Rule Adaptation in Collaborative Working Environments using RBAC Model

Ahmad Kamran Malik
Department of Computer Science
COMSATS Institute of Information Technology
Islamabad, Pakistan

Muhammad Anwar
Department of Computer Science
Federal Urdu University of Arts, Science &
Technology (FUUAST)
Islamabad, Pakistan

Abdul Mateen
Department of Computer Science
Federal Urdu University of Arts, Science &
Technology (FUUAST)
Islamabad, Pakistan

Wajeeha Naeem
Department of Computer Science
COMSATS Institute of Information Technology
Islamabad, Pakistan

Yousra Asim
Department of Computer Science
COMSATS Institute of Information Technology
Islamabad, Pakistan

Malik Ahsan Ali
Department of Computer Science
Federal Urdu University of Arts, Science &
Technology (FUUAST)
Islamabad, Pakistan

Basit Raza
Department of Computer Science
COMSATS Institute of Information Technology
Islamabad, Pakistan

***Abstract*—Collaborative Working Environments (CWEs) are getting prominence these days. With the increase in the use of collaboration tools and technologies, a lot of sharing and privacy issues have also emerged. Due to its dynamic nature, a CWE needs to adapt the changes into accordingly. In this paper, we have implemented the Adaptive Dynamic Sharing and Privacy-aware Role Based Access Control (Adaptive DySP-RBAC) model which provides user's information privacy to dynamically adapt the changes occurring in the system at any time. The proposed model has been implemented as a prototype and tested. Results have shown that our system efficiently and effectively adapts access rules according to the changes happening in a CWE along with preserving the user's information privacy in the system.**

Keywords*—*Dynamic Adaptation; RBAC; Privacy; Collaboration

I. Introduction

Today, mutual teamwork-based working environments and cooperation among the members have gained prominence. Now, people without working together at the same place can also collaborate with each other. This type of working environments is known as Collaborative Working Environment (CWE). With the help of CWE, people can share their ideas, efforts, results, inventions etc. and at the same time, they can change their locations as well. A lot of work has been done in order to improve CWEs. In [1], seven different collaboration factors related to collaboration have been discussed. Along with that, a collaborative working model has been presented by summarizing all those factors. With the increase in the use of CWEs, the need for privacy preservation and access control has also increased.

Dynamic Adaptation is yet another important characteristic of a CWE. In most of the collaborating environments, participating users may join or leave the group at any time. This causes the nature of groups to be dynamic. Due to which, rules and policies relating to the collaborating groups need to change as well. In [2], the concept of dynamic adaptation has been presented in the form of virtual teams. It advocates that virtual teams contain members who can geographically be located anywhere in the world. They can not only be linked but also participate equally with the help of emerging telecommunication technologies. In [3], an intelligent system for dynamic collaborations has been presented. In this system, changes frequently occur due to change in the users participating in the collaborative environment, which a system adapts effectively.

We aim to implement a CWE in which system adapt the related variations as and when any change in the system occurs. We have implemented the Adaptive Dynamic Sharing and Privacy-Aware Role-Based Access Control (Adaptive DySP-RBAC) model by extending the DySP-RBAC model [4] to incorporate dynamic adaptation of access control rules. We have monitored dynamic adaptation in different scenarios and recorded results to show the level of adaptation in the respective system. Since information sharing is inevitable in a CWE, that's why it has been considered a crucial step to provide privacy preservation of information. In [4], privacy and access control has been provided by creating the different type of policies. These policies have also been evaluated and compared with other access control models in [5].

Since CWEs usually have dynamic nature, user's personal and shared data or resources may also change with the change of user's participation in the collaborating environment. Context-awareness is yet another important feature of a CWE, so immense work has been done towards the improvement of context-awareness especially in collaborative environments. Such as in [6], context-aware computing has been defined along with its different categories. Cases of these categories have been prototyped and evaluated for results. Our system monitors user's information and performs adaptation more effectively according to the change in user's personal and shared resources information.

Rest of the paper has been organized as follows. Related work has been presented in Section II. Section III comprises of the architecture details. Results have been discussed in section IV. Section V concludes the paper and also describes future work.

