International Journal of Advanced Science and Technology
Vol.114 (2018), pp.59-68
http://dx.doi.org/10.14257/ijast.2018.114.06

Management of User Groups and Member Authorization with Various Access Rights on the Ethereum Blockchain

Jae-Hwan Jin[1], Wookun Lee[2] and Myung-Joon Lee[3*]

[1,2,3*]*Department of Electrical/Electronic and Computer Engineering, University of Ulsan, 93, Daehak-ro, Nam-gu, Ulsan 44610, Republic of Korea*
[1]*jjhok2000@gmail.com*, [2]*justinwk11@gmail.com*,
[3*]*mjlee@ulsan.ac.kr*

Abstract

Distributed applications based on the Ethereum blockchain have a great advantage in the analysis of the history of group work and the contribution factor to a group project of each group member. In these applications, multiple users share resources and their histories together in a robust way thanks to the characteristics of the blockchain that is practically impossible to be forged or modified. However, there has not yet been developed a scheme for providing a systematic management scheme of a group of users or authentication of the members of the group to allow the members to use shared resources through various access rights.

In this paper, we propose a user group and authorization management scheme over the Ethereum Blockchain. In the proposed scheme, group users can use shared resources by using various access rights. To do this, we define user groups as different types according to their access rights, and present the process of authenticating user groups according to their types. The proposed scheme is designed using smart contracts, being useful in various Ethereum distributed applications requiring group workspaces.

Keywords: *Bluetooth Beacon, Beacon Service, Button Beacon, Beacon Pattern, Beacon Data*

1. Introduction

Ethereum [1] is a decentralized application (DApp) [2, 3] platform based on the blockchain network [4]. Ethereum developers can easily develop decentralized applications that interoperate with the Ethereum blockchain network through the Ethereum platform. Since various kinds of resources generated through decentralized applications are stored in a distributed network based on blockchain technology, there is an advantage that it is impossible to forge or falsify those resources [5]. Based on the stability of blockchain, lots of researches on decentralized applications in various fields are actively being carried out. In particular, the blockchain structure has a great advantage in analyzing the history of group work in which multiple users share resources together and analyzing the contribution factors of group members. This allows users who perform group work to credibly concentrate on their work so that they can participate more enthusiastically without any concern to unfair profit sharing.

Ethereum-based group workspaces allow group members to safely work on the shared resources among group users through user group management because their work histories for shared resources are stored without falsification. User group management refers to systematic group authentication and resource access control to identify the identity and authority of users accessing shared resources. In this way, the members of the

Received (October 19, 2017), Review Result (January 20, 2018), Accepted (January 25, 2018)
* Corresponding Author

ISSN: 2005-4238 IJAST

group are not able to use the shared resources inappropriately because they perform tasks in a batch or partial manner by being assigned the access rights such as reading or writing tasks to the resources. Since the need and value for user group management in Ethereum decentralized applications are high, it is desirable to develop related techniques. As of now, although a scheme for providing a blockchain-based personal user authentication service such as UPort [6] and R2ID [7] have appeared recently, a method for managing user groups for shared resources in association with various access rights has not been developed.

In this paper, we propose a scheme to systematically manage user groups and access privileges over Ethereum Blockchain. In the proposed scheme, the user groups are defined as various types according to the purpose, and the access rights of the group members are assigned differently according to the type. To this end, we present the process of forming user groups in the Ethereum blockchain and how to manage the group information for the created groups. Also, we propose a group user authentication method that systematically manages the credential information for each group user and confirms the information about the group to which the user belongs and the credential information. The proposed technique is designed as the smart contracts for use in Ethereum decentralized applications. Ethereum Blockchain participants can use the smart contracts to create groups and perform group user authentication. Also, we present the process of authenticating user groups according to their types. The proposed scheme can be used for privilege management of shared resources even in group workspaces on a database or on off-blockchain storage.

The rest of this paper is organized as follows. We summarize background knowledge on the Ethereum Decentralized Application and the Group Workspace in Section 2. In Section 3, a method of Ethereum-based user group management is presented. In the next section, a process of group authentication and access control is described. Finally, the authors conclude the paper in Section 5.

