A Trusted Process to Digitally Sign a Document

Boris Balacheff
Hewlett-Packard Laboratories
Filton Road, Stoke Gifford
Bristol BS34 8QZ, UK
+44 117 3128002
boris_balacheff@hp.com

Liqun Chen
Hewlett-Packard Laboratories
Filton Road, Stoke Gifford
Bristol BS34 8QZ, UK
+44 117 3128217
liqun_chen@hp.com

David Plaquin
Hewlett-Packard Laboratories
Filton Road, Stoke Gifford
Bristol BS34 8QZ, UK
+44 117 3128801
david_plaquin@hp.com

Graeme Proudler
Hewlett-Packard Laboratories
Filton Road, Stoke Gifford
Bristol BS34 8QZ, UK
+44 117 3128753
graeme_proudler@hp.com

ABSTRACT

This paper describes a method of increasing the trust in open computing platforms, such that a person can have confidence in producing a digital signature using open platforms.

The process of using a digital signature to sign a digital document is well understood. Most descriptions assume the correctness of the process of signing a document within a computing platform. In an increasing connected world, this assumption is no longer true when open computing platforms are used. This paper proposes the signing of a document in a general-purpose computing platform using a trusted process. That trusted process creates a signature over a digital image that represents the document and uses a trusted display controller in the platform plus a smart card owned by the prospective signer. The trusted display controller is part of the video processing path, and can display video data on a monitor without interference or subversion by any software components at the platform. The smart card is able to authenticate the trusted display controller, and demonstrate to the signer the results of that authentication using the trusted display controller.

The most unusual aspects of the method are: (1) a thumbnail image is stored in the smart card, and used as a surround or background for an image (on a display) that is to be signed; (2) the smart card signs image data on the authority of the trusted display controller, without direct authorisation from the signer.

General Terms

Security, Human Factors.

Keywords

Digital signatures, smart card, trusted display, TCPA, authenticated image.

NSPW'01, September 10-13th, 2002, Cloudcroft, New Mexico, USA.

1. SUMMARY

Digital signatures are well known and well understood, and are legal in some countries. One worry with digital signatures is that a person might be duped during the signing process by a subverted platform, or by a platform that performs an inappropriate action. Such duplicity could commit the person to a legally binding contract that they would normally not have signed. Various groups have proposed closed platforms for performing digital signatures. Closed platforms reduce the scale of the problem, because a closed platform doesn't (by definition) execute arbitrary software and is consequently much harder to subvert. It is desirable, however, for open computing platforms to have similar levels of confidence as closed platforms. This is because open platforms can have a higher price/performance ratio than closed platforms. The problem is that (1) open platforms execute arbitrary software, (2) it is difficult to modify the architecture of ubiquitous open platforms, (3) the cost of any modification must be small, and (4) the modification must provide the necessary confidence.

The proposed solution is predicated upon the success of the Trusted Computing Platform Alliance. If TCPA is successful, open computing platforms such as the Personal Computer (and its numerous variants) will contain an extra piece of hardware (a "Trusted Platform Module"). TCPA provides greater confidence in the trustworthiness of a digital signature, but still has some reliance on correctly operating software. For the purposes of this paper, the most important point of the TCPA modification is that it sets a precedent for the installation in open platforms of tamper resistant hardware with cryptographic capabilities.

This paper postulates a modification of the TPM, so that it controls the video path in an open platform and is able to communicate with a smart card. Such a Trusted Display Controller (TDC) could be a bridge in a bus, to isolate existing video circuitry, or could itself include the video circuitry. This TDC has the property that it can force a particular image onto the computer's monitor, and can use its cryptographic capabilities to communicate with a smart card. It uses these properties to cooperate with a person and provide the necessary confidence in the process of digital signing. This provides a trusted platform facility for digital signing by a person. While the method requires some support from software executing normally in the platform, none of the trust in the signing process depends on that software. The method therefore cannot be subverted by software executing on the platform.

