RICH PRESENCE AUTHORIZATION USING SECURE WEB SERVICES

Li Li and Wu Chou
Avaya Research Lab, Avaya Inc., 233 Mt Airy Road, Basking Ridge, New Jersey, U.S.A.

Keywords: Role-Based Access Control (RBAC), Presence Authorization, Web Services.

Abstract: This paper presents an extended Role-Based Access Control (RBAC) model for rich presence authorization using secure web services. Following the information symmetry principle, the standard RBAC model is extended to support data integrity, flexible and intuitive authorization specification, efficient authorization process and cascaded authority within web services architecture. In conjunction with the extended RBAC model, we introduce an extensible presence architecture prototype using WS-Security and WS-Eventing to secure rich presence information exchanges based on PKI certificates. Applications and performance measurements of our presence system are presented to show that the proposed RBAC framework for presence and collaboration is well suited for real-time communication and collaboration.

1 INTRODUCTION

In communication and collaboration, presence information describes a person's availability and willingness to communicate or participate, as well as the environment, such as location. Many internet IM systems offer certain degrees of presence information, such as "available, busy, and away." Presence information is exchanged between a "presentity" who owns and discloses the presence states and a "watcher" who requests (subscribes) and receives the presence states. This information exchange occurs within a presence session until either the presentity or the watcher terminates the session. By this generalization, a presentity or a watcher can be a person, an automaton, or a group of such entities. Presence information in general improves social connections (keep in touch with family and friends) and collaboration within teams and organizations. When used properly, it significantly reduces the uncertainty and cost in communication and collaboration. For this reason, industrial standard organizations, e.g. IETF (Day 2000) and Parlay (Parlay X 2007), have begun to develop presence architectures and standard protocols.

Rich presence contains fine-grained and in some way private information about a presentity. For example, Parlay X Web Service Presence (Parlay X 2007) specifies 10 presence attributes, including Activity, Place, Sphere, Privacy, etc. where each attribute can have 20 to 30 values about the presence state in that category. Some of the presence values disclose the presentity's private life, e.g. the presentity is in a bar, prison or sleeping.

Albeit being useful, rich presence information may inevitably raise the privacy concerns about the presentity, and the needs to reveal or conceal such information have to be justified as appropriate. Negatively correlated with presence, privacy is a highly contextual and subjective issue and must be dealt with carefully to achieve a balance between ease of collaboration and personal freedom. Therefore, a privacy aware presence architecture must provide intuitive, uniform and flexible presence authorization model for the presentities to determine what presence information should be disclosed to which watcher and under what kind of context. For example, a person can expose different presence information to friends, parents, co-workers and managers. On the other hand, a company may enforce a central presence policy and leave the rest to the discretion of its employees.

Our observation is that the presence authorization to large extent is based on the social relation between the watcher and the presentity. This relation can be characterized by the role of the watcher from the perspective of presentity. The notion of role is an intuitive concept, which is meaningful to people even without technical background. For this reason,

Li L. and Chou W.
RICH PRESENCE AUTHORIZATION USING SECURE WEB SERVICES.
DOI: 10.5220/0001842700290036
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

we adopt Role-Based Access Control (RBAC) and apply the NIST standard RBAC model (Sandhu 2000) to presence authorization.

Various RBAC models have been traditionally used in business enterprises for access control to resources. The basic RBAC model assigns a user to roles which are associated with permissions. Such model grants a user only to permissions associated with the roles activated for him. Roles can be organized into hierarchies in which the permissions can be inherited. Constraints can also be added to various RBAC components to prevent undesired configurations. One typical constraint is to enforce Separation of Duty (SOD) such that mutually exclusive roles are not assigned to the same user at the same time. RBAC is a natural fit for enterprises where roles represent the tasks, responsibilities and qualifications that users assume within an enterprise. RBAC thus provides a valuable level of abstraction and modularization to specify and enforce authorization policies based on the Least Privilege Principle that cannot be achieved by other access control models.

