Trust Evaluation for Embedded Systems

Security research challenges identified from an incident network scenario

Christian Gehrmann
Swedish Institute of Computer Science
Isafjordsgatan 22
SE-164 29 Kista
E-mail: chrisg@sics.se

Jacob Löfvenberg
Swedish Defence Research Agency (FOI)
P.O. Box 1165
SE-581 11 Linköping
Sweden E-mail: jaclof@foi.se

***Abstract*—This paper is about trust establishment and trust evaluations techniques. A short background about trust, trusted computing and security in embedded systems is given. An analysis has been done of an incident network scenario with roaming users and a set of basic security needs has been identified. These needs have been used to derive security requirements for devices and systems, supporting the considered scenario. Using the requirements, a list of major security challenges for future research regarding trust establishment in dynamic networks have been collected and elaboration on some different approaches for future research has been done.This work was supported by the Knowledge foundation and RISE within the ARIES project.**

I. Introduction

Information and communication technologies have increasingly influenced and changed our daily life. They allow global connectivity and easy access to distributed applications and digital services over the Internet, both for business and private use, from on-line banking, e-commerce and e-government to electronic health care and outsourcing of computation and storage (e.g., data centers, grid, etc). Networks are steadily evolving and the numbers of IP based applications are growing rapidly. In the near future there will be billions of networked embedded devices, “things” and entities that are interacting all together forming systems of systems. These new existing and emerging end user devices, and the server resources, are interconnected by dynamic and heterogeneous networks. We know that malicious or counterfeit elements can be introduced to an electronic process through most of these devices and in most of the networks that these devices use for communication. Hence, preserving an acceptable level of security in such a diverse environment is a challenging task. Even if we agree on this, there are several different approaches to securing future devices and networks and not a consensus in the research community on which issues that are the most important to tackle.

The paper is organized as follows. First we give an overview of related work and in particular approaches to trust evaluation, trusted computing and embedded systems security. Next, in Section III, we describe the incident network scenario we have been working with and the needs we have identified by analyzing this scenario and related documents. Through a systematic analysis of the the identified needs, we have derived a set of security requirements presented in the form of a requirements hierarchy diagram. In Section IV, we discuss security challenges in relation to the security requirements we have identified. Different approaches to attribute-based trust evaluation as well as analytic evolution and comparison of trust models are discussed in more details. Finally, we conclude in Section V.

This work was supported by the Knowledge Foundation and RISE within the ARIES project.

II. Background and related work

A. Approaches to trust evaluation

Trust can be described as the degree to which confidence is placed in somebody or something. Encyclopædia Britannica gives one definition as: ”assured reliance on the character, ability, strength, or truth of someone or something”.

In computer science the research regarding trust is diverse and addresses how trust can be achieved, how it can be evaluated, how it can be effectively and efficiently managed, how it can be transferred (recommendation) and models for how multiple, partial trust values can be combined into a single trust level.

The idea of handling trust as a problem separated from the application in question was introduced in [1]. We are mainly interested in trust evaluation, i.e. how to automatically assign a trust level to something or somebody, based on available information about relevant properties. A good overview of this problem is given in [2]. An analytical model of trust establishment is given in [3], and [4] describes how context information can improve trust evaluation. How recommendations can improve efficiency in trust establishment is shown in [5]. The approach is interesting since it addresses the connection establishment problem, which normally uses a strong but cumbersome protocol, by introducing trust relevant knowledge to be able to slacken, and thereby simplify, the protocol. In [6] four of the most relevant trust and reputation models are described and an effectiveness comparison using simulations is presented. This is a relevant problem to address since there are few attempts at systematic comparisons between different solutions. An example of how trust evaluation can improve performance in an application is given in [7], where the increase in the resilience of an ad-hoc network to active attacks is analyzed.

B. Trusted computing

Lots of research have been done in the area of trust and trust attestation the past decade. So far, research focused on PCs rather than other mobile devices, although standards for mobile trust, e.g., the Mobile Trusted Module (MTM) [8], proposed by the Trusted Computing Group (TCG), have emerged for defining trust measurements for various classes of devices, including embedded devices. Trusted computing technologies as defined by TCG [9] are slowly starting to become adopted within the IT and telecommunication industry. Trusted computing is built around the usage of a dedicated hardware module, the Trusted Platform Module (TPM) [10], [11], supported by the majority of laptops on the market, and which is also starting to be a standard component on almost all x86 platforms. The TPM allows a user to securely create and store secrets, identify itself towards external parties and to report platform configuration status etc. The area of defining viable trust metrics and trust evidence for a broad set of embedded systems is still in its infancy and is currently subject to lots of research. This includes principles for measurement of trust levels as well as metrics to use for these measurements [12], [13]. Many recent papers also target different models and extensions to the TCG paradigm [14] [15].

