

3.2 Signing Static Formats

Documents which only require the most basic typesetting and formatting in order to be visualised can be called static file formats. Simple ASCII text represents an example of static format which requires relatively simple transformation for visualisation. PDF and PS documents actually represent dynamic formats that require relatively complex transformation in order to be visualised. Representing documents in static formats reduces the semantic distance between the bitmap representation and the semantic interpretation, but a certain distance remains. For example, even this approach would be vulnerable to the font replacement attack described in Sec.2.

3.3 Signing Bitmap Formats

An approach to reducing the semantic distance between the bit representation and the semantic interpretation even more is to sign a bitmap transformation of the digital document (Scheibelhoferm 2001). The font replacement attack of Sec.2 would for example not work when the document is visualised as bitmap.

More fundamental attacks would now be needed in order to be successful. The system platform itself still creates a certain semantic distance which can be exploited. For example malware can apply images over the displayed bitmap document (Lefranc & Naccache 2002) so as to create the impression that a different document is displayed. Not even bitmap formats can reduce the semantic distance sufficiently to ensure robust WYSIWYS.

Only trusted systems for displaying digital documents operating within highly controlled environment represents a protection against this type of attacks, and we consider such systems outside the scope of this study. Our focus is on solutions that can be commercially available to most users.

[1] The XSL family of standards consists of the following XML languages:
XSLT (XSL Transformations): for transforming XML documents,
XSL-FO (XSL Formatting Objects): for specifying the visual formatting of an XML document,
XPath (XML Path Language): a non-XML language used by XSLT

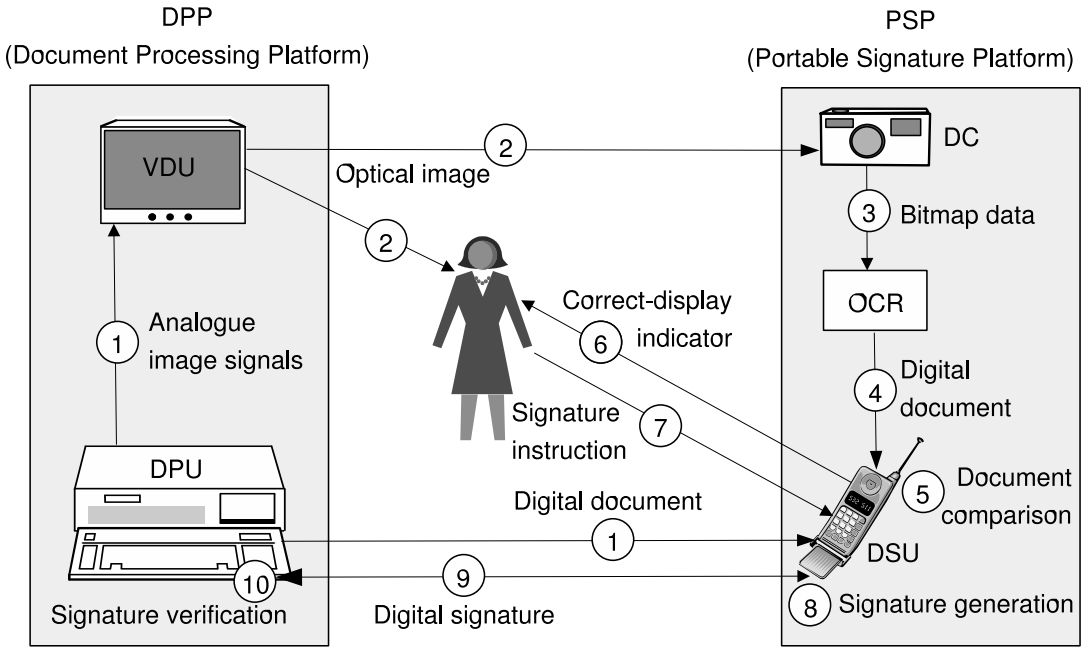

Figure 1: Components for a robust WYSIWYS system

4 The Robust WYSIWYS Solution

Our proposal is based on using standard computer systems with document processing software and a display monitor that can display digital documents. The display monitor will here be called an Visual Display Unit (VDU). The hardware and software for document processing will be called the Document Processing Unit (DPU). The VDU and the DPU together will be called a Document Processing Platform (DPP).