II. Related Work

A lot of work has been done in order to improve CWEs with respect to performance aspects as well as security. A framework named as "Distribute Cognition" has been presented in [7] to explain and analyze collaborative working. In this paper, some theoretical and practical issues regarding collaboration have been discussed. In [8], interoperability issues in CWEs have been focused. For this purpose, a generic CWE has been proposed which allows different types of groupware to collaborate easily and effectively using different Web services technologies. Similarly, in order to deal with new emerging challenges in CWEs, an approach named inContext has been proposed in [9]. This approach has been used to combine some collaboration services which are considered dissimilar with the help of web services. It also handles the CWEs that are considered of dynamic nature. In [10], privacy issues faced due to collaboration has been discussed and a privacy framework has also been given to improve privacy issues. This framework can be adapted for any type of domain since it contains generic privacy ontology. It contains privacy rules also, through which information access has been defined; this part has been named as reasoning engine. In [5], different collaborating environments such as RBAC, Team-based Access Control model (TMAC) and Extended RBAC model has been implemented and evaluated for sharing and privacy preserving rules and metrics. In [11], different challenges related to collaborative environments have been discussed. In the light of those issues, related solutions have also been proposed.

Dynamic adaptation in CWE has immensely been focused such as in [12], a REal-time Software Adaptation System (RESAS) has been presented. In this framework, a tool has been provided to programmers in order to adapt the changes in real time. Similarly, in [13], a policy-driven and context-aware dynamic adaptation framework named Chisel has been proposed. In this proposed system, with the change of user and application context, behavior of different service objects automatically adapted by the system. With respect to different context, a number of policies have also been associated. In [14], a model has been presented in which a user has been provided related policies whenever a change has occurred in the system. The proposed access control system has a feedback component which has been named as "know". Policy protection and level of feedback have been provided by this feedback component. Rules or policies have been efficiently implemented through Ordered Binary Decision Diagrams (OBDDs). In [15], an interactive access control model has been proposed to further improve autonomous computing systems. The idea is based on the interaction between the clients and servers in order to provide access to any resource. On the basis of credentials provided by the clients, servers grant or deny access upon evaluating predefined policies.

In [16], a context-aware access control mechanism has been proposed for ubiquitous applications, for this purpose standard RBAC model has been extended. In this mechanism, the system dynamically adapts changes and grants permissions accordingly as and when a change occurs in the context. Another access control policy model has been given in [17], which has focused context awareness for the sake of resource access and dynamic adaptation for accommodating changes caused in context. Along with that, semantic technologies have been used to specify context/policies in the system. In [3], an intelligent information sharing control system has been presented in which sharing and control policies have been dynamically adapted as and when a change occur in user context, relationships, activities, and interactions. In [18], definitions of context-awareness in Internet of Things (IoT) and Internet of Everything (IoE) along with their architecture have been presented. Similarly, current context-aware approaches in systems such as IoT and IoE have also been analyzed.

In next section, details of proposed model along with some scenarios have been explained.

III. Architecture

As mentioned earlier, we have implemented an extension of DySP-RBAC model and evaluated it how the model dynamically adapts the changes occurred in the system. Details of the model have been given below:

A. DySP-RBAC Model

DySP-RBAC is an extension of core RBAC model. Along with user roles, it focuses on teams and tasks as well, including other data elements such as user, session, and permissions called sharing and privacy aware permissions. This is because permissions have also been created with the help of sharing and privacy elements. Sharing elements are used to enhance sharing among collaborating users. Sharing elements include Collaborative Relationships (CR) and Access Level (AL). Privacy elements are used to preserve the privacy of user's personal and sharing resources. Privacy elements include Purpose (Pur), Condition (Con) and Obligation (Obl). In core RBAC permissions were based on only objects and operations. In Fig. 1, DySP-RBAC model is shown. Any person who is participating in the system is termed as a user while that user may be a member of one or more teams. A user can be assigned multiple tasks in each team he is participating in. Similarly, a user can have multiple roles according to which he will be assigned appropriate team and task. Objects contain user related information such as his teams, tasks, as well as personal information.

