Access Control for Local Personal Smart Spaces

Brian Greaves
Academy of Computer Science and Software Engineering
University of Johannesburg
Johannesburg, South Africa
brianrgreaves@gmail.com

Marijke Coetzee
Academy of Computer Science and Software Engineering
University of Johannesburg
Johannesburg, South Africa
marijkec@uj.ac.za

Abstract— **As computer systems grow more compact, powerful and cheap to produce, they become more pervasive in society. Smart devices enable users to compute and share resources on the go. Services such as Wi-Fi Direct allow for the creation of device-to-device networks, of a peer-to-peer nature, deemed "smart spaces". Smart spaces are capable of providing an access-point-less means to share information and resources between their peers. Recent research points to the personalisation of smart spaces, making their management more challenging. Personalised smart spaces, advanced as they may be, introduce new security challenges such as secure resource sharing. This paper consequently evaluates a family-related scenario then a LPSS access control framework is proposed, with a focus on the specific nature of LPSS environments namely, local and global sets of rules defined in local and global policies. Finally, access control rules are presented, with respect to the motivating scenario, to illustrate the operation of access control enforcement using local and global policy rules.**

Keywords; access control, context, smart space, trust, policy

I. INTRODUCTION

Today, the abundance of smart devices makes it possible to share information directly between two devices, and create entirely access-point-less networks of capable devices [1]. Users can be provided with an intelligent environment where services and resources on their devices are managed with ease, and minimal user intervention. Such developments bridge the gap between fixed and mobile pervasive systems [2].

Smart devices may serve as repositories of confidential information which must be protected from unauthorized access. As it is not always possible to verify someone's identity visually, due to the increasing strength of radio antennae, trust becomes an important aspect to consider when sharing resources with others [3]. As devices have evolved, so too has the amount of sensory data they have access to, due to built-in devices such as gyroscopes, Global Positioning Systems (GPSs) and cameras. This provides the ability for devices to measure the context of interactions, deemed context-awareness [1]. Devices can respond to their operational environments and change the parameters of their operation based upon their context. Even with these advances, these technologies still lack a reliable means to determine trust in order to facilitate access control decision-making mechanisms [3].

In the face of fast-evolving technology, the development of security mechanisms to protect resources shared across network-less device-to-device communication is struggling to keep abreast. To date, not much research on access control for local personal smart spaces has been done. The distributed location of information in a smart space across devices makes it difficult to control access to resources using well-known access control models, such as discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC). To meet the nature of smart spaces, access control should be dynamic in nature [12].

This papers makes a contribution by identifying access control requirements for local personal smart spaces and proposing a framework that uniquely addresses the local and global requirements of users who are in possession of various intelligent devices that they use in different groupings. The paper is structured as follows: The concept of the local personal smart space is discussed, followed by a scenario. A set of access control and other requirements are identified for the framework. Finally, the paper proposes a framework better geared to protect local personal smart spaces through a trust and context-aware access control model which focuses on two dimensions of policy, namely local and global. Then access control enforcement is described using scenario-based examples used to highlight access control policy usage. Finally, the paper is concluded.

II. LOCAL PERSONAL SMART SPACES

Technology capable of facilitating network-less device-to-device communication accommodates information sharing so that mobility is not limited. Devices that directly connect to each other are peers, unlike in network-centric architectures where there may be a hierarchy between devices. The only limitation of the connection is the ability of the device to connect to the desired network [4]. Devices interact using connections such as Wi-Fi [14] or Bluetooth [5], making them ideal for such ad-hoc networks. Thus, with the nomadic tendency of devices [6] there is no real limitation on where resource sharing can take place.

Personalisation adapts the behaviour of a system to meet the needs of individual users [2]. This ensures that a system behaves differently for different users or for different contexts. Taking this concept further, *a Local Personal Smart Space (LPSS)* is defined by this research as a set of services that are available within a dynamic space of connected devices that is

978-1-4799-7755-0/15/$31.00 ©2015 IEEE

owned, controlled and administered by a single user [2] - the *personal* dimension.

