Sharing but Protecting Content Against Internal Leakage for Organisations

Muntaha Alawneh and Imad M. Abbadi

Information Security Group, Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK
{M.Alawneh,I.Abbadi}@rhul.ac.uk

Abstract. Dishonest employees, who have privileges to obtain corporate critical information and access internal resources, cause the problem of internal leakage. Employees, who have such privileges and know from where to obtain corporate sensitive information, are far more dangerous than outsiders. This paper proposes a mechanism for protecting information inside organisations against unauthorised disclosure by internal adversaries. It mainly focusses on sharing and simultaneously guarding information assets from one another. This paper proposes a novel solution for binding sensitive content to organisation devices, thereby preventing uncontrolled content leakage to other devices. In the proposed solution we used trusted computing technology to provide a hardware-based root of trust on client side.

1 Introduction

Organizations consist of groups of employees performing business activities in order to achieve a particular goal [7]. There are different structures for organisations based on the type of the organisation [6]. Each has its own specific policy and process workflow. Organisations are generally divided into multiple groups/departments. Each group/department performs a specific function for the organisation. In addition, a department/group is typically structured into different levels specifying the seniority of employees, e.g. senior managers' level, managers' level, supervisors' level...etc. For example a department might has manager(s) who manages team leaders. A team leader supervises a group of employees. A group of employees with the team leader need to share specific organisational information assets. This sharing is required for accomplishing the organisational tasks assigned to this specific group. Such data needs to be shared but protected from getting accessed by other unauthorised employees. Sharing content is categorised as horizontal and vertical sharing.

- Horizontal sharing means providing the ability to share information by users at the same level; e.g. sharing information between all managers, between team leaders, between a group of employees performing a task for a specific project.
- Vertical sharing means providing the ability to share information between higher and lower levels; e.g. between managers and employees, between team

V. Atluri (Ed.): DAS 2008, LNCS 5094, pp. 238–253, 2008.

leaders and managers, and between a department manager and all employees in the department.

Proposing a solution that can directly be applied to different kinds of organisations is not practical. However, organisations have common requirements that would vary in specific details. Thus, for a solution to be practical it must focus on organisational common requirements. Each specific type of organisation then can customise the solution to satisfy its specific requirements. Organisations typically have the following common requirements specific for sharing its informational assets.

- Flexibility as organisations might change process workflow, employees, infrastructure, etc.
- Share but protect. As described above, organisations require sharing pools of content between different employees, and simultaneously protecting the content from getting transferred to others not authorised to access the pools of content. Content sharing could either be horizontal or vertical as defined above.

The main problem, which is the focus of this paper, is how to share content with a group of users, and simultaneously preventing a member user in the group, who is authorised to access the content, from accidentally or deliberately transferring the content to others not authorised to access such content. This is what we referred to as an internal content leakage. *"The quest for secure information sharing has been a central but elusive goal for information security for over three decades. The stumbling block is simple to understand but difficult to solve. Digital information is easy to copy and transport, and read access to any copy is as good as read access to the original"* [13]. Employees, who have privileges and know from where to obtain corporate content, are by far more dangerous than outsiders. Thus, the cost of insiders' threat exceeds outsiders' threat. Also, the greater an individual's authorisation for accessing corporate content, the greater the potential threat from that person. In this case, using password for user authentication is not enough for ensuring sharing and protecting content. This is because, a dishonest user can transfer his authentication credentials to others not authorised to access the content.

Satisfying the requirement of sharing but protecting content needs binding content to devices requiring access to content, and, simultaneously, ensuring that only authorised users having access to such devices using well know user authentication techniques; see, for example, [4,8].

Objectives: The main objectives for this paper are the following:

- Enabling sharing (the horizontal and vertical sharing) of content between a group of users in an organisation, and simultaneously protecting content from getting leaked accidentally or deliberately to unauthorised parties.
- Satisfying other organisation requirements as discussed above.

Internal content leakage has a major impact on organisations, for example, leaked information could be abused by committing an identity fraud or for marketing purposes. According to the 2002 CSI/FBI Annual Computer Crime and

Security Survey, "*insider misuse of authorised privileges or abuse of network access has caused great damage and loss to corporate information*" [10]. There are several real examples of information leakage, for instance, "*Jonathan Pollard, who had high-level security clearance, was arrested for passing tens of thousands of pages of classified U.S. information such as satellite photographs, weapon systems data, etc., to Israelis. A Libyan intelligence agent obtained the U.S. Military's officers' directory through his wife, who worked at the Department of Transportation and had access to the database of the Metropolitan Washington Council*" [9].