The main idea of our proposal is to use a personal portable platform that is able to convert the analogue visual representation of a document from the VDU into its original digital representation using OCR (optical character recognition) software. The conversion from analogue to digital form first requires the analogue optical image emitted from the VDU to be captured by a digital camera (DC). The bitmap representation of the analogue image produced by the DC is then translated to a digital document representation by Optical Character Recognition (OCR). The DC and the OCR are combined with a Digital Signature Unit (DSU) that is able to generate digital signatures on digital documents. The DC, the OCR, and the DSU together will be called the Portable Signature Platform (PSP). The combination of a DPP and the PSP can be considered as the Robust WYSIWYS system. The components of the Robust WYSIWYS system are illustrated in Fig.1 below.

Our method for Robust WYSIWYS requires communication between the DPP and the PSP through a digital channel, in addition to the visual channel. The user communicates with both units. With reference to Fig.1, the digital signature process goes as follows.

1. The DPU transmits the digital document to the DSU. Simultaneously, the DPU displays the document in analogue form on the VDU.
2. The analogue visual representation of the document is now visible for the user and for the the DC.
3. The DC captures the image displayed on the VDU and generates a bitmap image data which is transmitted to the OCR.
4. The OCR converts the bitmap image data into a digital document, which is transmitted to the DSU.
5. The DSU compares the two digital documents.
6. In case they are equal, the DSU sends a positive signal (audible, visual, physical or other) through the user interface, indicating that the document is correctly displayed, and the DSU will allow the document to be signed if the user decides to do so. In case they are not equal, the DSU sends a negative signal through the user interface, indicating that the document is incorrectly displayed, and the DSU will not allow a digital signature to be generated.
7. Assuming that the digital document was correctly displayed, and that the user decides to sign the digital document, and tells the DSU to apply the digital signature.
8. The DSU generates the digital signature.
9. The Digital Signature is transmitted to the DPU.
10. The DPU verifies the digital signature.

Portable communication devices such mobile phones are possible candidates for a PSP. In fact, most mobile phones have integrated digital cameras, so that they already have the necessary hardware to become a DSU. The inclusion of software for the OCR and for applying digital signatures is all that is needed.

Commercial and open source OCR software packages are available. In its simplest form, OCR software takes scanned documents and converts them into text files. More advanced graphical layout of digital documents will require a standard for geometrically formatting documents so that the translation from analogue bitmap format to

digital document format is unambiguous. Reliable translation will also require adequate visibility conditions to minimise optical distortion when the image is captured by the DC.

5 Towards a Layout Definition Language

Clearly the Robust WYSIWYS system is unable to apply digital signatures to any document representation. It would only be practical to apply digital signatures to static documents such as containing a standard set of characters without any advanced graphical formatting. Documents that only contain ASCII characters could for example be signed.

In order to allow some graphical formatting a new layout specification language would have to be defined. The fundamental requirement of such a language is that there must always be a bijective mapping between the digital bit representation and the analogue visual representation of a document. This is formally expressed below.

Let d represent a document in is digital bit format, and let a represent the same document in its analogue visual form, e.g. as displayed on the computer screen. Let V represent the visualisation process, i.e. the transformation of a digital document into an analogue representation. Let E represent the encoding of an analogue document into a digital document. The bijectivity requirement can then be expressed as:

$$d = \mathrm{E}(\mathrm{V}(d)) \tag{1}$$

In order for our system to be applicable to documents with higher complexity than pure ASCII, research effort is needed for developing a *layout definition language* (LDL). It would be possible to base such a language on the XML structure, but each document would have to be a standalone document without any externally linked style and formatting documents.

