USING NGSCB TO MITIGATE EXISTING SOFTWARE THREATS

Matt Barrett
mbar116@ec.auckland.ac.nz
Department of Computer Science
The University of Auckland

Clark Thomborson
cthombor@cs.auckland.ac.nz
Department of Computer Science
The University of Auckland

Abstract We introduce Microsoft's Next-Generation Secure Computing Base (*NGSCB*), and present a novel metaphor to describe it. An existing software application providing electronic legal document services is discussed, and results of a security analysis presented. The existing software architecture is extended with NGSCB to solve some noted security vulnerabilities. The novel architecture is then analysed for its successes and shortcomings.

1. Introduction

Microsoft is being watched with considerable interest as they continue to promote and develop their Next-Generation Secure Computing Base, alongside other industry heavyweights developing trusted computing platforms [Group, 2003]. Commentators are weighing in, with both technical [Spinellis, 2003] and philosophical [Stallman, 2002] arguments against the innovations being promoted by the Trusted Computing Group.

Whilst the debate surrounding the various philosophical and technical implications of NGSCB, and trusted computing as a whole, is being fiercely conducted, little has been said or done about its possible uses when integrated with existing applications.

It is expected by the authors that NGSCB will first appear in commercial desktops. This prediction is made based on the observation that the benefits of NGSCB will appeal most strongly to businesses that have an interest in conducting more secure electronic transactions with other businesses.

Conversely, it is also expected by the authors that NGSCB will find resistance in the home market. The same features that allow a company to secure its data and operations from an attack can be used to remove existing privileges that home users have with digital content. A home environment with an NGSCB-enabled system would enable content providers to implement powerful DRM systems, allowing secure end-to-end digital media delivery to the home [Brendan, 2001].

Previous attempts at securing digital content in a hostile environment have failed, and researchers attempting to secure software from modification or unauthorised duplication have stated that a successful solution without a hardware component is not possible [Martin and Marsden, 2003]. Microsoft will make use of the Trusted Platform Module (TPM) [Group, 2003] designed by the Trusted Computing Group, manufactured and provided by hardware vendors, including AMD [Strongin, 2003], to fill the role of this hardware component.

This paper presents a case study investigating the application of NGSCB to an existing software application. For NGSCB to prove useful to software developers with an existing product, it should be able to be integrated into an existing architecture without requiring extensive modifications. Indeed, if the modifications required to integrate NGSCB necessitated significant changes to the code base, it may be more cost effective for a development team to design and implement their own security enhancements, as opposed to using the more general toolset provided by NGSCB.

The application chosen for this case study was Electronic Legal (ELF) Forms [Forms, 2003], a product of the Auckland District Law Society (ADLS) [ADLS, 2003]. The Electronic Legal Forms application allows lawyers to work with electronic versions of pre-prepared legal documents. The Auckland District Law Society's hard-copy legal form products provide law firms in New Zealand with standardised, well-known documents to assist them in various legal transactions. It should be noted that the architecture described in this case study requires all involved parties to have NGSCB-enabled platforms.

This paper is organised as follows. The second section will introduce NGSCB, illustrating some features and describing its operation through use of a novel metaphor. The third section will describe the Electronic Legal Forms software. Its purpose, features and security vulnerabilities and the security goals of integration with NGSCB will be outlined. The fourth section presents an integrated architecture, and illustrates possible uses of NGSCB. Section five contains discussion of the success and shortcomings of the NGSCB integration.

2. Microsoft's NGSCB

A detailed analysis of the Next-Generation Secure Computing Base is outside the scope of this paper. A good introduction to NGSCB can be found on Microsoft's NGSCB webpage [Microsoft, 2003b].

2.1 Novel Security Primitives

NGSCB provides four new security primitives to Windows application developers, through a number of hardware modifications described in [Microsoft, 2003a] by Microsoft. These four new security primitives are sealed storage, secure IO, strong process isolation and attestation.

Attestation is the most novel of these four security primitives. It allows the security boundary from a nexus computing agent (NCA) running on one machine to extend to and include that of another NCA running on another machine, with communication taking place over an insecure channel such as the Internet. Further explanation of attestation, and nexus computing agents, can be found in section 2.2. This allows applications to trust a remote application to perform in a correct manner, despite it being located on a machine administered by possibly malicious users. Using this base, policy projection from one computer to another can occur, enabling Digital Rights Management (DRM) style applications to be built.

It is described as the most novel, because of all four new security primitives provided by NGSCB, it alone enables a new class of secure application to be built. Sealed storage simply advances upon features that come from file systems with access control, but shifts authorisation from being user-based to program-based. Secure input and output are novel features, but do not allow new applications to be built until secure peripherals, other than video, mouse, and keyboard become available. Additionally, there are already a number of software-only methods available to prevent screen scraping in Microsoft Windows. Strong process isolation merely provides in hardware what has been provided in some degree by operating systems for a number of years. Whilst it makes virtual memory protection much more secure, most current applications are written on the assumption that their memory address range is protected from other programs.

