Techno-Science Research Journal 10(2) (2022) 50-56

Content list available at ITC

Techno-Science Research Journal

Journal Homepage: http://techno-srj.itc.edu.kh/

Security Enhancement of Kubernetes Management in the Blockchain Platform for Building the Medical System for Information Exchange in Cambodia

Kokthay Poeng[1*], Dara Tith[1], Phutphalla Kong[1]

[1] *Department of Information and Communication Engineering, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

Received: 29 July 2022; Accepted: 19 October 2022; Available online: December 2022

Abstract: *Medical Information System represents a designed system that enables medical staff to collect, store, manage, and transmit data to strengthen the health sector. Healthcare data is sensitive and therefore, it requires protection from unauthorized access. Besides, this data should be shareable among all health organizations. Thus, the health information system enables multiple users from different healthcare organizations to join the network and have the privilege to access their resource. So, Blockchain, Kubernetes, and Access Control are technologies that fully fit all requirements to create the new medical information exchange system. This research is to investigate Kubernetes, Blockchain, Access Control, and Open Policy Agent to solve the research problem. With the investigation, this article also designs and deploys role-based access control with MAC address verification using Open Policies Agent to manage the accessibility of the resources in the Kubernetes cluster, which handled the blockchain network to distribute health care data using a new medical information system. The proposed method of research is to design the use case of the whole system, select a better access control model, and design the selected access control model to deploy to the system. After the implementation of the access control model, the researcher will get a policy evaluation engine that is set up in front of the infrastructure stands as a proxy server, to ensure that no one can break the rule and also to protect the infrastructure by preventing attackers from accessing the infrastructure directly. Moreover, we can get access control processing time through a set of experimental results.*

Keywords: Medical Information System; Blockchain; Kubernetes; Role-Based Access Control

1. INTRODUCTION

Over time, medical information was converted into electronic form, and the reliability of the information depended on the efficiency of the medical information system. Ideally, the system allowed hospitals and other relevant institutions to collect, store, manage, analyze, and optimize patient treatment histories and other data. Today, medical information system enables doctors, nurses, and other medical professionals to easily access patients' data to expedite the treatment process. Moreover, this technology has assisted the facilities to minimize the scope of human error with automation and many jobs ended up being software-operated. However, medical information system also requires a better security technique to protect the sensitive data of patients and needs to have the ability to transmit medical information of patients amongst all healthcare organizations.

The planning from the Department of Planning and Health Information, Ministry of Health of Cambodia [1] reported that, in Cambodia, the medical information of patients was being moved to digital form. For this reason, Cambodia's government encourages all hospitals and clinics to improve the way of storing and expressing medical information from paper to digital. Nevertheless, the Department of Planning and Health Information, Ministry of Health of Cambodia [2] reported that most hospitals in Cambodia are utilizing their independent systems, which means that there are limitations to sharing medical information among hospitals.

Regarding digital medical information exchange, Abimbola et al. [3] said that a decentralized system is a technology that enables multiple computer nodes to participate in a network to communicate together with various impressive features. Firstly, the communication of multiple computers in the decentralized system allows all nodes in the network to be able to share data with other nodes. Secondly, the scalability feature

* Corresponding author: Kokthay Poeng
E-mail: pkokthay@gmail.com; Tel: +855-11 525 656

provides the system with the ability to add more nodes easily. Lastly, within the connection of all nodes in the network, the data is shared with the other computer in the cluster; therefore, the data can still be accessed even if some computers were dysfunctional. Currently, such a popular technology of decentralized systems is called Blockchain.

Based on the above introduction, Blockchain is a suitable technology for creating a new medical information system that can exchange medical information among all health organizations in Cambodia. At the same time, blockchain technology requires multiple nodes to participate in the blockchain network. It needs a fully fit infrastructure, which has reliability, scalability, and high availability to host them. Thus, this article selected Kubernetes infrastructure, which consisted of those abilities to host the blockchain platform. The document provided by the Kubernetes Authors, Linux Foundation [4] mentioned that Kubernetes is an open-source software system used in microservices architecture such as cloud computing and the Internet of Things (IoT), and it has to merge Docker as the platform to manage the container life cycle and automate the management of computerized services. However, each organization that joined the Kubernetes cluster still obtained the privilege to manage and use their resources. Therefore, managing the access control with multiple organizations joined in a cluster is a priority task, which needs to be considered.