A local personal smart space is created between any two devices owned by the user in a direct peer-to-peer manner using ad-hoc wireless network technologies such as Wi-Fi Direct or Bluetooth - the *local* dimension. Important features of a LPSS is that it is owned by a specific user and maintains the preferences of that user; the physical boundary of the LPPS moves around with the user and his devices; and the LPPS must be able to identify and interact with other LPPSs. Unlike Personal Area Networks [13], where multiple devices connect to each other when they are in close proximity, a LPSS is a personal space that enables the creation of groups of devices that are governed by rules that have been defined by the owner of the group.

The relationships between users of the system, devices and files, and how they relate to each other and the smart space are now further investigated.

III. MOTIVATING SCENARIO

In order to outline the security and access control concerns of LPSSs, a scenario is presented to be used as a point of reference in this paper. The scenario allows for the extraction of the functional and access control requirements of the proposed framework later on.

John, his wife Mary and their son Peter are members of a family. As shown in Fig. 1, John owns three devices; a tablet, a work smartphone and a private smartphone. Mary has a smartphone and tablet, and Peter has a smartphone. Even though their devices are not necessarily made by the same manufacturer, they would like all devices in the family to be able to connect to each other to share files and resources securely. Furthermore, John would like the connection to not accrue any data costs as he does not have an Internet access point in his home.

Before sharing resources with his family, John's priority is to directly connect his three personal devices together in a group so he can share files between them using a set of rules designed for that purpose. Currently, he connects to a cloud application to synchronise resources between devices, but has limited control over settings and rules. He needs more fine-grained access control, for instance, when he is in the office, he wants to share his work documents with his colleagues, but when he is at home no-one may access those documents. Likewise, the rest of the family would also like to create groupings of their personal devices in order to set up how they would like to share their personal files.

His wife Mary needs to similarly group her devices so that she can control resources and file sharing between them. She may want to set rules on how resources are shared with groups of friends or colleagues. As Peter, their son, has just one device, he does not need to create a group. As a family, they need to connect to each other's devices to share resources and files when they are together in proximity. Within these groupings of family devices, John would like to create a set of global rules which dictate how file and resource sharing is controlled for his family. In this regard, John would like to ensure that the smartphone of his son Peter, is restricted in his actions. John wants to protect Peter from sharing files and resources with friends who are not trusted.

Even though Peter only has one device, he should still be able to manage rules relating to file sharing with his family. However, he should not be able to give himself, or any of his friends, access to files and resources that are private. For example, John and Mary do not want just anyone to be able to connect to the family's home group and share. They prefer that only people they find trustworthy can be granted access. Should Mark, one of the Peter's trusted friends who they know well, come over to visit, John would not mind giving him access to some of Peter's files and resources according to Peter's rules.

Finally, John would like the software which enables the connection, grouping, and rule set-up to be easy and flexible to use, with minimal user input after the initial setup.

Fig. 1 illustrates John and his family. On the left is John's two smart phones, *device_J1* and *device_J2* and his tablet, *device_J3,* grouped as *group_J*. Each device has a copy of its local set of rules shown as *LJ* and the global set of rules of the Home *GH* (*group_H*).

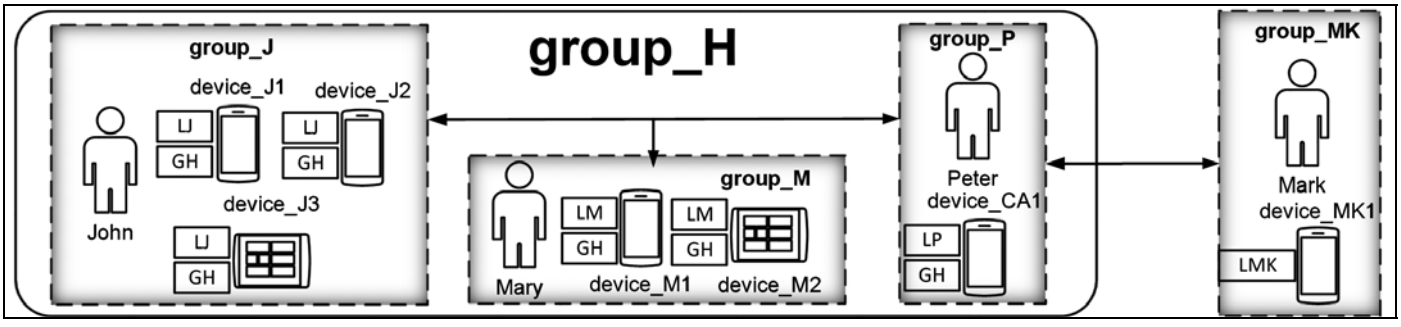

Figure 1. John's LPSS Group

Likewise, Mary, Peter and Mark have their own groups of devices with copies of their local set of rules *LM, LP,* and *LMK*. The family all share the rule set *GH* as it governs their home group. The challenges of John and his family's environment highlight the following functional and access control requirements of an LPSS framework:

