Secure Sharing of Electronic Patient Records

Surya Nepal, John Zic, Gregorie Kraehenbuehl, Frederic Jaccard

CSIRO ICT Centre
PO Box 76
Epping NSW 1710
Australia
{surya.nepal, john.zic}@csiro.au,
{gregorie.kraehenbuehl, frederic.jaccard}@csiro.au

Abstract: Sharing electronic patient records without compromising individual's privacy and confidentiality is a key challenge in adoption of distributed, electronic healthcare systems. To address this issue, it is necessary to ensure that a sender can trust the receiver and its environment to correctly implement its policy and vice versa, while sharing electronic patient records. This paper proposes an approach that can be used to establish a trust between two interacting parties based on trusted computing technologies, and describes its application and implementation in the context of distributed healthcare systems.

1 Introduction

There is a growing need of sharing patient records among healthcare facilities to provide effective, co-ordinated patient treatment and care. Due to the increasing use of information technologies in healthcare industries, patients' records are often easily produced and shared in an electronic form. However, the ease of sharing of personal medical information must be counterbalanced with the need to ensure the privacy and security requirements of the patient and facility. The lack of such mechanisms may have serious consequences for patients; for example, a leaked patient record to an insurance company may block them buying insurance products.

In order to address the issue of control of private medical information, the CSIRO ICT Centre developed an eConsent model [OK05] within the context of the Australian Government Department of Health and Aging (DoHA) Electronic Consent Project [DO02]. The term *eConsent* was coined to refer to a mechanism through which a patient can express their consent policies on their electronic records being accessed and shared between healthcare facilities. The eConsent model was developed and demonstrated to stakeholders from the Australian healthcare sector along with other three research and development projects commissioned by DoHA. The most important outcome of the project is the concept of "placeholders", which are used in novel, privacy-preserving, anonymous electronic record transfer protocols.

We make the following two observations on this eConsent model.

First, any underlying data model must support a good set of default consent statements and policies to ensure that eConsent model has a minimal impact on clinical workflow. The hierarchical data model used by the current eConsent model has limitations on expressing default set of policies as well as categorizing electronic patient records. This limitation led to the development of our new tag-based data model. This tag-based data model allows us store both electronic patient records and their consent policies [NE06] and helps categorize the patient information. It also allows the definition of default consent for a variety of domains such as individual, health care provider and facility. Further, the tag-based data model also supports patients' consent expression in terms of healthcare facilities, healthcare providers, their roles, and categories of medical records or any combination of them within a single framework.

Second, the sender facility has to trust the receiver facilities, and vice versa. Both facilities need to ensure and enforce the patients' consent for their electronic records when their records are transferred. That is, the sender, as well as authenticating the identity of the receiver, trusts the receiver has the right software and hardware infrastructure to enforce sender's policies on privacy and confidentiality at all times. Similarly, the receiver wants to authenticate the sender, and establish that it is always in the correct state when accepting any message. The current eConsent model relies on someone at the receiver facilities for appropriate expression of patient consent in the receiver facility's system. Anonymous transfer using placeholders allows the patient's consent conditions to be satisfied for records arriving in the receiving facility. However, the sender has to trust on someone in the receiving facility to appropriately express and attach the consent conditions to the "placeholders". This paper aims to address this shortcoming in the eConsent model by developing a mutual attestation protocol.

The issue of trust between interacting parties is not new, and there are many ways to address this problem. Most of the earlier works are based on software-based security, where the entire security features are implemented in software systems. These solutions are often sufficient, but not always adequate [TR03]. Problems with key storage and management, with secret keys that may be stored in an accessible persistent storage, still remain a challenge. In recent times, hardware-based solutions have gained momentum and a general, hardware based solution has been proposed by the Trusted Computing Group [TR03]. Similar, but specialised, systems have been developed, ranging from the use of a dedicated simple "dongle" attached to a serial port in early computers, to more sophisticated specialised military integrated hardware systems such as the Tenix "Coalition Network Exchange" [TE06].