C. Embedded systems security

Security for embedded systems cover everything from secure communication to protection of the embedded system execution environment. Secure communication is about the correct choice (considering the embedded system capacity and protection means) of algorithms and protocols for protecting the device to device or device to infrastructure communication. Most essential in this regard are efficient methods to distribute and generate the needed cryptographic keys, i.e., key management.

Secure execution in embedded systems is a difficult task and covers issues ranging from the previously mentioned trusted computing and a secure boot process to different methods to create secure isolation between security critical and non-security critical task on a single embedded device. In particular, we are interested in investigating pure software based isolation methods provided through a thin supervisor running at the most privileged level in the system. Virtual machine monitors or hypervisors have been found to be an apt base for security services, thanks to the hypervisor's isolation and high-privilege visibility into and control over its guests with research suggesting various novel applications such as intrusion detection, malware monitoring, kernel protection, I/O security, and system componentization [16], [17], [18]. Such security services can offer protection to commodity guest OSs. Hypervisor design for embedded systems, and in particular on resource constrained devices, is a rather immature but promising area. However, as was shown in [19], with the right trade-offs and with careful hardware/software co-design, very cost efficient embedded architectures with high security can be achieved with hypervisors.

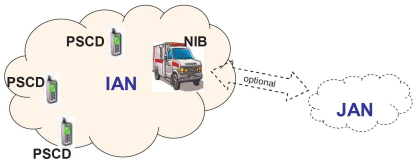

Fig. 1. Incident network scenario

III. An incident network scenario

A. Scenario overview

The need for trust evaluation arises in many different communication scenarios. In particular in situations where two previously unknown entities begin to communicate for the first time. Typically such communication is triggered by a certain need or network context. However, even if this is true, in order to get a realistic view of the security requirements for such events, one better starts an analysis from a concrete application scenario. Hence, we have chosen to base our analysis on a suitably demanding (from a trust perspective) scenario defined in the Euler project [20]. The situation we consider is the communication and collaboration between a collection of so-called Incident Area Networks (IANs), which are being deployed on the scene of a crisis as depicted in Figure 1. Each IAN is administered and owned by a public safety organization, and enables communication within this organization, in the area of the event. IANs may also use different waveforms for their internal communication. A typical IAN consists of

- A vehicle-based "Network-In-a-Box" (NIB), i.e., a fully autonomous and transportable network infrastructure, with base station and all necessary network switching and control functions. The NIB embedded radio unit is a Software Defined Radio (SDR) device that allows new radio wave forms to be configured "on the fly".
- A fleet of user terminals, called Public Safety Communications Devices (PSCDs)

Optionally, this elementary IAN is also connected to the organizations permanent terrestrial infrastructure, which is called a Jurisdiction Area Network (JAN). If all PSCDs in this scenario belong to the same organization, normal mutual authentication in combination with protected communication between the PSCDs and the NIB would be sufficient security measure. However, if PSCDs from *different* organizations also are allowed to connect to the NIB, the situation is rather different, as the hardware, platform and configurations of the visiting PSCDs are unknown to the IAN or JAN manger of the original incident network. This situation is depicted in Figure 2. Such a roaming scenario gives rise to a set of security requirements that we discuss in more details below.

B. Security requirements

In the scenario we consider there are *two* major stakeholders, the NIB manger and the roaming PSCD user. The security

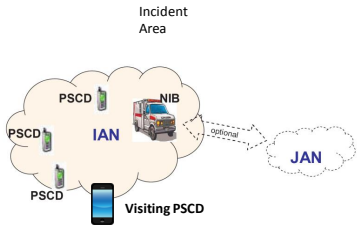

Fig. 2. Roaming scenario

expectations from these two stakeholders form the foundation for the requirements with respect to the trust evaluation that we can derive from the scenario that we are studying. In order to identify the most relevant requirements, we have used a methodology where we first have identified the most important security needs through analyzing a set of statements from the stakeholders [20], [21]. These statements have then been reformulated into basic requirements. The result is summarized in Table I. We have then complemented these requirements with a set of security requirements derived through direct analysis of the scenario in Figure 2 in Table II. Next we have sorted the list of needs into different security categories and presented those as a hierarchical diagram shown in Figure 3. Below, we discuss these requirements in more detail and identify major research challenges with respect to the security solutions that are needed to meet the requirements.