These fundamental RBAC security concepts and mechanisms are applicable to presence authorization and we do not have to reinvent the wheel. However, presence authorization cannot be regarded as a special case of the standard RBAC models. The differences are elaborated below.

First, the concept of role in presence authorization is different from the standard RBAC models. In presence system, roles represent relations between two users (watcher and presentity), whereas in standard RBAC model they represent functions of users within an enterprise. This implies that role assignment in presence system depends on both participants, instead of on one user as in the standard RBAC models. Furthermore, in presence system, the role assignment may depend on other contextual factors, such as the time and nature of the engagement.

Second, the standard RBAC models do not support cascaded authorization, where a central authority establishes common authorization rules and its subordinates can refine them. Cascaded authority is necessary to assert certain global control while permitting personal choices in organizational and social environment. For instance, an enterprise may have global presence rules that require or prohibit disclosure of certain activities, but the employees can exercise personal judgement within the perimeters of these rules. An online community may also mandate a minimal set of presence requirements and leave the rest to its members.

Third, the standard NIST RBAC model only defines the core concepts and semantics, but it leaves the user authentication, permission presentation and protocol integration to the implementations. In particular, we need to seamlessly integrate RBAC models with the web services architecture which is based on the notion of service contract. Since the contract exposes the data model about the presentities, the RBAC model can be used to grant contract access to only qualified watchers. Another reason to incorporate contract into RBAC model is to maintain data integrity such that the contract (presence data model) discovered by the watchers is identical to the one published and updated by the presentity. For permission representation, RBAC model should allow both watchers and presentities to select presence states at different level of granularity. Overall, the presence protocols and RBAC models should be combined to maximize symmetric information flow such that the watchers and presentities have equal knowledge and control over presence information.

To address these issues, we developed a secure web service based rich presence architecture using the extended RBAC model that we developed for presence authorization. The proposed RBAC based model extends the role assignment mechanism and adds new features to support cascaded authority. It emphasizes the importance of context and presence data model in order to facilitate web service discovery and invocations. In particular, we propose an abstract presence data model by combining Parlay X Web Service Presence (Parlay X 2007) and RFC5025 authorization (Rosenberg 2007). In this architecture, we use WS-Eventing (WS-Eventing 2006) as the presence protocol. We treat the presentity as event source and presence information that is sent to the watchers as events. WS-Eventing is composed with WS-Security (WS-Security 2006) for end-to-end user authentication, message integrity and confidentiality. Using WS-Eventing instead of the full-blown Parlay X Presence Web Service allows us to limit the number of running web services on communication endpoints, and in addition, provide a generic methodology for presence authorization in collaboration and social networks.

The rest of the paper is organized as follows: In Section 2, we survey some related work in RBAC, presence authorization and web services. In section 3, we outline the presence protocol over the Web in which the RBAC model operates. We then present our extended RBAC models in two steps. First, the basic model that incorporates the relation, context and data models is described to address the specific need of presence authorization. Second, we formalize the special constraints for role hierarchy to support cascaded RBAC models. In section 4, we discuss a prototype presence architecture based on

secure web services using WS-Security and WS-Eventing with PKI and UDDI servers. Within this architecture, we study the performance and applications of the proposed presence system for web services enabled softphones. The paper is concluded in section 5 with some thoughts on future work.

2 RELATED WORK

Privacy aware design has been an active research area in recent years (Hong 2004, Lederer 2004, Laugheinrich 2001, Jorns 2004). A recurring principle in these researches is "information symmetry" such that parties participating in information exchange should have equal awareness and control over the information flows.

The concept of role has shown up in several presence architectures, but none of them formally adopts RBAC models.

Hong et al (Hong 2004) proposes an analytical privacy risk model which considers the roles and social relationships between presentity and watcher as one of the input parameters to the model. The authors recommend "push" mode delivery of presence information to lower privacy risks, because the presentity initiates the data transmission.