In this paper, we enhance the trust between two interacting healthcare facilities in eConsent model by using a mutual attestation protocol that utilises the TPM devices and associated protocols between two server applications. We present the architecture of the new demonstrator built using TPM devices and explain the mutual attestation protocol between two healthcare facility server applications. We also report upon our integrity measurement architecture that uses existing hardware and software infrastructure. One of our system's key characteristics is its ability to support of different versions of software. Finally, we demonstrate the feasibility of mutual attestation protocol in distributed health care systems using some simulated functionalities.

2 Motivating Scenario

We consider an example distributed healthcare environment that includes two hospitals, one medical clinic and a government Department of Health as shown in Figure 1. The Department of Health issues credentials and permissions to operate the healthcare facilities such as hospitals and medical clinics. Each of these facilities operates and is managed autonomously, and has its own medical information systems. However, as is often required, these facilities share electronic patient records in order to provide effective services. For example, a doctor at a town's medical centre can refer a patient to a specialist at a regional hospital, and then transfer the patient's relevant medical records. Each facility has its own set of privacy and confidentiality policies for the patient's electronic medical records held in its information system.

Though the basic policies are setup by government rules and legislation, each facility may implement these policies differently. That is, each may have different set of policies for different categories of information. For example, Hospital A may have different set of policies for AIDS related records to that of another Hospital, B. However, it is a fundamental assumption of our paper and the eConsent work that a patient is an ultimate owner of their own electronic medical records, and their privacy and consent policies may differ from those of a particular hospital. For example, a patient can define a policy that all of his AIDS related records to be accessible to his doctor, and no other doctor in the hospital can access it except in case of emergency.

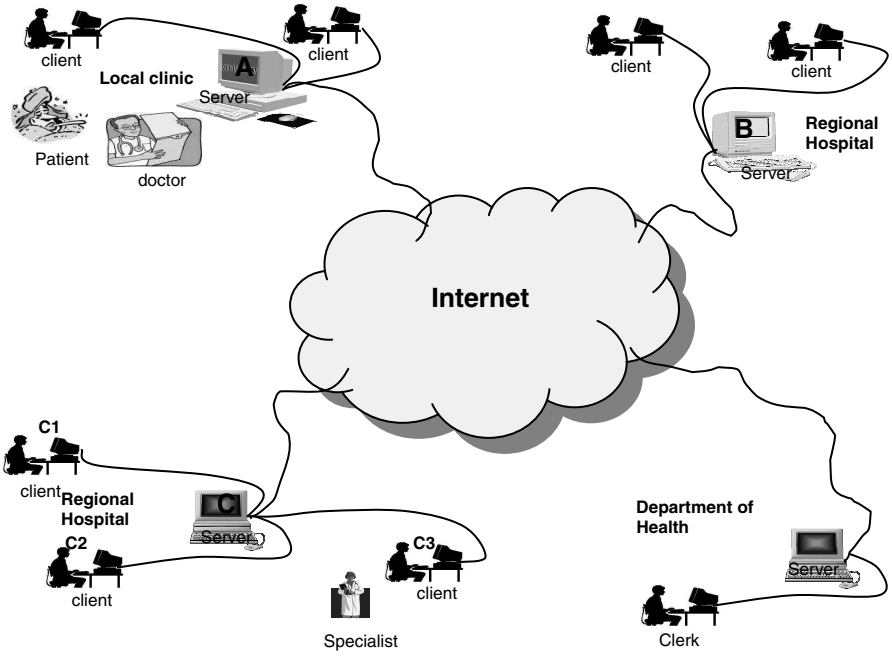

Figure 1: Distributed Healthcare Environment

We will further illustrate the problem of mutual trust using an example scenario where two healthcare facilities share electronic patient records. Sally is a regular visitor to a local town Clinic. Sally has given consent to access her medical records at the clinic to her family doctor. The doctor suspects that her recent illness is due to the rare defects on her DNA and takes her consent to refer her to the specialist at the nearest regional Hospital. She gives consent to transfer some of her previous test results to the specialist at the hospital and is asked to visit the specialist. When the records are transferred from the local Clinic to the regional Hospital, it is transferred in such a way that only permitted doctors, in this case the specialist, at the hospital can access the records. The local clinic trusts the host hospital information system of the specialist to implement Sally's policy correctly so that no other doctors can access her medical records.