2. Background

2.1. Ethereum Decentralized Application

Ethereum decentralized applications are created using the Ethereum platform and functions by interworking with the blockchain network. In addition, Ethereum decentralized applications generate transactions for storing resources and executing functions through smart contracts distributed in the Ethereum blockchain network. Smart contracts have member variables and methods as programmable contracts to perform specified functions for specific conditions. Ethereum decentralized applications are generally based on the web, working with the Ethereum distributed network through the Web3.js module of JavaScript [8]. Based on the use of the blockchain network, the Ethereum Decentralized Application has the advantage of being highly secure about the storage and functioning of resources and maintaining the work history. So, many distributed applications are coming up in various application domains such as game, virtual currency, auction, SNS, electronic voting as shown on the Ethereum official website [9].

2.2. Group Workspace

Members of a user group can effectively share resources with each other using the associated group workspace. Group workspaces provide management of group members' access rights, as well as basic functions such as uploading and downloading of files and documents. Access rights management is a function for assigning rights to members of a group workspace to perform tasks on shared resources. In general, access rights management can specify access rights based on the read and write permissions on shared

resources, and the scope can be adjusted to whole or partial resources. These access rights are specified by the owner of the group, and each group member performs operations based on the information specified. In recent years, research has been conducted to provide collaborative services where group members work together through handling shared resources on group workspaces [10, 11].

3. Ethereum-based User Group Management

A user group is composed of a group of users gathered for a specific purpose, and assigns roles and privileges to each member to systematically perform a group operation. In this section, we propose a method to create a user group that can be used in Ethereum decentralized applications for managing user group information in the blockchain network.

3.1. Ethereum-based User Group

The proposed Ethereum-based user group stores the information needed for user group management via Ethereum smart contracts. One user group is created as one UserGroup contract and the generated UserGroup contracts are managed through the GroupStore contract. Figure 1 shows the structure for storing user group information over Ethereum.