Lederer et al (Lederer 2004) discusses five pitfalls in privacy aware design based on their experience of developing a GUI for setting presence preferences. A preference rule has three parts: a *role*, *situations* and a *face*, meaning that if someone of this *role* (roommate, parent, boss) requests in these *situations*, it should show the presence states defined in the *face*. This clearly is a RBAC based approach, but the author does not make any reference to RBAC. Their usability studies showed that users expressed discomfort with hiding presences behind faces, which is a permission definition issue in RBAC models. The general lessons are that the privacy component should be simple, transparent, intuitive and easy to change. The paper recommends the practice of *plausible deniability* – people can choose to ignore presence requests or customize information disclosure without having to explaining why.

Jorns (Jorns 2004) provides an overview of location based systems in the face of privacy and proposes a pseudonym based scheme to enhance privacy in a mobile communication system. The scheme is based on "symmetric information data arrangement" where the presentity has control over to whom the location information is disclosed using pseudonym management, in contrast to "asymmetric arrangement" where a presentity has no such control. Hengartner et al (Hengartner 2004) proposes a digital certificate based access control framework for sharing personal location information in a heterogeneous environment. The authorization is achieved by checking the trust delegation relations expressed in certificates. The paper also uses "local names" to designate a group of people in a single certificate instead of creating certificates for each person. Local names can be regarded as roles in our RBAC approach.

RBAC model has been standardized by NIST (Sandhu 2004), and it is still a very active research area for improvements (Chen 2008, Ni 2007, Zhang 2008). Traditionally, it is used for enterprise resource protection. There are some recent efforts to extend RBAC to incorporate privacy (Ni 2007, 2008) and geospatial and temporal factors (Chen 2008). On one hand, however, these generic models are too complex for presence, and on the other they do not address the specific issues in presence and web services. For instance, the privacy aware RBAC model (Ni 2008) does not consider the relational information in role assignment. Its new features, such as action, purposes and obligations do not seem applicable to presence system. Still focusing on enterprise usages, this model does not address the data integrity or cascaded authority in web service based presence exchange. For this reason, we rely on the standard NIST RBAC model.

Godefroid et al (Godefroid 2000) developed a framework to ensure the correctness of presence authorization rules using model checking techniques. Each rule has a condition and an action. Conditions are defined in terms of the presentity's state, such as "user i's door is open" and the action includes invite, accept, reject, etc. However, these rules are not based on RBAC models.

There are many research work based on the SIP presence framework (Rosenberg 2004, 2007, Beltran 2008, Singh 2006). IETF RFC5025 (Rosenberg 2007) defines presence authorization rules for SIP Instant Messaging and Presence (SIMPLE). This specification is based on a three level presence data model. The top level consists of three entities: device, service and person. The second level is the attributes: activity, place, mood, etc., and the third level is a range of presence values. The authorization rule consists of condition, action and transformations, and it is applied to each presence subscription. The condition checks the identity of the watcher and the sphere of the presentity. If the condition matches, the action and transformations are carried out. The actions (block, confirm, polite-

block and allow) determine the status of the subscription. The transformations define which the presence entities and attributes will be included. However, the authorization rules of (Rosenberg 2007) are user centric instead of role based and the transformations do not apply to presence values. RFC3856 (Rosenberg 2004), the protocol for SIP presence subscription, does not provide a mechanism for a watcher to indicate the presence preference in subscriptions and to know the authorized presence attributes. This information asymmetry may lead to information overload to the watchers. In particular, Singh et al (Singh 2006) outlines some authentication mechanisms within SIP architecture, including using PKI certificates.

Parlay X Presence Web Service (Parlay X 2007) defines a two level presence data model. The top level is a set of attribute types and the second level is a set of values about the presence state of a person. The presence web service includes interfaces for watchers to subscribe or poll presence data. Similar to SIMPLE, a watcher can request a subset of presence attributes, and the presentity can grant permissions (yes or no) to the requested attributes. When the presentity changes the authorization of an existing presence subscription, the service will notify watchers accordingly. However, neither watcher nor presentity can select presence values from the attributes. Also, the standard does not specify a standard presence authorization process.