A. Local Personal Smart Space Functional Requirements:

- *A device-independent connection* – LPSS capable devices must be able to connect to each other regardless of the device manufacturer, using an access-point-less means.
- *LPSS group creation* – the LPSS framework must facilitate the creation of groups, and allow devices to be invited to join them.
- *Autonomous device and group management* – LPSS capable devices must identify groups as they come into range and connect to known groups automatically.
- *Minimal user input* - after the initial setup little user input must be required.
- *Easy to use* - the system should not be confusing, nor difficult to use at any level.

B. Access Control Requirements:

- *Simple policy management* – access control policies should be straightforward for group and device owners to implement and maintain.
- *Policy overriding* – local and global policy rules need to be processed together, and if conflicts exists, it should be solved appropriately.
- *Strict access control enforcement* - access to resources must be strictly enforced on each device so that access to them is limited. Policy must dictate what can and cannot be done so that unknown or distrusted entities are not granted access to resources.
- *Trust* – trust can be used to grant access to devices that are not members of a group, based upon the context of their actions.
- *Context measurement* – users, device interactions and the environment must be monitored in order to provide the basis for access control and trust computation.

In order to gain a better understanding of the requirements for LPSS access control, a review of related research on access control is given.

IV. Access Control for Local Personal Smart Spaces

Having extracted the access control requirements from John's scenario, the topics of *access control*, *trust, context* and *policy* have been brought to the forefront. Before defining access control, a formal definition of both context and trust is given, and how they relate to each other. The use of policy is discussed later.

A. Context

Context is anything that is used to characterize the situation of some form of entity [7]. An entity may range from a user of the system, to a resource being used or the very system itself. The context of an entity, therefore, is characteristic of what the entity is, is doing, or is affected by during its operation. Changes in context may lead to breaches in security that need to be addressed by revising the security policy. Thus, a cyclic relationship is formed between security context, security policy, the operational environment and changes in context [8].

B. Trust

Trust is the intention to rely on the ability, character, or integrity of another entity or party [9]. Trust describes a relationship between parties and is established after interaction in a LPSS has happened for some time with context measurement. Context allows for the interactions to be gauged as trustworthy or untrustworthy and alters the entity's trust computation. Once a trust relationship is formed, parties can share information and resources. If John trusts Mary, he will allow Mary access to files and services and he will accept her use of them. Ultimately, trust cannot be computed without taking into consideration the previous contextual actions of entities. First-time connecting devices will score a low trust value as there is no contextual basis for their trustworthiness.

C. Access Control

Access control is a security service responsible for limiting access to resources for legitimate users of a system [10]. The foundation of access control lies in the paradigm that *subjects*, entities capable of initiating action within the system, can perform *actions* on *objects*, representations of resources in the system [11]. For example, each device in John's group needs to make access control decisions about the protection of resources on that device. Thus, each device is responsible for its own access control decisions. In order for such devices to make accurate decisions, they need to manage their own policy rules and compute trust on per-device basis. For that to happen, each device must store user, device and environmental context for a fine-grained trust level to be computed and used when deciding how trustworthy non-group members are.

Next, the LPSS framework is proposed with respect to the real-world scenario and the findings of the state-of-the-art research.

V. Local Persoanl Smart Space Framework

The LPSS framework is now presented, with a focus on the access control and policy features. The framework is used to connect devices to each other and to share information and services within a LPSS, and to share between devices in different LPSSs. The framework needs to be installed on each device of Fig.1 in order for groups to be created.

Fig. 2 gives a high-level overview of the layers and components of the framework which interact in order to provide necessary functionality to the LPSS environment. The framework is divided into two primary segments, namely the

Device-independent Layer and Device-specific Application, rolled into a single application for installation. Finally, a high-level description of the access control model is given.