IV. Research challenges

The requirements identified in Figure 3 falls, as shown in the diagram, into three different classes. One class covers access control and authentication mechanisms on the involved nodes, the second class is related to the software installation and management and the third class is about secure execution. In the literature, and in most deployed systems, these three different security classes are treated *separately* and the measures taken to meet security requirements in these classes are done through orthogonal technical solutions. There are for example well established methods for authenticating devices in communication networks as well as very rich frameworks for access control. Trusted boot and software handling is the main contribution from TCG as we discussed in Section II-B and there also exists lots of proprietary solutions for secure software installation and upgrade. Secure execution environments can be achieved through physical or software based isolation and in particular secure execution in embedded systems is a very active research area as we discussed in Section II-C. The major technology challenge when it comes to offering secure execution is the ability to offer a high level of security at a reasonable cost. The latter is important, as if large resources are available the standard solution is separate physical execution environments that have gone through rigorous security evaluations and that provide strong isolation etc.

In a system context like the scenario we analyze in this paper, the orthogonal approach to the three basic security requirements classes mentioned above is not viable as there are close dependencies between the different classes. For example, a very strong authentication mechanism will not be secure

TABLE I
Identified security requirements from statements in [20]

Statement	No.	Requirement
Fear of unauthorized use of application and network services	1.1	Secure access control and authentication mechanism on PSCD and NIB units
Fear of unauthorized modification of software	1.2	Secure software installation and upgrade routines on PSCD and NIB devices
Fear of NIB or PSCD compromise through scripted attacks	1.3	Secure NIB and PSCD software execution environments
Fear of SDR NIB compromise through usage of unlicensed/unsupported OS and software	1.4	Strict control of NIB software installation and usage
We need protection against attacks that replace legal NIB software payload at software upgrade	1.5	Secure NIB software upgrade routines
Fear of downloading invalid NIB or PSCD software updates	1.6	Secure NIB and PSCD software upgrade routines
Fear of bugs in NIB or PSCD software	1.7	Strict verification of approved NIB and PSCD software such as formal verification/evaluation
Leakage of classified information from the internal IAN/JAN network to the visiting PSCD	1.8	The NIB node implementation must ensure that only necessary and authorized communications flow from one domain to the other
Develop a policy driven configuration framework for SDR that: • downloads policies on the fly • verifies their certification • parses, compiles and loads the policies • activates the desired radio device • provides attestation of its configuration to service providers	1.9	NBR SDR policy framework that: • downloads policies on the fly • verifies their certification • parses, compiles and loads the policies • activates the desired radio device • provides attestation of its configuration to service providers
Certify authenticity of configuration software and validity of the configuration to an external entity: • Prevent loading, installation, instantiation of unauthorized software • Verify that downloaded software origins from a trusted vendor • Ensure confidentiality and integrity of over-the-air software download and stored data • Ensure the NIB operates within allowed frequency bands and power levels specified by local regulators	1.10	Strict control over NIB and PSCD software installation, boot and upgrade procedure: • Prevent loading, installation, instantiation of unauthorized software • Verify downloaded software origins from a trusted vendor • Ensure confidentiality and integrity of over-the-air software download and stored data • Ensure the NIB operates within allowed frequency bands and power levels specified by local regulators

TABLE II
ADDITIONAL REQUIREMENTS IDENTIFIED FROM SCENARIO ANALYSIS

No.	Requirement
2.1	When a PSCD roams to a visiting IAN, it must be possible for the NIB in the visiting IAN to securely authenticate connecting PSCD and for the PSCD to secure authenticate the NIB.
2.2	When a PSCD roams to a visiting IAN, it must be possible for the NIB in the visiting IAN, to verify that the connecting PSCD is in a trustworthy state prior to giving access to the IAN and vice versa.
2.3	When a PSCD roams to a visiting IAN, it must be possible for the NIB in the visiting IAN to securely verify the detailed security policies that apply for the connecting PSCD and vice versa. Access decision shall be based on the policies.