WS-Security (WS-Security 2006) is an OASIS standard for securing end-to-end web services interactions. It provides support for user name tokens and password, message timestamp, digital signature of message, and message encryption based on a variety of cartographical schemes, including PKI architecture. WS-Eventing (WS-Eventing 2006) is a W3C submission and in the process of becoming a W3C standard recommendation. It defines a subscribe/notify web service interaction pattern for managing event subscriptions between event sources and event sinks. It has been applied in web service oriented communications as shown in (Chou 2007, 2008). UDDI (UDDI 2002) is another OASIS standard for web services to publish their structural information so that services can be discovered and consumed. A UDDI repository at the minimum contains the name and URI of web services, but it is extensible to store other type of information.

3 EXTENDED RBAC MODEL

In our web service based presence framework, watchers and presentities interact over the Web though the web service protocols. The high level interaction sequence that leads to a presence session is depicted in the following diagram (Figure 1). In this architecture, a presentity *s* first publishes its web services onto the Web (UDDI registry or Web server). A watcher *w* interested in *s* can discover the presentity's data model *d* over the Web, subject to the authorization approval. To obtain the presence, the watcher *w* subscribes to *d* with its preference *q*. The subscription is then subject to the authorization process. Upon approval, a filter *f* is returned along with the current presence states *v* of *s* and a reference *m* to the subscription. Subsequent presence event from *s* will be delivered to *w* only if it matches *f*. Either *s* or *w* can cancel the subscription *m* to terminate the presence session.

This sequence of operations is valid regardless if a presence server is deployed between *w* and *s* or there are peer-to-peer connections between them. For the same reason, the RBAC models can be deployed either in the presence server or in the presentity or both. If a presence server is used, the presentity can delegate part or all authorization to the server. For clarity, this diagram only illustrates the deployment of RBAC model on the presentity. Regardless where the RBAC model is installed, the authorization process will be the same.

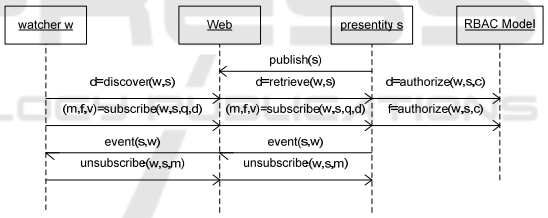

Figure 1: High level Web service based presence protocol.

Our extended RBAC model M consists of the following sets of entities: presentities (S), watchers (W), contexts (C), roles (R), presence data model (D), permission actions (A), and data model transformations (T). The sets S and W correspond to the users (U) of the standard RBAC model. The sets P and R have the same meaning as in the standard RBAC model.

The following additional entities are introduced in our extended RBAC model. The set C denotes a set of application dependent contexts. The set D is a set of presence data models (service contracts) of the presentity. Presentities S are represented as trees. The set T is a set of transformations that create valid subtrees from D. The subtree is a permission tree if the nodes are decorated with actions defined in the set A (Rosenberg 2007), e.g. allow (A), block (B), polite-block (L) and confirm (I). In permission trees,

a node without action will inherit its parent's action, whereas a node's own action overrides its parent's action.

The authorization specification of M is defined by the following mappings (Formula 1).

$$W \times S \times C \rightarrow [Role] \rightarrow R \rightarrow [Granted] \rightarrow C \times T(D) \quad (1)$$

$$f(w, s, c) = Requested(w, s, c) \cap Granted(Role(w, s, c)) \quad (2)$$

$$Received(w, s, c) = f(w, s, c) \cap Presence(s) \quad (3)$$

Here $Requested(w,s,c) \in C \times T(D)$ denotes the presence request from watcher *w* to the presentity *s* in the context *c*. The presence filter *f* for this request can be computed by Formula 2 following the relations in M. Formula 3 shows that the presence events from *s* is filtered by *f* before received by *w*.