2 Dynamic Domain Definition

A dynamic domain consists of one or more devices owned by a specific organisation. We used the word dynamic to refer to its nature of being flexiable for adding and removing devices from it, i.e. the dynamic domain can be moved across organisational devices based on the organisation needs. Each dynamic domain has a unique identifier i_D and a unique symmetric key k_D. The dynamic domain-specific symmetric key is used to protect the domain-specific pool of content that can only be accessed by the domain-specific set of devices. This key is only available inside member devices of the domain, thus only domain devices can access the pool of content bound to the domain. The dynamic domain creation process is performed by organisation authorised security administrators, who choose devices that need to be bound to a dynamic domain based on the organisation requirements. This binding is performed using an organisation-specific master control device, as will be explained later in this paper. For example, assume an organisation has a department or a group of users, which require its devices to access a specific pool of content, and it does not want the pool of content to leak to other departments/groups. In this case the organisation needs to create a dynamic domain consisting of all devices that need such an access, and simultaneously the organisation needs to bind the pool of content to the dynamic domain. Authorised users, who use member devices in a specific dynamic domain can access the protected content bound to that domain. On the other hand, users cannot access the protected content from devices not members in the dynamic domain even if they have a copy of the protected content. This is because devices not member in the domain do not possess a copy of the domain-specific key, and hence cannot decrypt the domain-specific content. The dynamic nature of the domain enables adding more devices to the domain, and removing member devices in the domain, which should be based on the organisation needs.

Organisation system administrators create dynamic domains, assign devices to dynamic domains and destroy dynamic domains based on organisation needs. A device can join multiple domains to access all content bound to these domains.

3 Hardware and Software Requirements

In this section we describe the main entities constituting the proposed model.

3.1 Organisation Devices

Software-only techniques cannot provide a high degree of protection for organisation domain credentials; for example, Apple FairPlay[1], which uses software-only techniques, has been hacked multiple times; see, for example, [11] and the Hymn project[2]. In the proposed solution we require that organisation devices to be compliant with the Trusted Computing Group (TCG[3]) specifications [17,18,19]. TCG compliant trusted platforms (TP) are not expensive, and are currently available from a range of PC manufacturers, including Dell, Fujitsu, HP, Intel and Toshiba [12]. In addition, since early 2006, all Intel-based Apple computers are TCG compliant [20].

3.2 TCG Overview

This section provides a very brief overview of the main entities in TCG compliant platforms, which are required in the proposed scheme. TCG is a wide subject and has been discussed by many researchers; we will not address the details of TCG specifications in this paper for space limitations. For further details about this subject see, for example, [12,16,17,18,19].

TPM. The TCG specifications require each TP to include an additional inexpensive hardware chip to establish trust in that platform. This chip is referred to as the Trusted Platform Module (TPM), which has protected storage and protected capabilities. In order to reduce the TPM cost, the TCG specifications only require the TPM to be used for functions requiring protected storage and capabilities. Functions that do not require protected storage and capabilities could run using the platform main processor and memory space. The TPM is typically implemented as a processing engine that is separate from the TP's main processing environment. A TPM incorporates various functional components and features including: I/O; a cryptographic co-processor that supports the following: asymmetric key generation, asymmetric encryption/decryption, hashing and random number generation; generation, storage and protection of symmetric keys; HMAC engine; SHA-1 engine; power detection; non volatile memory; volatile memory; platform configuration registers (PCRs), which are shielded locations inside the TPM used to store integrity measurements; and an opt-in component that provides mechanisms and protections to allow the TPM to be turned on/off, enabled/disabled, activated/deactivated.

Protected Storage. Once a TPM has been assigned an owner, it generates a new Storage Root Key pair (SRK), which is used to protect all TPM keys. The private part of the SRK is stored permanently inside the TPM. Other TPM objects (key objects or data objects) are protected using keys that are ultimately protected by the SRK in a tree hierarchy structure. The entries of a

[1] http://www.apple.com/lu/support/itunes/authorization.html
[2] http://hymn-project.org
[3] http://www.trustedcomputinggroup.org

TPM PCRs, where integrity measurements are stored, are used in the protected storage mechanism. This is achieved by comparing the current PCR values with the intended PCR values stored with the data object. If the two values are consistent, access is then granted and data is unsealed[4]. Storage, and retrieval are carried out by the TPM. Therefore, if a software process relies on the use of secrets, it cannot operate unless it and its software environment are correct. The latter ensures that the software process that implements this scheme is trusted to operate as expected.