Attestation, however, allows remote cryptographic verification of not only the executing program but also the call stack, all the way down to the hardware level. This is a powerful new security primitive, creating new levels of assurance for computations performed by remote computers, and allowing administrators to project policy to remote platforms. Attestation is described in various levels of detail by England et al and in the various white papers [Trusted Computing Group, 2003, England et al., 2003, Microsoft, 2003d] published by Microsoft [Microsoft, 2003b]. It makes use of a Trusted Platform Module

(TPM), along with various cryptographic certificates, to prove that a specific hardware and software stack is NGSCB-enabled, and can be trusted to operate as expected.

2.2 A Hierarchy of Trust

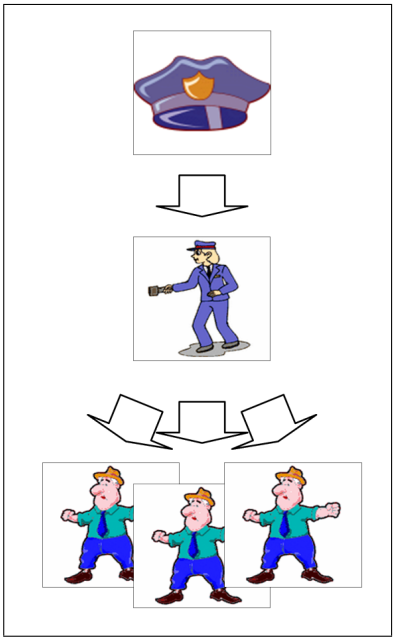

Figure 1. A Three-Layer Hierachy

Without a strong technical understanding of NGSCB, it can be difficult to imagine how the system will operate when in widespread use. As is often the case with technical subjects, it is useful to develop a metaphor for the NGSCB system. The NGSCB environment that runs inside a single machine can be thought of as a hierarchy of three enforcement agents. The enforcement agents running inside a system have differing levels of authority. They are identified by their badge numbers, which are generated for each agent, by the agent directly above it in the hierarchy.

The badge number can be thought of as a code ID or manifest. This is a hash of the binary executable of the program. The TPM hardware chip can be thought of as a police officer; an executing nexus can be thought of as a security guard; an executing NCA can be thought of as an ordinary citizen. This three-layer hierarchy is illustrated in Figure 1.

The nexus is the trusted kernel that hosts those programs specifically written to take advantage of NGSCB security primitives. These are referred to as

nexus computing agents (NCAs), and run separately from both other NCAs and normal programs.

The police officer can be trusted to enforce the operating restrictions of NGSCB as it is fixed in hardware. The police officer's validity is proven by the certificate of the manufacturer, which it can provide upon request. This certificate attests that it has been built to an approved, published standard.

The police officer, or TPM, will allow any security guard, or nexus, to run on the platform. The police officer can be guaranteed to provide, when asked, the correct badge number of the security guard, which can be used to ensure that the security guard is who he claims to be. In turn, the security guard will attest to the identity of any citizen, or NCA, it is watching or hosting.

When using the seal primitive, the three identities of the police officer, security guard, and citizen work together. In the common case, of a citizen sealing secrets to itself for later use, their unique identities ensure that the data can only be unsealed when the same police officer, security guard, and citizen are present. This means the data cannot be unsealed on a different NGSCB platform, where the police officer, or TPM, differs. It also means that the security guard and citizen will be the same executing binaries when the data was sealed, as when it is unsealed.

Despite the open nature of the NGSCB system, that according to Microsoft will allow anyone to write a nexus and NCA themselves [Grawrock, 2003], this hierarchy of identities will remove some of the freedom of choice that may initially appear to be present. Users must run security guards whose identities are known by the parties to whom they are attesting. This can be easily illustrated by describing a current application that could make use of attestation — securing online banking — described in [Microsoft, 2003d].

The bank server, running under a known police officer and a known security guard, receives the attestation vector from the remote client bank application. The vector contains the identity of the application, or citizen, as well as the identity of the security guard under which it is running. These identities are certified by the signature of the police officer, who is implemented in hardware and is considered trusted by all parties. An alternative is that the two identities are certified by a third party, who is trusted by the bank to only issue certifications to NGSCB platforms — this process allows some degree of anonymity. The bank server will only communicate with the remote application if both the identities of the security guard and the application are as expected.

In effect, NGSCB places a police officer inside the current PC architecture that can be trusted to ensure that the operation of any security guard or citizen is assured, and cannot be modified during execution. The end user is in full control over what security guards and citizens are allowed to run on their system, and what information the various entities can reveal. However, in order to make use of citizens, or NCAs, to obtain a service or perform some commu-

nication, a specific security guard, or nexus, must be running under the police officer. No details are available as to how many nexus will be made available for use. It is expected that a Microsoft-written nexus will be distributed, and used by default. It will be this security guard that citizens (provided by software vendors) will need to run under, in order to be identified correctly.

3. Electronic Legal Forms

3.1 Business Usage

The Auckland District Law Society's Electronic Legal Forms application is intended for use by lawyers and legal professionals to prepare legal forms for use in a variety of transactions. The product was developed seven years ago to allow law offices to move away from the dated practice of legal secretaries working with manual typewriters, filling in the various fields of a hard-copy legal form template. The user is able to work with an electronic version of the template, filling in the required fields and removing clauses that are not required for a specific document. By moving to an electronic format, forms can be partially completed, returned to later, and then finalised before printing and signing by the client.