The method itself relies upon the protected communications between the TDC and the user's smart card, and on privileged

access to the computer's display. The method is to sign an image on the computer's display. The TDC obtains a thumbnail image from the smart card and uses that thumbnail as a surround or background to the image (on the display) that is to be signed. The user knows that that his smart card reveals the thumbnail image to the TDC only if the smart card has been able to verify that the TDC is genuine. (This uses techniques described by TCPA and based on PKI.) The user can therefore believe that any image on the computer's display that is surrounded by the thumbnail image, or highlighted by the thumbnail image, is controlled by the TDC, and will be the image sent by the TDC to the smart card for signing. At the same time, the TDC generates a nonce pass phrase, which is displayed and highlighted by the thumbnail image. The signer (person) enters that pass phrase into the unprotected keyboard, to confirm to the TDC that the highlighted image should be signed.

2. INTRODUCTION

The use of images in security has been discussed in papers such as [17] (which discusses the fingerprinting and watermarking of documents) and [18] (which presents authentication information as images). This paper introduces the use of images as proof of trustworthiness of a signing process.

In the same way that conventional services rely upon a hand-written signature, E-services must rely upon a digital signature. The assertion in this paper is that digital signing on an open computing platform requires a modification to the architecture of standard computing platforms. Otherwise there is doubt in the validity of such signatures.

With the increase in commercial activity over the Internet, known as E-commerce, there has been much interest in enabling data origin authentication and data integrity. In particular, it is perceived to be important for users to be able to enter into binding contracts over the Internet. The governments of some countries, such as the US, the UK and France, have already encouraged the use of digital signatures [4, 6, 7]. Signing an electronic document through the use of a computing platform, instead of a conventional hand-written signature on paper, is critical for more ambitious types of E-commerce. Preferably, that computing platform is an open platform, rather than a closed platform.

Although digital signature and hand-written signatures play the same role for E-commerce and conventional commerce respectively, their implementations have very different properties.

In the conventional method of signing a document, a signer physically writes a signature on the medium (usually paper) upon which an image of a document is reproduced. This method has an obvious advantage: it is clear to a signer what is being signed, and if the graphical hand-signature and the signed document are not modified, they are proof of what has been signed.

Conventional electronic methods of digitally signing a document are well known. Essentially, digital data is compressed into a digest, for example by the use of a hash function [10]. Then that digest is encrypted by the use of some encryption method that has been initialised by a secret key [9, 11]. This can be done on an open computer platform, such as a PC, equipped with a smart card, which offers better protection of a private signature key [8]. The following are a number of examples of digital signatures using smart cards: the Endorse card (Barclays Bank [3]), the smart ID card (Lloyds TSB [13]), and the personal identifiers (Malaysia, Spain [3] and Finland [15]).

During conventional signing processes, a signer visually interprets a document as it has been rendered on the computer's monitor at normal magnification and resolution. Typically, the signer's smart card signs data in a format that is the representation of the document by the application used to create and/or manipulate the document (e.g. word processor, Email viewer, etc).

The potential problem is that the document that is presented for signing might not be the same as the one that is displayed to the signer. It is possible that the software used to display a document to the signer is malicious or just broken, and displays a document that is different from the one sent to the smart card for digital signing. It is therefore possible for a signer to unintentionally sign data, via their smart card, which is different from that which they intended to sign. Conversely, it is also possible for a signer to intentionally sign data and later fraudulently claim that the signed data does not correspond to that displayed to them by the computer platform at the time when they gave the instruction to the smart card to perform the signature. This reduces considerably the trust that can be put in a digital signature. A discussion on how the above issues challenge the very value of digital signatures, and their validity to be recognised as legally binding, can be found in [5].

The fundamental issue is that the signer has to trust the open computing platform. In particular, the signer must be sure that the display system and the user input system (i.e. the human and computer interface) within the computing platform will reliably interact with the smart card. This trust relationship is particularly difficult to establish with normal computing platforms due to their increasing connectivity. Indeed, most of the time it is nearly impossible to have a good knowledge of the history of the platform in order to determine the level of trust it can offer (possible virus, worms, rogue software, etc...). Under these circumstances and without the knowledge of these elements, the signer needs some other means to trust the behaviour of the platform. One approach to securing the interactions between a user and their smart card is to use a specific smart card reader with display and pinpad. These can be found in specific applications such as banking. This type of solution doesn't however extend to the more general use of digital signature on electronic documents, due to the limitations of the display properties of such "secure" smart card readers.