Blood samples and DNA tests are performed when Sally visits the specialist. After analysis, the specialist will send those results to her family doctor at the local clinic. When the DNA test results are transferred to the local clinic, it should only be accessible to her family doctor. If such reports become public, she may be declined to access certain insurance products. The specialist trusts the receiver facility system such that it will correctly implement the Sally's consent so that no other doctors can access the test results. One way of addressing this problem is mutually attesting each others system states such that they ensure that Sally's policy is always respected.

3 The Mutual Attestation Protocol

This section explains the mutual attestation protocol and its implementation between two facility server applications in a distributed healthcare system. Figure 2 depicts the overall architecture and components of the "MedicClient", a prototype system developed to demonstrate the trust enhanced eConsent model. We first discuss the attestation protocols and then describe the implementation of Integrity measurement and Attestation components of the architecture. We refer the reader to [NE06] for the description of other parts of the architecture.

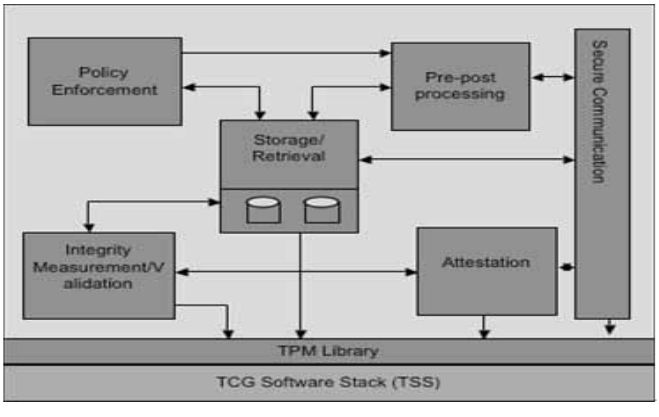

Figure 2: MedicClient Architecture Components

3.1 Attestation Protocol

The Trusted Computing Group (TCG) [TR03] is an industry standards body that defines a variety of specifications aiming to provide best practice software and hardware-based solution to "help users protect their information assets (data, passwords, keys, etc.) from compromise due to external software attack and physical theft" (from the TCG website). One of the TCG specifications defines the cryptographic hardware Trusted Platform Module (TPM), that serves as a root of trust for authenticating a host's software configuration and provides protection to data by never releasing a root key outside it.

Part of the TPM chipset contains a set of Platform Configuration Registers (PCRs) that are used in the measurement the platform environment. The measurement includes properties and characteristics of the programs such as integrity, configurations and running state. The PCR value keeps the integrity of the platform from boot time to loading operating system and finally loading applications. This is done by updating the PCR values, when the configuration is changed, applying SHA-1 to their current values concatenated with the new measured content.

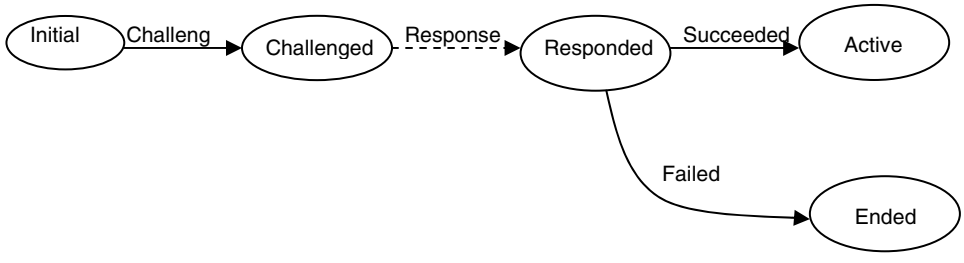

Figure 3: TCG Remote attestation protocol

The measured environment is exchanged between two parties using a remote attestation challenge-response protocol. This protocol is used to verify authenticity of the receiving party, as shown in Figure 3. During attestation, the challenger platform (sender) sends an attestation challenge to the receiving challenged platform. The challenged platform then signs one or more PCR values using an attestation identity key and sends this back to the challenger as a response. The attestation identity key is protected by the TPM and the signing is usually done using the special "quote" operation. This operation is supported by the software stack on top of TPM. The challenger then verifies the received signed values with the expected values to determine the authenticity of the challenged platform. After the successful attestation, the protocol reaches an active state, i.e., the challenger trusts the platform configuration of the challenged facility and that facility has a valid TPM; it can then send encrypted messages to challenged facility, knowing that only the challenged facility can decrypt them (through the use of its TPM).

