Chapter 14

COLLABORATIVE ACCESS CONTROL FOR CRITICAL INFRASTRUCTURES

Amine Baina, Anas Abou El Kalam, Yves Deswarte and Mohamed Kaaniche

Abstract A critical infrastructure (CI) can fail with various degrees of severity due to physical and logical vulnerabilities. Since many interdependencies exist between CIs, failures can have dramatic consequences on the entire infrastructure. This paper focuses on threats that affect information and communication systems that constitute the critical information infrastructure (CII). A new collaborative access control framework called PolyOrBAC is proposed to address security problems that are specific to CIIs. The framework offers each organization participating in a CII the ability to collaborate with other organizations while maintaining control of its resources and internal security policy. The approach is demonstrated on a practical scenario involving the electrical power grid.

Keywords: Access control, policies, models, collaboration, interoperability

1. Introduction

Critical infrastructures (CIs) are logical/physical facilities that are essential to public welfare; their disruption or failure could have a dramatic impact on the economy and social well-being of a nation. The most significant CIs are those dedicated to electricity generation, transport and distribution (electrical power grid), telecommunications, supply services (energy, food, fuel, water and gas), transportation systems (roads, railways and airports) and financial services (banks, stock exchanges and insurance companies).

Due to interdependencies existing between the various infrastructures, cascading and escalating failures are possible [15, 21]. A cascading failure occurs when a failure in one infrastructure causes the failure of one or more components in a second infrastructure. An escalating failure occurs when a failure in one infrastructure exacerbates an independent failure in a second infrastructure; this second failure has increased severity and/or requires significant

recovery or restoration time. A prime example is the North American blackout of August 14, 2003 [4]. A small failure in monitoring software prevented an electrical line incident from being confined; the failures propagated across the electrical power grid resulting in losses exceeding six billion dollars. In general, failures may occur as a result of accidental faults or malicious actions such as intrusions, denial-of-service attacks and worm propagation.

The North American blackout was caused by a failure in a computer system. Information technology and communications systems are used so widely in the various critical infrastructures that they have come to be known as the critical information infrastructure (CII). CIIs involve vulnerable information and communication technologies (ICT) that are easily compromised by attackers. Consequently, securing CII assets is an important component of any critical infrastructure protection effort.

In Europe, America and elsewhere, regional, national and multinational energy companies may be in competition, but they have to cooperate to produce, transport and distribute electric power. CIIs play a major role in these efforts – they are open, distributed systems that support collaboration between the various entities involved in operating critical infrastructures. CIIs are required to be flexible and extensible; it is equally important that they implement sophisticated access control strategies given the diverse entities that share the information and communication assets.

This paper focuses on security problems related to access control, collaboration and interoperability in CIIs. In particular, it shows how PolyOrBAC [2], a security framework based on the OrBAC access control model [1, 18] and web services technology, can be adapted to cope with CII security needs.

2. CRUTIAL Architecture

This section describes a generic CII architecture (CRUTIAL [10]), which models interconnected infrastructures such as those encountered in the electrical power grid, provides an excellent framework for applying PolyOrBAC to protect CIIs from accidents and as well as from malicious acts.

The CRUTIAL architecture is presented in Figure 1. It can be viewed as a WAN comprising multiple LANs interconnected by CRUTIAL information switches (CISs). A LAN incorporates various logical and physical systems; it has its own applications and access control policy, and provides services to other entities. Each LAN belongs to an organization involving different actors and stakeholders (e.g., power generation companies, power plants, substations, energy authorities, external maintenance service providers, and transmission and distribution system operators). Multiple LANs are connected by a single CIS if they are part of the same organization and located in the same area. In this case, each LAN is dedicated to a component (e.g., substation) in order to manage a different access control policy for each component.