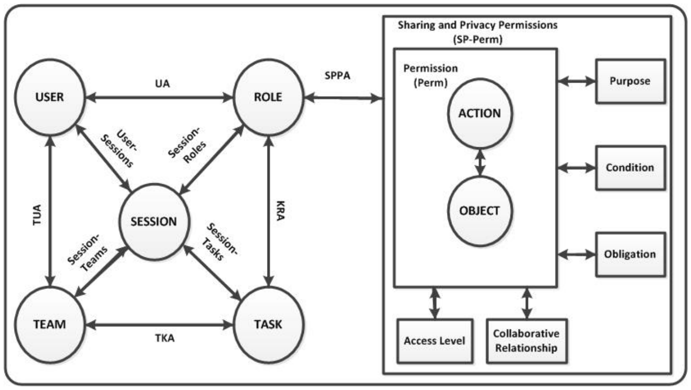

Fig. 1. DySP-RBAC Model [4]

B. Sharing and privacy

In order to provide access control, sharing and privacy based permissions have been used. It means that whenever a user wants to access other user's resource, the request will be evaluated according to the permissions created by its owner. Similarly, a user can create conditions to allow or restrict access to his resource. The level of information sharing has been determined through CR and AL. There are three types of collaborative relationships such as Mutual, Member, and Colleague. If some users are participating in the same team and they have been assigned the same task as well, their CR will be considered as Mutual, while users who are team members but they do not share the same task termed as Member, likewise users who neither have been on the same team nor assigned same tasks are considered as colleagues. Theses CR levels determine the level of information sharing at the different type of collaborative relationships. For example, users having CR as ”Mutual” will have more resource access as compared to ”Colleague”. Similarly, for users having different collaborative relationships or roles, three type of access levels have been used i.e. Level 1 (L1), Level 2 (L2) and Level 3 (L3). So, users assigned AL as L1 will have higher access as compared L3, which will have lowest access level.

Whenever a user wants to create permissions, he will select one or more elements, for example, a CR, an AL, a Team, a Task, a Role and a resource for which he will create permissions. For example, if a user has created a permission having Mutual (as CR), L2 (as AL), Team A (as Team), T1 (as Task), R1 (as Role) and Location (as Resource). This means when any other user will request to access this users ”Location”, it will be checked that requesting user must be his Mutual (in CR), he must have an L2 (in AL), he must be a member of the specified team (i.e. Team A in this case), he must be assigned the specified task (i.e. T1 in this case), and he must have a particular role (such as R1 in this case). If the requesting user fulfills any of the mentioned conditions he will be allowed access to the requesting resource.

C. Dynamic Adaptation

Our scenario is an enterprise-based system in which different users perform their assigned tasks in the form of teams. Each user will have assigned tasks on the basis of roles that they have been allocated. Users, roles, teams, and tasks have dynamic nature in the system. This means, when a user leaves a team, all his related information and permissions should be dynamically adapted according to the new situation. Similarly, when a task is finished within a team, this will cause the task related information to change. Our system is capable of accommodating all changes taking place as a result of dynamic adaptation. The whole adaptation process has been shown in Fig. 2. Four main steps are Monitor, Analyze, Plan and Execute. Our system continuously monitors the changes on the basis of its knowledge. System knowledge includes collaborative relationships among users, their access levels, permissions/policies and system entities such as teams, tasks, and roles etc. It then analyzes them so that it can be determined that how the changes, which have already being monitored, can be adapted to the new situation. The system plans accordingly and executes the adaptation of changes occurred.

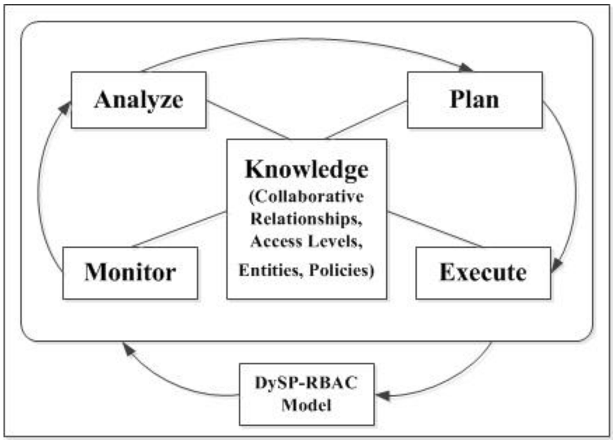

Fig. 2. Adaptive DySP-RBAC Model

1) Scenario: In this part, we have presented a scenario so that dynamic adaption in our system can be explained and understood more effectively. We have taken an enterprise-based CWE in which people collaborate in different teams and perform different tasks. These teams can be overlapping in nature because users can participate in more than one tasks at a time. Similarly, users can join or leave any team or task any time, making the whole scenario of dynamic nature. This can also happen due to the finishing/completion of a task within a team or any team/task can also be revoked from a user at any time. In order to perform a task, users may share and request to share each others resource information. It may be related to a user's personal information i.e. location etc. or team task related information. There are two types of sharing control policies which have been used in our proposed model; User-defined policy and Enterprise-defined policy. Users create permissions/policies to allow or restrict access to their personal or shared data. This is called user-defined policy. In order to control sharing of team related or task related information, enterprise-defined policies will be used [3].