The notion of a more powerful display and input capability for a signing device was, for example, addressed by Blafanz and Felten in [2]. Their solution was to use a hand-held computer instead of a smart card. This transforms the problem into "how to make the hand-held computer trustworthy". The hand-held solution is safe as long as the device is closed. In other words, that the device is dedicated to this purpose only, or is considered to be a constrained environment (one user only, no or very few controlled download of applications, etc...). But the evolution of PDA technology tends to evolve towards connectivity and open platforms, and hand-helds are evolving towards Internet peers. As a consequence, we start to see malicious code in these environments (e.g. the first virus

designed for PalmOS, documented in [12]). Today, the problem of being able to trust a platform to perform a digital signature (using a smart card to protect the sensitive signing key) is just as applicable to hand-helds as it is to PCs and any other open computing platforms.

This paper proposes the digitally signing of a document by means of a trusted process within an open computing platform. As part of the solution, we create a signature on a digital image representation of the document, with the property that the image is the same as the one displayed on a monitor of the platform. A similar approach is used in [16]. There, an image of a document is copied from a open PC (not necessarily belonging to the signer), displayed on a open PDA (belonging to the signer), and signed by a smart card. That approach is acknowledged to have the weakness that the trustworthiness of the signing process depends critically on the trustworthiness of the PDA, but it is argued that the PDA is more trusted than the PC.

If a signature is to be trusted, something in the signing equipment has to be trusted by the signer. This solution uses an architecture that minimises the number of things that must be trusted by the signer. The roots of trust are a trusted display controller in the platform and a smart card belonging to the signer. The trusted display controller is part of the video processing path, in order to display video data on the monitor without interference or subversion by any software components at the platform. The smart card is able to authenticate the trusted display controller and tell the signer the result of authentication. This trusted process provides the signer with confidence that the document they are seeing on the screen of the platform is in fact the document they are signing with their smart card. The data that is actually signed is the bit-map image of the document plus additional information about the rendering of the image on the monitor. The additional information is everything that is necessary to accurately reconstruct the image: the size of the image, the number and distribution and shape of the pixels, and so on. Such information is needed to reproduce exactly what the signer saw when the image was signed. Otherwise, the intent of the signer could be in doubt. For example, if some "small print" in a document is so small that it is displayed as dots on a screen, it is likely that a signer was unaware of that small print, and hence the signer could assert that the dots were not part of the agreement.

Note that the signed image can be automatically interpreted (as ascii, say) by submitting the image to Optical Character Recognition (OCR) software. Since the image is already an optimised "scan" of characters, the OCR should have a high probability of mapping the bitmap back into ascii. The signed image includes a description of the font used to construct the image, which further improves the success of the OCR process.

The organisation of the remaining of the paper is as follows. In the next section, we will introduce the concept of a trusted process for signing a document. We will then in Section 3 describe one implementation of the solution in detail. Then we give a security analysis of this solution in Section 4. The paper concludes in Section 5.

3. CONCEPT OF A TRUSTED PROCESS FOR SIGNING A DOCUMENT

In this section, we introduce the concept of signing a document using a trusted process within a computing platform. This means signing a document in a manner that provides a high level of confidence to the signing party that the document they think they are signing is in fact the document they are signing.

The term "trusted", when used in relation to a physical or logical component or an operation or process, implies that the behaviour thereof is predictable under substantially any operating condition and highly resistant to interference or subversion by external agents, such as subversive application software, viruses or some level of physical interference.

In this paper, we focus on one possible implementation of the solution involving an open computing platform with a smart card. The platform is a data processing system that generates a visual representation of a document to be signed. The smart card has a computing engine for receiving the visual representation of the document, and generating a digital signature on it. It is assumed that the signer trusts the smart card to follow the procedure of the trusted signing process properly, as described in the next section. However, it is not necessary for the signer to trust the platform, apart from its trusted display controller. This means that the signer does not need to implicitly trust that the platform has a correct software and hardware configuration.