All the CII organizations in the CRUTIAL architecture are interconnected by CISs. Therefore, to provide controlled cooperation, each CIS must contain mechanisms that enforce the local security policy of each collaborating

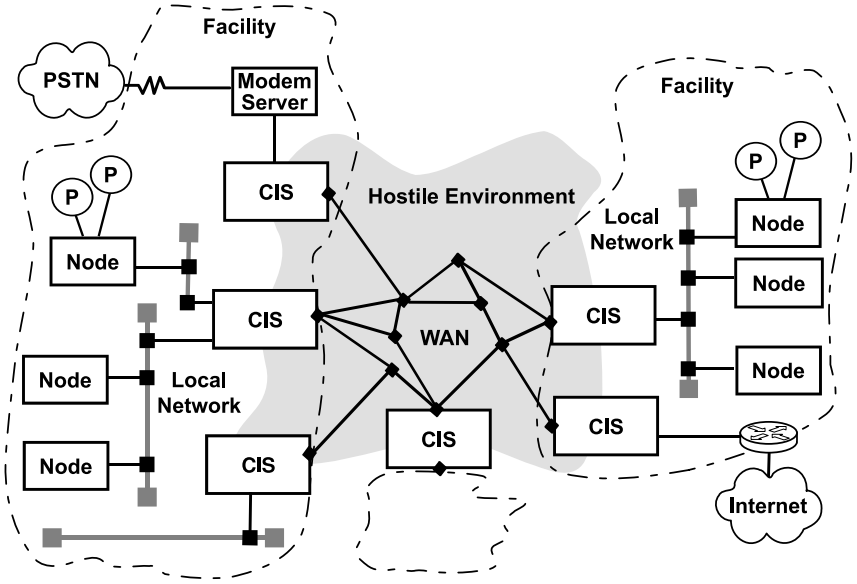

Figure 1. Generic CII architecture.

organization. Also, collaboration mechanisms should be implemented using web services. These policies and mechanisms must allow authorized access to resources and prevent all unauthorized access.

3. PolyOrBAC Framework

XACML is a popular language for specifying policies and managing access control in distributed and decentralized systems [16]. However XACML's flexibility and expressiveness comes at the cost of complexity and verbosity. It is hard to work directly with the language and policy files. Tools are being developed, but until they are widely available, it will be difficult for average users to work with XACML-based systems.

We use the OrBAC access control model as an alternative to XACML to implement access control for each CII component and to facilitate collaboration between components. Two approaches are possible. One approach is to impose a global, centralized security policy for all the organizations. This approach is not appropriate for a CII because of its dynamic character – organizations should be able to join or leave the CII without disturbing the rest of the architecture. Also, organizations are mutually suspicious. Each organization has specific rules and security policies and would prefer to maintain its autonomy. An organization would likely refuse to open its information systems to competitors or change its security policy.

The second approach is to use PolyOrBAC [2] to manage collaboration between CII organizations while maintaining the autonomy of individual organi-

zations. PolyOrBAC implements access control and security using local access control policies specified using OrBAC. Collaboration rules between the various organizations are managed using web services. Thus, the approach provides for interoperability, collaboration and secure sharing of information between CII components, actors and organizations.

3.1 OrBAC Access Control Policies

The OrBAC [1, 18] model is an extension of RBAC [11, 22]. It enables the structured, abstracted expression of security policies: subjects are abstracted using roles (as in RBAC), objects [5, 14] as views (as in VBAC [7, 12]) and actions [5, 14] as activities (as in TBAC [20]). Also, security policy specification is completely separated from its implementation, which reduces complexity.

In OrBAC, an organization is a structured group of active entities in which subjects play specific roles. An activity is a group of one or more actions, a view is a group of one or more objects, and a context is a specific situation that conditions the validity of a rule. The role entity is used to structure links between subjects and organizations. Similarly, objects satisfying a common property are abstracted as views, and actions are abstracted as activities.

OrBAC rules can express positive/negative authorizations (permissions/interdictions) and obligations. Security rules have the following form: Permission (org ; r ; v ; a ; c), Prohibition (org ; r ; v ; a ; c) or Obligation (org ; r ; v ; a ; c). The rules imply that in context c, organization org grants role r permission (or prohibition or obligation) to perform activity a on view v.