In Fig. 3, we have presented an example of proposed

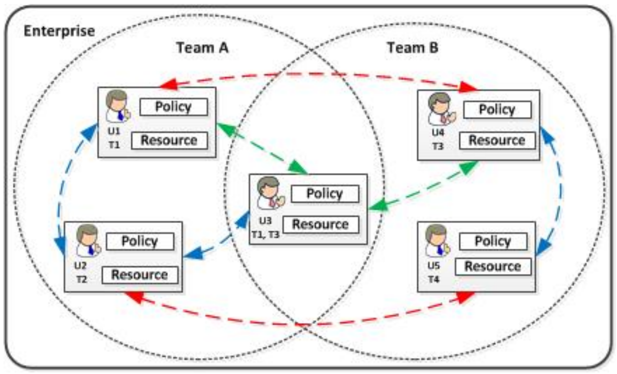

Fig. 3. Dynamic adaptation in a CWE (before) [3]

scenario. We have an enterprise in which users interact with each other in the form of teams and work on different tasks, which may be mutual or individual as well. We have two teams ”Team A” and ”Team B”. Tasks include T1, T2, T3, and T4, while five users are named as U1, U2, U3, U4, and U5. There are policies or permissions associated with each user. The users U1 and U2 are part of the same team, which is team A, but they do not share same tasks. Whereas users U4 and U5 are the member of the same team, which is team B, but they do not share tasks. The user U3 is participating in both teams (A and B) at the same time, since U3 has been assigned tasks common in both teams. Users U1 and U3 have been working on a mutual task T1 while users U3 and U4 have been assigned another same task which is T3. Users U1 and U2 are part of the same team but they do not share any task of the team. This also shows the level of a collaborative relationship among users i.e. Mutual, Member, and Colleague. Several colors have been used in Fig.3 to represent different collaborative relationships. Such as, green color represents a mutual relationship, blue color represents a member relationship and red color represents a colleague relationship.

The said collaborative relationships also determine the level of sharing information among users. As mentioned earlier, the highest level of collaboration is ”Mutual” which is being a member of the same team and being assigned the same task. In Fig. 3, we can see that users U1, U3, and U4 have been connected with green arrows because they have a mutual relationship. Medium level of collaboration is ”Member” in which users may be a part of the same team but do not share the same task. It can also be seen in the figure that users U1 and U2 have a relationship as a member as well as users U2, U3, and U4, U5 have been assigned member relationship. Moreover, the lowest level of collaboration is ”Colleague” which is neither being a member of the same team nor having assigned the same task. According to Fig. 3, the user U1 of team A and the user U4 of team B have colleague relationship, similar relationship exists between users U2 and U5.

Since we have a dynamic CWE, changes can occur anytime. According to the Fig. 3, Team A and B were having a mutual task T1. When the task T1 is finished, the collaborative relationships of participating users are also changed. This has been shown in Fig. 4. Now, users U1 and U3 do not have mutual relationship so the green arrow joining them before has been removed. But, users U3 and U4 still share the same task (T3) and are members of the same team (B) they still have a mutual relationship that is why they are still connected with a green arrow. Likewise, users U2 and U3 were sharing member relationship in Fig. 3, when the task T1 is finished user U3 is not a member of the ”team A” anymore so users U2 and U3 are not sharing member relationship anymore. This is how dynamic adaptation takes place in our system. As and when a change occurs in CWE, related policies are changed accordingly by the system. The scenario explained above has also been tested on a prototype model of the system. The results have been explained in next section.

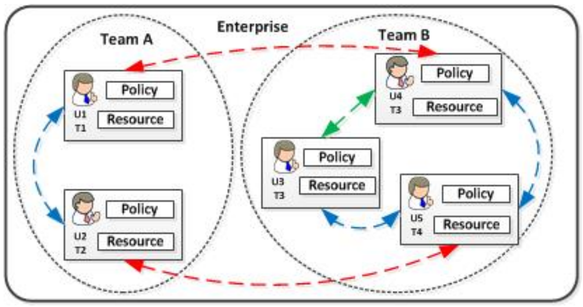

Fig. 4. Dynamic adaptatin in a CWE (After) [3]

IV. Results and Discussion

In order to test the proposed model, we have taken some empirical data set in which a different number of teams, tasks, users, roles have been taken to evaluate different scenarios such as finish team, finish task, task revocation, team revocation. There are 16 users and 20 roles, each user has been assigned one or many roles at a time and on the basis of his roles, he has assigned related team or task. There are 8 teams and 9 tasks. There are 4500 access control permissions which are used in sets of 1500, 3000, and 4500 permissions to test the system with increasing number of permissions. Each aforementioned scenario has been evaluated through these numbers of permissions separately. Details of said scenarios are provided as follows.