In general, the process follows these steps:

- The smart card authenticates a trusted display controller within the computing platform.
- After authentication of the display controller, the smart card indicates to the signer that the authentication is successful.
- The signer is informed of the representation of the document that will be sent to his smart card for signing.
- If the signer believes that the representation of the document is what he wants to sign, he sends confirmation to the display controller.
- The display controller sends confirmation to the smart card.
- Upon receiving confirmation from display controller, the smart card signs the document and releases the signature to the platform.

A detailed account of this solution will be described in the next section. We now give a very brief introduction.

The TDC in the platform controls the production of images on the monitor, and is protected against interference.

The smart card holds trusted image data, which we call a "seal", since it performs a function similar to the seals of old (which sealed documents with wax). The seal image is passed to the TDC over a logically protected channel and displayed by the TDC during the signing procedure. A seal image is typically unique to the signer.

The display of a "seal" image on a monitor controlled by a TDC provides the signer with the confidence that the TDC has

been authenticated by the smart card, and that the TDC is in control of the signing operation. The seal image highlights the image that is to be signed, and also highlights an ascii nonce string that is generated by the TDC. The technology of using a smart card to authenticate a component of a computing platform has been discussed in [1]. When the signer sees the highlighting by the seal image, the signer knows that his smart card has verified that the TDC is trustworthy, and that the highlighted image is genuinely the thing that is being proposed for signature, and that the ascii nonce string has been generated by the TDC. If the signer is certain that he wishes to sign the highlighted image, the signer types the highlighted ascii nonce to confirm the processing of signature. The signer knows that it is safe to enter the ascii nonce through the normal keyboard because that string will be used only once – it doesn't matter whether the platform snoops on that string. After receiving this nonce, the TDC tells the smart card to sign the document, and the smart card releases the signature to the TDC.

The logical components used to achieve the trusted process of signing a document are shown in Figure 1.

be accessed by the other platform's components. Of course, the platform must still have access to the normal display facilities offered by the TDC, so that the OS and other various applications can carry on displaying normal information, but all the sensitive data (such as private keys and loaded seal images) must be kept protected in the TDC.

3. The TDC must provide a protected communication link with the smart card. This can be achieved using cryptography, for example (see description of the implementation in the next section).

The TDC should have the same level of protection as that provided by a smart card. This is acceptable because the TDC does not contain global secrets. Any level of hardware protection can be broken (given sufficient time and money), but a TDC attacker gains only the secrets belonging to an individual TDC. A global secret is a valuable prize, practically guaranteed to attract unwelcome attention if the protecting hardware left unattended.