The TCG attestation protocol can support the remote authentication of a specific, challenged platform. However, in a distributed healthcare environment, it is necessary to ensure *both* healthcare facilities in a transaction have valid TPM and are in the correct software configuration. That is, both the sender trusts the receiver and the receiver trusts the sender for each possible transaction.

We therefore extend the current remote attestation protocol and propose a mutual attestation protocol that permits two facilities to prove their validity to each other as shown in Figure 4. The full and dotted lines indicate the messages sent by sender and receiver facilities respectively. The mutual attestation protocol is as follows. First, facility A sends an attestation challenge to facility B. B then signs one or more PCR values representing its platform state using an attestation identity key and sends these values to A as a response along with B's attestation challenge to A A then verifies the received signed values with the expected values to determine whether the challenge is successful or not.

If the challenge is successful for A, it then signs one or more PCR values representing its platform state using an attestation identity key and sends to facility B as a response along with the successful result for its response to the earlier challenge. The receiver facility verifies the received signed values with the expected values to determine whether the challenge is successful or not. The successful mutual attestation ends at active state where they can exchange electronic patient records, whereas unsuccessful attestation ends at ended state. We next describe the implementation of this protocol in our MedicClient prototype demonstrator.

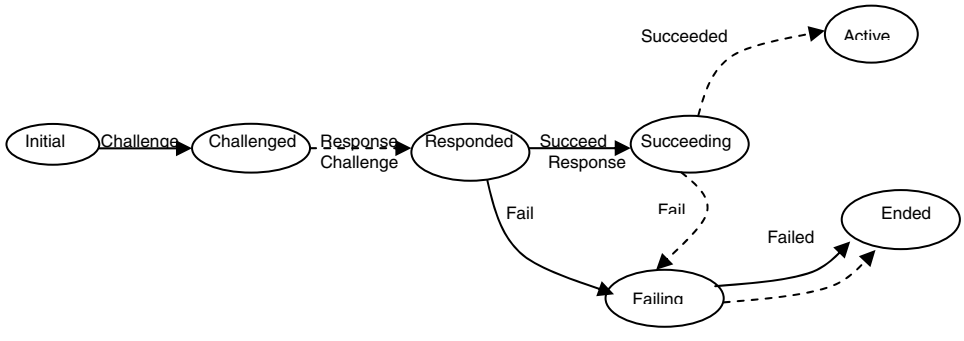

Figure 4: Mutual attestation protocol

3.2 Implementation

3.2.1 Facility Credential

As explained earlier, in our scenario distributed healthcare system, a government health department provides the operational credentials and licences for healthcare facilities. Our attestation protocol design requires both parties to validate the issued credentials. In order to support this, we introduce a *facility credential*, digitally signed by the Department of Health or authority, within the TPM credential architecture as shown in Figure 5. This credential is given to all facilities by the external authority after registration, and is linked together with the endorsement credential of the registered facility. The facility credentials include information such as registration date, expiry date, facility public key, facility registration number, and facility identification. They are used by Privacy CA to certify the facility during mutual attestation.

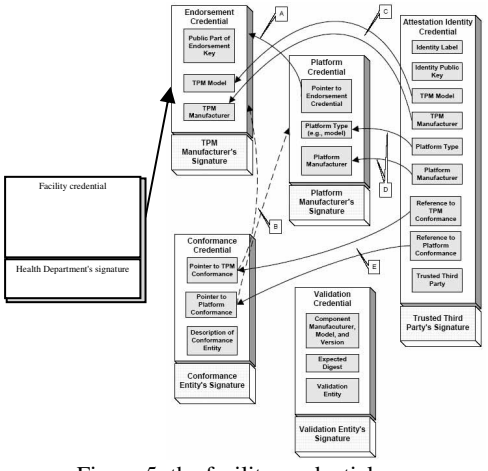

Figure 5: the facility credential

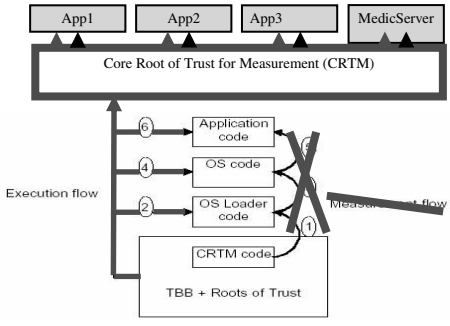

Figure 6: raising the level of trust (for measurement)