To solve the above problem, our research aims to investigate, design, and deploy role-based access control that is expressed by Open Policy Agent to the Kubernetes infrastructure, which handles the medical information exchange system in the Blockchain platform.

2. BACKGROUND AND RELATED WORKS

In this section, we explain the background of four technologies including Blockchain, Kubernetes, Access Control, and Open Policies Agents, while shedding light on any other related work

2.1. Blockchain

Blockchain is a distributed ledger technology that shares a ledger with multiple inter-connected devices (phones, computers, or embedded systems) in a public or private network, and it is not restricted by geography. The reliability between nodes and nodes in this technology depends on the system design. For instance, in a "Block" content Hash, Previous Hash, and Data are distinct blocks that are connected to each other by Hash and the Previous Hash (the Hash of the previous Block), and it is called a "Chain" [5]. With this data structure, fraud cannot happen in the blockchain network because of its immutability feature. Before new transactions are added to the ledger, participating nodes have to reach a consensus to ensure that there exists only one single valid copy of the record shared by all the nodes, despite the attacker's capacity to control a few nodes within the network. Transparency is a salient feature of Blockchain, which enables all participants to build trust among all parties and fully audit all transactions in the network [6,7].

2.2. Kubernetes

Kubernetes is the most popular container orchestration tool available and maintained by The Linux Foundation, which is one of the fastest-growing open-source communities [4]. The Kubernetes project originated within Google, a long-time user of a massive number of containers. To manage these containers well, one needs to develop a system for container orchestration. Kubernetes combines the lessons that Google learned from years of container usage into a single tool with an array of features that make container orchestration simple and adaptable to the wide variety of use cases in the technology industry. Since it became open source in July 2015, the capabilities of Kubernetes have continued to grow [8].

Kubernetes orchestration can build application services that span, schedule, and scale several containers across the cluster. Additionally, it can control the health of those containers. It is easy to achieve high availability with Kubernetes. Kubernetes supports multi-tenancy, with which we can develop and test blockchain applications. To provide a comprehensive container infrastructure, Kubernetes requires integration with security, storage, networking, and telemetry, among other services. With Kubernetes, one can arrange containers across multiple hosts and make good hardware use to increase the resources required to execute the enterprise apps.

2.3. Access control

Access Control (AC) is the most fundamental technique of data security in the information technology world. AC enables organizations and systems to manage who is authorized to access corporate data and resources. This way, AC seeks to prevent activity that could lead to a breach of security [9]. AC can be enforced through many types of AC models such as Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC), Attributed-Based Access Control (ABAC), etc [10].

RBAC is a technology that provides security to admins, simplified by using roles, hierarchies, and constraints to organize privileges [11]. This technology limits user access to the resources by their role. RBAC is organized into four main components including Core RBAC, Hierarchical RBAC, Static Separation of Duty Relations, and Dynamic Separation of Duty Relations [12]. Each component requires three functions Administrative Functions, Supporting System Functions, and Review Functions.

ABAC, on the other hand, enables or denies user access requests based on the attributes of users and the resources that they want to access. Additionally, ABAC is a flexible technology that can provide access control for many different types of resources, accessed by varying types of users [13].

Common to all models of ABAC, there are two types of attributes including Subject Attributes and Object Attributes. Notably, the components in ABAC that contribute to evaluating the policies are; Policy Enforcement Point, Policy Decision Point, Policy Information Point, and Policy Administration Point. There are two standards to express the ABAC model including Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC).