OrBAC considers two different levels for the security policy: the abstract level and the concrete level. At the abstract level, the security administrator defines security rules using abstract entities (roles, activities, views) without worrying about how each organization implements these entities. At the concrete level, when a user requests an access, an authorization is granted (or not) according to the relevant rules, organization, role, instantiated view/activity and parameters. The derivation of permissions (i.e., instantiation of security rules) is formally expressed as:

$$\begin{aligned}&\forall org \in Organizations, \forall s \in Subjects, \forall activ \in Activities,\\ &\forall o \in Objects, \forall r \in Roles, \forall a \in Actions, \forall v \in View, \forall c \in Contexts :\\ &\text{Permission } (org,\ r,\ v,\ activ,\ c) \wedge \text{Empower } (org,\ s,\ r) \wedge\\ &\text{Consider } (org,\ a,\ activ) \wedge \text{Use } (org,\ o,\ v) \wedge\\ &\text{Hold } (org,\ s,\ a,\ o,\ c) \Rightarrow \text{is-permitted}(s,\ a,\ o).\end{aligned}$$

The security rule specifies that if in organization org, role r can perform activity $activ$ on view v when context c is True; and in organization org, r is assigned to subject s; and in organization org, action a is a part of activity $activ$; and in organization org, object o is part of view v; and context c is True for (org, s, a, o); then subject s may perform action a on object o.

3.2 OrBAC Limitations

OrBAC can be used to specify complex security policies that are encountered in real-world information technology systems. Also, it facilitates security policy management and updates.

However, in the CII context, it is necessary to specify security requirements and rules for each CII organization or subsystem, manage the collaboration between organizations, and enforce (within each CIS) the different security policies. OrBAC can handle the first requirement, but not the second. In particular, it is not possible to specify rules pertaining to multiple independent organizations in a collaborative system using a single OrBAC policy. Also, permissions cannot be associated with users belonging to other partner organizations or sub-organizations. Consequently, while OrBAC can express the security policy of an organization, it is inadequate for modeling collaboration and interoperability between multiple organizations.

3.3 Web Services

Web services technology offers powerful mechanisms for implementing collaboration. Software applications written in different programming languages and running on diverse platforms can use web services to exchange data over computer networks in a manner similar to inter-process communication on a single computer. Web services use well-known open standards and protocols such as XML [6], SOAP [19], WSDL [17] and UDDI [8], which are readily used with current web interfaces. Since web services are based on the HTTP protocol, they can cross firewalls without changing established security requirements. Moreover, the execution of a web service does not require substantial resources (memory, power and CPU time) and a small quantity of code is sufficient for implementation. Finally, web services can be easily coupled with OrBAC.

3.4 PolyOrBAC Framework

PolyOrBAC uses OrBAC to specify local access control policies (for each organization) and collaboration rules involving multiple organizations. Web services are used to enforce collaboration. A CII component (organization or subsystem) may have its own security objectives and can cooperate with the other components. Each organization has its own resources, services and applications with its own objectives, operations, security rules and policy. Figure 2 shows a scenario where Alice from Organization A wishes to invoke web service Service1 offered by Organization B.

During the initial publication step, each organization determines which resources and/or services it will provide to external partners. Web services are developed on the organization's application servers; they are defined in the organization's OrBAC security policy and are referenced in the organization's CIS to be accessible to external users.

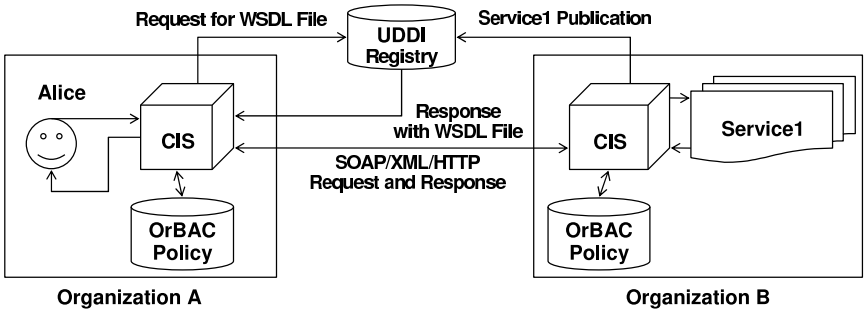

Figure 2. PolyOrBAC framework.