A TPM can generate two types of keys, known as migratable and non-migratable keys. Migratable keys can be transmitted to other TPs if authorised by both a selected trusted authority and the TPM owner. A non-migratable key is bound to the TP that created it. The TP associates the current platform Software State, which is stored in PCRs, with the non-migratable key, and then protects them using the SRK. Stored secrets are only released after the platform's PCRs have been compared with the values associated with the stored key. Data encrypted using a non-migratable key can leave the TP if and only if the software agent (whose execution status matches the one associated with the non-migratable key, i.e. is authorised to read data encrypted using the non-migratable) authorises the release of the data to other platforms.

Attestation. Establishing trust in a TP is based on the mechanism that is used for measuring, reporting and verifying platform integrity metrics. TP measurements are performed using the RTM (Root of Trust for Measurement), which measures software components running on a TP. The RTS (Root of Trust for Storage) stores these measurements inside TPM shielded locations (i.e. the PCR). Next, the RTR (Root of Trust for Reporting) mechanism allows TP measurements to be reliably communicated to an external entity in the form of an integrity report. The integrity report is signed using an AIK[5] (Attestation Identity Key) private key, and is sent with the appropriate identity credential. This enables a Verifier to be sure that an integrity report is bound to a genuine TPM[6].

3.3 Trusted Software Agent

Trusted software agents act as trusted reference monitors that need to be installed into domain devices and the master control device, and which are required

[4] Seal/unseal are TCG terms used for encrypting/decrypting a data object. Seal binds a data object with an integrity measurement that must match the platform PCR value when unsealing the object. Also, a data object must be unsealed on the same TPM that sealed the object.

[5] AIKs are signature key pairs function as aliases for the TP; they are generated by the TPM, and the public part is included in a certificate known as an Identity Credential, signed by a trusted third party called a privacy certification authority (privacy CA). The identity credential asserts that the (public part of the) AIK belongs to a TP with specified properties, without revealing which TP the key belongs to.

[6] One might argue that the device states might change after getting attested. This is solved by using the new generation of Intel/AMD hardware technology that stops DMA or by using Virtualisation technology as has been described in [12].

to implement the proposed scheme; i.e. creating and managing dynamic domains, protecting content and binding it to a specific dynamic domain, permitting the creation and accessing of content on member devices of a specific dynamic domain.

We require that each organisation possesses three different types of trusted software agents[7]. The first is to be used by the master control device for implementing its functionality as discussed in section 3.4; the second is to be used for devices requiring creating and binding content for dynamic domains; and the last is to be used for accessing content. These agents are the only entities authorised to read 'data protection keys' encrypted using a **non-migratable key**[8] specific to each device in a domain. This is because the non-migratable key object is sealed with the integrity metric of the trusted software agent. The trusted software agents need to be implemented so that they will not release the 'data protection keys' to others. Also, they should be designed in such a way they will not release the data protected using these keys 'unprotected' outside the TP boundaries[9].

TCG compliant hardware ensures that the only means to access the protected content is through the trusted software agent. The trusted software agent, in turn, is responsible for ensuring that access to cleartext content is provided only to authorised users.

TCG compliant hardware provides the main functions required by the trusted software agent, e.g. basic cryptographic functions, local and remote platform and application attestation, and sealed storage for 'content protection keys'. The hardware-based root of trust provides the trustworthiness of the software agent. In this a challenger can verify that a platform is trustworthy by validating the platform integrity metrics. The TP measures the integrity of software executed from platform start-up and stores the result in the platform's PCRs; this provides assurance to the challenger that the OS, and of any other measured software, is running as expected on the platform. The trusted software agents are considered to be trusted if their PCR values are as expected. Therefore, if the OS, running applications, and the trusted software agents are as expected, then the execution environment of the TP is trusted. Hence the secrecy of organisation data is subsequently guaranteed.

3.4 Master Control Device

The master control device is a trusted device that has all TP features, as defined in section 3.1. Each organisation has a specific master control device in charge of managing the organisation dynamic domains and all devices membership in

[7] The three types of software agents could be integrated in one package or three packages. The way this is designed and implemented is outside the scope of this paper.

