Context-Aware Access Control Model for Smart-M3 Platform

Alexey Kashevnik, Nikolay Teslya
SPIIRAS
St.Petersburg, Russia
{alexey, teslya}@iias.spb.su

Abstract

One of the main security problems of Smart-M3 platform is a lack of dynamic security management support. In particular, a new access control model for resource sharing is needed. The access control model should describe the current situation via a context. The paper proposes a model of the context-based access control for the information shared in a smart space based on the Smart-M3 platform. Micro virtualization mechanisms represented by virtual private smart spaces are the basis for the model, which is built on the combination of the role-based and attribute-based access control models. Roles are assigned dynamically based on the smart space participant's trust level. The role separation allows simplifying policies and makes them human-readable and easy to configure. The trust level calculation is based on the participant's context, which includes identification attributes; location; current date; device type, etc. Also, three kinds of security policy rules have been proposed. These rules are used to calculate the trust level, to assign roles based on the trust level, and to grant permissions to the smart space resources.

Index Terms: Access control, Security, Context, Smart space, Smart-M3.

I. Introduction

The cyber physical environment (such as smart building, smart car, etc.) encapsulates both information and physical spaces. It provides shared use of information and allows devices to join and leave the environment [1]. Thereby, smart space can be considered as a part of cyber physical environment, where acting, computational and information resources and virtual community members interact with each other as services to share information (Fig. 1).

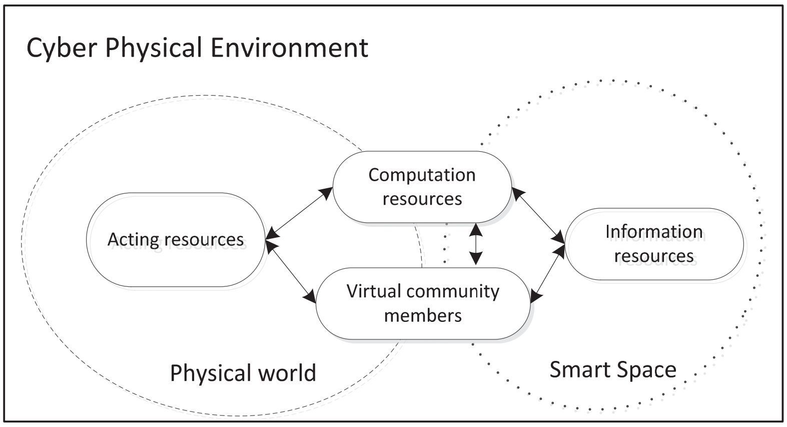

Fig. 1. Smart space as a part of cyber physical environment

The smart space concept helps to make daily human life easier through automation of the routine actions. It allows multiple devices to provide coordinated support to users based on their preferences and current situation in the cyber physical environment (formalized by the context). The smart space based on the Smart-M3 platform is an evolution of the cloud computing concept, which combines the ideas of distributed computing and Semantic Web. In [2] the following features of the smart space are presented and compared with those of cloud computing (see Table I).

TABLE I
COMPARISON OF CLOUD COMPUTING AND SMART SPACE PARADIGMS

Cloud computing paradigm	Smart Space paradigm
Vendor Specific	User specific
Centralised to user (but distributed across provider servers)	Distributed across space devices
Requires network	Network not required continuously
Data privacy and ownership issues	Data is private but some ownership issues (sharing, citation, accreditation)
Unlimited computing resources Unlimited storage resources Cost	Computational and storage capacities are limited by those of space devices and services·(but can extend to clouds)
Not personal , vendor controlled	Personal, user controlled
Partial user responsibility·see licensing agreement, T&C’s	User responsibility
Applications decided by vendor	Flexible applications
Interoperable within vendor’s context	Interoperable

From Table I it can be allocated the following features of smart spaces that affect the information security, and require the development of approaches to achieve security:

- information distribution across space devices. The distribution of information in the smart space makes it difficult to provide access to resources using the existing classical access control models, such as discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC);
- ownership issues in information sharing. It is hard to trust the shared information, when it is impossible to find its source. Also some services can be configured to designedly provide a false information;
- computational and information storage capacities are limited by those of space devices and services. Limited storage and computational capacities of space devices may be the object of denial of service (DoS) attacks;
- user controlled information sharing. People can provide access to personal information because of forgetfulness, negligence, carelessness or ignorance;
- large amount of applications and services operating in the smart space. A large amount of unverified applications may be dangerous, because they may include unknown vulnerabilities or backdoors, which may enable access to private information for unauthorized participants.