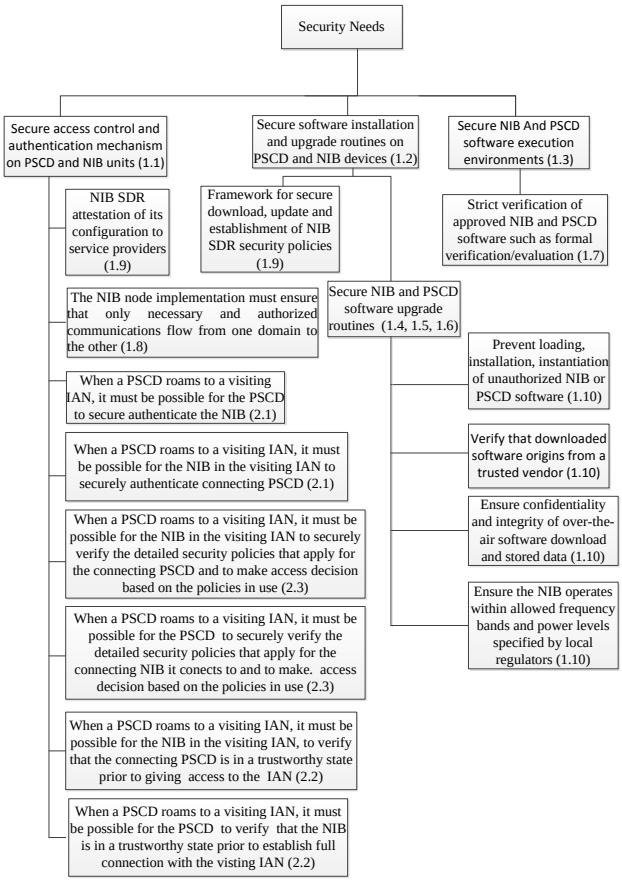

Fig. 3. Requirements represented as a hierarchy diagram

unless the software that actually *implements* the authentication algorithm is trustworthy *and* runs in a secure execution environment. In particular, in a scenario like our incident network scenario where units from different administrative domains and with different manufacturers need to communicate, there is a basic need to combine traditional security properties such as authentication and access control with well defined methods to evaluate the level of trust that can be put into a connecting unit or network and this level clearly depends on the software/hardware that constitutes the connecting unit as well as its execution environment. This leads to a set of new research challenges that we discuss in more detail below.

A. Attribute-based Trust Evaluation

Assume that a mobile unit A comes in contact for the first time with another unit B that it has never met before, and that it wants to initiate a communication session with. At this point A knows very little about B. Of course B also knows very little about A, but this problem is symmetric so we will not consider it explicitly.

In a case where there is a large number of units and a multitude of unit owners and issuers the level of trust that A has in B can vary a lot, depending on who B is, what attributes it has and what credentials it can show. We are interested in the problem of how trust can be created from attributes of the communicating parties. This question can be divided into four parts:

1) What attributes are relevant for trust building?
2) How can A be securely convinced of the values of B's attributes?
3) How are security attributes obtained or measured in dynamic systems where units enters and leaves on a regular basis?
4) How are access control security policies based on trust attributes defined?

With respect to the first and third questions, the previously discussed TCG framework (including extensions) constitute a good starting point, but as the TCG framework mostly deals with software integrity measurements, it needs to be complemented with additional trust attributes and in particular attributes that applies to *dynamic systems*. When looking into the scenario we have analyzed, candidate attributes to consider include (but are certainly not limited to):

- device/domain integrity at a certain point of time
- proof of the authenticity of a device, virtual machine or a virtual set of machines
- verifiable software/OS version
- verifiable hardware identities
- device design certificate (e.g., based on Common Criteria or approved self-tests)
- validity of manufacturing practices
- proof of device/domain ownership
- device/domain security policies
- communication and access policies

The first bullet above covers the TCG integrity attributes, i.e., hash of software binaries and signatures over these hashes. However, in order to handle our dynamic scenario, it needs to be complemented with properties such as proof of the identity of the integrity attribute evaluator, the time of evaluation, which require secure clock values including proof of time value correctness as well as run time evaluation criteria. The latter may include such things as cryptographic attestations of policies and/or protocol transcripts. Other important attributes to consider for trust evaluation include authenticity of device, software, hardware or complete execution environments as

indicated in the second to fourth bullets. For example, the promising recent research with respect to hypervisor based protection of embedded systems that we discussed in Section II-C, should be complemented with mechanisms that allows a remote entity to actually verify the authenticity of the hypervisor that is currently running. Not only the authenticity is important. In some circumstances one cannot assume that it is possible to make a trust decision based on the authenticity of a software or hardware component, but we would need more generic attributes that can give evidence on the security quality of the whole device design such as a device design certificate or proof that important manufacturing principles have been followed. Obviously, important attributes include the traditional security attributes such as proof of device or domain ownership. Finally, in order to make trust decisions an important security attribute to consider is that the security policies that applies within a device as well as for information exchange between the devices and external entities.