During the discovery step, when Alice wishes to use Service1, Organization A contacts the UDDI web service registry to search for Service1 (which is published beforehand by the offering organization (Organization B)). Then, Organization A receives the WSDL file with the description of Service1 and the URL of the site (in Organization B) that hosts Service1.

During the negotiation step, Organizations A and B mutually authenticate each other, they negotiate and come to an agreement, establish a contract and jointly define security rules governing access to Service1. These rules and contracts are registered in the OrBAC format in each CIS database that contains the security policy.

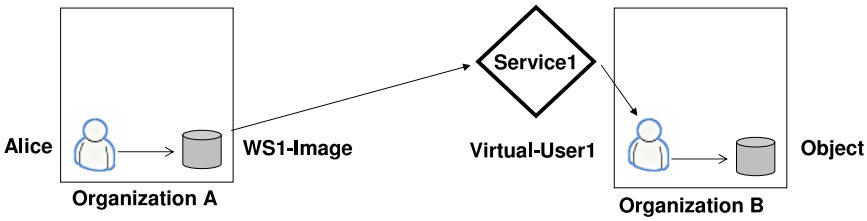

Figure 3. Virtual user and web service image.

During the invocation step, when Alice wishes to use Service1, she is first authenticated by Organization A. Organization A might wish to avoid using the UDDI registry every time it invokes Service1. Also, it might want to have a local representation of Service1 to manage the access control policy locally. To accomplish this, Organization A creates a local WS1-Image accessed by authorized users in A (Figure 3). Organization B may wish to have a local representation of (the remote) Organization A that requests Service1 to virtualize the distant access from Organization A and to manage it like a local access in Organization B. To accomplish this, Organization B creates a local Virtual-User1 with an OrBAC role that enables it to perform the activity corresponding to Service1. Each organization thus controls access to its own resources and services. It is responsible for authenticating its own users when they use services hosted by other organizations.

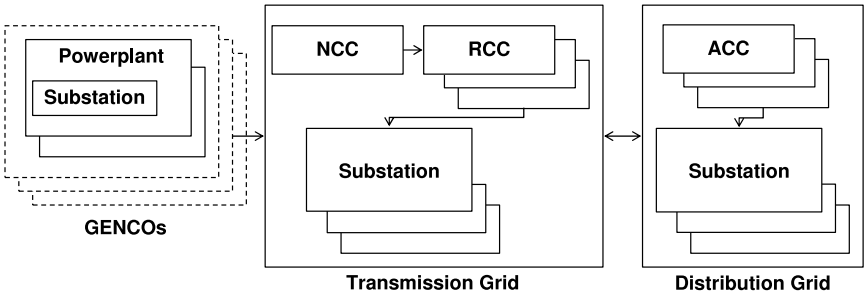

Figure 4. Generic electrical power grid architecture.

The request involves an access (controlled by A's OrBAC policy) to the local object WS1-Image representing Service1 as well as a remote access (controlled by B's OrBAC policy) to B's actions corresponding to Service1. The CISs belonging to Organizations A and B check that all exchanges between A and B are compatible with the agreed contract, and maintain logs of all exchanges to serve as evidence in case of dispute. If the invocation of Service1 is authorized, A sends a SOAP/XML request to the URL of Service1, B executes the request, and sends the result of Service1 to A, which transmits it to Alice.

4. Case Study

This section descibes a case study involving the application of PolyOrBAC to an electrical power grid CII.

4.1 Electrical Power Grid Scenario

Figure 4 presents a generic electrical power grid architecture. One or more electricity generation companies (GENCOs) – each operating several power plants — are connected to transmission grids. Each transmission grid, which is managed by transmission system operators (TSOs), comprises transmission substations monitored by one national and several regional control centers, and is connected to one or more distribution grids. Each distribution grid, which is managed by distribution system operators (DSOs), is composed of distribution substations monitored by area control centers; the grid provides electricity to subscribers (industries and residences) over distribution lines [13].