3.2.2 Integrity Measurement

The general principle of integrity measurement using the TPM is that any software is measured using the SHA-1 digest of the executable code before being loaded and stored in PCR registers. The first (loaded) software that does the measurement is called Core Root of Trust for Measurement (CRTM), which in our case, is the machine BIOS. Our development platform was based on Microsoft's current version of the Windows XP operating system, with the application development being done under the .Net environment. Despite the fact that the TPM chipset is found in most new PCs, the current version of Windows XP does not support integrity measurement from CRTM. Further, and as a consequence of this fact, there is no chain of trust that ties together the hardware BIOS to the application software.

As a consequence of the lack of support at the OS level, we simulated the core root of trust by developing a specialised loader, called the MedicLoader, as shown in Figure 6. This loader intercepts calls to the loaded and unloaded applications and performs measurements on the executables before their execution. A fundamental assumption to our implementation is that the MedicLoader can be trusted to securely measure itself and the processes it loads before executing them. This means all components that are under this root of trust must also be trusted. This is useful for a concept demonstration, but it is not a realistic solution for commercial applications. Realistic solutions must rely on OS support of both integrity measurement and being able to respond to a chain of trust.

Our integrity measurement mechanism using the MedicLoader as CRTM is as follows. For each executable file, we uniquely identify the executable, and capture whether it is loaded or unloaded. We do this by extending the PCR values as a triple: the state of the code ("load" or "unload"), the file name of the executable, and SHA-1 digest value of the executable. Our implementation of this triple is simply a concatenation of these three values and each of these resulting triples is also kept as a list as shown in Figure 7.

```
.....
load#WZQKPICK.EXE##7A3525B07DBB074B50B1A517C7EF67235B43E51A
unload#IGFXTRAY.EXE##94FE661D6661D87A921B152570DC246969BF50D7
load#IEXPLORE.EXE##7BBA110F375883290475D815C5E1128765092ABE
unload#IEXPLORE.EXE##427AAE0C922872D6DFB25C5A9C3FF0F959751E79
load#MSPAINT.EXE##908A8DBFC8C9ACD3E3071B3B5D4DE4A8F1E5F48E
.....
```

Figure 7: example of list of loaded and unloaded processes

Separately from this list, each facility keeps a database of processes that represents a register of "acceptable" processes states. Each entry consists of the name, a calculated hash value for the executable if it is loaded or unloaded and the desired executable load state ("can not", "can" "must"). An example database of processes is shown in Figure 8. 'Can' signifies that the process can be loaded and 'Must' signifies that the process imperatively must be loaded.

	processName	processHashLoad	processHashUnload	mode
28	IEXPLORE.EXE	7BBA110F3758832904 75D815C5E11287650...	427AAE0C922872D6DFB25C5A9C3FF0F9597...	Can
29	JUSCHED.EXE	2CFCD80C433ED6B317912347BE5A041F9D5...	8894FC922C841E9552086B3236B1B50D143...	Can
30	UPDATERUI.EXE	7A0D0E9B4BD6E4C2B752CA9D8E0238C3B84...	9FC73F9C8A4EAEECBD8C541AD603BEFD724...	Can
31	VPNGUI.EXE	2D4C118E0CF28BE51A1EB9F2E6F7EE4BEE4...	CE60DE1791EF09D6BB632DB4DDA15BFCD6A...	Can
32	OSA.EXE	41F3B0ECF31A924EE030483C6A25CCAB849...	30B8E3DEDDFFC41F445F1ABBD9058B8A1B1...	Can
33	READER_SL.EXE	5CB9D38E835A63AC38BA144FE995B50FD16...	567CCCDE6C092DA3F78A79D52C84170BE48...	Can
34	PRIVACYCA.EXE	FDFD67EAB3B7CBC3E4A0FC2C14070217249...	3C08E8B25E121EC9F6C508D5843E6AA2E2B...	Can
35	MEDICSERVER.EXE	33D259457BF070B8D17FCF3EBAD0AEE6BD1...	0E55058868B6FC12A23E8B277E7B816199F...	Must

Figure 8: example of the list of acceptable processes states.