In the cloud computing, solving similar problems is the responsibility of the provider. For the users, the cloud computing resources are provided as services, such as information as service, platform as service, software as service, etc. The access control system is included into the cloud service infrastructure and all client applications are verified for the potential vulnerabilities and backdoors by the provider.

In the proposed approach acting, computational and information resources and virtual community members are considered as smart space participants. Every participant is characterized by a context, which describes its activities in the smart space. The context is defined as any information that can be used to characterize the situation of an entity, where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [3]. For example, the context can include a type of the network which using to access to the smart space, date and time of activity, company and/or community for which coalition belongs to, position of the participant in company, etc. The union of contexts of all participants is the context of the corresponding cyber physical environment.

Considering the described above features of the smart space it can be concluded, that one of the main information security problems in coalition operations is a support of the dynamic security management. In particular, it is needed to develop a new access control model based on the coalition operation participant's context. It is proposed to use micro virtualization mechanisms including a virtual private smart space for this purpose. This space is a smart space available only for two participants used for private information sharing between them. It is named virtual, because it is created and used only for information transfer between two participants. After that the space is destroyed.

The paper proposes a model of the context-based access control for the information shared in a smart space. The model is built based on the combination of the role-based and attribute-based access control (ABAC) models. Roles are assigned dynamically based on the user trust level and help to manage access to the resources. The trust level calculation is based on the participant's context, which includes attributes, identifying the user (user ID and public key); user location; current date; device, which requests the information, etc. A special smart space service has been proposed for this model. This service grants access to the resources for the smart space services guided by the security policies. It is needed to note that the public information can be published to smart space and processed by all participants, but the private information is provided only for appropriate participants through the virtual private smart spaces when the corresponding access permissions are granted.

The rest of the paper is organized as follows. Section II describes the smart space Smart-M3 platform features and presents requirements to the smart space security. Section III presents some existing works that introduces access control in Semantic Web and smart spaces based on the context of the participant. Section IV introduces the proposed model and general scheme of the context-based access control for the smart space based on Smart-M3 platform. Section V presents main characteristics of the access control module, based on the presented approach.

II. SMART-M3-BASED SMART SPACES

Smart spaces extend computing to physical spaces, thus, information and physical security become interdependent. Moreover, the dynamism and interoperability that smart spaces advocate can give additional leverage for cyber-criminals, techno villains, and hackers by increasing opportunities to exploit vulnerabilities in the system without being observed. In [4] the following requirements to security and privacy in the smart space are proposed:

- The security service itself has to be distributed, non-intrusive, and transparent.
- The security has to be multilevel, i.e., able to provide different levels of security services depending on security policies, environmental situations and available resources.
- The security system has to support a security policy language that is descriptive, well-defined, and flexible.
- The language should be able to incorporate rich context information as well as physical security awareness.
- Finally, in an open, massively distributed, space-based computing system, authentication should not be limited to authenticating human users, but rather it should be able to authenticate mobile devices that enter and leave the smart spaces, as well as applications and mobile code that can run within the smart spaces.

Presented work is based on the open source Smart-M3 platform [5], [6], which provides implementation of the smart space methodology. This platform was first released at the NoTA conference in October 1, 2009 in San Jose. The Smart-M3 is being developed at ARTEMIS JU programme in SOFIA (smart objects for intelligent applications) [7] and in Finnish national DIEM (Device interoperability ecosystem) research projects. It was applied in other European projects, for example, eHealth, eMobility.

The key idea of this platform is that the formed smart space is device, domain, and vendor independent. Smart-M3 assumes that devices and software entities can publish their embedded information for other devices and software entities through simple, shared information brokers. Information exchange in the smart space can be implemented via different protocols. For example HTTP protocol and Uniform Resource Identifier (URI) [8] can be used for information exchange. Semantic Web technologies have been applied for decentralization purposes. In particular, ontologies are used to provide for semantic interoperability.