PolyOrBAC is useful when infrastructure components are required to execute remote actions and access resources from other partner organizations. We employ a real-world scenario to illustrate the application of PolyOrBAC to the electrical power grid CII. The scenario considers the possible cascading effects caused by attacks on the communication channels between TSO/DSO control centers and their substations during emergency conditions (e.g., line overloads). We assume that during emergency conditions a TSO is authorized by a DSO to activate defensive actions that include load shedding. The scenario involves four classes of organizations: transmission regional control centers (TS

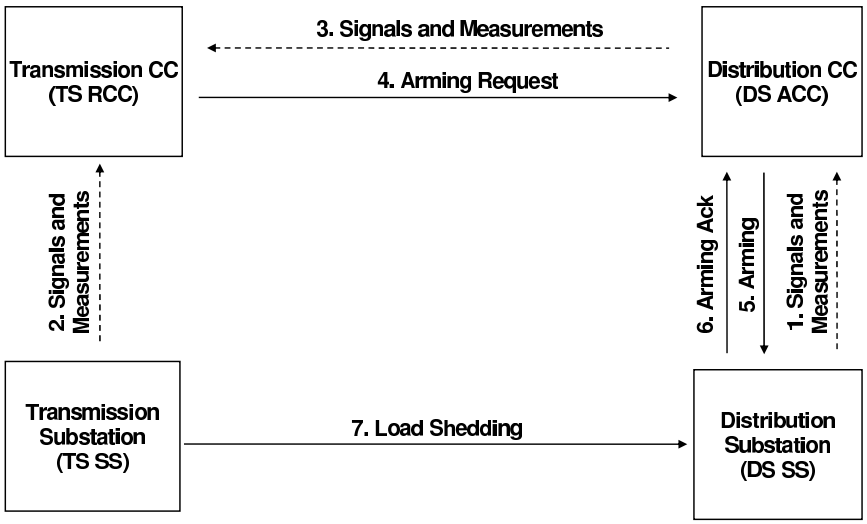

Figure 5. Exchanged commands and signals.

RCCs) managed by TSOs, transmission substations (TS SSs), distribution area control centers (DS ACCs) managed by DSOs, and distribution substations (DS SSs). Figure 5 presents the commands and signals exchanged between these organizations in normal and emergency situations.

During normal operations, the DS SSs send several signals and measurements (power, voltage, frequency) to the TS RCC via their DS ACCs (1 and 3). On the other hand, the TS SSs send signals and measurements (power, voltage, frequency) to their TS RCC (2). The TS RCC monitors the electrical power system and identifies emergency conditions that could be remedied with opportune load shedding applied to specific areas of the grid. To actuate a defensive action, the TS RCC chooses a subset of HV (high voltage) or MV (medium voltage) DS SSs from the list of participating DS SSs in the emergency plan; these DS SSs are subsequently armed by the TS RCC.

In the arming step, the TS RCC sends the requests to preventively arm the selected DS SSs to the concerned DS ACCs (4) to prepare for load shedding. The DS ACCs send the arming order to DS SSs that arm the appropriate monitoring control and defense terminal unit (MCDTU) (5); the armed DS SSs then send acknowledgements to the DS ACC (6). When an emergency situation is detected, the TS SS sends a load shedding command to all the DS SSs participating in the emergency plan, at which point only the previously armed DS SSs perform load shedding over their MCDTUs (7).

4.2 Scenario Interpretation

This section details the web service invocations that are involved in the scenario based on the PolyOrBAC security policy.