The first and second questions above are closely connected. The potential attributes that we have listed cannot be verified through a single verification mechanism. This implies that we will need combinations of traditional authentication mechanisms, certificates and novel attestation principles.

Probably, the most important and also most difficult problem, is the fourth question on how to combine different trust attributes in order to make a trust decision. A trust decision will define to what extent a device is trusted and hence will determine which information and/or resources a device shall be allowed to share with other devices. This needs to be formulated as security policies that can be transferred to a unit and constitute the basis for how the unit will act and protect its resources in dynamic communication scenarios. Even if there already exist well defined syntax languages for the description of access control policies such as XACML [22], the major issue lies in in how to combine different security attributes into well defined policies that are easy to understand and implement in real systems.

B. Analytic Evaluation and Comparison of Trust Models

In [6] is presented a simulation based effectiveness comparison of four trust and reputation models. Such comparisons are valuable in practice since when designing a system one eventually has to decide on a model to use, and the better the model the better the system. From a theoretical point of view comparisons based on simulation are often suboptimal in the sense that they usually have very little of explanatory capacity. For this reason we would like to evaluate and compare trust and reputation models using analytical tools rather than simulation. We believe that by doing so it will be possible to gain a deeper and more intuitive understanding of how and why different models differ in performance. Our ultimate motive is to be able to design new models with even better performance using the insight gained in this analysis.

We realize that to be able to address the evaluation and comparison analytically we will probably have to use simplifications and approximations to get expressions that are possible to work with. In doing so there is of course a risk of loosing accuracy to such a degree that the results will be incorrect. For this reason we still want to verify any findings using simulations. That is, the analytic approach is not a way of reaching greater accuracy, it is a way of gaining insight that we can use for improved constructions.

As a next step we would like to investigate how the identified new trust models can be combined with the previous discussed trust evaluation based on attributes. In particular we would like to investigate how different models can be ”translated” into a trust attribute context and policies based on trust attributes.

V. Conclusions

In this paper we have investigated a dynamic incident network scenario and in particular identified that to meet the security needs in such scenario, new research is needed with respect to how to evaluate the level of trust that can be given to another entity when two or several units are connected ad hoc. We need both new trust models and new tools for trust evaluation. A common framework based on trust attributes that allows efficient trust decision and security policy expression on embedded devices lies at the top of the “wish list”. In static environments and for computing entities where one has the possibility to have tight control of the hardware and the software, there is a very limited need for advanced trust evaluation methods. On the other hand, the current trends speak toward usage of standard hardware components and open software environments in combination with increased network connectivity and interconnection of information systems. Hence it becomes critical to provide new tools that can be used to configure advanced security policies such that systems can still interact efficiently without getting compromised as for the incident scenario we have presented. We believe that trust evaluation is a key problem and we have pointed out two research paths which we think are important to secure future networked, embedded systems.

Acknowledgment

The authors would like to thank Ahsant Mehran at Saab in Järfälla for good discussions and input with respect to the incident network scenario analyzed in this paper.

References

[1] M. Blaze, J. Feigenbaum, and J. O. Lacy, “Decentralized trust management,” in *IEEE Symposium on Security and Privacy*, 1996, pp. 164–173.

[2] D. O’Callaghan, “Trust evaluation for the grid,” Ph.D. dissertation, University of Dublin, Dublin, Apr. 2007.

[3] Y. L. Sun and Y. Yang, “Trust establishment in distributed networks: Analysis and modeling,” in *Proceedings of the IEEE International Conference on Communications, 2007 (ICC ’07)*, Glasgow, Jun. 2007, p. 1266–1273.

[4] S. Toivonen, G. Lenzini, and I. Uusitalo, “Context-aware trust evaluation functions for dynamic reconfigurable systems,” in *Proceedings of the Models of Trust for the Web Workshop (MTWS06), held in conjunction with the 15th International World Wide Web Conference (WWW2006)*, Edinburgh, Scotland, May 2006.