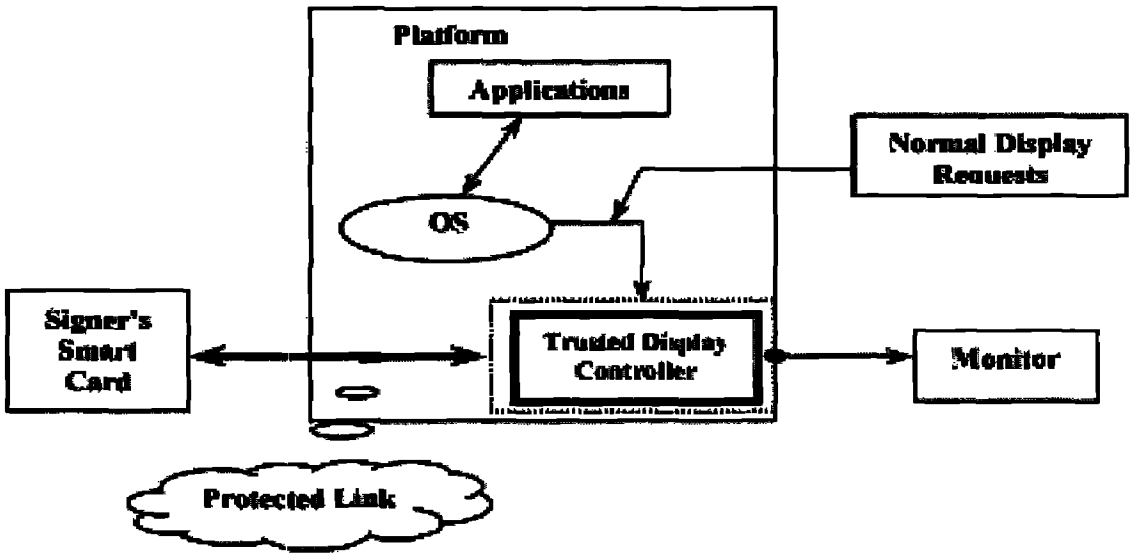

Figure 1. Logical components to achieve trusted signing

The trusted process relies on the design of the TDC and on its integration in the platform. Therefore the following criteria must be checked before a certificate authority issues a certificate for the identifier of a TDC, i.e., the asymmetric key pair owned by the TDC.

1. The TDC must be connected to the monitor in a proper manner, i.e. no malleable component is between the TDC and the monitor. This is necessary to prevent modification of the video data after it has left the TDC.
2. The TDC must have properties of tamper resistance. It must be isolated from the rest of the platform so that any private information held inside the TDC cannot

This fact, plus the cost constraints of a ubiquitous open platform such as the PC, makes it essential that a TDC does not store global secrets.

4. ONE IMPLEMENTATION OF THE SOLUTION

In this section, we describe an implementation of this trusted signing process in more detail. There are three entities involved in this implementation: the signer, the TDC, and the signer's Smart Card (SC).

As mentioned earlier, the TDC must meet certain criteria in terms of tamper resistance (both physical protection and protection against software attack). More particularly, the TDC

must be immune to unauthorised modification or inspection of internal data. It has crypto functionality to securely communicate with the SC. The TDC is associated with video data at a stage in the video processing beyond the point where data can be manipulated by standard computer software. This allows the TDC to display data on a monitor without interference or subversion by any software components of the computer platform. Thus, the TDC can be certain what image is currently being displayed to the signer. This is used to unambiguously identify the image that a signer is signing.