A. The Device-specific Application

At the top of Fig. 2, the device-specific application accommodates the different types of mobile operating system platforms that are available. A *User Interface* enables interaction with the user of the device. The *Resource Manager,* is a middle-man in the framework that is responsible for interfacing with the device's storage for storing and retrieving various types of information.

B. The Device-independent Layer

At the next layer, the device-independent layer houses most of the functionality of the framework. The *Security Manager* contains all of the management components required by the framework. The *Access Control Manager* is responsible for making decisions about which entities are trusted adequately to gain access to protected resources. It primarily interacts with the other components of the *Security Management* layer to make decisions using identities, trust levels and context as provided by the *Trust, Context and Identity Managers*. The *Access Control Manager* refers to the *Policy Manager* to look up access control and policy rules. The *Policy Manager* supports the creation and alteration of policies affecting the system. Policies can be dynamically adapted by changes in state as detected by the *Context Manager.* It also allows for users to alter their local policy.

The *Network Manager* is responsible for abstracting the semantics of interfacing with lower layers such as the device's protocols and radio to pass requests and information up and down between devices. The *Client Manager* deals with connections, groups and passes requests to other components to be processed. The *Context Monitor* is an ever-present watcher that observes the context of interactions and stores them in the *Context Database,* via the *Resource Manager.*

C. Framework Operation

In order for devices to connect to one another and share resources, they first need to have the *device-specific LPSS application* installed. The operation of the *device-independent layer* is abstracted away from the user, as they will only need to manage interactions thought the easy-to-use user interface.

John and his family members set their application preferences. Each person creates a local group for their devices using a username/password credential and then group files and resources into categories on their devices to specify access rules. They invite each of their devices to their personal group and verify their credentials in order to be successfully added. Once the rules are set they are propagated to each device in the personal group. Any changes made on any of the group owner's devices will result in the change being propagated to each device.

After John has created his personal group, he creates the family group. He needs to provide credentials as the group owner. He invites his entire family, including his own personal devices, to the group. John sets up rules specific to the family group. These rules are then propagated to each device in the group in the same manner as the personal group rules. All devices thus have the access control policy of each group they belong to.

In order to ensure that access control is always enforced by the LPSS framework, the device-specific application is installed with root access on each device, to ensure access to resources are routed through the device-independent layer.