3.2 ELF Features

In the current implementation, electronic stock is stored on a hardware dongle attached to the user's computer, or in the case of larger firms on a network server. This dongle is referred to as a Software Authentication Button (SAB). The Electronic Legal Forms application connects to the SAB whenever a final copy of the document is printed. The software ensures that sufficient reserves of stock are present on the SAB. The stock is decremented by the appropriate amount, and the legal form is printed. This process allows collection of revenue by the ADLS.

The Electronic Legal Forms package allows users of the software to send under-revision, or finalised, copies of documents to each other. The editing process is conducted through a GUI, shown in Figure 2. User-modifiable fields are shown in grey, and boilerplate legalese is viewable as black-on-white text.

After creating or editing a legal form, users simply transmit a string, obtained from ELF, which encodes only the fields that contain user-modifiable text, through e-mail to the other party. There, the encoded string is inserted into Electronic Legal Forms. For example, the name and address of one party can be entered by their lawyer, after which the document can be sent to the lawyer of another party, who can update the required fields with their own client's name and address. Once the legal form has been finalised to all parties' satisfaction, the form is marked as such. After this point in time, ELF no

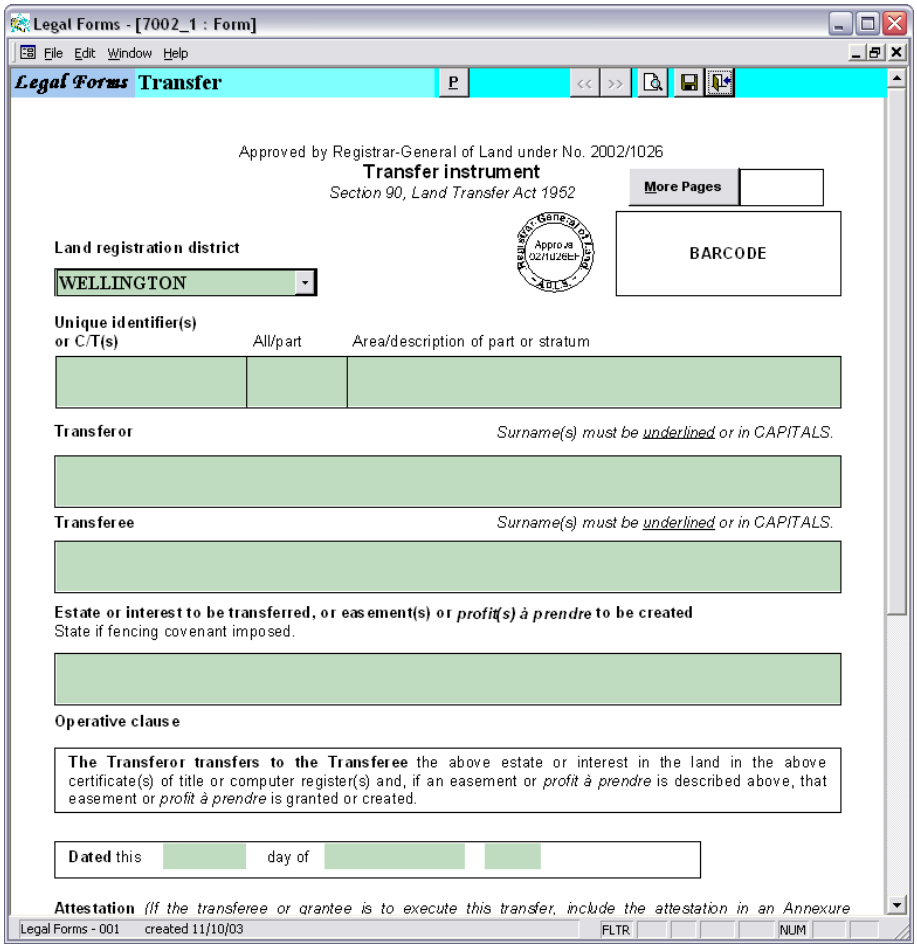

Figure 2. Electronic Legal Forms interface

longer allows modification to the user-modifiable fields. This process requires a copy of Electronic Legal Forms at both ends of the communication, as the information that makes up the body of the legal form — information that should be a facsimile across all versions of that legal form — is not transmitted.

3.3 Security Goals

The implementation described above illustrates an important design goal of Electronic Legal Forms. The exact wording and formatting of a legal document is of the utmost importance.

EMAILING ELECTRONIC LEGAL FORMS

Many users have asked whether it is possible to email a legal form to a non Legal Forms user. The answer is yes, but for security reasons the facility is not included within the package.

If you have access to the full version of Adobe Acrobat, or a simple PDF printer driver, then all you need to do is change your default printer in Legal Forms to use this instead, and print the file as normal. A PDF version of your file will be created, and can be emailed to anyone. Please note that by default, there is no security on a PDF file, this means that with access to the full version of Acrobat the content of the PDF can be altered, which in this case could possibly be the clauses within a form. The amended file can then be re-saved with the alterations intact.

For ADLS it is paramount that any forms distributed between law firms are as complete as when they were first printed. This ensures that you, the lawyer, can be confident that a particular word of a particular clause will always appear in the same place on the same page each and every time. Being able to guarantee this reduces the need to carefully re-read the clauses on forms that may be printed or received from other lawyers, unless absolutely necessary.