[8] See section 3.2.

[9] Such trusted Software agents can easily be designed to cover the assumptions, as DRM techniques has designed their own agents based on similar assumptions; see, for example, [3].

each dynamic domain. The trusted software agent in the master control device is in charge of creating and managing dynamic domains involving the following:

- Securely generating and storing each dynamic domain-specific unique identifier, protection key, and a public key list which includes the public keys for all member devices in each dynamic domain.
- Attesting to the execution environment status of devices added to a dynamic domain, ensuring they are trusted to securely store domain keys and execute as expected.
- Adding devices to a dynamic domain by releasing the domain-specific key (i.e. the content protection key) to member devices in the domain.

4 Process Workflow

The workflow of the proposed system is divided into the following phases (for simplicity we refer to the trusted software agent on a device performing certain action, by just using a device performing certain action i.e. we implicitly assuming that trusted software agents discussed in section 3.3 perform the proposed scheme functionality).

4.1 Master Control Device Initialisation

This section describes the process of initialising the master control device. The first time the master control device is initialised, it instructs the organisation security administrators to provide their authentication credentials. The master control device then stores in its protected storage[10] the authentication credentials of the organisation security administrators associated with its trusted execution environment state (i.e. the integrity measurement, which is stored in the TPM's PCR as described in section 3.2). The authentication credential[11] is used to authenticate security administrators before using the master control device. The master control device is used each time the security administrators want to create, expand, shrink or change a dynamic domain.

4.2 Dynamic Domain Establishment

Whenever an organisation wishes to protect a type of content in such way it only can be accessed by a set of devices, it needs to create a dynamic domain consisting of these devices. The process of creating a dynamic domain is done as follows (figure 1 summarises the protocol for this stage).

1. The organisation decides how many devices need to access a specific type of content, say N. N would be the initial size of a dynamic domain. The organisation also decides which devices will access this type of content. This should

[10] We mean by storing data in a protected storage is 'sealing data' in TCG terms, as described in section 3.2

[11] User authentication mechanism is outside the scope of this paper, and it has been discussed in [2].

be based on organisational needs. For example, a dynamic domain could consist of devices used by managers' level, devices used by supervisors' levels. This case covers horizontal sharing. A dynamic domain could be selected to cover vertical sharing. In this case the dynamic domain would consist of devices mixed between different levels. Each group of devices constitutes a specific dynamic domain.
2. The security administrators instruct the master control device to create a new dynamic domain. The master control device then authenticates the organisation security administrators, e.g. using a password.
3. If authentication succeeds, the master control device instructs the security administrators to provide the number N, and the public keys of devices that will be in the dynamic domain.
4. The master control device then securely generates a dynamic domain specific symmetric key k_D, and a dynamic domain specific identifier i. The master control device creates a public key list for this domain consisting of the provided public keys. It then ensures that the size of the public key list equals to N. k_D and i are associated with the public key list and the value of N, and then stored in the master control device protected storage and bound to a trusted execution environment based on TCG specifications; see, for example, section 3.2.

4.3 Adding Devices into a Domain

This section describes the process for adding a device into a dynamic domain, which is performed as follows (figure 2 summarises the protocol for this stage).

1. From each device in the public key list, the organisation security administrators sends a join domain request to the master control device to install the dynamic domain specific key. This request includes the dynamic domain specific identifier i identifying which domain to join.
2. The master control device and the joining device mutually authenticates each other conforming to the three-pass mutual authentication protocol de-