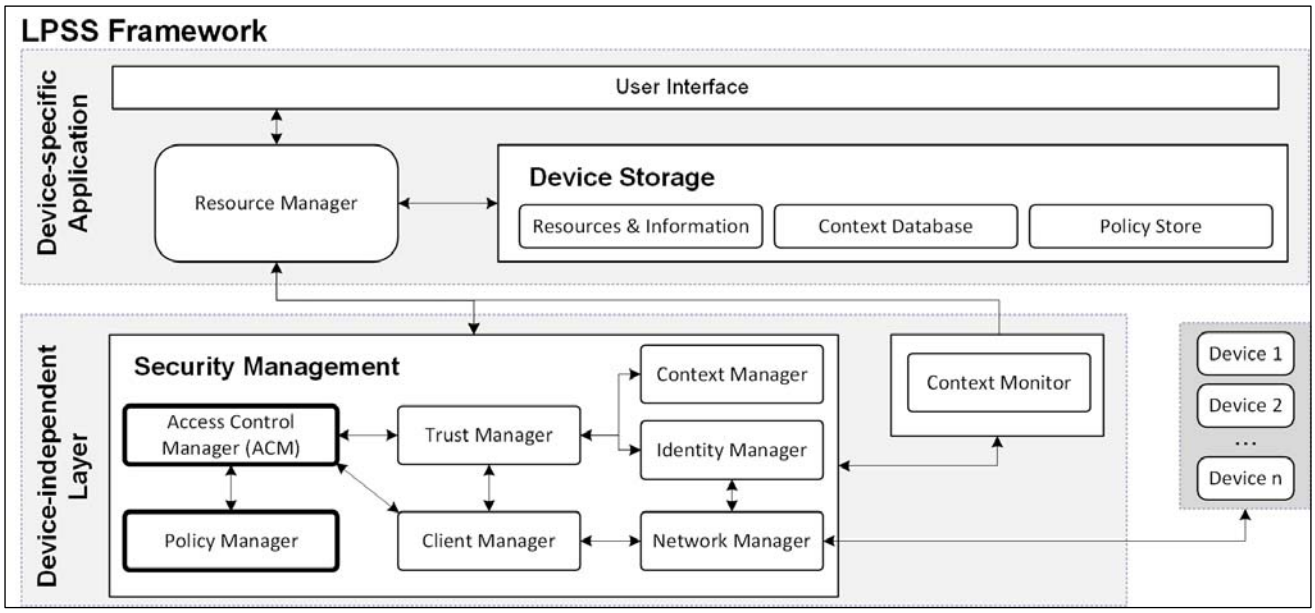

Figure 2. Local Personal Smart Space Framework

Having the application installed in root mode on a device gives the application greater control over access to the device and its available resources. John can for example assign specific resources to be protected by the framework on his son Peter's device. The remaining resources on the device can be used by Peter as he sees fit. This research investigates a method along the lines of [15] and [16] in order to prevent users, such as Peter, gaining access to resources that John does not want them having access to.

In the next section, the access control model used in the LPSS framework is defined.

VI. Access Control Model Definition

This research proposes an access control model for LPSSs defined by local and global policy rules, context constraints and the trustworthiness of requesters and the trust requirement of resources. In a LPSS, user and device identification is available and the initial trust value for every resource such as files or services is created. All access permissions and rules for both the personal and family group are stored in the policy store. Access control policies of the LPSS are discussed next.

A. Access Control Policies

Access control policy rules are set when group owners create groups or as resources are added to them. The *Policy Manager* supports a flexible, descriptive and well-defined policy language that can take into consideration context information. The *Policy Manager* makes decisions following the closed policy [10] approach that allows access if there exists a positive authorization for it, and denies it otherwise. Hence, actions can only be performed if there is a rule stating so. Furthermore, the *Policy Manager* needs to distinguish between two types of access control policies as illustrated in John's family example. First are the private group policies, named *local* policies and second are the policies of the family, or global group, named the *global* policy:

- *Local policies* - govern personal preferences relating to private resources and the access control limitations imposed upon them for the devices in the personal group such as the devices in John's personal *group_J*.
- *Global policies* - govern rules that are applied to all group members in the global group. The global policy is copied to each group member's device. Any changes to the global policy results in immediate updates to each group member device's global policy. For example, the global policy *GH* propagates to each device in the family *group_H*.

B. Model Definition

In order to fully define the access control model, additional terms are defined as follows:

1) User – A user is a human being who owns, or manages, any number of devices which are capable of connecting to a LPSS in order to share information resources.

2) Device – A device owned by a user, *U*, is denoted by *device_i* and is any form of electrical equipment capable of connecting to a LPSS through an installed LPSS application, facilitated by a wireless networking technology such as Wi-Fi Direct. Furthermore, devices provide some form of user interaction and can be grouped together.

3) Local Group - A set of devices owned by a user, *U*, which have been grouped such that:

$group\_L = \{device\_0, device\_1, \dots, device\_i\}$

4) Global Group – The group, *G*, is a set of devices, not necessarily owned by the same user, added to a group such that:

$group\_G = \{device\_0, device\_1, \dots, device\_i\}$

5) Group Member – Any device, *K*, that is a member of a group, *J*, such that:

$\{\exists\, device\_K \mid device\_K \in group\_L\}$

6) Group Management Device –The group owner, *GO*, is a user who is capable of performing management of a group through any of his/her personal devices connected to the group such that:

$\{\exists\, device\_GO \mid device\_GO \in group\_GO \cap group\_i\}$

In the next section, a set of example rules are presented to illustrate the model in use within the LPSS access control framework.

VII. Access Control Rules

The access control model supports a number of rules capable of defining access, ownership and trust within the model. This section uses example scenarios relating to John and his family to illustrate the rules in practical use. First, the use of local policies within a local group are illustrated.

A. Scenario 1 : Local Group with Local Policiy

The first scenario deals with the creation of a *local* group, with local rules that are used to determine access, as shown in Fig. 3.