Unfortunately with there being no default security on a PDF file, ADLS can no longer guarantee that this is going to be the case. It is therefore essential that if anyone wishes to create a PDF version of the form using one of these methods, that they independently set the security passwords on each form that is produced. This will prevent any potential conflicts that may arise over a form being signed that could otherwise differ from that originally generated. Setting the security options will prevent any unauthorised access to the content of the form, and therefore once again ensure that the content is accurate.

It must be stressed though that the setting of the security is up to each individual who generates a PDF version of a form from within Legal Forms, and ADLS are not responsible for the content of the form once it leaves the package in PDF format.

Figure 3. Instructions to ELF users taken from August 2003 Auckland District Law Society newsletter

Legal forms that include the Auckland District Law Society letterhead are widely accepted amongst lawyers as being correct replications, and are favoured because they do not require proof reading at each use. The Auckland District Law Society guarantees the veracity of their legal forms supplied in hard copy format, and of final copies that are printed directly from the Electronic Legal Forms package.

This is one of the primary goals of the ADLS, and can be found in their own words in paragraph 4 of Figure 3. From this, a security goal that users of ELF, after NGSCB integration, are able to transmit documents in a format whose veracity can be guaranteed by the ADLS is derived. This is defined as $G1$ in table Table 1. This is a goal of the (non-malicious) lawyers who use ELF (hereafter referred to as primary users). It is also a goal of the (non-malicious) secondary users, defined as those with whom a primary user communicates but does not have an ELF installation. It is not strictly a goal of the ADLS, but due to interest in their clients' satisfaction, may be considered one.

The ADLS also requires the payment of an appropriate fee for each legal form that is printed. Again, the implementation described in section 3.2 allows the ADLS to collect the appropriate fees for each printed hard copy of a legal form. This is defined as $G2$ in Table 1. This goal is only pertinent to the ADLS, as primary or secondary users of ELF are not strictly concerned that revenue is collected for each print.

The Auckland District Law Society recently noted an increase in requests from clients to be able to email copies of legal forms to users who do not have ELF. This issue was addressed in an August 2003 newsletter, the relevant parts of which are reproduced in Figure 3. The ADLS is concerned that PDF is seen by many as a way to send a high quality document that cannot be easily modified to others. These requests lead the ADLS to look for a document format that can be transmitted like a PDF, yet retain goals $G1$ and $G2$. This goal is defined as $G3$, in Table 1. This is a goal of both primary and secondary users of ELF as it allows them to communicate. Again, this is not strictly a goal of the ADLS, but for the same reasons as $G1$, may be considered one.

Currently, printing to PDF from Electronic Legal Forms is possible using standard PDF printer drivers. Paragraph 2 of Figure 3 instructs ELF users how they may print to the PDF format. Paragraph 3 points out to users that there is no default security in a PDF file. Despite user education, the Auckland District Law Society is concerned that its users may enjoy a false sense of security regarding the static nature of a document printed to PDF. Without the appropriate security restrictions put in place at the time of authoring, a PDF can be modified with ease. The Auckland District Law Society is aware of this, and is not willing to provide the same guarantees to a document's veracity once it has been transmitted in PDF format over an open channel. Paragraphs

4 and 5 of Figure 3 show that ADLS strongly deprecate the use of PDF to transmit legal forms.

An additional security concern, inherent in electronic communication, is the ease with which a confidential legal document can be sent via e-mail to unauthorised parties. Working with only hard copies of legal documents severely restricts the distribution of highly confidential information to unauthorised parties, both by accident and through malice. If the use of PDF to store and transport legal documents via email increases, mistaken or malicious transmission to unauthorised third parties will also increase. It is viewed as highly beneficial by the ADLS [Martin and Marsden, 2003] to be able to impart DRM-style viewing restrictions to an authored document. Ideally, a closed set of relevant parties could be added to a legal document, with other parties unable to view the document. This is defined as $G4$ in Table 1. This is a goal of both the primary and secondary users of ELF. Again, it is not strictly a goal of the ADLS, but due to interest in their clients' satisfaction, may be considered one.

It is worth drawing comparisons between the security goals outlined in Table 1, obtained by analysis of ELF, and Pfleeger's [Pfleeger, 1997] three arms of computer security: confidentiality, integrity, and availability (CIA). Goal $G1$ maps directly to the integrity of the legal form. Goal $G2$ is a special case of Pfleeger's availability — a restricted DRM-style availability. Goal $G3$ is a standard availability goal. Goal $G4$ is a confidentiality goal. This CIA mnemonic will be used in section 5.1 to draw conclusions about the success or failure of the proposed architecture to satisfy the stated security goals.

3.4 Security Threats

In the current ELF architecture, certain threats to the defined security goals arise due to an ability to print to the PDF format from within Electronic Legal Forms. Additionally, a number of threats arise from the manner in which a legal form is transmitted between two users of ELF, as described in 3.2 above.

It is possible to prevent the installation of printer drivers on an administered Windows machine, and thus restrict the ability to print to PDF from Electronic Legal Forms through a PDF printer driver. However, this form of restriction is not possible when the program is installed on machines not under the administration of the Auckland District Law Society.

A PDF document can be re-printed without any limitations by anyone who obtains it. As described previously, the transferral to hard copy of an electronic legal form is a considerable and important source of revenue for the Auckland District Law Society. The ease of printing to PDF from Electronic Legal Forms allows two paths for violation of $G2$ (Table 1).