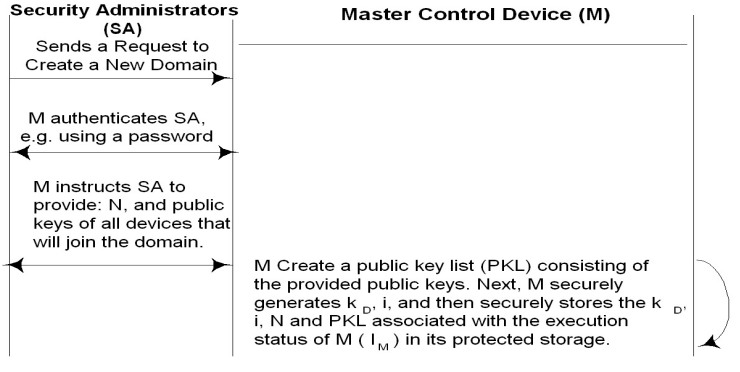

Fig. 1. Dynamic Domain Establishment Protocol

scribed in [5]. The master control device then attests to the execution environment of the joining device and validates its trustworthiness; as described in section 3.2.
3. If the joining device execution environment is trusted, the master control device checks if the device's public key is included in the public key list for the dynamic domain (as specified in step (1) above). If so, it securely releases the dynamic domain specific key to the device.
4. The device stores the domain key in its protected storage, and binds it to a specific execution environment. This device is now part of the domain, as it possesses a copy of the domain key and its public key matches the one stored in the master control device.
5. Now, all member devices of the domain can access the encrypted pool of content related to that domain. All these devices have a copy of the dynamic domain-specific key k_D. Therefore, these devices can access the domain-specific content as protected using the key k_D.

4.4 Binding Content to a Specific Domain

Most organisations create and manage their own content, e.g. creating patient records in clinics, creating bank accounts for customer. As we described in this paper there are different kinds of organisations, each has its own requirements and process workflow. Such requirements and process workflow determine who would create content, and how content should be bound to a domain. Usually departments in organisations create their own content by a group of users in the organisation. These users might be in one department or split across deferent departments. For simplicity, in this paper we consider a single case, which could be easily modified to be suitable for other kinds of organisations. Herein, we assume that an organisation firstly needs to define a group of devices that need

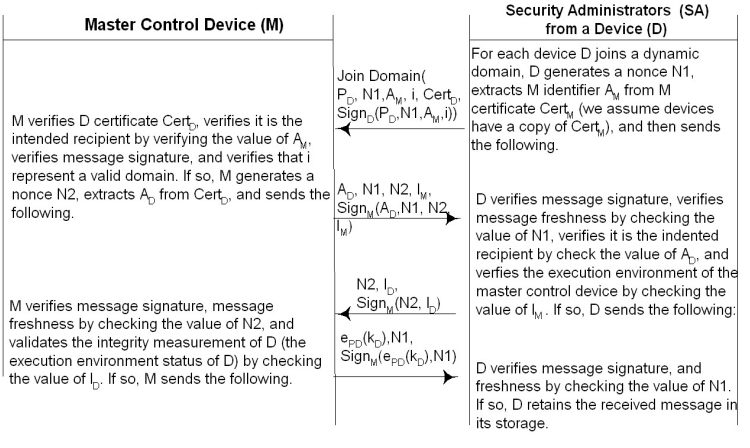

Fig. 2. Adding a Device into a Domain Protocol

to be in a domain to share a specific pool of content. Security administrators then instruct the master controller to create a dynamic domain for this group. Secondly, the third type of the trusted software agent (as described in section 3.3) is used to create content and to specify for which dynamic domain the content belongs. Authorised users (who are allowed to access the trusted software agent) have the ability to create content and assign it to the domain.

We now describe the process for adding content into a domain in a context of a scenario. Assume an organisation needs to work on a new project. This project requires sharing a specific pool of content. Employees working on this project need to share the pool of content, in such a way the content is protected against internal leakage. In this case, the organisation security administrators create a dynamic domain identified by an identifier i. This dynamic domain consists of all devices that need to share the pool of content specific for this project. Authorised employees, which either could be from this group or from a different group create content for this project. Next, the trusted software agent transfers the created content associated with the domain identifier i to the master control device. The master control device identifies the dynamic domain using i, and then encrypts the received content with the dynamic domain-specific key. The encrypted content is typically stored in a dynamic domain-specific location[12] (e.g. a relational database management system, a shared network file system, or others, which should depends on organisational policy.). If someone copied such content he/she will not be able to access it except on devices holding the content-specific dynamic domain key, i.e. member devices in the content-specific dynamic domain.

Next, each member device in a dynamic domain can download the protected content belonging to this domain, typically, from a dynamic domain-specific location or receive it from another device. In this case, only member devices in the same domain i.e. hold a copy of the dynamic domain-specific key k_D, can decrypt and then access the dynamic domain content. As we described earlier, different departments/groups in an organisation, sometimes, require sharing but protecting information. Our solution considers this requirement by allowing devices, which need to share content with other departments or other dynamic domains, to be able to join multiple dynamic domains. Therefore, a single device could join, for example, three domains and so having three dynamic domains-specific keys enabling it to access these dynamic domains content.