To elaborate the authorization process, Figure 2 shows that a watcher *w*, subscribing certain presence data represented as a tree *q*, is assigned to role *r*. The role *r* is associated with a permission tree *t* decorated with some actions. In this case, the watcher *w* requests presence values *v11* and *v12* of the attribute *a1* and all values of *a2*, whereas the permission tree grants access to values *v11* of the attribute *a1* and all values of *a2* only if it is confirmed by *s*. Assuming the presentity rejects attribute *a2*, then we have *f={a1/v11}*. If there is a presence event *{a1/v11,a1/v12}*, only *{a1/v11}* will be delivered to the watcher due to the filtering.

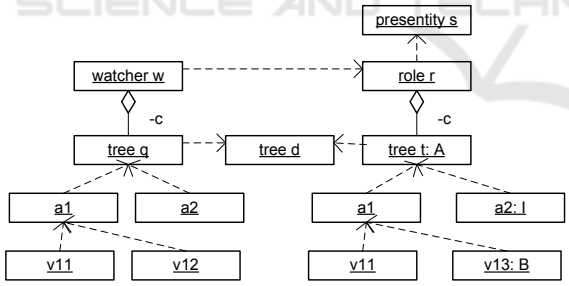

Figure 2: A RBAC model with watcher request and role permission represented as trees created from the same data model.

The data security and integrity is maintained in this model because the watchers can only discover elements in D and that all permission trees are created from D. By using permission trees, the presentity can designate presence disclosure with different granularity, ranging from the entire tree to individual presence values. The information symmetry is achieved by the same data model and the presence filter shared between the watcher and presentity during the presence session. In other words, the shared filter guarantees that no presence values will ever leak to the watcher without the consensus of the presentity. By the same token, no granted presence values will be concealed from the watcher. The only exception to this rule is polite-block, where the presentity hides its presence from the watcher. However, this is a well accepted privacy practice following the Plausible Deniability principle (Lederer 2004) that favours privacy over presence.

This extended RBAC model is particularly suitable for creating cascaded authorities using role hierarchy. Our goal is, given a RBAC model M_1 representing a central authority, to derive a new model M_2 that maintains all the mandatory aspects of M_1. To support this, we propose new constraints on standard role hierarchy to prevent the mandatory aspects from being overridden. Since what constitutes the mandatory aspects is domain dependent, we introduce a new attribute, *final* (F), to decorate the permission trees in M_1. This attribute separates nodes that must be inherited by M_2 from those that can be overridden, thus providing protection as well as flexibility in cascaded models.

In this process, we leave out the role assignments until the final RBAC model is assembled. More formally, let $M_1 = (R_1, C_1, T_1 (D_1), A_1)$ and $M_2 = (R_2, C_2, T_2 (D_2), A_2)$, representing the roles, context, transformations, and actions of two RBAC models respectively. M_2 is a valid derivation of M_1 iff the following conditions hold:

$$D_2 \subseteq D_1 \wedge A_2 \subseteq A_1 \wedge C_2 \subseteq C_1 \quad (4)$$

$$\forall (c,t_2) \in C_2 \times T_2(D_2) \exists (c,t_1) \in C_1 \times T_1(D_1) \\ (d(t_1) = d(t_2) \rightarrow (role(c,t_1) \in junior(role(c,t_2)) \\ \wedge final(t_1) \in t_2)) \quad (5)$$

By the first condition, the data models, actions and contexts of M_2 are sanctioned by M_1. The second condition ensures that if a tree t_2 in M_2 has the same data model as a tree t_1 in M_1, then t_2's role inherits t_1's role and all final nodes of t_1, if any, are inherited by t_2 as well. The final model M3 is assembled by $M_3 = M_1 \cup M_2$.