The first is the casual printing of a legal form that has been sent to a secondary user by a primary user, or a malicious third party who happens upon

the document through other channels. They are able to print a copy for themselves, indistinguishable from a copy printed directly from Electronic Legal Forms for which revenue was collected. The print operation occurs outside the control of an ELF installation, resulting in inability collect revenue for the print. This is noted as $T1$ in Table 1. A print operation is considered controlled if the appropriate fee is paid at some point.

The second comes from the removal of the personalised text, such as names and addresses, from the PDF. This process, performed only once, creates a blank template. This template can be used to avoid the need to purchase legal forms from the Auckland District Law Society. This threat is noted as $T2$ in Table 1. This threat comes from primary and secondary users, as well as from malicious third parties.

In addition to allowing printing without restriction, a standard PDF file also allows modification of the document itself. This opens the document up to threats $T3$ (modification of the legalese boilerplate, as defined in section 3.2), and $T4$ (modification of the user-modifiable fields). These threats come from a malicious third party who is able to intercept, modify and re-inject the document on its way from a primary to a secondary user through an insecure channel. Additionally, primary and secondary users are able to modify the document, calling into question the accuracy of both parties' copies.

In comparison, transmission between two or more users of ELF (section 3.2), where the legal form is never printed to PDF, results in only $T4$ able to occur. In this situation, only primary users are involved in the transmission of the legal form. In fact, it should be noted that if a form is never printed to PDF threats $T1$ and $T2$ can not occur. However, as previously mentioned, it is impossible to prevent a legal form from being printed to PDF. This issue is addressed in section 5.1. It is possible, however, for the string transmitted between two primary users across an insecure channel to be modified by a malicious third party able to intercept, modify and reinject it. This is noted as $T4$ in Table 1.

4. Integrated Design

To make full use of the security that can be implemented with NGSCB, a Public Key Infrastructure (PKI) built around NGSCB is proposed. Then an NGSCB-enabled legal form viewer that allows controlled distribution of legal forms is described. Finally, a simple architecture for allowing controlled remote printing to occur is described.

4.1 PKI and Attestation

Many of the weaknesses of a PKI come from being unable to control the enrolment of parties into the scheme, and being unable to verify their identi-

GOALS		
	Description	*Goal Of*
$G1$	*"...a particular word of a particular clause will always appear in the same place on the same page..."*	Non-malicious primary and secondary users
$G2$	Every print operation of a legal form is controlled by the ADLS, allowing collection of the appropriate revenue	ADLS
$G3$	A legal form can be viewed on a computer without an ELF installation.	Primary user, secondary user
$G4$	A legal form can only be read by the intended recipients(s)	Non-malicious primary and secondary users

THREATS		
	Description	*Threat From*
$T1$	Uncontrolled printing of a finalised form	Primary and secondary users, and third parties
$T2$	Creation of an electronic template of a legal form	Primary and secondary users, and third parties
$T3$	Modification of the legalese boilerplate on a legal form.	Malicious third parties, and malicious primary and secondary parties
$T4$	Modification of the user-modifiable fields on a legal form	Malicious third parties, and malicious primary and secondary parties.

Table 1. Security goals and threats of ELF

ties when doing so. With Electronic Legal Forms, administered enrolment is possible when a copy of the software is purchased and installed by a law firm. When discussing the new design of the ELF architecture it will be referred to as New ELF (NELF).

The root of trust is a master server Lf, or certificate authority, administered by the Auckland District Law Society. The installation procedure of a copy of NELF at a primary user's site involves the generation of a public/private key pair, k_i/k_i'. This key pair is for the sole intended use of participating in the ADLS controlled PKI. The private key is stored, using the NGSCB seal primitive [Microsoft, 2003d], on the local computer, C_{local}.

Once this is done, the newly installed copy of NELF contacts the ADLS server. The procedure for establishing a trust relationship between two NCAs on different computers is described in the Microsoft white paper concerning software authentication [Microsoft, 2003c]. In this situation, the two NCAs in question are the ADLS administered NGSCB-enabled ADLS server, and newly installed NELF NCA on the primary user's computer.

One difficulty with automatically creating a trust relationship between a NELF installation and the ADLS server is establishing network communication in heterogeneous corporate environments. A secure communication is required between the two parties who are expected to be located behind various layers of network and application security. It is feasible to perform the required communication over the HTTPS protocol — which is widely available on corporate desktops, and allowed through corporate firewalls.

An initial communication takes place, most likely over HTTPS. The HTTPS protocol ensures the integrity of the communication, and the confidentiality. The primary user, through checking the server's PKI certificate, will authenticate the server. For this communication, the previously generated public key, k_i, is attested by the nexus running on the primary user's computer. It is then transmitted, along with other cryptographic information used to verify the NGSCB platform itself and the NELF installation program to the ADLS server. This extra information is used to verify that the NCAs that are communicating with each other can be trusted to operate as expected, i.e. they are executing on a valid NGSCB platform, as described previously in section 2.1 above.