We next describe the mutual attestation protocol between two server applications using this integrity measurement technique.

3.2.3 Mutual attestation

The TrouSers FAQ [TR06] shows a dependency graph of elements required to support remote attestation. Importantly, there are key elements that are missing in current infrastructure, and so it is not possible to use some of the core functions provided by TCG specification for remote attestation such as `CollateIdentityRequest` to collect TPM credentials or the PCR extend operation. Because of this, we have simulated these missing elements in our current version of the prototype demonstrator. However, our simulated functions can be easily removed when the TCG-enabled infrastructure is available.

TCG supports two remote attestation methods: one by use of a Privacy CA, and the other by Direct Anonymous Attestation (DAA) [YU05]. We chose the former and Figure 9 illustrates the implemented mutual attestation protocol. The following explains the steps perform in the protocol when facility A wants to send a message to facility B.

1. Facility A sends the message *challenge* to facility B with a random non-predictable nonce (*nonceA*).
2. Upon receiving the challenge, Facility B gets the TPM credentials of the facility B using TPM function `CollateIdentityRequest`. This function also generates an Attestation Identity Key (AIK). The TPM credentials, Facility Credential together

with the public part of the AIK are signed by the private part of the TPM Endorsement Key and then encrypted using the public part of the CA key.

3. The encrypted blob is then sent to the CA as an identity credential request.
4. The CA decrypts this blob with its private key. It verifies the correctness of the signatures on the credential request and then creates and sends an identity credential to the facility. This credential is a digital certificate containing the public part of the AIK together with a nonce, *nonceA*, signed by the CA private key.

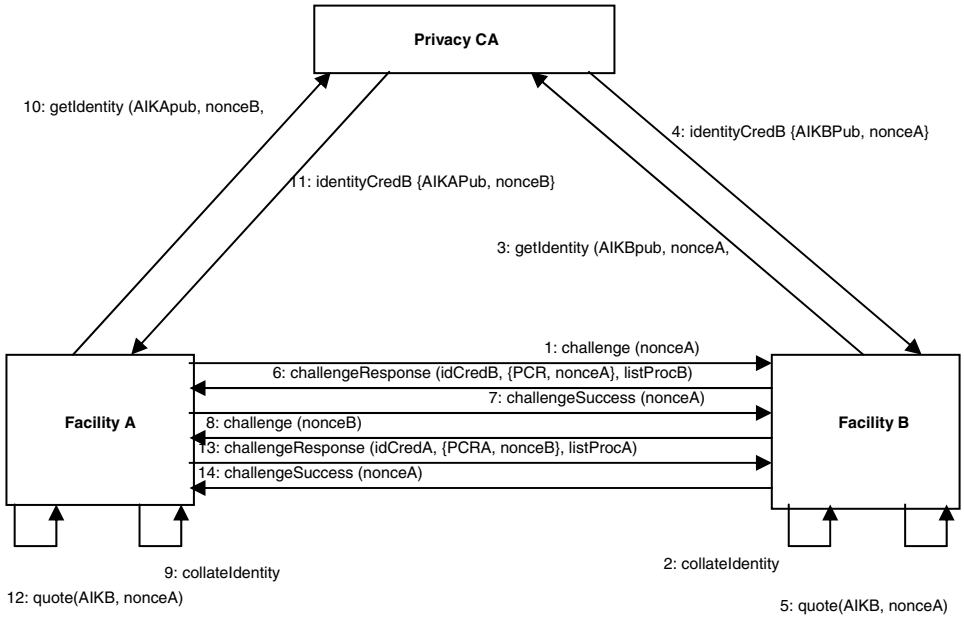

Figure 9: Mutual attestation protocol

5. Facility B then performs an integrity measurement using the TPM "quote" function using the AIK, *nonceA* and one or more PCR as input parameters. The AIK can only be used to sign the quoted PCR registers.
6. The message containing the output of quote function, the identity credential received from the CA and the list of recorded loaded and unloaded processes (as shown in Figure 7) is then sent to Facility A from Facility B. Facility A first checks the identity credential, verifying that the Privacy CA has signed it. This also reveals that B has a TPM and AIK is an identity key in the TPM of B. With this information, it can then verify the signature of the PCR values and *nonceA*. If the signature is successfully verified, then A knows the PCR values of B. Finally, Facility A does the following three verifications:
    a. All processes in the received list are in the database of acceptable processes (as shown in Figure 8) with corresponding hash code.
    b. The expected PCR value is computed from the list of processes and compared with the PCR values of B.
    c. All processes that are marked as MUST in the acceptable process database (see Figure 8) are loaded.