The following diagram (Figure 3) illustrates two models: M_1 with role manager and M_2 with role director that satisfy these conditions. In this case, M_1 marks a1:A (allow attribute a1) as *final* but leaves a2 open. The derived M_2 changes a2 from I (confirm) to A (allow), and adds a new node a3. If t_2 had a node a1:B (block attribute a1), it would violate the conditions. The final RBAC model contains both roles. This derivation process can be repeated as long as necessary.

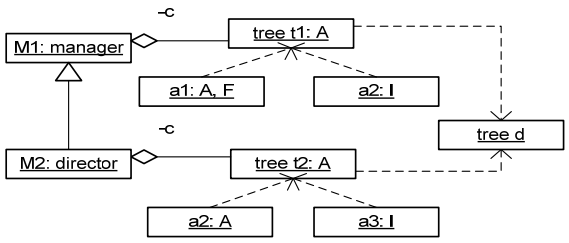

Figure 3: Cascaded RBAC models using constrained role hierarchy.

This role hierarchy can be interpreted by standard RBAC semantics, as either permission hierarchy or activation hierarchy (Sandhu 2000). To enable efficient computation of permissions, the cascaded models can be flattened into one final RBAC model whose permission trees are the unions of all permission trees along the inheritance relations. The permission tree of a senior role will inherit nodes from the junior role if it does not have them, and it will overwrite nodes from the junior role by its own. Following this rule, the collapsed tree for the role director in M_2 is {a1:A, a2:A, a3:I}.

4 IMPLEMENTATION

The presence architecture is illustrated in Figure 4, which consists of a UDDI server, presentity, watcher and a Certificate Authority (CA) server. Each presentity and watcher in our study is a softphone hosting identical set of web services (WS-Eventing and WS-Security). PKI certificates (public and private keys) generated from the CA are administrated to each softphone. Each softphone publishes or deletes its web services registry and its public key in the UDDI server, when the user logs in or off the softphone. The UDDI server thus acts as a coarse grained presence server and a public key repository. To start a presence session outlined in the previous section, the watcher's softphone finds the presentity web service location from the UDDI and invokes them accordingly.

The security for all message exchanges, including presence subscriptions and events, is achieved by WS-Security based on the PKI certificates. Figure 5 shows the WS-Security component intercepts outgoing and incoming SOAP messages to add and check security features. On the presentity side, once the presence subscription passes the security check, it is submitted to the presence authorization module which employs the RBAC model to authorize the request as discussed in previous section.

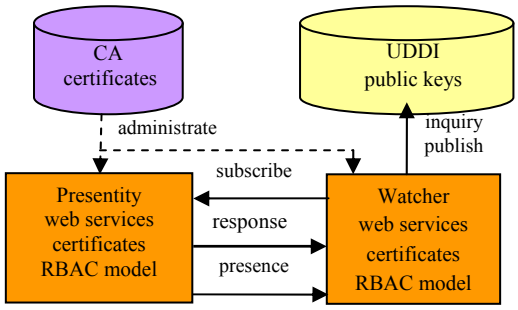

Figure 4: High level presence architecture based on web services.

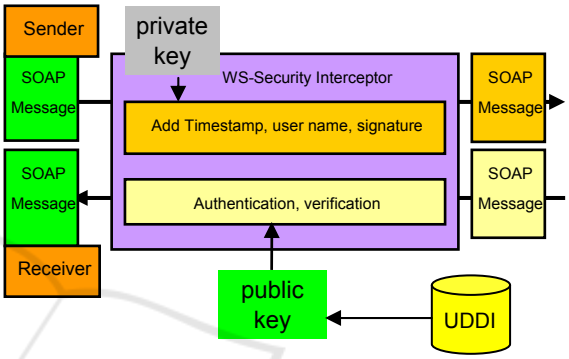

Figure 5: SOAP message security procedure using WS-Security.