The ADLS server signs a certificate, C_k, identifying the public key, along with information concerning the primary user itself, A_i, most likely a contact address or other information of interest to users. This certificate is returned to the NCA at the primary user's site. In order for identification to be established in both directions, the process is repeated, with the two NCAs reversing their roles. Once this process has been completed for the primary user, it is considered enrolled into the ADLS PKI. This protocol is outlined more succinctly in Table 2, as PKI Enrolment and Attestation.

The NELF application is modified to present the identities of other primary users, A_x, who are enrolled into the PKI, when preparing a legal form for electronic transmission. This directory listing would be retrieved from the ADLS server when the primary user's NELF installation first enrols into the PKI, and periodically thereafter, to maintain a fresh listing. A legal form would then be encrypted with the published public key(s), k_x, of the respective primary user(s), A_x, to which it addressed.

Due to the forced enrolment during installation, and the inclusion of suitable identification information, future communications are able to take place between law firms in confidence. Revocation is controlled by the ADLS. Regular updates by primary users of their local certificate stores will reduce the likelihood of a compromised key continuing to be trusted.

4.2 Widget

Further integration of NGSCB with NELF occurs through the development of a widget, similar in functionality and use to Adobe's Acrobat Reader. This widget re-uses the internal document format and existing form editor of Electronic Legal Forms as shown in Figure 2.

As described previously, the current version of Electronic Legal Forms allows two users of the product to transfer under-revision or finalised legal forms between themselves. This functionality would not be removed when integrated with NGSCB, but would be restricted in order to address $G4$ — preventing viewing of a legal form by unauthorised parties — with a PKI as described in section 4.1.

4.3 Lightweight DRM Wrapper

Currently, when a user wishes to send a legal form by email to a client they print the form to PDF, which is then emailed to the client. Under the new architecture, this process is still the same. However, instead of a PDF being generated, an encrypted version of the legal form is generated. This legal form can only be viewed with the NELF widget.

This NCA widget has a limited set of functions, and can enforce a number of restrictions, such as an inability to print the form. It is similar in appearance and usability to the document viewing and editing component of Electronic Legal Forms.

In order to ensure the confidentiality of the transmitted document ($G3$) additional trust relationships must be established. A trust relationship is established between a primary user (a law firm), Lf that uses NELF and any secondary users (clients), Cl_{1-n}, to whom a legal document needed to be distributed. This would take a similar form as between the PKI rooted at the ADLS administration server, and primary users using ELF.

PKI Enrolment and Attestation	
1	Root of trust created on ADLS administered server Lf. Public/private key paid j/j' generated.
2	Installation of NELF at law firm. Public/private key pair k/k' generated, and stored on C_{local} with NGSCB *seal* command.
3	New NELF install contacts ADLS server over HTTPS. C_{local} nexus attests to k, Lf nexus attests to j.
4	Lf signs certificate C_{kn}, including A_{kn} and k. A_k contains enough information to uniquely identify the law firm, most likely with name and addresses.
5	C_{kn} returned to C_{local} along with all other C_k certificates created for other law firms.
Message Transmission	
1	User picks certificate C_k of intended recipient from list presented, using A_k to identify them.
2	Legal form is encrypted with public key C_k, and emailed to electronic address specified in A_k.

Table 2. Protocol Steps

The widget installation file from the primary user's NELF computer is distributed to the client. The client, $C1$, upon reception through email of the widget from a trusted party — namely their law firm Lf, simply executes it. It is expected that local user interaction and authorization will be required to allow an NCA to execute on a computer. The exact manner in which this will occur has not yet been finalised by Microsoft. Additionally, NCAs are likely [Cram and Kaplan, 2003] to execute in a sandboxed environment, with a user-customisable set of restrictions placed upon them.

Once the user authorises the execution of the widget, a trust relationship must be formed between the secondary and primary users. The secondary user's NELF widget installation generates a public/private key pair, k/k'. The public key k of this pair is presented in an emailtransmittable form to the user. It is then emailed to the primary user Lf, which records the public key in their local ELF system. This process could easily be automated, so to appear transparent to the primary and secondary users. No trust relationship is established in the reverse direction, as none is required. The NELF viewer widget serves only to display the documents; it does not allow any editing or formatting to take place.

At this point, the widget has been installed, and the newly generated public key returned to Lf. The main NELF installation at Lf can then send legal forms encrypted with the appropriate public key of the intended recipient.

Documents are prepared for sending to secondary users with the installed widget, just as a form is currently prepared for printing to PDF. A primary user can create a copy of the legal document encrypted for the relevant secondary user. The NELF program would present a list of known widgets that have been distributed and installed. A primary user can select the secondary users to whom they want to distribute the document. The document would be encrypted, and the primary user would simply email the file to the secondary user. There, the preinstalled NCA widget would be used to display the document securely.

This procedure illustrates the ability to create a secure, one-way trust relationship between two NGSCB platforms without the need for a hierarchical PKI that is created by attestation. Once again, however, the NGSCB platform verifies the NCA has not been modified, and can be trusted to maintain the policies applied to any legal documents sent to it.

The architecture described here presumes all parties involved have NGSCB platforms upon which to execute the NCAs. This is a major shortcoming, and is noted in section 5.2.

This architecture illustrates the use of NGSCB to project policy restrictions onto a remote computer to protect an electronic document. It can be seen as a lightweight DRM application, capable of protecting high value documents, the integrity of which both parties have an interest in.