John, as specified in his family scenario, creates a group *group_J* for his devices, *device_J1*, *device_J2* and *device_J3* before he can start sharing files and resources. He assigns himself as the group owner through the *device-specific application*. In the background, the rule to accomplish this for an owner, *O*, and a group, *G*, is defined by:

$groupOwn(O, G)$

Once John has created the group, he can invite his other devices to the group. Once John provides the group credentials on each of his devices, they are added to the group by the rule for devices, *S*, and group, *G*, as follows:

$in(S, G)$

John's *device_J1*, *device_J2* and *device_J3* are the subjects of the rule and the group is his personal group, *group_J*. Once the devices are all assigned to the group, John can create access rules for subjects, *S*, objects, *O*, and actions, *A* where "-" indicates deny and "+" indicates permit as follows:

cando(S, O, ± A)

This rule grants access to subjects to perform actions on objects. As the framework uses positive authorisations for access to be granted to an object the following must evaluate to "true" – a decision is represented by “do” and an access control rule is defined by “cando”.

do(S, O, + A) ⇐ cando(S, O, + A)

The rule needs to further limit control to member devices in the local group, *group_J*. Thus, the access control rule is augmented with a group check as follows:

do(S, O, + A) ⇐ cando(S, O, + A) ∧ in(S, group_J)

For example, if John wants to share an object *pic_1* between his devices, he sets the following rule:

do(S, O, + A) ⇐ cando(S, pic_1, + A) ∧ in(S, group_J)

```
GroupOwn (John, group_J)

In (device_J1, group_J)
In (device_J2, group_J)
In (device_J3, group_J)

cando (S, pic_1, + A)

do (S, O, + A) ⇐ cando(S, pic_1, + A)
                ∧ in (S, group_J)
```

Figure 3. Local group policy group_J

Fig. 3 lists the local access control policy of *group_J*. The devices that are part of the group are defined, a single access control rule grants access to subjects to access *pic_1*, and a decision rule states that access can be granted to *pic_1* if the subject in *group_J*.

B. Scenario 2 : Global Group with Local and Global Policiy

This next scenario deals with the creation of a *global* group for John's family to share files and illustrates how their *local* and *global* policies are used together.

After the family members have set up their personal groups, John creates the *global* group, *group_H*, for his family. He sets the group owner as *groupOwn(John, group_H)* as he did for his personal group and invites all the family devices to the group.

To protect group resources, John sets rules that allow only group members to access certain files and resources within the home group, *group_H*. The rule is identical to the one used in his local group as follows:

do(S, O, + A) ⇐ cando(S, O, + A) ∧ in(S, group_H)

When any member device in the family group, *group_H*, requests to read *file_a* from one of John's devices, which is also a member of his personal group, *group_J*, access is granted as the rules in the local policy *group_J* does not deny it, as shown by the global group policy in Fig. 4.

```
groupOwn (John, group_H)

In (device_J1, group_H)
In (device_J2, group_H)
In (device_J3, group_H)
In (device_M1, group_H)
In (device_M2, group_H)
In (device_P1, group_H)

cando (S, file_a, + A)

do (S, O, + A) ⇐ cando(S, file_a, + A)
                ∧ In (S, group_H)
```

Figure 4. Global group policy group_H

C. Scenario 3 : Local Group with Global Policy Override Uisng Contextual Information

This scenario now illustrates how a *global* policy denying access, overrides a *local* policy granting access. One of John's smartphones, *device_J2,* is used for work purposes where he stores documents *work_doc1* and *work_doc2* that are assigned to a group of objects called *work_docs.*

John’s work device, *device_J2,* is part of a global group *group_W,* set up by his manager at work to share documents. All devices belonging to this group are restricted to only grant access to devices of colleagues belonging to the global group *group_W,* during working hours as follows:

do(S, O, +A) ⇐ cando(S, work_docs, + A)

∧ in(S, group_W)

∧ (09:00 ⩽ time_of_day() ⩽ 17:00)

The *global* policy rule denies access to any file in *work_docs* if the subject is not in John's group of colleagues, *group_W*, or the request is out of hours, denoted by the environmental context returned by *time_of_day()*, or if there is no rule allowing access to it. Appendix A documents all access control rules applicable to John’s device, *device_J2.*

Should John inadvertently give Mary access to *work_doc1* at home, she is denied access to the file because the *global* policy of *group_W* explicitly denies access by specifying two additional decision rules below. The first rule derives a “deny” if there exists a rule that grants access, but the subject is not in the valid group. The second rule grants access as there are no rules denying access, which is not the case for Mary.

dercando(S, O, -A) ⇐ cando(S, work_docs, + A)

∧ ¬in(S, group_W)

do(S, O, +A) ⇐ cando(S, work_docs, + A)

∧ ¬ dercando(S, O, -A)

The *Policy Manager* of the framework processes applicable local and global policy rules to ensures a deny overrides, so that the local policy is overridden to protect sensitive information, out of the hands of John. Thus, the work documents are protected even if the *local* group, *group_J*, has defined an invalid policy rule as shown by the access control policy in Appendix A, where both local policy of *group_J* and global policies of *group_H* and *group_W* are applied to the files belonging to John.