III. State-of-the-art

J. Al-Muhtadi et al. [4] propose a mechanism that integrates context-awareness with automated reasoning to perform authentication and access control in space-based computing environments. The authors use this mechanism in the core service of the Gaia project, which provides the infrastructure for constructing smart spaces. The access control is based on the user's confidence value calculation. This value is calculated by the user's context (using simple probabilities, Bayesian probability, and fuzzy logic) and associated with different strengths of authentication which allows different activities in the smart space. Such approach is rather flexible and suitable for dynamic system like the smart spaces.

D. Kuhn et al. [10] propose to integrate two access control models: RBAC and ABAC. Three ways of integration are discussed: (i) with dynamic roles, where user's roles are set by attributes, (ii) attribute-centric, where roles are just attributes, not a set of permissions, (iii) role-centric, where attributes are added to constrain of RBAC. Constraint rules that incorporate attributes can only reduce permissions available to the user, but cannot expand them. The integration of roles and attributes in one model enables to grant access depending on the current situation (context), for example, date and time or location of the user.

Extending this idea, A. Mohammad et al. [11] propose an ontology-based access control model. Usage of ontologies enables access level decisions and provides automated search of information related to the access control.

B. Carminati et al. [12], [13] propose an access control system based on the Semantic Web technologies for social networks. The approach presented in the paper enables granting access based not only on "friendship" relation with the resource owner but also on evaluation of the confidence level of the user. The authors propose policies for filtering available resources specified both by the rules and access control policies. With these policies, the person providing the access can control the information provided to the target users.

Semantic Web technologies are also used by Z. He et al. [14]. They propose access control based on the model of the RBAC using some of the ideas of attributive control, namely, the extending the RBAC with attributes of identity (certificates X.509 [15], public key, etc.). The authors propose the system architecture which implements the described model and discuss its implementation.

S. Verma et al. [16] compare RBAC and ABAC models with respect to the Semantic Web. The authors describe each model and analyze its strongest and weakest features. One of the advantages of the attribute-based access control model noticed by the authors is the support of context by attributes, which enables considering the current situation for granting the access permission.

K. Yudenok in [17] proposes an access control model for the smart spaces which are based on the Smart-M3 platform. The author describes algorithms of the identification, authorization and access control. For the identification and authorization the usage is of the Host Identity Protocol (HIP) [18, 19] is proposed. For the access control the author proposes creation of the mapping between the smart space resources and virtual file system with further usage of the discretionary access control model for granting the access permissions. In this file system every term from the smart space is mapped to the file and the term's hierarchy is represented by the folder structure. A module which implements this model author embeds in the Smart-M3 platform.

All of the reviewed models except one described in [4] are aimed to adaptation of existing access control models to the Semantic Web technologies specifications. Smart space combines the ideas of the distributed computing and Semantic Web, thus, its access control model should provide for interoperability, flexibility and simplicity of the access control rules, decentralization of the resources and access permission based on the semantic attributes from the user's context. All above requirements are met by the model based on the combination of the RBAC and ABAC models and by the scheme proposed by J. Al-Muhtadi et al. [4]. The model proposed in [17] cannot provide support for the user's context and it is very difficult to configure because it uses the discretionary access control model. Moreover, mapping smart space resources to the virtual file system requires significant computational capacities and will certainly affect the system performance.

IV. CONTEXT-BASED ACCESS CONTROL MODEL FOR THE SMART SPACE RESOURCES

As it has been noted, the following specific features of the smart space affect the information security: distribution across user devices, ownership issues computational and storage capacities are limited by those of space devices, and user controlled information sharing. The mechanisms addressing these issues are presented in (Table II).

TABLE II
SECURITY MECHANISMS FOR THE SMART SPACE SECURITY

Smart space specific features	Security mechanisms
Distribution across user devices	Share encoded information
Ownership issues	Context management
Computational and storage capacities are limited by those of space devices and services	Access control and context management
User controlled	Context management

All these mechanisms require introduction of the identification and authentication techniques for the services which request information. The participant is identified by the system when registering in the smart space. At this step the unique identifier is generated and saved in the Security Broker. At the next steps this identifier is used as a part of the participant's context to authorize in the smart space. Additionally, the public and private keys are generated (for example using the RSA algorithm). These keys are needed for participant's authentication in the smart space and providing private information through the virtual private smart space.