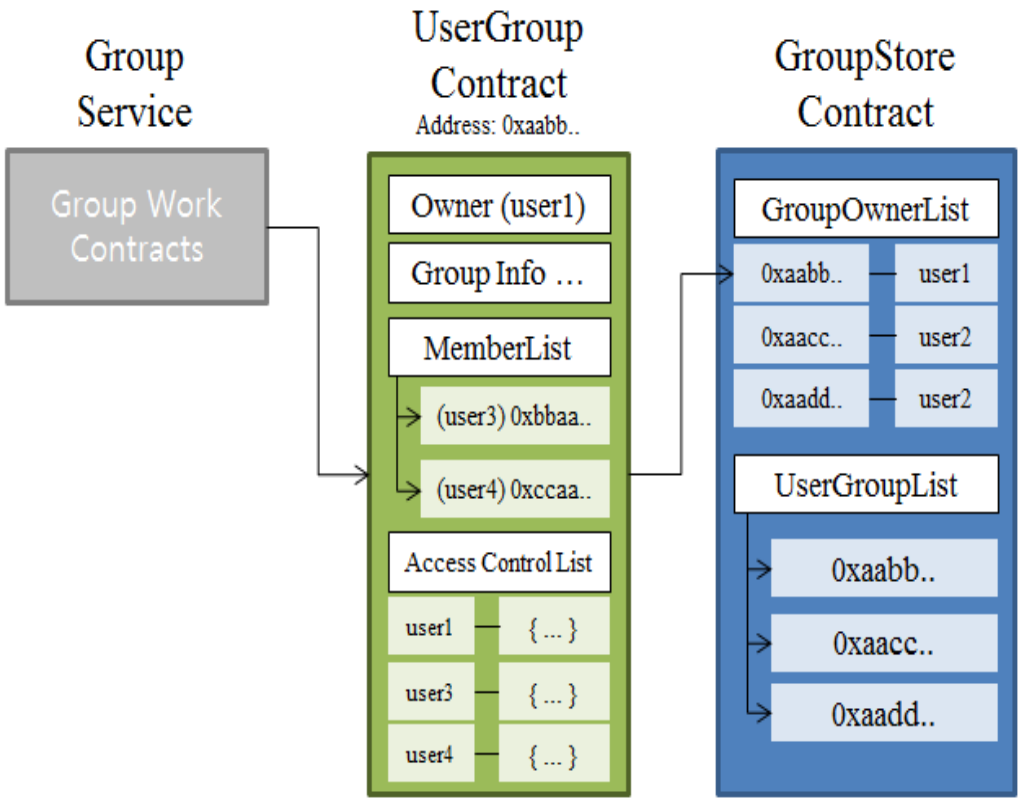

Figure 1. Ethereum Smart Contracts for User Group

The GroupStore contract stores the address list of the UserGroup contracts for all groups to access the generated UserGroup contracts. In addition, the GroupStore contract manages a hash table with the address of the UserGroup contract as key and the group owner's account address as value, so that the owner of each group can be identified. The UserGroup contract manages detailed information about the group as well as meta information about the resources shared by the group and information about the access rights to the resources of members. Table 1 shows the main information of groups stored in GroupStore and UserGroup.

Table 1. Data Fields for User Group Contracts

Smart Contract	Field Name	Data Type	Description
GroupStore Contract	UserGroupList	Array	List containing UserGroup contract addresses
	GroupOwnerList	HashTable	Hashtable with the address of the UserGroup contract and the account address of the group owner
UserGroup Contract	Owner	Address	Owner's account address of the group
	MemberList	Array	List storing account addresses of group members
	ACL	HashTable	A hash table having account addresses of group members and members' rights to shared resources

3.2. Type of User Groups

Ethereum users can create new groups through the GroupStore contract. The user who requests the creation of the user group becomes the owner, and the GroupStore creates a new UserGroup contract for the request, adding it to the group list. A user group is classified into four types of groups according to the method of assigning access rights of group members. Table 2 shows the characteristics of 4 types.

Table 2. Type of Ethereum-based User Group

Group Type	Description
Public Group	All group members have full access to all shared resources
Anonymous Group	All group members have the same permissions set by the group owner
Owner-defined Group	The group owner can set different permissions for group members
User-defined Group	Group members can add their own temporary accounts to distribute their privileges

The type of user group can be selected when the group owner creates the group and cannot be changed after it is created. A user-defined group uses a privilege distribution scheme to allow users who are not registered in a user group to easily access the group's shared resources with a simple authorization specification. The members of the user-defined group are assigned the initial authority to the resource from the owner of the group in the same way as the owner-defined group. The group member then specifies the rights for each unregistered user to distribute his/her authority. After the group members have distributed their privileges, an Ethereum temporary account is created for each

distributed privilege and the privilege is assigned to the account. Figure 2 shows the operation of the proposed access rights distribution function.