6 Discussion

6.1 Security Considerations

The WYSIWYS property is based on using a digital camera which "sees" the digital document to be signed exactly as the user sees it. The bitmap image is then converted to the original digital document using OCR techniques. This bridges the semantic distance between the digital document in its binary form and the analogue visualisation of the document. It basically guarantees that what you see is what you sign.

While the security of the Robust WYSIWYS system is independent of the integrity of the DPP, it does depend on the integrity of the PSP. However, assuming that no both the DPP and the PSP have been compromised simultaneously, it is possible to verify that the PSP indeed has created the digital signature correctly. This cross check is possible precisely because of the bijective property of the document format.

6.1.1 Compromised PSP

Assuming that some element in the PSP has been compromised to that the digital signature has been applied to the wrong document, this fact will be noticed when the DPP receives and verifies the digital signature. It must be assumed that the DPP has not been compromised. Messages (9) and (10) in Fig.1 allows the DPU to verify the correctness of the DSU's signature. DPU will detect when the signature has been generated over the wrong document.

6.1.2 Compromised DPP

Let us now assume that the DPU has been compromised, so that the wrong digital document is sent in message (1). The comparison between the received digital document and digital document converted from the analogue image will be detected by the DSU in step (5).

6.1.3 Simultaneous Compromise of DPP and PSP

Let us now assume that the DPU has been compromised so that it send the wrong digital document to the DSU in (1), and that the DSU has been compromised so that it wrongfully indicates positive comparison in step (5). In this case, the user will instruct the DSU to generate a digital signature over the wrong document.

It thus requires simultaneous compromise of the DPP and the PSP in order to break the security of our system.

6.2 Possible Applications

Given the limited flexibility in document formats that can be digitally signed with our system, the applications will also be limited. However, the simplicity and portability of our system can make in practical in many situations.

When conducting transactions online, documents to be digitally signed can be presented as simple frames with minimal formatting. For example, when conducting online purchase, the amount, date, name of buyer and seller and a simple product description can be sufficient.

The product can be anything from standard consumables to shares and horse race bets.

The Robust WYSIWYS solution enables people to commit to online transactions from any terminal without fear of applying the wrong signature because the terminal is compromised.

6.3 Advantages and Disadvantages

The advantage of the Robust WYSIWYS method is that the digital signature process is separated from the standard computer platforms where documents are normally processed. Such platforms are designed with priority on flexibility and functionality, which unavoidably results in security vulnerabilities.

The security of the Robust WYSIWYS method only depends on either the DPP or the PSP being secure. In fact, one of them can be compromised without causing a risk of tricking the user into applying a digital signature to the wrong document, so the security of our method is totally independent of the security of the DPP. The PSP can be designed with priority on security, and with limited functionality and flexibility. The PSP will be controlled by the user, so she does not have to rely on systems outside her control when digitally signing documents, even when the documents are stored on systems outside her control.

The Robust WYSIWYS method also allows mobility of the digital signature technology. The PSP can be a portable device, that e.g. could be integrated with a mobile phone. Users can then carry with them a device that allows them to apply digital signatures to documents stored on any system anywhere, as long as that system is able to display the document, and to send it in digital format to the PSP. This also allows for flexibility and usability, as

the digital documents can be stored and processed on any system.

The main disadvantage of the proposed method is the limitation to simple ASCII documents. This limitation can be reduced by developing a new document layout definition language. However the requirement of having a bijective mapping between the digital and the visual representation of documents makes it impossible to use the rich formats of modern word processing tools and Web page designs. In our view, digital signatures on documents with such rich formats can never be made secure, except by signing the raw display image bitmap. However, digital signatures on raw image bitmaps would prevent any further processing of the digitally signed document such as search and data mining. Digitally signatures on image bitmaps would also be vulnerable to image overlay attacks, whereas our method is resistant against such attacks.