4.4 Printing and Replay Attacks

The design is further extended to allow $G2$ (the collection of revenue for all printed legal forms) to occur at secondary users' sites, as well as primary users'.

The secondary user is able to print a restricted number of copies of the legal form under certain conditions. When a document is being prepared for transmission to a secondary user, a certain number of print credits must be attached to the document by the primary user, if the secondary user is to print that document. The ADLS collects revenue for these credited prints from the NELF primary user's account. The primary user can collect the cost of these prints from their secondary user through their regular accounting channels with that user. When the document is viewed by the widget, the print credits allow the secondary user a set number of prints. When a copy is printed, the credits are decremented, and the document securely updated with the new value.

This method of enabling remote pay-per-print is vulnerable to a form of replay attack. A secondary user who receives a document that contains a certain number of print credits may simply exhaust those credits, then replace the exhausted copy with the document they were originally sent.

A solution to this attack is to force the widget to contact a server, run by the primary user that distributed the document, in order to verify every print command. There are a number of problems with this solution.

Firstly, a primary user may not want, or have the capability, to maintain a permanent presence on the Internet. Secondly, even if each primary user provided such a server, each print operation would require a network connection to the server, which may not be possible for a number of reasons. An ideal solution would have some form of offline printing capability, as well as still ensuring that $G2$ is maintained.

In the system described, the NCA widget is able to store some uniquely identifying attribute of any document for which it generates a printed copy in its configuration set. Future attempts to print the same file will be caught by matching the unique attribute previously stored. It can be seen that this merely shifts the target of any attempted replay attack. Now, the configuration set, which has data concerning the number of times a certain document has been printed, is simply replaced with an earlier copy.

Discussions with members of the NGSCB development team [Ray and Cram, 2003] regarding this problem revealed a number of solutions under development. One interesting idea was the development of an encrypted NGSCB registry, which NCAs could use to store persistent state. If this was modified or deleted, the NGSCB platform itself could be engineered to stop working, preventing further access to the legal documents. In addition, counters such as those required by the NELF widget could be stored in multiple places, increasing the difficulty of simply replacing them with earlier values. While these solutions would not make the described replay attack impossible, it would increase the difficulty of such an attack.

5. Discussion

5.1 Satisfaction of Goals

In order to ascertain the success, or otherwise, of the NELF design after integration with NGSCB, we can review the original security goals as defined in Table 1.

Goal $G1$, concerned with the integrity of a legal form, is assured with public key encryption. All legal forms are encrypted with the public key of their intended recipients before transmission. The goal is met for all concerned parties: the ADLS, and primary and secondary users. Integrity is assured through the cryptographic strength of the underlying encryption scheme.

Goal $G2$, concerned with the DRM-restricted availability of printing a legal form, is the most difficult to satisfy. As noted by the ADLS in their newsletter (Figure 3), every user of ELF is able to print, through the addition of the appropriate print driver, a legal form to PDF. As printing to PDF is unable to be

prevented, it is important that the ADLS continue to inform primary users of the weaknesses of the PDF format.

Given the design described in this paper, which successfully reproduces the functionality given by using PDF — the ability to send legal documents to users with out an ELF installation — it is hoped that users will reduce their use of PDF. With increased use of the system outlined in this paper, with its high degree of confidentiality and integrity, it is hoped that any use of PDF to store or transport ADLS legal forms would be seen as malicious, or at the least, ill-informed. Given the legal community's noted [Martin and Marsden, 2003] willingness to report firms or individuals using obviously unauthorised hard-copy forms, it is reasonable to assume the same would occur with PDF forms, especially once informed about the risks inherent in the PDF format. Goal $G2$ can only be met for the ADLS if primary users discontinue their use of PDF.

Goals $G3$ and $G4$ are met for both primary and secondary users. The PKI established during the administered enrolment of primary users enables those users to encrypt legal forms with the public keys of their intended recipients. The PKI described has a tightly controlled enrolment process, increasing trust in the identities of those enrolled. It serves a primary user's interests to keep their key pair secret as a third party can use it to create legal documents purporting to come from them. Should a key pair be compromised, revocation is handled at a central site by the ADLS. This PKI allows the method of legal form transmission as described in section 3.2, to continue to be used amongst primary users.

To meet $G3$ and $G4$ for transmission between primary and secondary users, a trust relationship is established between every secondary user with whom a primary user wishes to communicate. A primary user is then able to encrypt legal forms with the public key of the specific secondary user to whom they wish to transmit a legal form. Confidentiality is strictly enforced by the system, as the private key generated during the secondary user's NELF widget installation is never released outside the NGSCB platform by the NELF widget itself.

5.2 Shortcomings

To find and evaluate shortcomings, it is possible to evaluate the initial threats against the new architecture.

As noted in section 5.1, it is currently impossible to restrict the ability to file. Despite the ADLS being able to collect a single charge for any form printed to file, this file (in PDF, PS or PCL format) can then be used to generate any number of hard copies. It should be noted that once in any of these formats, threats $T2$ (creation of electronic template), $T3$ (modification of legalese) and $T4$ (modification of user-modifiable fields) cannot be mitigated. However, if

the ADLS is successful in creating an aversion to using any format of electronic form transmission other than the NELF system described, these threats can be reduced. For example, all the noted threats from malicious third parties will be reduced, as they will not be able to obtain a copy of any legal form (guaranteed through attainment of the confidentiality goal). Without access to a copy, none of the denoted threats can occur from a third party.