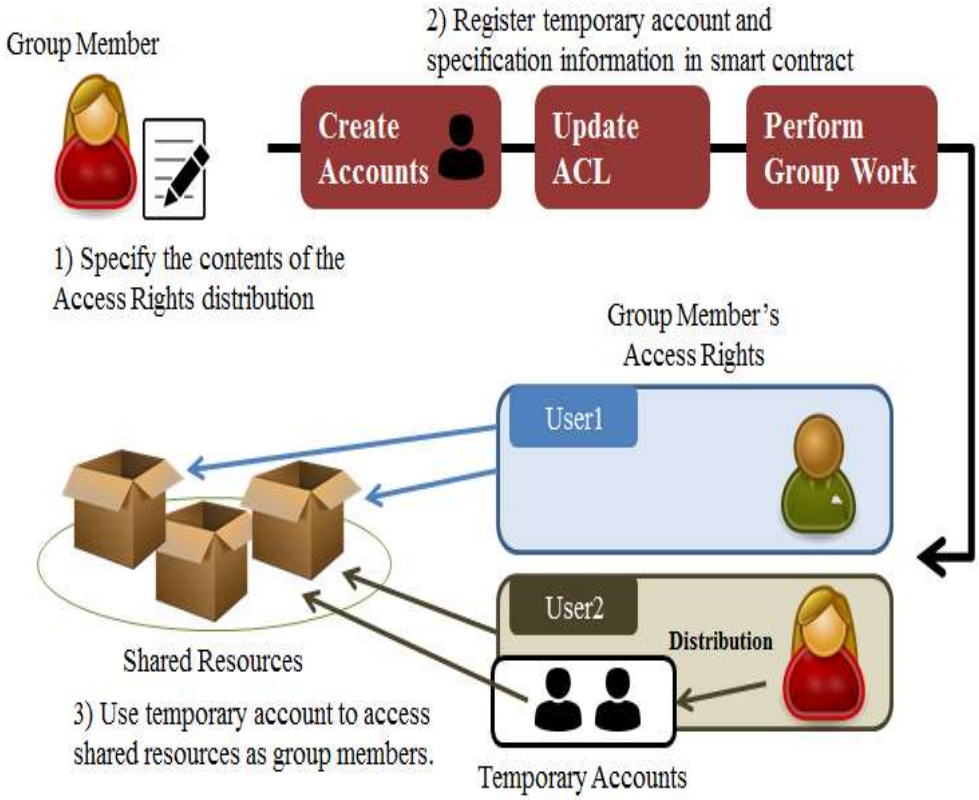

Figure 2. Process of Access Rights Distribution Function

The registered temporary account is authenticated by confirming the contents of the authority distribution stored in order to perform the group operation. In this way, users can easily perform group work by using the access authority without registering them in the user group separately. The UserGroup contract generated through the user group creation request manages a list storing the account addresses of the group members. Also, the UserGroup contract stores the owner information of the group, and if the owner of the group is changed, the group owner information stored in the GroupStore is also changed to maintain consistency.

4. Group Authentication and Access Control

After the user group is formed, the group members can be authenticated as a group member through the created smart contract. Also, the authenticated group member can perform group work based on the assigned authority. This section describes the group authentication method using the Ethereum-based user group and the method of managing the access rights to the shared resources using the smart contract.

4.1. User Group Authentication

Group members access shared resources or services through other decentralized applications that require user groups. At this time, group members must obtain group certification through the GroupStore contract and the UserGroup contract. First, the user accesses the UserGroup contract previously deployed in the Ethereum blockchain network to receive group authentication for the user group. The address of the UserGroup contract is provided when group members join the group. A user can be authenticated as a

group member via the *verifyGroupMember* method in the UserGroup contract as shown in Figure 3. The *verifyGroupMember* method verifies that the user's address is in the list of group member addresses stored in the UserGroup contract. The temporary accounts of the user-defined group are checked from the list of temporary account addresses that only the UserGroup contract of the user-defined group has. The *verifyGroupMember* method then returns whether the requesting user is a member of the group. This authentication method is also used as a modifier for conditional checking and execution guard in all methods associated with group work. These methods are executable only when they are group members, so they go through this authentication method.