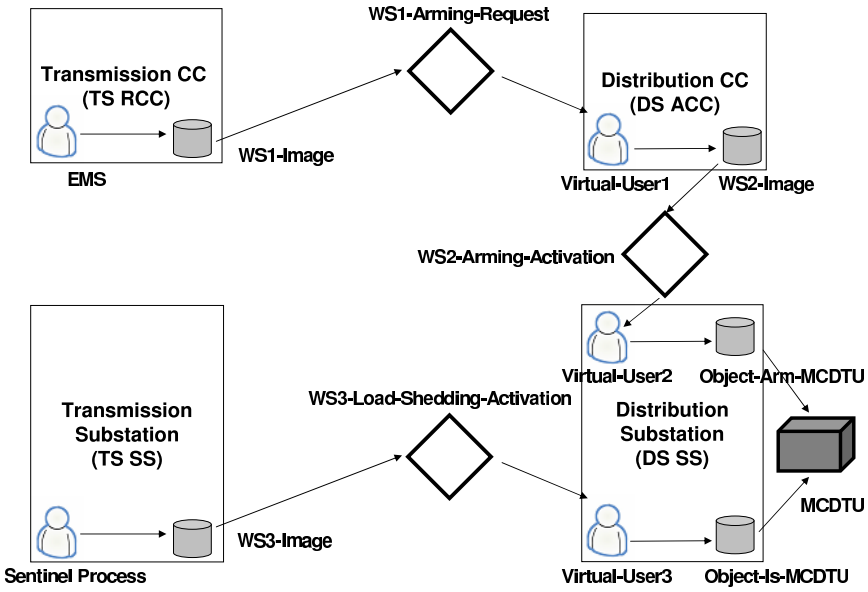

Figure 6. PolyOrBAC applied to the electrical power grid scenario.

Table 1. Instantiated web services.

Service	Provider	Client
WS1-Arming-Request	DS ACC	EMS in the TS RCC
WS2-Arming-Activation	DS SS	Virtual user in the DS ACC
WS3-Load-Shedding-Activation	DS SS	Sentinel process in the TS SS

Figure 6 summarizes the web services, virtual users representing remote organizations that can request web services, ws-images (local images of remote web services that can be invoked), and resources involved in the scenario. The scenario involves four organizations (TS RCC, DS ACC, TS SS and DS SS) and three web services (Table 1).

We assume that the electrical management system (EMS) in the TS RCC orders the DS ACC to arm its DS SS MCDTUs. When the EMS activates WS1-Image, the execution of WS1-Arming-Request is automatically activated. The access of WS1-Image by the EMS is checked against the TS RCC policy and is granted according to the OrBAC rule (Table 2) that manages access control for the Arming Request web service at the level of the organization that invokes the service (i.e., TS RCC).

On the DS ACC side, WS1-Arming-Request tells Virtual-User1 in DS ACC to access object WS2-Image. This access (WS2-Image by Virtual-User1) is checked against the DS ACC policy and is granted according to the OrBAC rule

Table 2. Arming request OrBAC rule at TS RCC.

Rule
Permission(TS RCC, TSO for RCC, Access, RCC Dist. Circuits, Emergency) ∧ Empower(TS RCC, EMS(Subject), TSO for RCC) ∧ Consider(TS RCC, Activate(Action), Access) ∧ Use(TS RCC, WS1-Image(Object), RCC Dist. Circuits) ∧ Hold(TS RCC, EMS, Activate, WS1-Image, Emergency) ∧ ⇒ is-permitted(EMS, Activate, WS1-Image)

Table 3. Arming request/activation OrBAC rule at DS ACC.

Rule
Permission(DS ACC, DSO for ACC, Access, ACC Dist. Circuits, Emergency) ∧ Empower(DS ACC, Virtual-User1(Subject), DSO for ACC) ∧ Consider(DS ACC, Activate(Action), Access) ∧ Use(DS ACC, WS2-Image(Object), DS ACC Dist. Circuits) ∧ Hold(DS ACC, Virtual-User1, Activate, WS2-Image, Emergency) ∧ ⇒ is-permitted(Virtual-User1, Activate, WS2-Image)

(Table 3) that manages access control for the Arming Request and Activation web services at the level of the organization that provides the service (i.e., DS ACC).

Table 4. Arming activation OrBAC rule at DS SS.