The context of the smart space participant consists of the physical and virtual components (Fig. 2). The physical component is due to the fact that each participant in the smart space is also represented in the physical environment, which requires the processing of its properties from that environment.

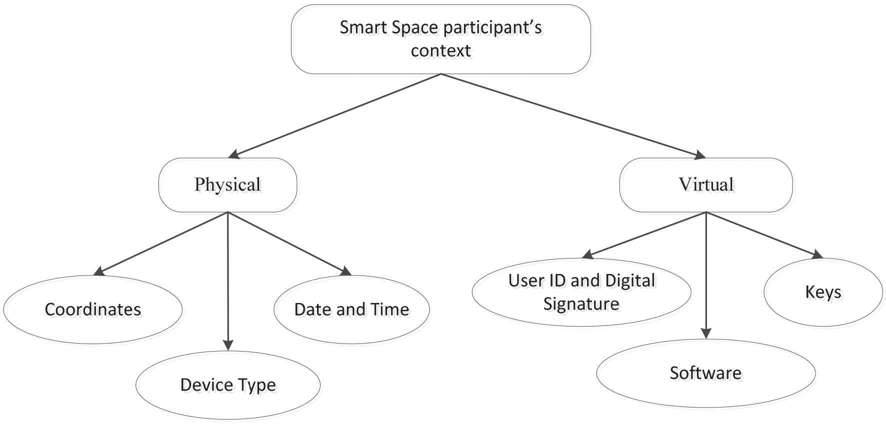
Smart Space participant's context
Physical
Virtual
Coordinates
Date and Time
Device Type
User ID and Digital Signature
Keys
Software

Fig. 2. Components of smart space participant's context

The physical component includes: geographical location of the device, date and time, type of a device. Using this information, the smart space services can determine the current network type of the device, and time of the information access. It enables granting different access permissions from the corporate and public networks in different ways.

Virtual component of the context is due to the fact that each smart space participant interacts with others, and is characterized by a set of attributes that characterize it in a smart space. This component includes software used by the participant for accessing the smart space, digital signature (the participant's identifier and the identifier encoded by the private key), and public key. This information enables authentication and authorization of the participant and provides encoding of the private data.

For the web-community the participants add a social component to the context (Fig. 3).

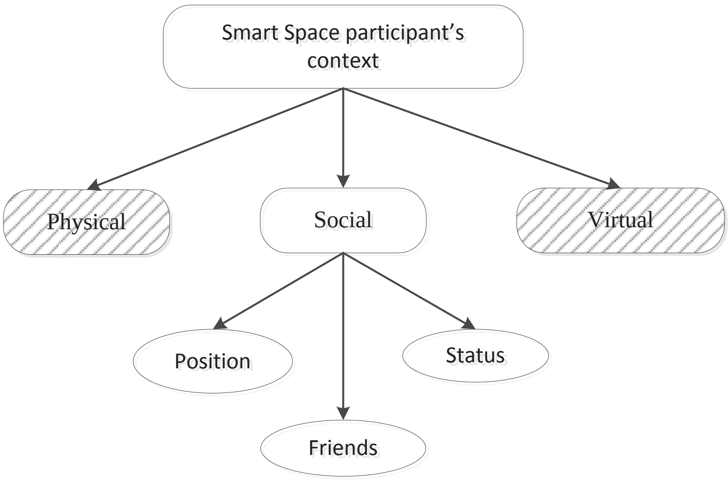

Fig. 3. Components of smart space participant's context in case of participant is a member of web-community

This component includes, for example, position in the company, social relationships. The social component of the context enables granting access to the employees at different positions with the different trust levels, some private data can be shared only between friends, etc. All components of the context are collected and stored on the smart space devices. They become available upon the request of the Security Broker.

Participant's context is used to define the trust levels assigned with its role. The role separation allows simplifying policies and makes them human-readable and easy to configure. Each component of the context is associated with the trust level. The level is represented by a number in the range [0, 1] and depends on the context of the current situation. For example, the trust level of “0.1” and “0.9” can be assigned for access from the public network and from the private network respectively. The logical function taking into account trust levels of all appropriate context components is used to assign a role to the participant. For example the role “trusted_participant” can be assigned only if the participant is authenticated, its network trust level is in the range [0.8, 1] and current time is in the range [0.3, 1]. According to this, there are three sets of security policy rules.