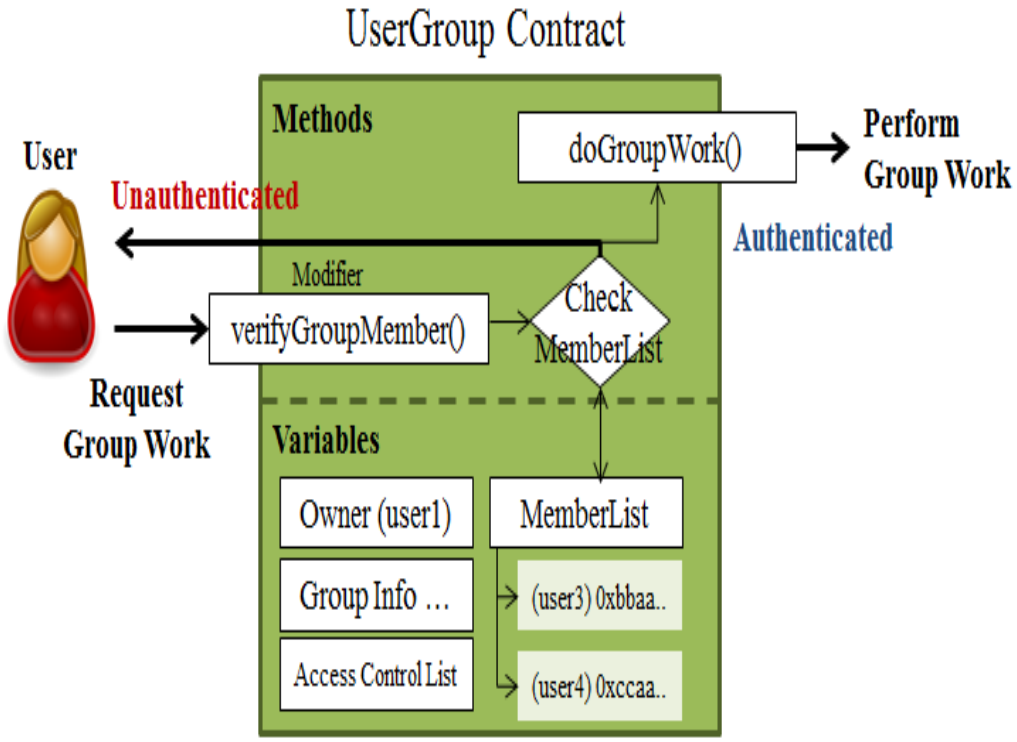

Figure 3. Process of Group Authentication

4.2. Access Control of Group Member

If shared resources are used together by group members, the roles of the group members should be distinguished by specifying access rights to the shared resources. The proposed technique adopts the access control method that the UserGroup contract controls access rights to the shared resource. The UserGroup contract stores an access control list (ACL) for shared resources. The access rights information of the UserGroup contract can be changed by the group owner and managed differently according to the type of user group.

The public group does not manage an ACL because all members have full privileges. The anonymous group manages only one access information because all members use the same access rights specified by the group owner. The owner-defined group uses an ACL for each member to have different access rights for each member. Finally, the user-defined group manages an ACL for all members the same as the owner-defined group and additionally manages privilege distribution information. The user-defined group can check the privilege information of the temporary account accessing the shared resource based on the privilege distribution information. Each group member can access the resource only by checking the access rights to the shared resource based on the ACL of

the UserGroup contract. If a group member wants to access a specific resource through other decentralized application, you can verify access to the resource using the *verifyPermission* method of the UserGroup contract. The *verifyPermission* method checks the ACL according to the type of user group and returns whether the corresponding group member can access the resource. Figure 4 shows the process of confirming access rights for a group member to use shared resources.