Finish Team:

In the proposed scenario, there are a number of teams in which different users are participating to accomplish different tasks. So, whenever any team is finished, its related permissions and information are also removed using the dynamic adaptation system i.e. whenever a team finish will occur the system will automatically accommodate related changes. For 1500 permissions, upon finishing a team, there are 426 permissions that have been changed. For 3000 permissions, in a finish team scenario, there are 576 permissions that have been changed or removed. Similarly, in the case of 4500 permissions, there are 720 permissions which have been changed. This has also been shown in the given Fig. 5. We have compared the number of permissions affected when a team has been finished in three different number of permissions sets. We can see that as the number of permissions increases the system adapts the change accordingly.

Finish Task:

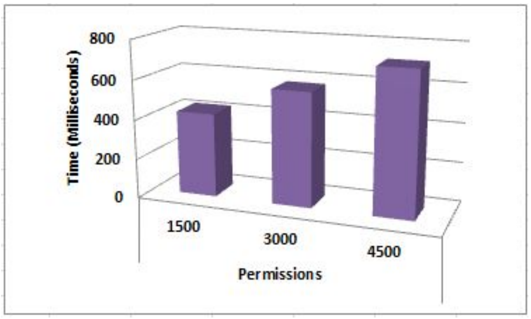

Fig. 5. Adaptation in Finish Team

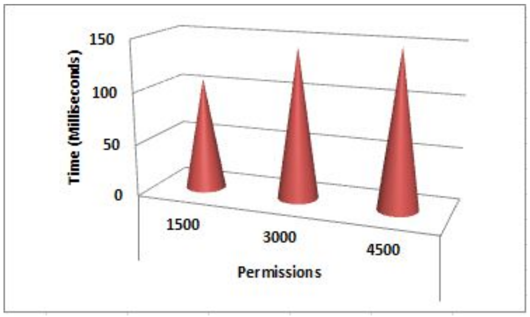

Fig. 6. Adaptation in Finish Task

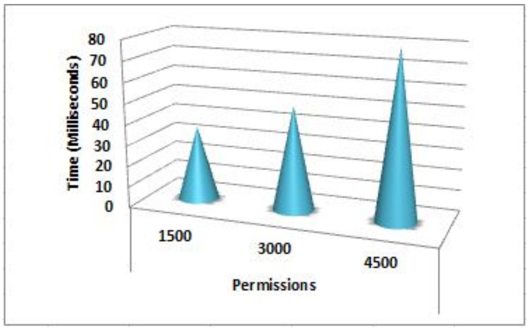

Fig. 7. Adaptation in Revoke Task

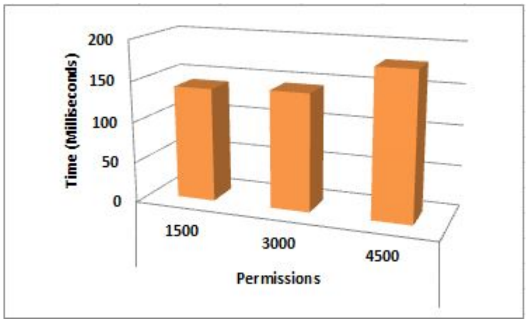

Fig. 8. Adaptation in Revoke Team

In this scenario, the case of finishing a task is discussed. In other words, whenever any task assigned to any team will be completed/finished, its related access permissions will also be removed. A single task may be assigned two multiple users but when the task has been completed all associated changes will be accommodated for each and every concerned user. This is how adaptation takes place in this scenario. In Fig. 6, we have shown how adaptation has taken place with different number of permissions by outlining change in the number of permissions due to task finish. We can see in the Fig. 6, for 1500 permissions, the number of permissions that have been affected are 108, for 3000 permissions the number is 144 and for 4500 permissions it reaches to 150.