The general problem of replay attacks outlined in section 4.4, causing threat $T1$ to occur from primary and secondary users, arises because the NGSCB platform has no form of persistent, secure storage. Discussions with Geoffrey Strongin, Platform Security Architect for Advanced Micro Devices [Strongin, 2003] revealed that a working group has recently been set up within the Trusted Computing Group to develop trusted mass storage. Persistent storage that protects files stored by an NCA from modification or deletion, unless authorised by that same NCA, would enable a general solution to replay attacks.

Threats $T3$ and $T4$, described in section 3.4, are minimized as much as possible by using a public key encryption standard. If, as hoped, no legal form is ever released outside a primary user's NELF installation without encryption, $T3$ and $T4$ from malicious secondary users can be reduced.

It should be noted that the initial release of NGSCB would not allow NELF to secure printed output. Discussions with Microsoft security staff [Cram and Kaplan, 2003] indicated that improvements in this area are expected. Such secure printing will not come directly from Microsoft, but from other vendors in the printer marketplace. It is hoped that this will allow the restriction of printed output to a hard copy printer.

The NELF architecture proposed relies on NGSCB to be present on all systems in the distributed environment. How soon, if ever, that this will occur is a question that cannot be answered in this paper. As stated in the introduction, NGSCB is expected to make inroads in the corporate marketplace first. As such, the ability to secure high value legal documents could be one of the killer applications needed to drive NGSCB uptake.

6. Conclusion

We have introduced Microsoft's NGSCB technology, and discussed it by way of a novel metaphor. It is hoped that this metaphor will be useful in explaining the concepts and architecture of NGSCB to those without a firm technical grasp of computer security. Trusted computing represents a fundamental shift in the way applications may operate, and it brings a number of dangers and benefits. It is imperative that consumers are able to make informed decisions about their use of the technology.

We have performed a detailed security analysis of an existing software application, and shown the source of a number of threats. After discussion with

relevant parties, we have arrived at a number of security goals. A system architecture was then developed to meet these goals, through mitigation of the noted threats. We have shown it is possible to reduce various security threats to an existing application by way of integration with Microsoft's NGSCB.

Such integration illustrates that it is possible to redesign an existing application to make use of the new security primitives provided by NGSCB, without being forced to redesign completely, discarding the existing usability and strengths of an application.

Acknowledgments

The authors would like to gratefully acknowledge the assistance and willingness of the Auckland District Law Society, as well as correspondence received from various members of the NGSCB development team.

References

[ADLS, 2003] ADLS (2003). Auckland district law society. Available from `http://www.adls.org.nz/`.

[Brendan, 2001] Brendan, C T S (2001). Protecting digital content within the home. *Computer*, 34(10):42–47.

[Cram and Kaplan, 2003] Cram, Ellen and Kaplan, Keith (2003). Next-Generation Secure Computing Base - Overview and Drilldown. Presentation at Microsoft Professional Developers Conference.

[England et al., 2003] England, P., Lampson, B., Manferdelli, J., and Willman, B. (2003). A trusted open platform. *Computer*, 36(7):55–62.

[Forms, 2003] Forms, Electronic Legal (2003). Electronic legal forms. Available from `http://www.adls.org.nz/shop/elf.asp`.

[Grawrock, 2003] Grawrock, David (2003). Tpm main part 3 commands. Available from `https://www.trustedcomputinggroup.org/downloads/tpmwg-mainrev62\_Part3\_Commands.pdf`.

[Group, 2003] Group, Trusted Computing (2003). Trusted computing group. Available from `http://www.trustedcomputinggroup.org/home`.

[Martin and Marsden, 2003] Martin, Marcus and Marsden, Simon (2003). Meeting.

[Microsoft, 2003a] Microsoft (2003a). *Hardware Platform for the Next-Generation Secure Computing Base*.

[Microsoft, 2003b] Microsoft (2003b). *NGSCB Product Information*.

[Microsoft, 2003c] Microsoft (2003c). *NGSCB: Trusted Computing Base and Software Authentication*.

[Microsoft, 2003d] Microsoft (2003d). *Security Model for the Next-Generation Secure Computing Base*.

[Pfleeger, 1997] Pfleeger, C. (1997). Is there a security problem in computing? In *Security in Computing*, pages 1–19. Prentice Hall, 2nd edition edition.

[Ray and Cram, 2003] Ray, Kenneth and Cram, Ellen (2003). Interview at Microsoft Professional Developers Conference.

[Spinellis, 2003] Spinellis, Domidis (2003). Reflections on trusting trust revisited. *Communications of the ACM*, 46(6):112.

[Stallman, 2002] Stallman, Richard (2002). Can you trust your computer? Available from `http://www.gnu.org/philosophy/can-you-trust.html`.

[Strongin, 2003] Strongin, Geoffrey (2003). Platform Security Architect, Advanced Micro Devices. Interview at Microft Professional Developers Conference.

[Trusted Computing Group, 2003] Trusted Computing Group (2003). TPM main specification. `https://www.trustedcomputinggroup.org/downloads/tpmwg-mainrev62_Part3_C%ommands.pdf`.