5 Domain Management

In order for a solution to be accepted and be widely used, it should adapt with organisations dynamic structure; for example, an organisation might need to change its strategy, layout, business work flow, and/or replace its devices. In this section we discuss how the proposed scheme covers these requirements, i.e.

[12] We assumed in this paper that content are stored in a dynamic domain specific location. This is because this way is the most commonly used in practical life. However, our solution does not make this as mandatory assumption, i.e. content could be stored anywhere based on the organisational policy.

removing a device from a dynamic domain, adding a device into a dynamic domain, and key revocation.

5.1 Domain Shrinking

An organisation might need to enable accessing for a pool of content on fewer number of devices than it is currently use, or it might need to replace its devices for several reasons, e.g. a hardware failure and the device cannot be recovered, or replace the device with newer technology. In these cases the organisation should be given the flexibility to do these changes.

The way to remove a device from a dynamic domain is as follows. The master control device needs to attest to the execution status of the device ensuring it is trusted to remove the dynamic domain key from its storage (based on TCG specifications; see, for example, section 3.2). If the device is trusted, the master control device instructs the leaving device to delete all dynamic domain keys for which the device is leaving. The master control device then removes this device public key from the public key list of the dynamic domain, and decrements the value of N. On the other hand, if the execution status of the device is not trusted, the master control device will not remove this device; i.e. it will not decrement the value N, will not remove the device public key from the dynamic domain-specific public key list. Also, security administrators should still know that this device is still have the content.

5.2 Domain Expansion

An organisation can expand a dynamic domain, for example, when adding more employees to perform a new business requirement or to help existing employees if business expands. In this case, the master control device instructs the security administrators to provide the public keys of the new devices. The master control device then adds the number of the new devices to N. The master control device securely stores the new value of N and updates the public key list with the added values, and finally it allows the new devices to join the domain as described in section 4.3.

5.3 Key Revocation

Hacking a dynamic domain specific key only affects the dynamic domain-specific pool of content. As a precautionary measure, security administrators need to revoke the dynamic domain key, and generate a new domain key, which can be done as follows. The security administrators instruct the master control device to change the key for a specific dynamic domain. The master control device then authenticates the organisation security administrators. If authentication succeeds, the master control device generates a new domain-specific key, and then replaces the old copy of the domain key with the new domain key in its protected storage. The master control device then reinstall this key on domain devices; the master control device identifies devices using their public keys, which

are securely stored inside the master control device, as described in section 4.2. For each device, the master control device releases the new value of the domain key encrypted using the device public key. The device replaces the domain key with the new value in its protected storage and binds it to the same execution environment used for the old key, as it has already been verified as trusted; see section 4.3.

6 System Analysis

In this section we discuss how the proposed solution meets our objectives defined in section 1.

- ▷ The proposed solution allows content sharing but protection against internal leakage. Authorised users can freely transfer content amongst each other and share it. Our solution protects content leaks accidentally or deliberately to unauthorised users. We achieve this requirement by using two security levels, the first is device base level and the second is user level. Device based level means binding content to a domain where authorised users can freely access content. Each device in the domain possesses a copy of the domain key, which is used to protect a pool of content that needs to be shared between the dynamic domain devices. In the proposed solution we ensure that the domain key will not be released to unauthorised devices by securely generating it, transferring it and storing it. Content cannot be transferred unprotected to other devices in the organisation, which means devices only receive protected content. In this case the recipient device either could be an authorised device for accessing content or it could be a device that is not authorised to access the content. Authorised devices can decrypt the content and access it because they already possess a copy of the content protection key. However, unauthorised devices are not capable to access the content because they do not have a copy of the key. For achieving user level protection a specific mechanism needs to be integrated with the proposed scheme ensuring only authorised users accessing devices. The details of this important point is outside the scope of this paper and has been discussed elsewhere; see , for example, [4].
- ▷ Allows horizontal and vertical content sharing across organisation structure. Devices require accessing shared content must join all dynamic domains where shared content is bound. For example, for a chief information officer (CIO) of an organisation to be able to access all organisation shared but protected information, the CIO device needs join all organisation dynamic domains.
- ▷ Flexibility. This is realised as follows.
  - As it is known, organisations have different layers, e.g. managers, seniors. In addition, organisations are organised into different business processes, e.g. a newspaper type of organisation has an editorial work flow, a publishing workflow and page layout. A dynamic domain can contain devices

from a single layer, or from different layers, based on organisation requirements. This provides an organisation the flexibility to bind content on devices based on the organisation functionality.