Revoke Task:

Task revocation means a task which has previously been assigned to a user is withdrawn from that user. There are some cases in which some users might not be able to produce expected results in a task. In this situation, tasks can be revoked from users. Our system is capable of acclimatizing changes associated with that task. In Fig. 7, we have shown how many numbers of permissions have been affected due to revocation of user assigned task. It shows that, whenever a user is revoked his assigned task, his related permissions are also removed from the system, regardless of what the number of permissions is. Such as for 1500 permissions, there are total 36 permissions that have been affected. For 3000 permissions, there are 50 permissions that have been changed or removed. Similarly, for 4500 permissions, we have 80 permissions that have been removed.

Revoke Team:

Another scenario is revocation of a team from a user. This means, due to any circumstances a user can be considered incapable of being a part of a team. So the user will be removed from corresponding team. This scenario is different from finish task because in that one, a user will be finishing task assigned to him. While in this scenario, it's not the case, a user may not be able to finish his task and he may be released. In this situation the permission which he has already created will also be removed by the system automatically. The Fig. 8 shows the numbers of permissions changed due to revocation of a user assigned team. This is shown for three different sets of permissions. For 1500 permissions, there are 140 permissions that are changed. For 3000 and 4500 permissions there are 144 and 180 permissions that are changed, respectively.

Running Time Comparison:

We have also compared run time taken by our system for all said scenarios. Fig. 9 shows the comparison graph displaying time taken by each scenario.

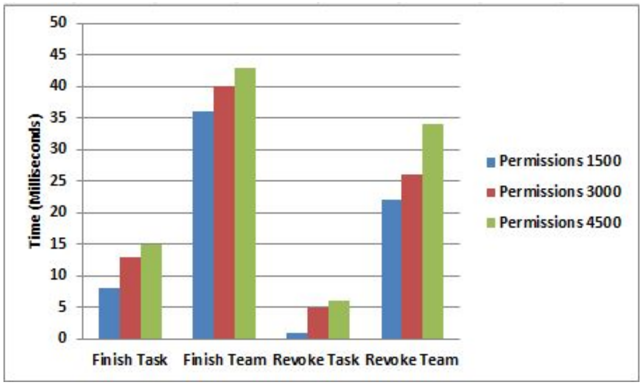

Fig. 9. Comparison of running time for different scenarios

In case of 1500 permissions, time taken by Finish Task is 8 milliseconds, while for Finish Team, Revoke Task and Revoke Team it is 36, 1 and 22 milliseconds respectively. In case of 300 permissions Finish Team has taken maximum time which is 40 milliseconds. While, time taken in Finish Task, Revoke Task and Revoke Team is 13, 5 and 26 milliseconds respectively. In case of 4500 permissions, maximum time has been taken by Finish Task which is 43 milliseconds. However, Finish Task, Revoke Task and Remove Team has been taken time as 15, 6 and 34 milliseconds respectively. We can see that ”Finish team” has taken the maximum time among others while ”Revoke task from user” has taken the less time among all. This is because a team may contain a good number of users and each user will create different number of permissions, so when that team will be finished all related permissions will be removed and that will be a large number. Similarly, a user will individually be assigned any task and he will create his task related permissions accordingly. So for that task, only those permissions will be removed which are related to that particular user only.

With the help of all results stated above, we have presented how the proposed system efficiently and effectively implements dynamic adaptation in a CWE so that users as well as administrators do not have to worry about managing their large number of access permissions manually.

V. Conclusion

We have implemented an Adaptive DySP-RBAC model in which sharing and privacy of user information is presented using two types of sharing control policies. The Enterprise-defined policy is used to prevent user’s shared information (such as their teams, tasks, roles etc.) from being revealed to unauthorized users. Similarly, the User-defined policy is used to prevent user’s personal information (i.e. his location, personal details). Users create permissions to control access to their resources. A large number of such access permissions are hard to manage. For this purpose, our system executes in a dynamic working environment in which rules are dynamically adapted at runtime as and when a relevant change is detected. Our system successfully adapts the changes caused by such cases, for example, when a team or task is finished, a user may be revoked a team or task which has been assigned him previously. A prototype of this model is implemented and it has been evaluated for dynamic adaptation of access control rules. In future, this work is to be extended for more than one enterprise, since we have focused on within a single enterprise scenario.

Acknowledgment

This research is funded by Higher Education Commission (HEC), Pakistan, through the project ”Information Privacy in Collaborative Environments” approved for CIIT, Islamabad.