Rule
Permission(DS SS, DSO for SS, Access, DS SS Dist. Circuits, Emergency) ∧ Empower(DS SS, Virtual-User2(Subject), DSO for SS) ∧ Consider(DS SS, Activate(Action), Access) ∧ Use(DS SS, Object-Arm-MCDTU(Object), DS SS Dist. Circuits) ∧ Hold(DS SS, Virtual-User2, Activate, Object-Arm-MCDTU, Emergency) ∧ ⇒ is-permitted(Virtual-User2, Activate, Object-Arm-MCDTU)

When Virtual-User1 activates object WS-Image2, WS2-Arming-Activation is automatically activated, then Virtual-User2 activates Object-Arm-MCDTU in DS SS, and the physical arming command is executed over the MCDTU. This access (Virtual-User2 to Object-Arm-MCDTU) is checked against the DS SS policy and is granted according to the OrBAC rule (Table 4), which manages access control for the Arming Activation web service in DS SS. WS3-Load-Shedding-Activation is negotiated and activated in the same way.

5. Conclusions

The PolyOrBAC security framework provides excellent support for the access control and collaboration requirements in CIIs. Web services are leveraged to provide decentralized management of access control policies and to enable organizations to mutually negotiate contracts for collaboration. Organizations retain their own resources, services, applications, operating systems, functioning rules, goals and security policy rules. However, each organization is responsible for authenticating its users when they use other organizations' services. The framework also facilitates the management and integration of new organizations in a CII and ensures user privacy and non-disclosure of data and services. Moreover, the framework can handle hardware and software heterogeneities; network segments and physical/logical equipment belonging to the collaborating CII organizations can be integrated seamlessly [9].

The framework is currently being implemented in a Java environment using IBM Eclipse IDE and SWI-Prolog. Our future research will focus on meeting availability requirements using obligation rules and integrity requirements by monitoring information flows of different degrees of criticality [24]. Our research will also attempt to incorporate results from trust negotiation [23] and trust-based access control [3] in the PolyOrBAC framework.

Acknowledgements

This research was partially supported by CRUTIAL, a European FP6-IST Project and by PolSec, a LAAS Project. The authors also wish to thank Fabrizio Garrone and Giovanna Dondossola for their contributions to the case study described in this paper.

References

[1] A. Abou El Kalam, S. Benferhat, A, Miege, R. El Baida, F. Cuppens, C. Saurel, P. Balbiani, Y. Deswarte and G. Trouessin, Organization based access control, *Proceedings of the Fourth IEEE International Workshop on Policies for Distributed Systems and Networks*, pp. 120–134, 2003.

[2] A. Abou El Kalam, Y. Deswarte, A. Baina and M. Kaaniche, Access control for collaborative systems: A web services based approach, *Proceedings of the IEEE International Conference on Web Services*, pp. 1064–1071, 2007.

[3] W. Adams and N. Davis, Toward a decentralized trust-based access control system for dynamic collaboration, *Proceedings of the Sixth Annual IEEE SMC Information Assurance Workshop*, pp. 317–324, 2005.

[4] M. Amin, North America's electricity infrastructure: Are we ready for more perfect storms? *IEEE Security and Privacy*, vol. 1(5), pp. 19–25, 2003.

[5] D. Bell and L. LaPadula, Secure Computer Systems: Unified Exposition and MULTICS Interpretation, Technical Report ESD-TR-75-306, MTR-2997 Rev. 1, MITRE Corporation, Bedford, Massachusetts, 1976.

[6] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, F. Yergeau and J. Cowan (Eds.), Extensible Markup Language (XML) 1.1, Recommendation, World Wide Web Consortium, Cambridge, Massachusetts (www.w3.org/TR/2004/REC-xml11-20040204), 2004.

[7] G. Brose, A view-based access control model for CORBA, in *Secure Internet Programming: Security Issues for Mobile and Distributed Objects (LNCS 1603)*, J. Vitek and C. Jensen, Springer-Verlag, London, United Kingdom, pp. 237–252, 2001.

[8] L. Clement, A. Hately, C. von Riegen and T. Rogers (Eds.), UDDI Version 3.0.2, Organization for the Advancement of Structured Information Standards, Billerica, Massachusetts (uddi.org/pubs/uddi_v3.htm), 2005.