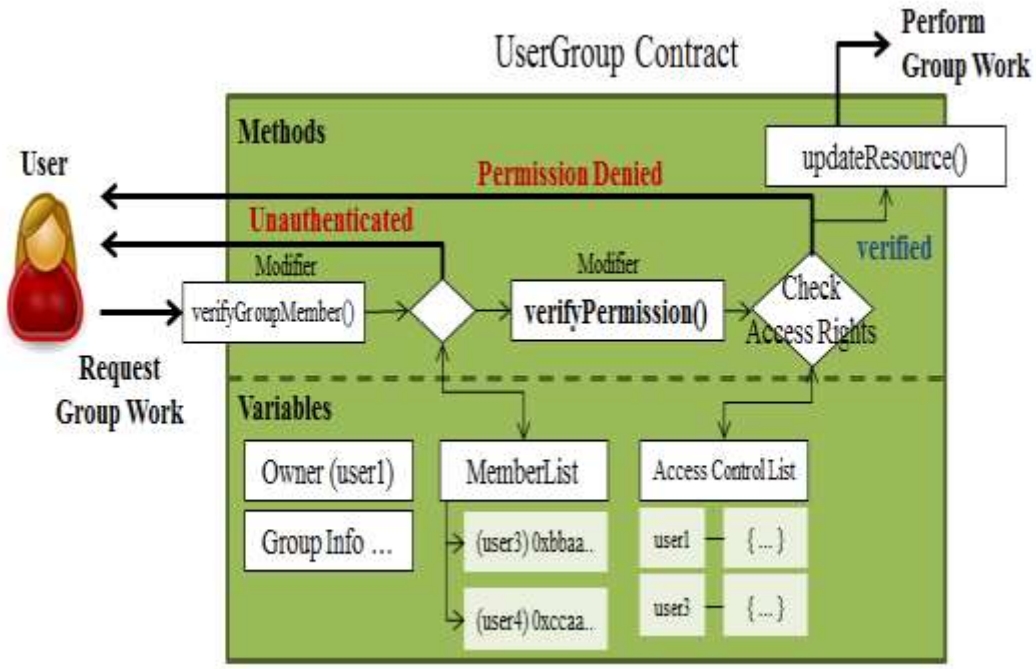

Figure 4. Process of Confirming Access Rights

The owner-defined group checks the privilege distribution information to verify the access rights of the temporary account. At this time, since the access right of the temporary account is created based on the access right of the group member who has distributed the access right, the group member's authority is checked before confirming the access right of the temporary account. This step is necessary to prevent a temporary account from having more access rights than the group member. The method for checking access rights works in the *verifyPermission* method just like any other type of user group. The *verifyPermission* method is also used as a modifier for conditional checking and execution guard in all methods associated with group works. So, when a user requests a group work on a resource, the UserGroup contract performs group authentication and access authorization checking through a modifier. After the two verification processes are completed, a group work desired by the user is performed on the resource.

4.3. Process of User Group Management with DApp

The proposed access control scheme is used in decentralized applications that perform operations on group workspaces. In this kind of decentralized applications, group members are subjected to group authentication to perform operations on group workspaces. Figure 5 shows the process in which a user is granted permission to access a resource through group authentication in Web-based collaborative applications associated with the Ethereum blockchain.

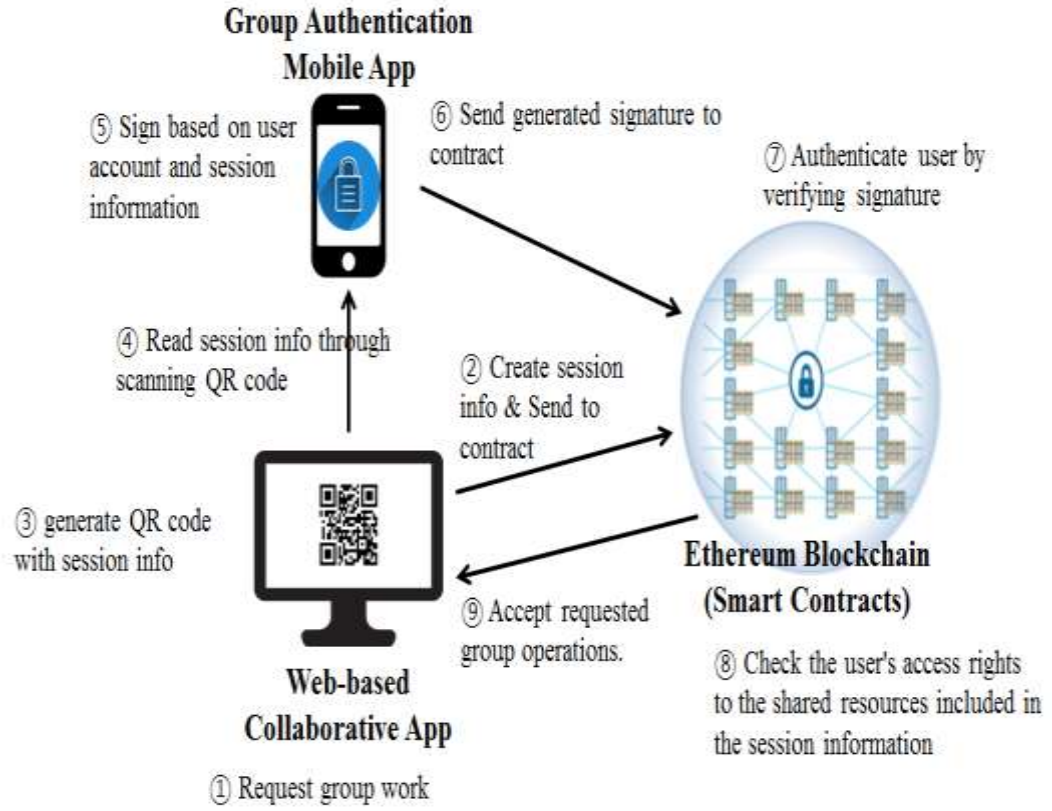

Figure 5. Process of Group Authentication using Collaborative DApp

In the proposed group authentication process, the user has a mobile device that includes a group authentication application, initiating a Web-based collaboration application to perform group operations. The group authentication application has personal information including the user's Ethereum account address, public key, and private key.