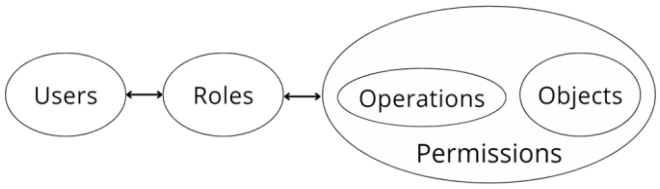

Fig. 1. Role-based access control architecture

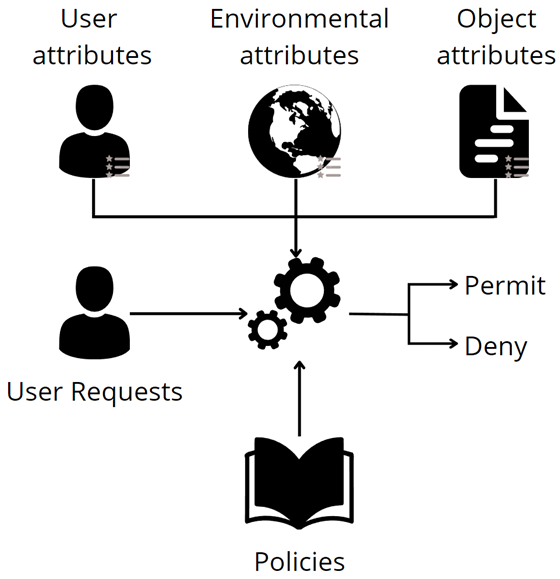

Fig. 2. Attribute-based access control architecture

2.4. Open policy agent

The Open Policy Agent (OPA) is an open-source and general-purpose policy engine that unifies policy enforcement across the stack (Cloud Native Computing Foundation, 2022). OPA provides a high-level declarative language that specifies policy as code and simple APIs to offload policy decision-making from any software. OPA uses Rego language to express policies and JSON to store data and input. This tool can enforce policies in microservices, API gateway, Database, CICD pipelines, and more. Moreover, OPA generates policy decisions by evaluating the query input against policies and data. OPA and Rego are domain-agnostic, so users can describe almost any kind of invariant in the policies. Almost all access control models can enforce their rule in OPA. Besides, OPA provides a declarative language, where policy is specified as code and APIs offload policy decision-making from software. OPA lets multiple teams contribute independent policies that provide the combination to make an overall decision. One or more policies can merge different packages to make a decision. The package can be ABAC, RBAC, XACML, and other written policies

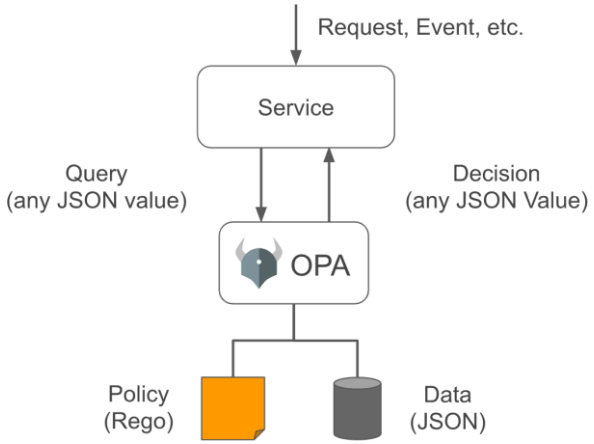

Fig. 3. Open policy agent workflow

2.5. Related work

To collect our internet artifacts, the researcher will use the Google search engine, Google Scholar, Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers, and Association for Computing Machinery (IEEE) with two search strings: "Kubernetes security" and "Kubernetes access control".

Shamim et al. [14] mentioned that the security vulnerabilities in the default Kubernetes deployments are susceptible. So, they have recommended 11 security practices that include 1st-Authentication and Authorization, 2nd-Implementing Kubernetes-specific Security Policies, 3rd-Vulnerability scanning, 4th-Logging, 5th-Namespace separation, 6th-Encrypt and restrict access to etcd, 7th-Continuous update, 8th-Limit CPU and memory quota, 9th-Enable SSL/TLS support, 10th-Separate sensitive workload, and 11th-Secure metadata access. 1st-Authentication and Authorization are the first recommendations that are needed for implementation of Kubernetes to secure authorization is Access Control.