The first set of rules is used to assign the context component trust level to the numeric trust value. The examples of this rule type are the following:

TrustValue(public_network) = 0.1;
TrustValue(“08:00” < current_time < “17:00”) = 0.6;
TrustValue(current_time > “17:00”) = 0.1….

These values are set by the security service and based on the estimations of the security service provider's experts according to the features of the particular smart space service.

The second set of rules is used at the time of logging in or authentication. This set includes rules in the form of logic equations:

Assign_role(some_rule) = (TrustValue(network) $\in$ *(0.8, 1)) &*
(TrustValue(current_time) $\in$ *(0.3, 1)) & …*

The last set of rules contains access control policies, which determine whether a participant with a certain role is allowed to access a particular resource type or not:

Permission(role, resource type);

General scheme of the request process is presented in Fig. 4 and described below.

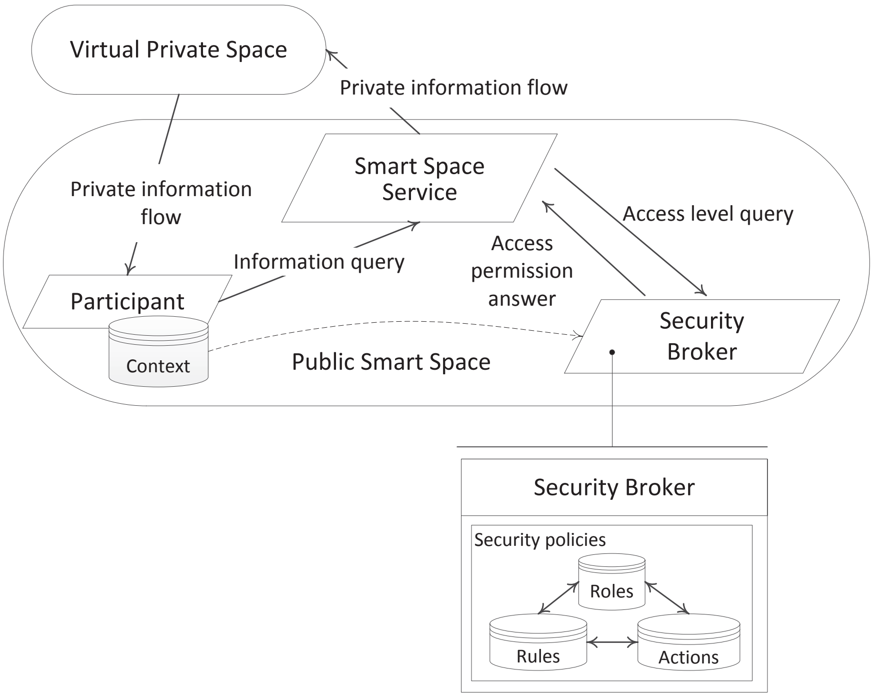

Fig. 4. General scheme of context-based access to Smart space resources

Fig. 4 shows smart space consisting of: participant, which requests the information, some service, which provides this information, and security broker, which provides access permission to participant based on its context. Information flow between participant and service is private due to virtual private smart space which has got no intersections with public space. Query process shown using the sequence diagram on Fig. 5.

A participant sends the request to access a private information (in the RDF notation) to the public smart space and subscribes to the corresponding response about the access granting (Fig. 5):

participant.smart_space.insert("participant_ID", "request", "resource");
participant.smart_space.subscribe("participant_ID", "access_granted", None);

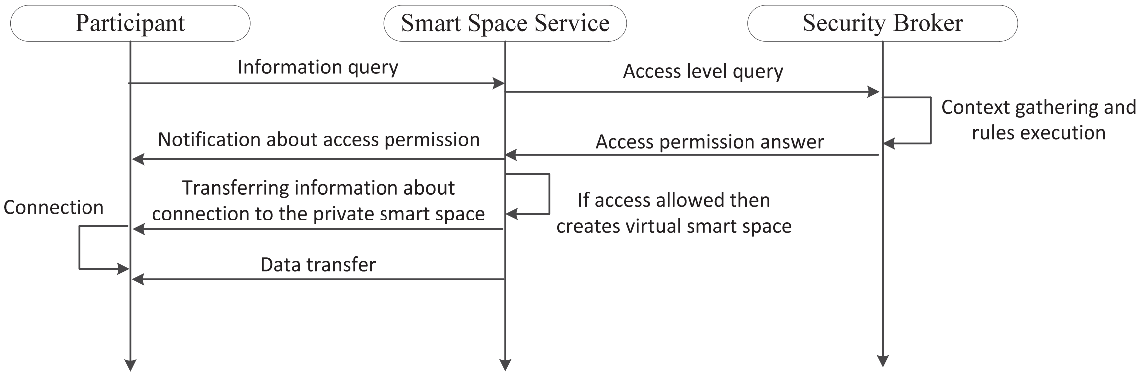
Participant
Smart Space Service
Security Broker
Information query
Access level query
Context gathering and rules execution
Notification about access permission
Access permission answer
Transferring information about connection to the private smart space
If access allowed then creates virtual smart space
Connection
Data transfer