- An organisation can dynamically move devices between dynamic domains based on changes in its needs. For example, if an organisation requires changing its layout, say after one year, this might require content re-binding. When a device is reallocated to be used by a new layer (i.e. different business process) that requires accessing different kind of content, it can join all dynamic domain where the content is bound. The device also needs to remove all dynamic domains specific keys it no longer authorised to access its content (the device will remove the domain keys, as it is trusted to perform as expected).

▷ Reduces the impact if a domain key is revealed. Because we are using trusted computing that provides hardware based root or trust, it is very unlikely for the domain key to be revealed. In the unlikely event of hacking a domain key, only it affects content bound to a single specific domain, i.e. it does not cause a global impact on other domains content.

7 Related Work

In our proposed solution we mainly focused on achieving two main goals:

- Enabling sharing for content by a group of devices, and simultaneously preventing internal information leakage.
- Satisfying organisation requirements such as: vertical and horizontal 'sharing and protection' of content across groups of devices, adapting with changes in organisational business processes, e.g. adding more employees, reducing the number of employees, changing the IT infrastructure.

In the following paragraphs we analyse current solutions based on the above two points. Current access control techniques such as Discretionary Access Control (DAC) and Role Based Access Control (RBAC) are based on the standard assumption that users are trusted and they will not misuse their authorisation. Also, access control is only enforced at content source. Moreover, DAC and RBAC techniques have security flows and usability limitations when talking about information sharing but protection, as has been widely discussed in many literatures (see, for example, [13]).

The second approach is generally called MAC, as has been analysed by Sandhu et al.[13] which attempts to "solve" the secure information sharing problem in a very specific and rigid framework. This makes it to be not very common over the past three and a half decades. In addition, MAC only allows objects to flow in one direction in a lattice of security labels, i.e. MAC does not allow object owner to share and protect an object amongst other users at the same or higher security levels. This means it does not provide flexible vertical information sharing and protection. MAC can be characterised as a coarse-grained one-directional secure information sharing. Therefore, it is clear that MAC does not satisfy organisation dynamic structure.

DRM schemes proposed in [1,2,3] involve creating a domain owned by a single owner, where all devices joining the domain are bound in some way to the domain owner. These schemes allow secure content sharing between devices in a domain, and prevent the illegal copying of content to devices outside the domain. These schemes focus on protecting copyrighted content in personal network. Organisation networks have different requirements than personal networks. These are as follows. (i) A personal network is composed of a single domain, on the other hand, an organisation consists of multiple domains. Consequently, a device in a personal network should be bound to a single domain. However an organisational network requires that each device to join multiple domains managed by the organisation security administrators. (ii) In personal networks each domain is bound to a single user; however, in the organisation multiple employees are members in an organisation domain, and each employee can be member in multiple domains. (iii) Devices in a personal network need to share but protect content between its all devices. On the other hand, an organisational network needs to share but protect pools of content across groups of devices, each (group) forming a dynamic domain. Most importantly, personal network does not have the concept of internal leakage.

There is another technique attempting reducing content leakage once the content in the hands of authorised individuals by proposing a method for monitoring the activities actioned on content. Park et al. [9] *provides scalable and reusable mechanisms to monitor insiders' behavior in organizations, applications, and operating systems based on insiders' current tasks.* This is by monitoring if an authorised user is performing an abnormal activity on content. Although this method attempts to detect information leakage, however it does not provide mechanism for preventing internal and external leakage, which we have addressed in this paper. We believe preventing information leakage should come before detecting a leakage. However, this is not to lower the importance of detection, which should follow the prevention as there is nothing like hundred percent secure system; i.e. any one attempts tampering with the system, he/she will be discovered at an early stage. Such a mechanism could be integrated with our proposed scheme to achieve other objectives.

The work done in [21] proposes a solution for content sharing, where content is accessed from a centralised location in a read only mode. This solution is useful in organisations that constitute one group and where data is located in a centralised location. Our proposed solution is for different kind of organisations, which have multiple groups each of which share a specific data, and also multiple combination of users within these groups might need to share specific data.