Another study proposed 10 Kubernetes security best practices reported by practitioners that can take enterprise security to the next level such as Authentication and Authorization Role-Based Access Control (RBAC), Private Kubernetes API Endpoint, Kubernetes-specific Security Policies, Audit Logging Monitor Network Traffic to Limit Communications, Namespace separation, Isolate Kubernetes Nodes, Keep Kubernetes Version Up to Date, Encrypt and restrict access to etcd, Limit CPU and memory quota and Enable

SSL/TLS support. The authors suggested that "Authentication and Authorization using RBAC" is the first security practice, which is required to implement in the Kubernetes cluster.

To build a zero-trust architecture to secure Kubernetes, D'Silva & Ambawade [15] proposes to deploy a stand-alone machine that installed policy decision points by using XACML to express a hybrid RBAC model in front of Kubernetes infrastructure. This proposed method prevents attackers from compromising the entire system by gaining access to the network, masquerading themselves as legitimate users to access control policy.

Minna et al. [16] explored the security implications of Kubernetes networking and provided many ideas to prevent Kubernetes from unexpected attacks. One of their ideas mentioned that Kubernetes does not automatically audit the security level of policies in a cluster and the potential risks and vulnerabilities that may result from them. In particular, authentication and authorization such as role-based access control (RBAC) and service accounts, secrets management, network policies, pod security policies, general policies handling the use of namespaces, and security options should be analyzed before deploying the cluster and exposing its services to the outside.

Based on the above works, the research will focus on Role-Based Access Control as the priority of Kubernetes security enhancement.

3. PROPOSED METHOD

This section will segregate methodology into two parts including (i) proposing a scenario and (ii) designing and expressing RBAC with MAC address verification (M-RBAC) using OPA with the proposed scenario to the infrastructure.

3.1. Proposed scenario

In the medical field, sharing medical information using private blockchain requires multiple hospitals to join the network and needs one or more organizations to manage, develop, and maintain the entire network and all participating hospitals.

In the designed scenario, the research proposes one organization that is responsible for managing, developing, and maintaining the whole blockchain platform with four main hospitals. Each hospital provides its services to clinics or other hospitals in their local area. The designed Scenario showed in Fig. 4 below.

The designated organization has two types of users, first is the super admin who manages everything in the cluster, and the other is the developer's team that is responsible for developing and maintaining the blockchain in a testing environment. Each main hospital has their network administrator who can access only their resources. Moreover, when a hospital joins the Kubernetes cluster, it will have one or more servers in its local area to join as worker nodes, and they can still use the remaining resources of their server to support their services.

Regarding the whole infrastructure, the researcher proposes to place the OPA engine between the connection user and the whole cluster to stand as a proxy server, so everyone has to be authenticated and authorized from this server before access to the cluster. Within this design, every single user must respect the rule that is enforced and the OPA machine reduces Kubernetes Master Node's workload with an access control mechanism. Moreover, the attackers can only communicate with the OPA engine, not directly with the Kubernetes network, so the infrastructure is still safe even if the policy engine is down.