Fig. 5. UML sequence diagram of the information request process

The smart space service accepts the request and calls the Security Broker for the access permission.

service.smart_space.insert("service_name", " participant _requested", "user_ID");

service.smart_space.insert("service_name", "resource_type", "type");

The security broker reads the participant's context and verifies its digital signature using the open key. If the signature is correct, the broker confirms that this user is authenticated and applies the rules from the security policies to assign the role to the participant. The access permission is granted based on the role of the participant and then is sent to the smart space service, which requested it.

security_broker.smart_space.insert("Security Broker", "participant", "participant_ID");

security_broker.smart_space.insert("Security Broker", "access", "granted" or "denied");

If the access to the resource is granted, the smart space service creates a virtual private smart space. The information requested by the participant is transferred to this private smart space. The connection information (space IP, space port and space name) is encrypted via the open participant's key and is sent to the public smart space.

service.smart_space.insert("participant_ID","access_granted", "Encrypted(IP,Port,Name)");

If the access was denied, the service sends the corresponding notification to the smart space participant.

service.smart_space.insert("participant_ID","access_granted","Denied");

Participant, who sends the information request, gets the notification via the subscription. If access is granted the participant decodes the encoded data with its private key and creates a connection to the specified virtual private smart space. When the requested information is transferred the virtual private smart space is destroyed.

V. TESTING OF A CONTEXT-BASED ACCESS CONTROL SERVICE FOR THE SMART SPACE RESOURCES

The basic ideas of context-based access control model for smart space resources have been implemented in security service prototype. This prototype has been evaluated by the following main parameters:

- Response time means the total time spent by the system, starting from the moment of sending the user's query and ending with answer of the service with obtaining information.
- Used RAM indicates total cost of the memory on one user’s device user and one security broker.
- Network load indicates the number of calls to the smart space using SSAP protocol for response time.

Test results shows (Table III) that for information exchange between participant and smart space service is 20 ms for the security service prototype. For real systems this value will be depended on participant’s context and rules database of security broker.

Approach, proposed in the paper aims to increase information security in smart spaces based on the Smart-M3 platform. It has been achieved by introducing dynamic access control model based on smart space participants' context. For the model has been implemented the basic rules, associating the value of context in accordance with the peer review, defining the roles and the access rights based on role. Additionally has been implemented the secure transfer of information between users on the basis of virtual private spaces, similar to the technique of virtual private networks.

TABLE III
THE MAIN PARAMETERS OF THE ACCESS CONTROL MODULE WORKING

Parameter	Value
Response time	20 ms
Used RAM	Client software additionally needs 1.1.Mb Security Broker - 4.5 Мб
Network load	4 additional queries from the client software 3 queries from the security broker

VI. CONCLUSION

Tests conducted during the verification of the developed prototype have proved the thesis put forward in development of the concept. All rules are human readable form and easy to set up in a fairly wide range. The rules are quite strict: non-compliance with at least one of the terms of appointment of the role will be assigned to a different role, more precisely satisfying for smart space participants' context. Computation resources used by prototype are not so high and it is possible to optimize its using.

ACKNOWLEDGMENT

The paper presents research results partly developed under projects funded by grants #13-01-00286 and #13-07-00271 of the Russian Foundation for Basic Research; project #213 of the Russian Academy of Sciences; and project #2.2 of the Nano- & Information Technologies Branch of the Russian Academy of Sciences.