[5] B. Liu, “Efficient trust negotiation based on trust evaluations and adaptive policies,” *Journal of Computers*, vol. 6, no. 2, p. 240–245, Feb. 2011.

[6] F. G. Mármol and G. M. Pérez, "Trust and reputation models comparison," *Internet Research*, vol. 21, no. 2, p. 138–153, 2011.

[7] Z. Yan, P. Zhang, and T. Virtanen, "Trust evaluation based security solution in ad hoc networks," Nokia Research Center, Helsinki, Finland, Technical Report, 2003.

[8] TCG Mobile Phone Working Group, "TCG Mobile Trusted Module Specification, Version 1.0," 2008. [Online]. Available: http://www.trustedcomputinggroup.org/files/resource_files/87852F33-1D09-3519-AD0C0F141CC6B10D/Revision_6-tcg-mobile-trusted-module-1_0.pdf

[9] Trusted Computing Group, "TCG Architecture Overview, Version 1.4," 2007. [Online]. Available: http://www.trustedcomputinggroup.org/files/resource_files/AC652DE1-1D09-3519-ADA026A0C05CFAC2/TCG_1_4_Architecture_Overview.pdf

[10] ——, "TPM Specification, TPM Main Part I-III Design Principles," 2007. [Online]. Available: http://www.trustedcomputinggroup.org/resources

[11] S. L. Kinney, *Trusted Platform Module Basics: Using TPM in Embedded Systems (Embedded Technology)*. Newnes, 2006.

[12] M. Nauman, M. Alam, X. Zhang, and T. Ali, "Remote Attestation of Attribute Updates and Information Flows in a UCON System," in *Trusted Computing, Second International Conference, Trust 2009, Oxford, UK, April 6-8, 2009, Proceedings*, ser. Lecture Notes in Computer Science, vol. 5471. Springer, 2009, pp. 63–80.

[13] F. Baiardi, D. Cilea, D. Sgandurra, and F. Ceccarelli, "Measuring Semantic Integrity for Remote Attestation," in *Trusted Computing, Second International Conference, Trust 2009, Oxford, UK, April 6-8, 2009, Proceedings*, ser. Lecture Notes in Computer Science, vol. 5471. Springer, 2009, pp. 81–100.

[14] J. Li and A. Rajan, "An anonymous attestation scheme with optional traceability," in *Trust and Trustworthy Computing*, ser. Lecture Notes in Computer Science, A. Acquisti, S. Smith, and A.-R. Sadeghi, Eds., vol. 6101. Springer Berlin / Heidelberg, 2010, pp. 196–210.

[15] S. Bratus, M. Locasto, and B. Schulte, "Segslice: Towards a new class of secure programming primitives for trustworthy platforms," in *Trust and Trustworthy Computing*, ser. Lecture Notes in Computer Science, A. Acquisti, S. Smith, and A.-R. Sadeghi, Eds., vol. 6101. Springer Berlin / Heidelberg, 2010, pp. 228–245.

[16] A. Seshadri, M. Luk, N. Qu, and A. Perrig, "Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses," 2007.

[17] J. Yang and K. G. Shin, "Using hypervisor to provide data secrecy for user applications on a per-page basis." in *VEE'08*, 2008, pp. 71–80.

[18] H. A. Lagar-Cavilla, L. Litty, and D. Lie, "Hypervisor support for identifying covertly executing binaries," in *17th USENIX Security Symposium*, San Jose, CA, August 2008.

[19] C. Gehrmann, H. Douglas, and D. K. Nilsson, "Are there good reasons for protecting mobile phones with hypervisors?" in *The 8th Annual IEEE Consumer Communications and Networking Conference (CCNC'2011)*, Las Vegas, NV, USA, January 2008, pp. 493–498.

[20] The Europen FP7 project Euler, "Deliverable 3.2, General Design, Version 1.0," 2009. [Online]. Available: http://www.cwc.oulu.fi/euler/EULER_D3_2.pdf

[21] ——, "Deliverable 2.3, End-users requirements, Version 1.1," 2010. [Online]. Available: http://www.cwc.oulu.fi/euler/D2_3End_users_requirements.pdf

[22] S. Godik and T. Moses, "eXtensible Access Control Markup Language (XACML)," Organization for the Advancement of Structured Information Standards (OASIS), Standard, February 2003, http://www.oasis-open.org/committees/xacml.