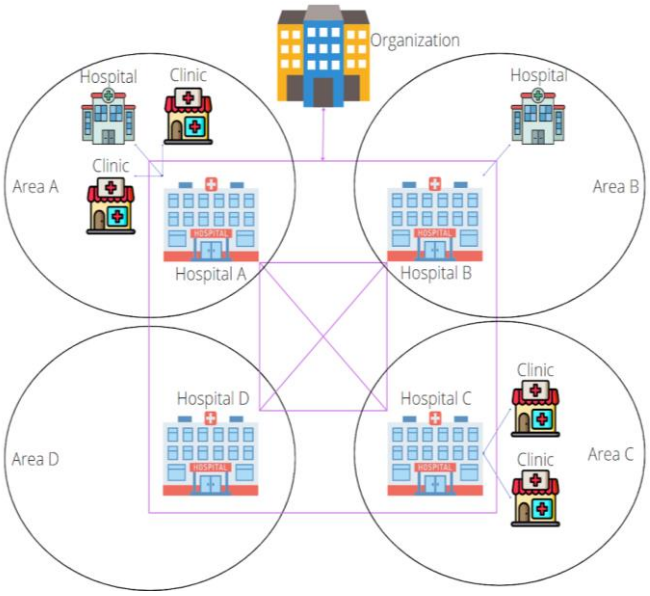

Fig. 4. Proposed Scenario

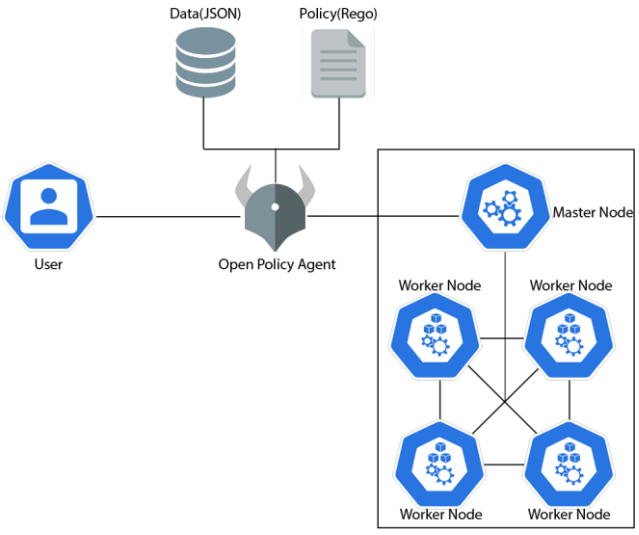

Fig. 5. Logical Infrastructure

3.2. Design and express RBAC with MAC address verification (M-RBAC) to the infrastructure

The proposed scenario above includes 5 types of roles namely; Developers, Admin Hospital A, Admin Hospital B, Admin Hospital C, and Admin Hospital D. This matched the other 5 types of resources such as Testing, Resource of Hospital A, Resource of Hospital B, Resource of Hospital C and Resource of Hospital D. The Table 1 below shows the Permission to Role assignment.

Table 1 Role-Permission assignment

Role	Operation	Resource
Developers	Full	Testing
Admin Hospital A	Full	Resource of Hospital A
Admin Hospital B	Full	Resource of Hospital B
Admin Hospital C	Full	Resource of Hospital C
Admin Hospital D	Full	Resource of Hospital D

Based on the design, users input their username, operation, resource, and MAC address to the policy decision point, and then the system matches the username with the role and checks the role with permission. Lastly, the system will check the user's MAC address on their computer to ensure that only the registered computer can gain the privilege to access the resources. The process of checking Role, Permission, and MAC address is shown in the diagram below.

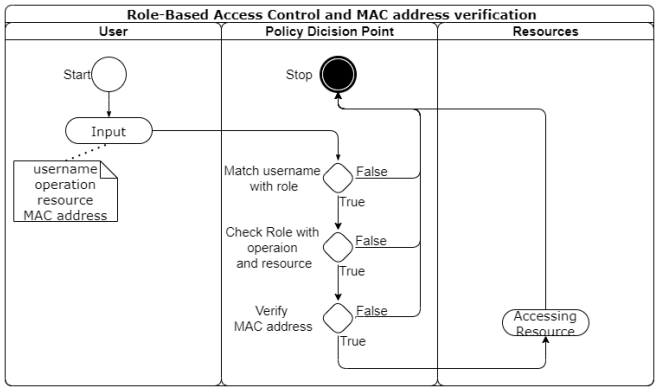

Fig. 6. M-RBAC activity diagram

4. IMPLEMENTATION

This part will present the systematic implementation of RBAC with MAC address verification (M-RBAC), using OPA as a policy decision point with the 5 types of roles to the Kubernetes platform to get the processing time of policy evaluation.