7. If all checks are successful, Facility A sends a challengeSuccess message to facility B. Otherwise, the attestation fails.
8. Facility B receives the challengeSuccess message, and initiates the same procedure (steps 8-14 in Figure 9). At the end of these steps, B knows the PCR values of A. and both A and B are authenticated.

4 Related Work

This section presents a review of related recent work on remote attestation. There have been recent works on architectures that support enforcement of an object owner's policy in client applications by attesting the platform and integrity of the requesting applications [SA05, SA04]. The policy enforcement architecture uses policy agent to enforce policies at the client. They have implemented such systems in a Linux environment using TPM remote attestation and integrity measurement technologies. Several problems have been identified related to remote attestation [HA04, RE03] such as (a) it says nothing about program behaviour, (b) it is static, inflexible and inexpressive (c) upgrades and patches to program are hard to deal with (d) it is fundamentally incompatible with a widely varying, heterogeneous computing environment and (e) revocation is a problem. There are many techniques that have been proposed to overcome these limitations such as semantic attestation [HA04] and property-based attestation [SA04b].

One of the problems associated with hardware-based solution is "isolated" execution of the programs. This problem has been partly addressed by the virtual machine concepts [BA03]. Garfinkel et al. [GA03] proposed a system, called Terra, which achieves the isolated execution of the programs on a separate, dedicated, tamper-resistant hardware platform using trusted virtual monitor. Another problem with remote attestation is variability of software versions and discrepancy in time-of-use and time-of-attestation. BIND [SH05] addresses these issues by (a) fine grain attestation that measures only the required code and (b) narrowing the gap between the time-of-use and time-of-attestation.

TCG first proposed remote attestation performed using trusted third party, called Privacy CA. An alternative approach to Privacy CA called Direct Anonymous Attestation (DAA) has been proposed as the method for remote authentication of a hardware module [BR04]. Yung [YU05] presented a through analysis of implications of remote attestation in trusted computing including Privacy CA, DAA and combination of both.

The most relevant work to our paper is the integrity measurement and attestation mechanism proposed by IBM in [SA04a]. To perform an integrity measurement, IBM has modified the Linux boot loader and the Linux kernel, to maintain the chain of trust from TPM to operating system to applications. The kernel maintains a list that represents the history of loaded processes and libraries. During attestation, the processes list is sent to the challenger, together with the reported PCR. The challenger can verify that whether the list is valid or not.

Our protocol differs from IBM proposal as follows. First, our approach is implemented in Windows environment to see what is possible to achieve within the current infrastructure. Second, our approach suppresses the need for the facility to load the processes in a particular order, as the order is impossible to maintain when software components are often loaded and unloaded by applications. Third, our approach gives more flexibility as we can state which process can, must or cannot be loaded for the attestation to be successful. Finally, our approach can be used (not) to force the facility to load the correct version of the software, as we permit multiple valid digests for a process in order to support multiple versions of the same software.

5 Conclusions and Future Work

We believe that our mutual attestation protocol between two server applications provides a reflexive form of trust between two parties needing to support enforcement of a facility's policies. We explored the bounds of what is possible in the current Windows environment, although the current Windows infrastructure lacks necessary elements for remote attestation such as chain of trust from hardware to operating system to applications.

We have developed a prototype system and demonstrated to CeNTIE [CE06] enterprise system focus group. The demonstration has led us to the belief that it is feasible to deploy such mutual attestation protocols between two server applications in a distributed healthcare environment to enhance the trust while sharing electronic patient records. We also observed that the current implementation of mutual attestation protocol suffers from poor performance as the protocol has to be run for each message sent or received by a facility. In future, we plan to tackle this problem by developing an application session-based mutual attestation protocol. Our preliminary study shows that such protocol will also address the problem of discrepancy in time-of-use and time-of-attestation.