REFERENCES

[1] M. Mohsin Saleemi, Natalia Diaz Rodriguez, Johan Lilius and Ivan Porres. "A Framework for Context-aware Applications for Smart spaces," *Smart spaces and Next Generation Wired/Wireless Networking. 11th Int. Conf., NEW2AN 2011, and 4th Conf., ruSMART 2011*, St. Petersburg, Russia, August, 2011, pp. 14-25.

[2] Ian Oliver. *"Clouds, Spaces and Information Sharing - A Future for the Semantic Web,"* 5th Conf. of Open Innovations Framework Program FRUCT. [Online]. Avaliable: http://www.fruct.org/sites/default/files/files/seminar5/ s5_Fruct_IanOliver_29April2009.pdf.

[3] Dey, A. K., Salber, D., and Abowd G. D. (2001). *"A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications,"* Context-Aware Computing, A Special Triple Issue of Human-Computer Interaction, 16. Retrieved August 13, 2012. [Online]. Available: http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf, Lawrence-Erlbaum.

[4] Al-Muhtadi, J. Ranganathan, A. Campbell, R. Mickunas. "Cerberus: a context-aware security scheme for smart spaces," *Pervasive Computing and Communications, 2003. (PerCom 2003). Proc. of the 1st IEEE Int. Conf.*, pp. 489-496, 23-26 March 2003.

[5] Honkola, J., Laine, H., Brown, R., Tyrkko, O.: *"Smart-M3 Information Sharing Platform,"* Proc. IEEE Symp. Computers and Communications (ISCC'10). IEEE Comp. Soc.; Jun. 2010, pp. 1041-1046.

[6] *Smart-M3 at Sourceforge*, 2012. [Online]. Available: http://sourceforge.net/projects/smart-m3

[7] Petri Liuha, Antti Lappeteläin, Juha-Pekka Soininen. "Smart Objects for Intelligent Applications," *ARTEMIS mag.*, no. 5, pp. 27-29. October 2009.

[8] Berners-Lee, T., Fielding, R., Masinter, L.: *RFC 3986 – Uniform Resource Identifier (URI): Generic Syntax*. [Online]. Available: http://tools.ietf.org/html/rfc3986.

[9] *Resource Description Framework (RDF)*. W3C standard, 2004. [Online]. Available: http://www.w3.org/RDF/.

[10] D. R. Kuhn, E. J. Coyne, T. R. Weil. "Adding Attributes to Role-Based Access Control." *IEEE Computer*, vol. 43, no. 6, pp. 79-81, 2010.

[11] A. Mohammad, G. Kanaan, T. Khdour, S. Bani-Ahmad, "Ontology-Based Access Control Model for Semantic Web service", *J. of Inform. And Computing Sci.*, vol. 6, No. 3, pp. 177-194, 2011.

[12] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, B. Thuraisingham. "A Semantic Web Based Framework for Social Network Access Control," *Proc. of the 14th ACM symp. on Access control models and technologies*, pp. 177-186, 2009.

[13] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, B. Thuraisingham. "Semantic Web-based social network access control," *Comp. & Security*, vol. 30, issues 2–3, pp. 108–115, March–May 2011.

[14] Z. He, L. Wu, H. Li, H. Lai, Z. Hong. "Semantics-based Access Control Approach for Web Service," *J. of Comp.*, vol. 6, no. 6, pp. 1152-1161, 2011.

[15] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. *"RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,"* [Online]. Available: http://tools.ietf.org/html/rfc5280 .

[16] S. Verma, M. Singh, S. Kumar. "Comparative analysis of Role Base and Attribute Base Access Control Model in Semantic Web," *Int. J. of Comput. Applicat.*, vol. 46, No.18, pp. 1-6, 2012.

[17] Kirill Yudenok. "Smart-M3 Security Model,". *Proc. of 11th Conf. of Open Innovations Assoc. FRUCT*, April, 23-27, St.Petersburg, Russia, pp.210–211, 2012.

[18] A. Gurtov, M. Komu, and R. Moskowitz, "Host Identity Protocol (HIP): Identifier/locator split for host mobility and multihoming," Internet Protocol Journal, vol. 12, no. 1, pp. 27-32, Mar. 2009.

[19] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson. *"RFC 5201: Host Identity Protocol,"* [Online]. Available: http://tools.ietf.org/html/rfc5201.