The first step of implementation is to install an OPA engine. OPA supports almost all operating systems such as Windows OS, Linux OS, Mac OS, and container virtualization. After deploying the OPA engine, the researcher will have to write the M-RBAC policy in Rego format and data in JSON format. The last step is to enforce the written policies on the OPA engine. The policy specification shown in Fig. 7.

```
package abac.authz

    default allow = false
    allow {
        # matching user with role
        roles := data.user_roles[input.user]
        r := roles[_]
        # lookup the permissions list for role r
        permissions := data.role_permissions[r]
        # check if the permission is match to the user's request
        input.action == permissions.action[_]
        input.resource == permissions.namespace[_]
        input.mac == permissions.mac[_]
    }
```

Fig. 7. Policy specification

Below is the experiment, which is demonstrated to get the processing time of policy evaluation. The experiment was evaluated 150 times and the researcher divided it into three stages, which are shown in Table 2.

- In the first stage, the admin accessed their organization's resources correctly 10 times per admin, so the researcher got 50 processing times from 5 admins.
- In the second stage, the admin also accesses their organization's resources 10 times per admin but with the incorrect MAC address. Therefore, the researcher can get 50 more processing times.
- In the last stage, each admin accesses the other organization's resources in total 10 times per admin. Therefore, there are 50 more processing times from 5 admins.

The researcher used one computer to implement and test the access control which included CPU Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz, Memory 32 GB, and Windows 11 operating system.

Table 2a Correct evaluation

Resource	Org1	Org2	Org3	Org4	Testing
Admin Org1	10	0	0	0	0
Admin Org2	0	10	0	0	0
Admin Org3	0	0	10	0	0
Admin Org4	0	0	0	10	0
Devolopers	0	0	0	0	10
	Total			50 Times	

Table 2b Incorrect MAC address evaluation

Resource	Org1	Org2	Org3	Org4	Testing
Admin Org1	10	0	0	0	0
Admin Org2	0	10	0	0	0
Admin Org3	0	0	10	0	0
Admin Org4	0	0	0	10	0
Devolopers	0	0	0	0	10
Total					50 Times

Table 2c Incorrect MAC address evaluation

Resource	Org1	Org2	Org3	Org4	Testing
Admin Org1	0	3	2	3	2
Admin Org2	3	0	2	3	2
Admin Org3	3	2	0	3	2
Admin Org4	3	2	3	0	2
Devolopers	0	2	3	2	0
Total					50 Times

5. RESULTS AND DISCUSSION

Depending on the above literature review, the RBAC features specifically configured for the application are systematically incorporated into the designed model. The flexibility of OPA allows the researcher to enforce set policy on the infrastructure easily. Moreover, the proposed method by combining M-RBAC ensures no one is unable to gain illegal permission and prevents the cluster resources from unauthorized access.

The chart below illustrates the average processing time of M-RBAC after installing it in the system.

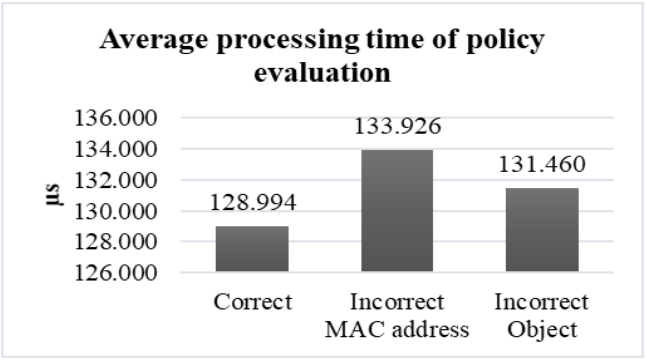

Fig. 8. Average processing time of policy evaluation

Within the processing time, the longest gap between these three cases is five µs and the shortest gap is 2.5µs. The correct evaluation is a case in which every user correctly accessed the resources. Therefore, the evaluating policy process in the policy decision point consumes less time than the other two. The Incorrect MAC address evaluation case took the longest time than other cases because the process to verify the MAC address was at the last point of all policy checking. Lastly, the consumed time of the Incorrect Object case standing between the correct case and the incorrect MAC address takes less time than the Incorrect MAC address because the process of checking the object in the policy is required before validating MAC address. Hence, if the object was not matched, the policy will deny access without checking the MAC address.