Acknowledgements

The work is supported by the Australian Government through the Advanced Networks Program of the Department of Communications, Information Technology and the Arts. Discussion with Paul Greenfield of the prior e-Consent model greatly assisted us in developing the mutual attestation protocol. We would also like to thank Michael Cox from NTRU for helping us on using NTRU TSS driver, Alan Fekete (Sydney University) and Andre Schiper (EPFL) for encouraging Greg and Fred to take this project at CSIRO.

References

[BA03] Barham P., Dragovic B., Fraser K., Hand S., Harris T., Ho A., Neugebauer R., Pratt I., and Warfield, A. Xen and the art of virtualization**,** 19th ACM Symposium on Operating Systems Principles, pp. 164-177, NY, USA , 2003.

[BR04] Brickell E., Camenisch J. and Chen L. Direct Anonymous Attestation, Technical Report HPL-2004-93, 2004.

[CE06] CeNTIE project: The Centre for Networking Technologies for the Information Economy. http://www.ict.csiro.au/page.php?cid=22

[DO02] Australian Government Department of Health and Aging Project. Consumer consent in electronic health data exchange – e-consent. www.ahic.org.au/downloads/ Foundations%20for%20the%20future.pdf

[GA03] Garfinkel T., Pfaff B., Chow J., Rosenblum M., and Boneh D. Terra: A Virtual Machine-Based Platform for Trusted Computing. 19th ACM Symposium on Operating Systems Principles, pp. 193-206, Bolton Landing, NY, USA , 2003.

[HA04] Haldar V. and Franz M. Symmetric Behavior-Based Trust: A New Paradigm for Internet Computing. New Security Paradigms Workshop, Sept 2004;

[NE06] Nepal S., Zic J., Jaccard F. and Krachenbuehl G. A Tag-based Data model for privacy-preserving medical applications. EDBT IIHA Workshop, Munich, Germany, 2006.

[OK05] O'Keefe, C.M., Greenfield, P., and Goodchild, A. A Decentralised Approach to Electronic Consent and Health Information Access Control. Journal of Research and Practice in Information Technology, Vol. 37(2):161-178, May 2005.

[RE03] Reid J., Juan M., Nieto G., Dawson E. and Okamoto E. Privacy and Trusted Computing, 14th International Workshop on Database and Expert Systems Applications (DEXA'03) pp. 383, 2003.

[SA04] Sailer R., Jaeger T., Zhang X. and Doorn L. Attestation-based policy enforcement for remote access. Proceedings of the 11th ACM conference on Computer and communications security, pp. 308-317. Washington DC, USA, 2004.

[SA04a] Sailer R., Jaeger T., Zhang X. and Doorn L. Design and Implementation of a TCG-based Integrity Measurement Architecture. 13th Usenix Security Symposium, San Diego, California, August, 2004.

[SA04b] Sadeghi, A. and Stüble, C. Property-based attestation for computing platforms: caring about properties, not mechanisms. Proceedings of the 2004 workshop on New security paradigms, pp. 67 - 77 , Nova Scotia, Canada, 2004

[SA05] Sandhu R. and Zhang X. Peer-to-Peer Access Control Architecture Using Trusted Computing Technologies. Proceedings of the tenth ACM symposium on Access control models and technologies, pp. 147-158, Stockholm, Sweden, 2005.

[SH05] Shi, E., Perrig, A. and Van D. L. BIND: a fine-grained attestation service for secure distributed systems. IEEE Symposium on Security and Privacy, pp. 154- 168, 2005

[TE06] Tenix: Corporation's "Coalition Network Exchange": Australian Government Defence Depeartment web site describing Tenix product: http://www.defence.gov.au/teamaustralia/index1532.html

[TR03] Trusted Computing Group. Trusted Computing Platform Alliance (TCPA) Main Specification Version 1.1b, https://www.trustedcomputinggroup.org/groups/tpm/, 2003.

[TR06] IBM. Trousers, A TSS implementation for Linux. http://trousers.sourceforge.net/.

[YU05] Yung M. Privacy Implications of Remote Attestation in Trusted Computing, CSIRO ICT Centre Technical Report 05/146, 2005.