D. Scenario 4 : Local Group with Global Policy Override Using Trust Measurement

In this final scenario, the *global* policy grants access by overriding a *local* policy using trust.

John would like to ensure that his son Peter is prevented from giving himself, or his friend Mark greater access to resources than John has allowed. Should Peter add his friend Mark's *device_MK1* to a local group, *group_P*, so that Mark can view his pictures, Mark's device could also access all other resources on Peter's device. John adds a rule to the global policy of *group_H*, of which Peter is a member, to deny access to resources for anyone in Peter's group. The new global rule is defined by:

$$dercando(S, O, -A) \Leftarrow cando(S, O, +A)$$
$$\wedge\, in(S, group\_P)$$
$$do(S, O, +A) \Leftarrow cando(S, work\_docs, +A)$$
$$\wedge\, \neg\, dercando(S, O, -A)$$

Peter can add any device to his group, but they will not be able to access objects that are protected.

To be able to grant others access to specific objects such as pictures, the concept of a trust level is rather used, as John prefers to grant trusted friends access to his child's resources. A trust level needs to be assigned to both subjects and objects so that access can be granted. In order for that to happen, John sets a required trust level for the pictures folder for the group as follows:

trustObject(O, O_TL)
-- Where O_TL is a numeric trust score in the range [0, 1].

The pictures folder is assigned an arbitrary required trust level of 0.65, as shown in Appendix A. Furthermore, John needs to manually assign Mark's *device_MK1* a trust level, or allow it to be computed over time using the *Trust Manager* of the framework. In either case, the resultant value can be assigned to *device_MK1* as follows:

trustSubject(S, S_TL)

Thus, John assigns *device_MK1* a trust level of 0.7. A global rule is defined as follows:

$$do(S, O, +A) \Leftarrow cando(S, O, \pm A),$$
$$(trustSubject(S, S\_TL), trustObject(O, O\_TL)$$
$$\wedge\, ((S\_TL > O\_TL) \vee (S\_TL = O\_TL))$$

Appendix A includes the trust assignment and associated rules for the global policy of *group_H*. Access is granted if there is a rule granting access and the trust level of the subjects is more or equal to the trust level of the object. The *global* policy gives John greater peace of mind about what his child and his friends are able to do with their smartphones.

VIII. Conclusion and Future Work

This paper has proposed the concept of the LPSS and how it can be used to better share information between smart devices that are not able to connect to an access point. Based upon the proposed scenario, current LPSS access control research was evaluated for areas that need addressing and a set of requirements were drawn up.

Next, a high-level framework was presented. The importance of local and global access control policies were identified and example access control rules showed their use. The framework, including the use of the Policy Manager, allows for each of the requirements of John's family scenario to be addressed. The device-specific application addressed the functional requirements of the scenario while the device-independent layer caters towards the access control requirements. When working together, the requirements for the protection of LPSSs for John's scenario are all covered by the LPSS framework.

Future work for this research involves a more detailed analysis of the implementation of local and global access control policies and their interaction. Furthermore, a prototype of the framework and the Access Control Manager and all other relevant components is to be developed. Thereafter, the development of test cases will be conducted to test the operation of the prototype.

References

[1] M. Adiba, C. Labbe, C. Roncancio, P. Serrano-Alvarado, "Context aware mobile transactions," Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on , pp.167, 2004.

[2] S. Gallacher, E. Papadopoulou, N. Taylor, M. Williams, "A personal smart space approach to realising ambient ecologies," Pervasive and Mobile Computing, Volume 8, Issue 4, pp 485-499, August 2012.

[3] A. Manaf, S. Movahednejad, M. Sharifi, S, Tabatanaei, "A security conscious service discovery framework in pervasive computing environments," Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBICOMM '09. Third International Conference on , pp.256-262, 11-16 Oct. 2009.