Below is the comparison between M-RBAC with Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Attribute-Based Access Control (ABAC) [17].

Table 3 Comparison of access control model

Access Control Model	DAC	MAC	ABAC	M-RBAC
Ease of Usage	High	Varies	High	High
Performance	Low	Varies with Security Levels	High	High
Reusability	Yes	No	Yes	Yes
Single Point Failure	Authorization failure	Less	Less	Less
Authentication Failure	Less	Varies	less	Based on role

Next step, the study will investigate the XACML and NGAC to express our M-RBAC in these two forms to make a comparison

6. CONCLUSION

In summary, this research started with multiple computer nodes joined in a Kubernetes cluster to share medical data using a blockchain platform. Nevertheless, the access control to each resource of each node is a complication, which needs to be considered. Therefore, after the literature review, RBAC is suitable access control that can solve the problem. Thus, this proposes a design of RBAC and MAC address verification and then expresses it with OPA to secure the infrastructure, which hosts the blockchain platform to handle medical information exchange systems.

ACKNOWLEDGMENTS

This research is fully funded by the Research Fund Institute of Technology of Cambodia. Moreover, ERASMUS+ Program provide a fully fund research internship in University of Namur in InfoSec lab for 3 months.

REFERENCES

[1] Department of Planning and Health Information, Ministry of Healths of Cambodia (2017), "Health Information System Master Plan 2016-2020"

[2] Department of Planning and Health Information, Ministry of Healths of Cambodia (2016), "Health Strategic Plan 2016-2020 Quality, Effective and Equitable Health Services".

[3] Abimbola, S., Baatiema, L., & Bigdeli, M. (2019). The impacts of decentralization on health system equity, efficiency, and resilience: a realist synthesis of the evidence. Health policy and planning, 34(8), 605-617.

[4] Kubernetes Authors, Linux Foundation (2022) Kubernetes <https://kubernetes.io/docs/>

[5] Xu, M., Chen, X., & Kou, G. (2019). A systematic review of blockchain. Financial Innovation, 5(1), 1-14.

[6] Maesa, D. D. F., Mori, P., & Ricci, L. (2019). A blockchain based approach for the definition of auditable access control systems. Computers & Security, 84, 93-119.

[7] Kritikos, M. (2018). What if blockchain offered a way to reconcile privacy with transparency?

[8] Kang, H., Le, M., & Tao, S. (2016, April). Container and microservice driven design for cloud infrastructure devops. In 2016 IEEE International Conference on Cloud Engineering (IC2E) (pp. 202-211). IEEE.

[9] Sandhu, R. S., & Samarati, P. (1994). Access control: principle and practice. IEEE communications magazine, 32(9), 40-48.

[10] Karp, A. H., Haury, H., & Davis, M. H. (2010). From ABAC to ZBAC: the evolution of access control models. Journal of Information Warfare, 9(2), 38-46.

[11] David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli (2007). Role-Based Access Control, Second Edition. Artech House.

[12] Hu, V. C., Ferraiolo, D. F., Chandramouli, R., & Kuhn, D. R. (2017). Attribute-Based Access Control. Artech House.

[13] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC), 4(3), 224-274.

[14] Shamim, M. S. I., Bhuiyan, F. A., & Rahman, A. (2020, September). Xi commandments of kubernetes security: A systematization of knowledge related to kubernetes security practices. In 2020 IEEE Secure Development (SecDev) (pp. 58-64). IEEE.

[15] D'Silva, D., & Ambawade, D. D. (2021, April). Building a zero trust architecture using Kubernetes. In 2021 sixth international conference for convergence in technology (i2ct) (pp. 1-8). IEEE.

[16] Minna, F., Blaise, A., Rebecchi, F., Chandrasekaran, B., & Massacci, F. (2021). Understanding the security implications of kubernetes networking. IEEE Security & Privacy, 19(05), 46-56.

[17] Chinnasamy, P., & Deepalakshmi, P. (2017). A survey on enhancing cloud security through access control models and technologies. International Journal on Computer Science and Engineering (IJCSE), 9(5), 326-331.