7 Conclusion

Current technologies for digital signatures have limited trustworthiness due to the vulnerability of the computing platforms used for applying the digital signatures. We have shown that it is possible to make the security of digital signatures independent of the security of the platform. This is achieved by using a portable digital signature device in combination with the traditional digital signature system. The WYSIWYS property is based on using a digital camera which "sees" the digital document to be signed exactly as the user sees it. The bitmap image is then converted to the original digital form using OCR techniques. While this approach currently is limited to simple ASCII documents, it can be made more general by specifying a layout definition language.

References

Alsaid, A. & Mitchell, C. (2005), 'Dynamic Content attacks on Digital Signatures', *Information Management & Computer Security* **13**(4), 328–336.

Arnellos, A., Lekkas, D., Spyrou, T. & Darzentas, J. (2005), 'A Framework for the Analysis of the Reliability of Digital Signatures for Secure E-commerce', *The electronic Journal for e-commerce Tools & Applications (eJETA)* **1**(4).

Bartel, M. et al. (2002), *XML-Signature Syntax and Processing - W3C Recommendation 20 February 2002*, W3C (World Wide Web Consortium).
URL: *http://www.w3.org/TR/xmldsig-core/*

Diffie, W. & Hellman, M. E. (1976), 'New directions in cryptography', *IEEE Transactions on Information Theory* **22**(6), 644–654.

ISO/IEC (2006*a*), *14888. Information technology - Security techniques - Digital signatures with appendix - Parts 1,2,3*, ISO/IEC JTC1.

ISO/IEC (2006*b*), *9796. Information technology - Security techniques - Digital signature scheme giving message recovery - Parts 1,2,3*, ISO/IEC JTC1.

Jøsang, A., AlFayyadh, B., Grandison, T., AlZomai, M. & McNamara, J. (2007), Security Usability Principles for Vulnerability Analysis and Risk Assessment, *in* 'The Proceedings of the Annual Computer Security Applications Conference (ACSAC'07)'.

Jøsang, A., Povey, D. & Ho, A. (2002), What You See is Not Always What You Sign, *in* 'Proceedings of the Australian UNIX and Open Systems Users Group Conference (AUUG2002)'.

Kain, K., Smith, S. & Asokanm, R. (2002), Digital Signatures and Electronic Documents: A Cautionary Tale., *in* 'Proceedings of IFIP Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security.'.

Kubbilun, W., Gajek, S., Psarros, M. & Schwenk, J. (2005), Trustworthy Verification and Visualisation of Multiple XML-Signatures, *in* 'Proceedings of the IFIP International Conference on Communications And Multimedia Security (IFIP CMS 2005)'.

Lefranc, S. & Naccache, D. (2002), 'Cut and paste attacks with java', Cryptology ePrint Archive, Report 2002/010. http://eprint.iacr.org/.

Mitchell (ed.), C. (2005), *Trusted Computing*, IEE Press.

NIST (1994), *Digital Signature Standard (DSS)*, National Institute of Standards and Technology (NIST).

Scheibelhoferm, K. (2001), Signing XML Documents and the Concept of What You See Is What You Sign, Masters thesis, Graz University of Technology, Austria.

Spalka, A., Cremers, A. B. & Langweg, H. (2001), The fairy tale of 'What You See Is What You Sign - Trojan Horse Attacks on Software for Digital Signatures, *in* 'IFIP Working Conference on Security and Control of IT in Society-II (SCITS-II)'.

UniTech-MyTools (2007), 'TrueType Font Namer', URL: http://www.mytools.com/fontnamer.html (visited 12 February 2007.

Weber, A. (1998), See What You Sign: Secure Implementations of Digital Signatures, *in* 'Proceedings of the International Conference on Intelligence and Services in Networks', pp. 509–520.