The problem of information sharing has also been addressed by other approaches, such as Windows folder sharing[13] (and windows domains), Network File System (NFS) [15], and resource sharing in P2P networks [14]. Although these approaches proposes different mechanisms for sharing content between groups of users; however, these mechanisms do not address internal content leakage (as defined in this paper). For example, a member in a group who is

[13] www.microsoft.com

authorised to access content shared using any of these techniques can transfer the shared content to others.

8 Conclusion

In this paper we propose a solution for protecting content against leakage in organisations. The proposed solution uses dynamic domains, consisting of devices owned by an organisation. Devices can be dynamically reallocated between dynamic domains based on the organisation needs. This protects content against leakage, and simultaneously allows content to be shared amongst devices in the same domain.

Acknowledgment

The author would like to thank Jason Crampton and Allan Tomlinson for their useful discussion, which have improved the paper.

References

1. Abbadi, I.: Authorised domain management using location based services. In: Cheak, A.D., Chong, P.J., Seah, W., Ping, S. (eds.) Mobility 2007: proceedings of the 4th International Conference on Mobile Technology, Applications & Systems, September 2007, pp. 288–295. ACM Press, New York (2007)
2. Abbadi, I.: Digital rights management using a master control device. In: Cervesato, I. (ed.) ASIAN 2007. LNCS, vol. 4846, pp. 126–141. Springer, Heidelberg (2007)
3. Abbadi, I., Mitchell, C.: Digital rights management using a mobile phone. In: ICEC 2007: Proceedings of the ninth international conference on Electronic commerce, pp. 185–194. ACM Press, NY (2007)
4. Ferraiolo, D., Chandramouli, R., Kuhn, R.: Role-Based Access Control. Artech House, Norwood (2003)
5. International Organization for Standardization. ISO/IEC 9798-3, Information technology — Security techniques — Entity authentication — Part 3: Mechanisms using digital signature techniques, 2nd edn. (1998)
6. Miles, R.E., Snow, C.C. (eds.): Organizational Strategy, Structure and Process. Stanford University Press (2003)
7. Oh, S., Sandhu, R., Zhang, X.: An effective role administration model using organization structure. ACM Trans. Inf. Syst. Secur. 9(2), 113–137 (2006)
8. Park, J., Sandhu, R.: Towards usage control models: beyond traditional access control. In: SACMAT 2002: Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, pp. 57–64. ACM, New York (2002)
9. Park, J.S., Ho, S.M.: Composite role-based monitoring (CRBM) for countering insider threats. In: Chen, H., Moore, R., Zeng, D.D., Leavitt, J. (eds.) ISI 2004. LNCS, vol. 3073, pp. 201–213. Springer, Heidelberg (2004)
10. Power, R.: CSI/FBI computer crime and security survey. Computer security issues & trends (2002)
11. Rowell, L.F.: The ballad of DVD JON. NetWorker 10(4), 28–34 (2006)

12. Sadeghi, A.: Trusted computing — special aspects and challenges. In: Geffert, V., et al. (eds.) SOFSEM. LNCS, vol. 4910, pp. 98–117. Springer, Berlin (2008)
13. Sandhu, R., Ranganathan, K., Zhang, X.: Secure information sharing enabled by trusted computing and pei models. In: ASIACCS 2006: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, pp. 2–12. ACM Press, New York (2006)
14. Schoder, D., Fischbach, K.: Core concepts in peer-to-peer (p2p) networking (2005)
15. Inc. Sun Microsystems. NFS: Network File System Protocol specification. RFC 1094, Internet Engineering Task Force (March 1989)
16. Trusted Computing Group. Infrastructure Working Group Architecture, Part II, Integrity Management. Specification version 1.0 Revision 1.0 (2006)
17. Trusted Computing Group. TPM Main, Part 1, Design Principles. Specification version 1.2 Revision 94 (2006)
18. Trusted Computing Group. TPM Main, Part 2, TPM Structures. Specification version 1.2 Revision 94 (2006)
19. Trusted Computing Group. TPM Main, Part 3, Design Principles. Specification version 1.2 Revision 94 (2006)
20. Weiss, A.: Will the open, unrestricted PC soon become a thing of the past? Journal of Trusted Computing 10(3), 18–25 (2006)
21. Yu, Y., Chiueh, T.: Display-only file server: A solution against information theft due to insider attack. In: Feigenbaum, J., Sander, T., Yung, M. (eds.) Proceedings of the 4th ACM workshop on Digital Rights Management, pp. 31–39. ACM Press, New York (2004)