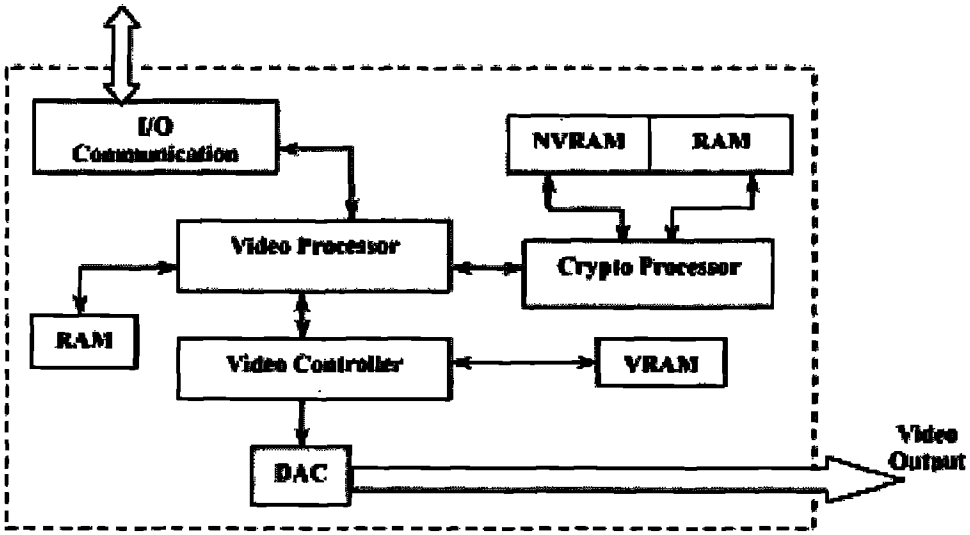

Figure 2. Structure of the TDC
© Hewlett Packard 2001

The figure 2 shows the internal architecture of a possible TDC. It includes the standard components of a video card and some specific cryptographic components. The I/O Communication component allows the TDC to communicate with the rest of the platform while enforcing a physical and logical isolation from the platform. This communication channel is used by the platform to send normal display requests and also to provide the communication mechanism with the signer's SC. The Video Controller is responsible for controlling access to the Video RAM (VRAM). It also transfers the content of the VRAM to the Digital to Analogue Converter (DAC) component that transforms the digital information into an analogue signal (Video Output) that can then be displayed on the monitor. The Video Processor implements most of the security functions of the TDC. It handles every request coming from the platform. It transforms all display requests into graphic primitives that describe how the VRAM should be filled. Based on this description, it uses the Video Controller to properly fill the VRAM. The Video Processor can also communicate with the Crypto Processor when interacting with the SC. The Crypto Processor deals with signing or confidentiality requests from the Video Processor. For example, when an encrypted seal image is received from the SC, the Video Processor first sends it to the Crypto Processor to decrypt it, and then stores the seal image into the Video Processor RAM so that it can later be rendered and displayed.

The implementation's protocols are designed with the following assumptions:

1. The TDC has two asymmetric key pairs, for signature and encryption respectively.
2. The SC has an asymmetric key pair for signature.
3. Both the TDC and the SC have access to each other's public keys. If this is not true, these two entities must exchange certificates describing their public keys before or during the protocols described below. In that case, they must be able to verify validation of these certificates using, for example, some trusted certificate authorities.
4. When authentication of the TDC is complete, the signer believes that the TDC is genuine, and hence that its design and implementation meet the criteria described in Section 2.
5. The signer believes that the smart card is well designed and implemented and will properly follow the protocols of the trusted signing process.

Before using the computer platform to sign a document, the signer chooses an unpredictable seal image and stores it in the SC. This should be done securely, and could be in advance on the signer's own trusted computing platform, for example.

Before the signer initiates a signature process, the TDC and the SC run the following protocol to introduce the TDC and the SC to each other. The protocol activates the SC, authorising it to accept "sign this data" commands from the TDC. The TDC permits the protocol to execute only when the platform is in a suitably trusted state – only a restricted range of software has executed on the platform since boot, for example. The process that permits a TPM to know whether a platform is in a trusted state is a TCPA method, and is not discussed further in this paper.

Broadly speaking, the activation protocol works as follows:

1. The TDC sends the SC a request to initiate the protocol.
2. Upon receipt of the signature request, the SC generates a challenge to the TDC.
3. The TDC generates a reply to the challenge, signs it using its private signature key, and sends it to the SC.
4. The SC verifies the signature. If the verification succeeds, the SC encrypts the seal image under the TDC's public encryption key, signs the encrypted image and the nonces from the challenge, and sends the data back to the TDC.

Otherwise, the SC refuses the protocol by doing nothing. After a timeout (say, ten seconds), the signer will recognise that the protocol has been aborted.

5. The TDC first decrypts the seal image, and then verifies the signature. If the verification succeeds, the TDC displays the seal image around or over a message that requests the user to unlock the SC.

6. Upon seeing the seal image, the user knows that the mutual authentication has succeeded. The user enters the SC's passphrase using the normal keyboard. The platform passes the passphrase to the SC, unlocking the SC, so it will accept commands from the TDC to sign data.

This protocol achieves the following two things.

1. Firstly, it includes mutual authentication between the SC and the TDC (the SC reveals the seal image only if the authentication to the TDC passes). According to Assumption 4 above, a successful authentication to the TDC implies that the desired level of trust for the system is met, enforced by the level of tamper protection of the TDC. As a result, the signer knows that the TDC is a genuine one and its design is suitable for digital signing.

2. Secondly, it ensures that only the TDC can know the seal image, since it is encrypted under the TDC's public key.

At some later time, perhaps when the platform is in a less trusted state than that required to activate the SC, the user needs to sign a document. An application that is "TDC aware" cooperates with the TDC to automatically frame the document, by requesting the TDC to display the seal image around or over the document that the signer wants to sign. (Figure 3 illustrates a framed document.) Alternatively, the user interacts with the TDC via software and frames the document with the seal image.