Initially, a user requests work on a shared resource in a Web-based collaborative application. At this time, the collaboration application generates session information for the requested work and delivers it to the smart contract managing session information. The session information includes the shared resource information and the type of operation to be performed. When the session information is transmitted, the collaboration application generates a QR code based on the session information and displays it on the web screen. Then, the user scans the QR code on the web screen through the group authentication application. The group authentication application generates signature information by signing with user account information and session information. The generated signature information ensures that the user of the account signs for the requested work. The signature information is passed from the group authentication application to the smart contract for group authentication. The smart contract verifies the signature information and extracts the user's account address and session information. After that, the smart contract performs group authentication using the user's account address and verifies the access rights by checking the requested operation contained in the session information. If the group authentication is successful, the collaboration application performs the requested operation in the session. With this process, collaborative applications can manage group authentication and access rights. Figure 6 shows an example of smart contracts for operating it.

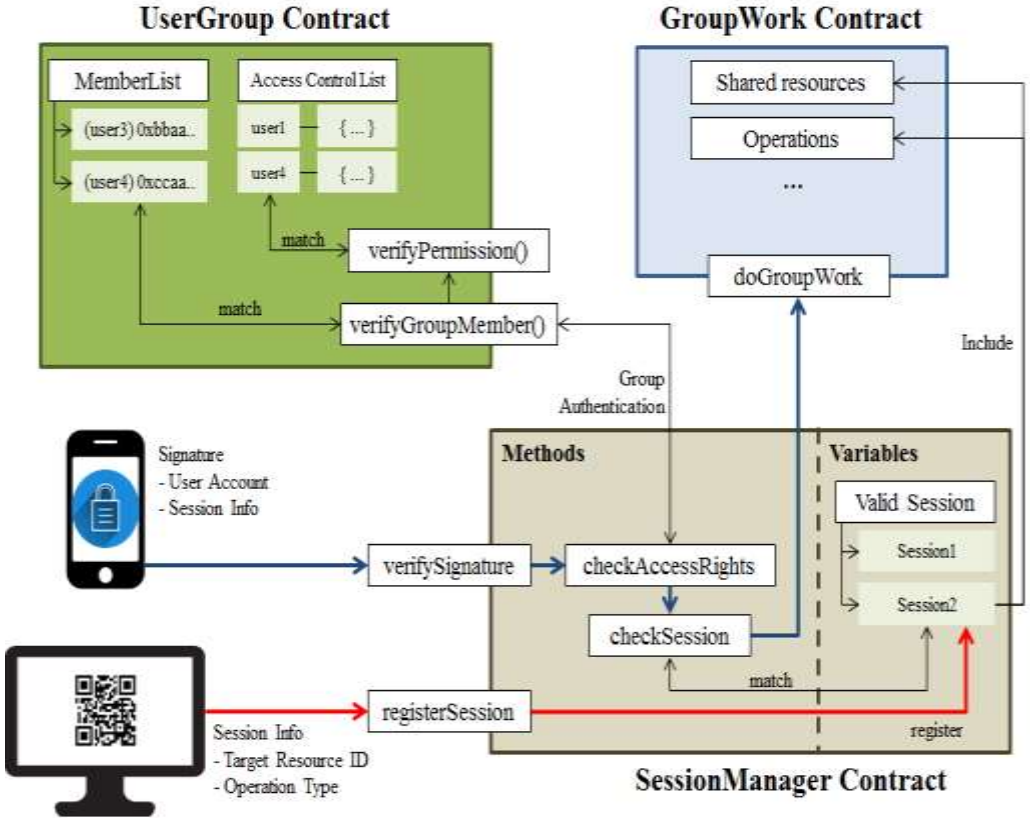

Figure 6. Smart Contracts for Collaborative DApp

5. Conclusion

In this paper, we presented a group authentication and authorization management method based on the Ethereum blockchain. In this method, the user groups are defined as various types according to the authority management principle, and the formation method of each user group is described. Also, we organized the information about the formed group as the Ethereum smart contracts. Using the method, the group users can access the user group contract through the proposed group authentication process, and the group members can be assigned the access right based on the privilege information stored in the user group contract to access the shared resource. The developed group authentication scheme can be utilized in various decentralized applications that require sharing of resources among multiple users with various access rights or task history analysis. Also, the scheme can be used in various types of workspaces because it can control not only the shared resources of blockchain but also the data stored in the off-blockchain. So, based on the presented method, it is possible to easily develop a high-level group service that performs group work systematically over the Ethereum blockchain.

Acknowledgments