The proposed RBAC based presence architecture prototype was implemented on top of the Google Android mobile phone platform (Android 2008). We developed WS-Eventing and utilized the WS-Security infrastructure provided by gSOAP (gSOAP 2008). Our prototype system has four roles: manager, peer, subordinate and anonymous, with some predefined permissions against the Parlay X Presence. To demonstrate the approach, four presence attributes: Activity, Privacy, Place and Sphere, of the Parlay X Presence data model are selected. The role anonymous, inherited by other roles, grants everyone access to the presence data model and some rudimentary presence information. Figure 6 shows a screenshot of the presence subscription interface used by the watchers. Figure 7 shows the presence states displayed in the softphone.

The presence information is combined with the call management of the softphone to enable presence aware communication. One application is presence aware calling that can put an active call on hold when it is not answered and automatically reconnect when the callee presence state changes to available. This avoids the hassle for the caller to frequently try the luck of getting connected with the callee.

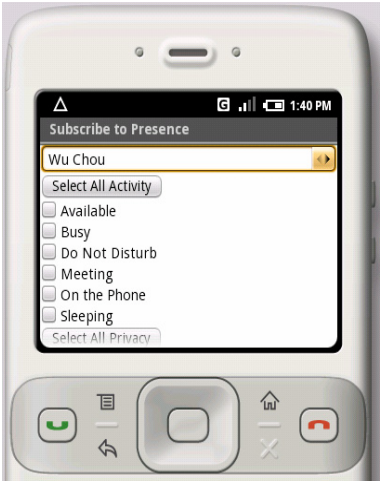

Figure 6: Screenshot of presence subscription interface on the softphone.

Figure 7: Screenshot of a user's presence state displayed in the contact list on watcher's softphone.

The performance measurement of the presence system is summarized in the following table (Table 1). The time (millisecond) were averaged over 10 to 100 trials on a notebook computer with 1.79GHz CPU and 1G RAM running Windows XP Professional in a LAN environment.

The "Watcher" column is the round-trip latency of presence subscription messages measured at the watcher side, with and without WS-Security. Similarly, the "Presentity" column is the total subscription processing time on the presentity side, with and without WS-Security. The total processing time includes SOAP engine and WS-Eventing module. The "WS-Eventing" is the time spent in executing the WS-Eventing code, which included the presence authorization. These results show that the overall performance is acceptable for real-time telecommunication.

Table 1: Performance measurement of the prototype.

Component	Watcher	Presentity
with WS-Security	20.35	7.45
no WS-Security	17.27	5.77
WS-Eventing	N/A	4.76

It should be pointed out that WS-Security ensures end-to-end security that is agnostic of transport protocols. This level of security allows sensitive presence information to be passed between value-add 3rd party applications without sacrificing message integrity and confidentiality. The PKI certificates also eliminate the need for shared secret, which is difficult or even impractical in open-ended communications.

The web service based approach is also extensible by service composition. For instance, if we need message reliability, we can compose a proper web service for reliability with the existing ones without any change to the existing service implementations.

5 CONCLUSIONS

This paper presented an extended RBAC model for presence authorization and a presence architecture using this model with secure web services for privacy protection. The standard RBAC model is extended in two ways. The first is to incorporate presence relation, context and data model to support data integrity, flexible authorization specification and efficient authorization process within web services architecture. The second is to introduce the constraints for cascading RBAC models using role hierarchy to support central authority. The effectiveness of the extended RBAC model is illustrated in facilitating information symmetry in rich presence exchange. A prototype implementation using WS-Security, WS-Eventing, UDDI registry and PKI CA server and application of the proposed presence architecture was demonstrated based on web services enabled softphones. The experimental results indicated that the system performance is well suited for real-time communication and collaboration.

For future work, we will study the combination of RBAC models with the context of enterprise organizations and social networks, to enable presence aware communication and collaboration efficiently without loosing privacy.

ACKNOWLEDGEMENTS

The authors would like to thank Wei Zhang and Yanbing Yu for their contributions to this project.