[9] F. Cuppens, N. Cuppens-Boulahia, T. Sans and A. Miege, A formal approach to specify and deploy a network security policy, in *Formal Aspects in Security and Trust*, T. Dimitrakos and F. Martinelli (Eds.), Springer, Berlin-Heidelberg, Germany, pp. 203–218, 2004.

[10] G. Dondossola, G. Deconinck, F. Di Giandomenico, S. Donatelli, M. Kaaniche and P. Verissimo, Critical utility infrastructural resilience, *Proceedings of the Workshop on Security and Networking in Critical Real-Time and Embedded Systems*, 2006.

[11] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn and R. Chandramouli, Proposed NIST standard for role-based access control, *ACM Transactions on Information and System Security*, vol. 4(3), pp. 224–274, 2001.

[12] T. Fink, M. Koch and C. Oancea, Specification and enforcement of access control in heterogeneous distributed applications, *Proceedings of the International Conference on Web Services*, pp. 88–100, 2003.

[13] F. Garrone, C. Brasca, D. Cerotti, D. Codetta Raiteri, A. Daidone, G. Deconinck, S. Donatelli, G. Dondossola, F. Grandoni, M. Kaaniche and T. Rigole, Analysis of New Control Applications, Deliverable D2, The CRUTIAL Project, CESI Ricerca, Milan, Italy (crutial.cesiricerca.it/content/files/Documents/Deliverables%20P1/WP1-D2-final.pdf), 2007.

[14] M. Harrison, W. Ruzzo and J. Ullman, Protection in operating systems, *Communications of the ACM*, vol. 19(8), pp. 461–471, 1976.

[15] J. Laprie, K. Kanoun and M. Kaaniche, Modeling interdependencies between the electricity and information infrastructures, *Proceedings of the Twenty-Sixth International Conference on Computer Safety, Reliability and Security*, pp. 54–67, 2007.

[16] M. Lorch, S. Proctor, R. Lepro, D. Kafura and S. Shah, First experiences using XACML for access control in distributed systems, *Proceedings of the ACM Workshop on XML Security*, pp. 25–37, 2003.

[17] N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, Y. Lafon, and C. Barreto (Eds.), Web Services Choreography Description Language Version 1.0, Candidate Recommendation, World Wide Web Consortium, Cambridge, Massachusetts (www.w3.org/TR/2005/CR-ws-cdl-10-20051109), 2006.

[18] A. Miege, Definition of a Formal Framework for Specifying Security Policies: The OrBAC Model and Extensions, Ph.D. Thesis, Department of Computer Science, Ecole Nationale Superieure des Telecommunications (TELECOM ParisTech), Paris, France, 2005.

[19] N. Mitra (Ed.), SOAP Version 1.2, Recommendation, World Wide Web Consortium, Cambridge, Massachusetts (www.w3.org/TR/2003/REC-soap12-part0-20030624), 2003.

[20] S. Oh and S. Park, Task-role-based access control model, *Information Systems*, vol. 28(6), pp 533–562, 2003.

[21] S. Rinaldi, J. Peerenboom and T. Kelly, Identifying, understanding and analyzing critical infrastructure interdependencies, *IEEE Control Systems*, vol. 21(6), pp. 11–25, 2001.

[22] R. Sandhu, E. Coyne, H. Feinstein and C. Youman, Role-based access control models, *IEEE Computer*, vol. 29(2), pp. 38–47, 1996.

[23] K. Seamons, T. Chan, E. Child, M. Halcrow, A. Hess, J. Holt, J. Jacobson, R. Jarvis, A. Patty, B. Smith, T. Sundelin and L. Yu, TrustBuilder: Negotiating trust in dynamic coalitions, *Proceedings of the DARPA Information Survivability Conference and Exposition*, vol. 2, pp. 49–51, 2003.

[24] E. Totel, J. Blanquart, Y. Deswarte and D. Powell, Supporting multiple levels of criticality, *Proceedings of the Twenty-Eighth Annual Symposium on Fault Tolerant Computing*, pp. 70–79, 1998.