[4] M. Weiser, "The computer for the twenty-first century," Scientific American, Vol265, No.3, pp94-104.

[5] S. Poslad, "Ubiquitous computing smart devices, smart environments and smart interactions, Chippenham:Wiley, 2009.

[6] P. Mahalle, N. Prasad, R. Prasad, A. Pravin, P. Thakre, "A fuzzy approach to trust based access control in Internet of Things." Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2013 3rd International Conference on , pp.1,5, 24-27 June 2013.

[7] Coetzee M, Eloff J (2007). Web services access control framework architecture incorporating trust. Internet Res., 17(3):291-305

[8] P. Brezillion, G. Mostefaoui, "Context based security policies: a new modelling approach," Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on , pp.154,158, 14-17 March 2004.

[9] J. Dalziel, A. Lin, E. Vullings, "A trust-based access control model for virtual organisations," Grid and Cooperative Computing Workshops, 2006. GCCW '06. Fifth International Conference on , pp.557-564, Oct. 2006.

[10] R. Sandhu, P. Samarati, "Access control: principle and practice," Communications Magazine, IEEE , vol.32, no.9, pp.40-48, Sept. 1994

[11] B. Quing-hai, Z. Ying, "Study on the access control model," Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2011 , vol.1,, pp.830-834, 26-30 July 2011.

[12] A. Kashevnik, N. Shilov, A. Smirnov, N. Teslya, "Context-based access control Model for smart space," Cyber Conflict (CyCon), 2013 5th International Conference on, pp.1,15, 4-7 June 2013.

[13] M. Bourgeois, E. Callaway, J. Gutierrez, B. Heile, V. Mitter, M. Naeve, "IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks," *Network, IEEE* , vol.15, no.5, pp.12-19, Sept.-Oct. 2001.

[14] B. Crow, G. Jeong, P. Sakai, Widjaja, I, "IEEE 802.11 Wireless Local Area Networks," *Communications Magazine, IEEE* , vol.35, no.9, pp.116-126, Sep 1997.

[15] X. Du, W. Xiaojiang, H. Zhang, "An effective access control scheme for preventing permission leak in Android," Computing, Networking and Communications (ICNC), 2015 International Conference on , pp.57-61, 16-19 Feb. 2015.

[16] B. Qin, D. Wang, A. Yu, Y. Zhang, B. Zhao, Q. Zhou, "ChainDroid: safe and flexible access to protected Android resources based on call chain," Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on , pp.156-162, 16-18 July 2013.

Appendix A

group_W members

```
in(device_J2, group_W)
…..and other group assignments….
```

group_J members

```
in(device_J1, group_J)
in(device_J2, group_J)
in(device_J3, group_J)
```

group_H members

```
in(device_J1, group_H)
in(device_J2, group_H)
in(device_J3, group_H)
in(device_M1, group_H)
in(device_M2, group_H)
in(device_P1, group_H)
```

Trust levels group_H

```
trustObject(pictures, 0.65)
trustSubject(device_MK1, 0.7)
```

Local access control rules group_J

```
cando(S, pic_1, + A)
```

Global access control rules group_H

```
cando(S, file_a, + A)
cando(S, work_docs, + A)     (invalid rule)
```

Decisions for group_J

```
do(S, O, + A) ⇐ cando(S, pic_1, + A)⋀ in (S, group_J)
```

Decisions for group_H

```
do(S, O, + A) ⇐ cando(S, file_a, + A) ⋀ In (S, group_H)
do(S, O, +A) ⇐ cando(S, O, ± A),(trustSubject(S, S_TL), trustObject(O, O_TL )
             ⋀ ((S_TL > O_TL) ∨ (S_TL = O_TL))
dercando(S, O, -A) ⇐ cando(S, O, + A) ⋀ in(S, group_P)
do(S, O, +A) ⇐ cando(S, work_docs, + A) ⋀ ⌐ dercando(S, O, -A)
```

Decisions for group_W

```
do(S, O, +A) ⇐ cando(S, work_docs, + A) ⋀ in(S, group_W)
             ⋀ (09:00≤ time_of_day()≤ 17:00)
dercando(S, O, -A) ⇐ cando(S, work_docs, + A) ⋀ ⌐in(S, group_W)
do(S, O, +A) ⇐ cando(S, work_docs, + A) ⋀ ⌐ dercando(S, O, -A)
```