References

[1] H. Patel, M. Pettitt, and J. R. Wilson, ”Factors of collaborative working: A framework for a collaboration model,” Appl. Ergon., vol. 43, no. 1, pp. 1-26, 2012.

[2] A. M. Townsend, S. M. DeMarie, and A. R. Hendrickson, ”Virtual teams: Technology and the workplace of the future.,” Acad. Manag. Perspect., vol. 12, no. 3, pp. 17-29, Aug. 1998.

[3] A. K. Malik and S. Dustdar, ”An intelligent information sharing control system for dynamic collaborations,” Proc. 8th Int. Conf. Front. Inf. Technol., p. 30:1-30:6, 2010.

[4] A. K. Malik and S. Dustdar, ”Enhanced sharing and privacy in distributed information sharing environments,” Proc. 2011 7th Int. Conf. Inf. Assur. Secur. IAS 2011, pp. 286-291, 2011.

[5] W. Naeem, M. A. Shah, and A. K. Malik, ”Privacy-Preserving in Collaborative Working Environments,” in IOARP, 2015, pp. 18-19, March 2016.

[6] B. Schilit, N. Adams, and R. Want, ”Context-Aware Computing Applications,” in 1994 First Workshop on Mobile Computing Systems and Applications, pp. 85-90, 1994.

[7] Y. Rogers and J. Ellis, ”Distributed cognition: an alternative framework for analysing and explaining collaborative working,” J. Inf. Technol., vol. 9, no. 2, pp. 119-128, June 1994.

[8] M. A. Martinez-Carreras, A. Ruiz-Martinez, F. Gomez-Skarmeta, and W. Prinz, ”Designing a Generic Collaborative Working Environment,” in IEEE International Conference on Web Services (ICWS 2007), pp. 1080-1087, 2007.

[9] H.-L. Truong, S. Dustdar, D. Baggio, S. Corlosquet, C. Dorn, G. Giuliani, R. Gombotz, Y. Hong, P. Kendal, C. Melchiorre, S. Moretzky, S. Peray, A. Polleres, S. Reiff-Marganiec, D. Schall, S. Stringa, M. Tilly, and H. Yu, ”inContext: A Pervasive and Collaborative Working Environment for Emerging Team Forms,” in 2008 Intl. Symp. Applications and the Internet, pp. 118-125, 2008.

[10] D. S. Allison, A. Kamoun, M. A. M. Capretz, S. Tazi, K. Drira, and H. F. ElYamany, ”An ontology driven privacy framework for collaborative working environments,” Int. J. Auton. Adapt. Commun. Syst., vol. 9, no. 3/4, p. 243, 2016.

[11] A. Mohamad, N. Yusoff, S. Aris, D. Bismo, and M. Regan, ”Dare to Change: An Approach to Implement Enterprise Level of Real Time Well Solution for Collaborative Working Environment,” in IADC/SPE Asia Pacific Drilling Technology Conference, 2016.

[12] T. E. Bihari and K. Schwan, Dynamic adaptation of real-time software, ACM Trans. Comput. Syst., vol. 9, no. 2, pp. 143-174, May 1991.

[13] J. Keeney and V. Cahill, ”Chisel: a policy-driven, context-aware, dynamic adaptation framework,” in Proc. POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks, pp. 3-14, 2003.

[14] A. Kapadia, G. Sampemane, and R. H. Campbell, ”Know why your access was denied: Regulating feedback for usable security,” Proc. ACM Conf. Computer and Communications Security. pp. 52-61, 2004.

[15] H. Koshutanski and F. Massacci, ”Interactive access control for web services,” IFIP Intl. Inf. Secur. Conf. , vol. 3, no. 3, pp. 1-16, 2004.

[16] Y.-G. Kim, C.-J. Mon, D. Jeong, J.-O. Lee, C.-Y. Song, and D.-K. Baik, ”Context-Aware Access Control Mechanism for Ubiquitous Applications,” pp. 236-242, 2005.

[17] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila, ”A semantic context-aware access control framework for secure collaborations in pervasive computing environments,” ISWC’06 Proc. 5th Int. Conf. Semant. Web, p. 14, 2006.

[18] E. de Matos, L. A. Amaral, and F. Hessel, ”Context-Aware Systems: Technologies and Challenges in Internet of Everything Environments,” Springer Intl. Pub., pp. 1-25, 2017.