REFERENCES

Android - An Open Handset Alliance Project. In: http://code.google.com/android/

Beltran, V. and Paradells, J.: Middleware-Based Solution to Offer Mobile Presence Services. In: Mobileware'08, February, 2008 (2008)

Chen, L. and Crampton, J.: On Spatio-Temporal Constraints and Inheritance in Role-Based Access Control. In: ASIACCS'08, March, 2008, pages 205-216 (2008)

Chou, W., Li, L., and Liu, F.: Web Services Methods for Communication over IP, ICWS 2007, pages 372-379, Salt Lake City, July 2007 (2007)

Chou, W. and Li, L.: WIPdroid – a two-way web services and real-time communication enabled mobile computing platform for distributed services computing, Proceedings of International Conference on Services Computing 2008, July 2008, Vol. 2, pages 205-212, July 2008

Day, M., Rosenberg, J., and H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, February 2000 (2000)

gSOAP. In: http://gsoap2.sourceforge.net/

Godefroid, P., Herbsleb, J.D. Jagadeesan, L.J., and Li, D.: Ensuring Privacy in Presence Awareness Systems: An Automated Verification Approach. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work, pages: 59-68, 2000 (2000)

Hong, J.I., Ng, J.D., Ledere, S., and Landay J.A.: Privacy Risk odels for Designing Privacy-Sensitive Ubiquitous Computing Systems. In: DIS2004, August 1-4, 2004, pages 91-100 (2004)

Jorns, O.: Privacy Enhancing Architectures Overview. In: Intensive Program on Information and Communication Security: Secure Embedded Systems (IPICS'04), November 25, 2004 (2004)

Langheinrich M.: Privacy by Design – Principles of Privacy-Aware Ubiquitous Systems. In: Proceedings of the 3rd international conference on Ubiquitous Computing, pages: 273-291, (2001)

Lederer, S., Hong, J.I., Dey, A.K., and Landay J.A.: Personal privacy through understanding and action: five pitfalls for designers. In: Personal and Ubiquitous Computing, Volume 8, Issue 6, pages: 440-454 (November 2004)

Ni, Q. and Trombetta, A.: Privacy-aware Role Based Access Control. In: SACMAT'07, June, 2007, pages 41-50 (2007)

Ni, Q. et al.: Conditional Privacy-Aware Role Based Access Control, Computer Security ESORICS 2007, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, pages 72-89, 2008.

Parlay X: Draft ETSI ES 202 391-14 v0.0.8 (2007-06), Open Service Access (OSA), Parlay X Web Services, Part 14: Presence (Parlay X 2) (2007)

Rosenberg, J., Request for Comments: 3856, A Presence Event Package for the Session Initiation Protocol (SIP), August 2004 (2004)

Rosenberg, J., Request for Comments: 5025, Presence Authorization Rules, December 2007 (2007)

Sandhu, R., Ferraiolo, D., Kuhn R.: The NIST Model for Role-Based Access Control: Towards A Unified Standard. In: Proceedings of 5th ACM Workshop on Role Based Access Control, July 26-27, 2000 (2000)

Singh, V.K. and Schulzrinne, H.: A Survey of Security Issues and Solutions in Presence. In: http://www1.cs.columbia.edu/~vs2140/presence/presencesecurity.pdf (2006)

UDDI Version 2.04 API Specification, UDDI Committee Specification, 19 July 2002 (2002)

Urs Hengartner and Peter Steenkiste: Implementing Access Control to People Location Information, SACMAT'04, page 11-20, June 2004.

Web Services Security: SOAP Message Security 1.1, (WS-Security 2004), OASIS Standard Specification, 1 February 2006 (2006)

Web Services Eventing (WS-Eventing), W3C Member Submission, 15 March 2006 (2006)

Zhang, Y. and Joshi, J.B.D.: UAQ: A Framework for User Authorization Query Processing in RBAC extended with Hybrid Hierarchy and Constraints. In: SACMAT'08, June, 2008, pages 83-91 (2008)