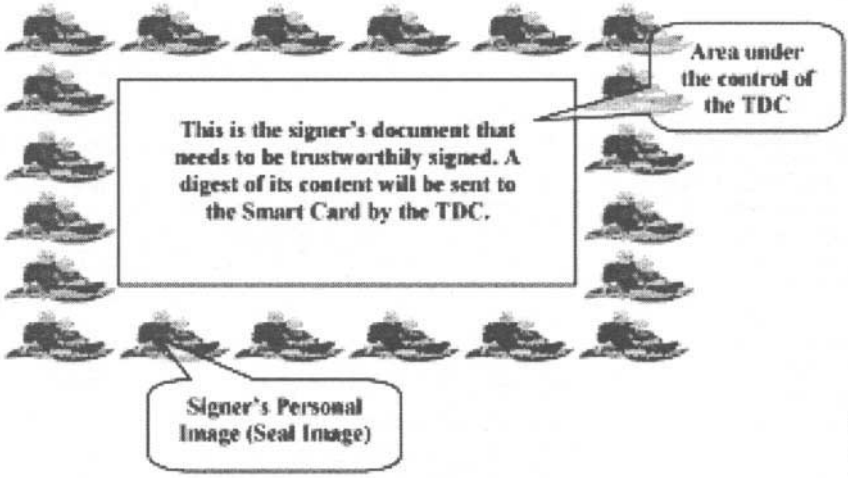

Figure 3. A document displayed by the TDC
© Hewlett Packard 2001

Upon seeing the seal image, the signer knows that the TDC is controlling the section of the display highlighted by the seal. Furthermore, the signer knows that a digest of the document highlighted by the seal image will be sent to the SC for signing.

Figure 4 illustrates a request from the TDC, asking for the signer's confirmation that they are satisfied with the document

Figure 4. Confirmation of the signature process
© Hewlett Packard 2001

that is going to be signed. If the signer is satisfied what they see, they type the nonce pass phrase. Since the pass phrase is a nonce, it doesn't matter whether the platform snoops on the pass phrase. If the TDC receives the correct nonce, the TDC communicates with the smart card and commands the smart card to sign the appropriate bit map.

Protocol 2 works as follows:

1. The TDC sends the SC a protocol-2 request.

2. The SC generates a random number and sends it back to the TDC.

3. The TDC signs the concatenation of the random number and the digest of the document image, marks the message as a "confirmed signature request", and sends the result to the SC.

4. The SC verifies the signature and observes that the message is a "confirmed signature request. If verification succeeds, the SC signs the digest of the document image and sends it to the TDC. Otherwise, the SC will refuse the signature process by doing nothing. After a timeout (say, ten seconds), the signer will recognise that the protocol has aborted.

5. After receiving the digest of the image, the TDC appends the signature to the video image and stores them.

In the above process, each video image normally displays one page of the document. If the document to be signed has more than one page, the process has to make a proper linkage between different pages and images. There are a number of

obvious ways to do this, such as incorporating the digest of the previous page in the digest of the following page and so on, and the process is not described here.

In order to verify a signed document, all of the signatures on each individual page and the summary must be verified.

5. SECURITY ANALYSIS

The main thrust of this paper is the creation of a trusted process for signing a document within an open computing platform. The process makes use of a trusted display controller and a smart card.