This paper is a revised and expanded version of a paper entitled "Managing User Groups and Access Rights on Ethereum Blockchain" presented at the 14th 2017 International Interdisciplinary Workshop Series, which was held on 21st-23rd December in Daejeon (Korea). This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A3B03931976).

References

[1] W. Gavin, "Ethereum: A secure decentralised generalised transaction ledger", Ethereum Project Yellow Paper, vol. 151, **(2014)**.
[2] K. Lee, J. I. James, T. G. Ejeta and H. J. Kim, "Electronic Voting Service Using Block-Chain", The Journal of Digital Forensics, Security and Law: JDFSL, vol. 11, no. 2, **(2016)**, pp. 123-135.
[3] S. C. Cha, W. C. Peng, Z. J. Huang, T. Y. Hsu, J. F. Chen and T. Y. Tsai, "On Design and Implementation a Smart Contract-Based Investigation Report Management Framework for Smartphone Applications", International Conference on Intelligent Information Hiding and Multimedia Signal Processing, **(2017)**, pp. 282-289.
[4] A. Kosba, A. Miller, E. Shi and Z. Wen, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts", Security and Privacy (SP), 2016 IEEE Symposium on IEEE, **(2016)**, pp. 839-858.
[5] P. Jiang, F. Guo, K. Lian, J. Lai and Q. Wen, "Searchain: Blockchain-based private keyword search in decentralized storage", Future Generation Computer Systems, **(2017)**.
[6] uPort Whitepaper, http://whitepaper.uport.me/uPort_whitepaper_DRAFT20170221.pdf.
[7] W. K. Lee, J. H. Jin and M. J. Lee, "A Blockchain-based Identity Management Service Supporting Robust Identity Recovery", International Journal of Security Technology for Smart Device, vol. 4, no. 1, **(2017)**, pp. 29-34.
[8] A. Bogner, M. Chanson and A. Meeuw, "A decentralised sharing app running a smart contract on the ethereum blockchain", Proceedings of the 6th International Conference on the Internet of Things, **(2016)**, pp. 177-178.
[9] E. Dapps, http://dapps.ethercasts.com/.
[10] J. E. Park, J. M. Park and M. J. Lee, "DirectSpace: A Collaborative Framework for Supporting Group Workspaces over Wi-Fi Direct", Mobile, Ubiquitous, and Intelligent Computing. Springer, Berlin, Heidelberg, **(2014)**, pp. 55-61.
[11] H. C. Lee, J. E. Park and M. J. Lee, "C3ware: A Middleware Supporting Collaborative Services over Cloud Storage", The Computer Journal, vol. 57, no. 2, **(2013)**, pp. 217-224.

Authors

Jae-Hwan Jin, is a Ph.D. student in the School of IT Convergence at the University of Ulsan in Korea. He received his M.S. from the University of Ulsan in Korea. His research interests include Beacon-based service, cloud storage service, and collaborative system.

Wookun Lee, is a M.S. student in the School of IT Convergence at the University of Ulsan in Korea. He received his B.S. from the University of Ulsan in Korea. His research interests include blockchain-based service, mobile web application and collaborative system.

Myung-Joon Lee, is a professor in the School of IT Convergence at the University of Ulsan in Korea. He received his Ph.D. from the Kaist (Korea Advanced Institute of Science and Technology) in Korea. He has (co-)authored more than 200 research publications including numerous works on collaborative system, distributed system and biological system.