The trusted process is designed to ensure that the document displayed on the screen is the same as the document sent to the smart card for signing. The process fails in (at least) the following situations:

1. The platform has a mistrusted display controller, so that there is no guarantee that the process of signing a document is invulnerable to malicious software on the platform. In this solution, the authentication to the TDC can help the signer to distinguish between a trusted display controller and a mistrusted one. As mentioned in Section 2, a certificate authority must check the criteria of the TDC design and implementation before issuing an assertion of a TDC. This will guarantee that a TDC with a valid certificate is a genuine TDC. If the signer is aware that the platform has a dubious display controller (for example Protocol 1 in Section 3 aborts), the signer should not use the platform for signing.
2. There is an insecure channel between the TDC and the smart card. It is not necessary to assume that the communication channel between the platform and the smart card is invulnerable to interference by a malicious entity. But, in the solution, the channel should be protected for the purpose of confidentiality, if requested, as described in Section 3. If the malicious entity only listens to the information through the channel, they cannot get any useful information during the signing process, because all message flows are protected with encryption. If the malicious entity is able to block the channel and modify the information through the channel, they cannot provide any information that will be accepted by either the trusted display controller or the smart card since all information is protected with data origin authentication.
3. The seal image may be compromised before execution of the protocols in Section 3. In the solution, we emphasize that it is the signer's responsibility to update the seal image securely. It is recommended that the signer uses an unpredictable seal image, and that the signer changes the seal image for particularly sensitive signatures. However, in some applications, it may be impractical to change them often. The frequency of upgrading should be dependent on the sensitiveness of the signature and confidence to the signer in the platform and the environment of the platform.
4. An alternative method of obtaining signer confirmation is to use a hardware switch directly connected with the TDC instead of the pass phrase. When the signer is asked to make a confirmation of the signature, they simply operate the switch. The TDC receives this message and passes it to the SC.

6. CONCLUSION

There is a potential problem with the validity of a signature of a document signed within an open computer platform. This paper contains a proposal for a trusted process of signing a document. The solution necessarily involves the reliable display of data and a trusted interaction between a user and their smart card, which can be used for other applications

7. REFERENCES

[1] B. Balacheff, D. Chan, L. Chen, S. Pearson, and G. Proudler. Securing intelligent adjuncts using trusted computing platform technology. In *the Proceedings of the Fourth Working Conference on Smart Card Research and Advanced Applications*, pages 177-195, Kluwer Academic Publishers, 2000.

[2] D. Balfanz and E.W. Felten. Hand-held computers can be better smart cards. In *the Proceedings of the 8th USENIX Security Symposium*, pages 15-24, August 1999.

[3] R. Banerjee, et al. The case for smart cards (third edition). CRL Digital Limited, 1999.

[4] BBC. http://news.bbc.co.uk/hi/english/sci/tech/newsid_813000/813437.stm.

[5] Bruce Schneier. Digital Signatures are not signatures. In *Crypto-gram*. Nov 2000. http://www.counterpane.com/crypto-gram-0011.html#1.

[6] DTI. http://www.dti.gov.uk/cii/datasecurity/electronicsignatures/.

[7] Gouvernement Francais. http://www.internet.gouv.fr/francais/index.html.

[8] ISO/IEC 7816 (all parts), Identification cards - Integrated circuit(s) cards with contacts.

[9] ISO/IEC 9796 (all parts), Information technology – Security techniques – Digital signature schemes giving message recovery.

[10] ISO/IEC 10118 (all parts), Information technology – Security techniques – Hash-functions.

[11] ISO/IEC 14888 (all parts), Information technology – Security techniques – Digital signatures with appendix.

[12] MCCafee Information library. September 2000. http://vil.nai.com/vil/virusChar.asp?virus_k=98836.

[13] Schlumberger. Lloyds TSB chooses schlumberger smart ID cards for Internet based banking. www.slb.com/smart cards/news.

[14] TCPA. Trusted Computing Platform Alliance Main Specification Version 1.0. In www.trustedpc.org.

[15] Teleware. http://netnews.teleware.fi.

[16] Kehr, Posegga & Vogt (Deutsche Telecom); PCA: Jini-based Personal Card Assistant. See http://gemini.iti.informatik.tu-darmstadt.de/~kehr/research.html#cqre99

[17] Bern, BreidenBach, & Goldberg (Xerox PARC); Trustworthy Paper Documents.

[18] Dhamija & Perrig (U.California at Berkley); Déjà vu – A User Study using Images for Authentication; See http://paris.cs.berkeley.edu/%7Eperrig/projects.html