# Electronic Access Control System Based On Blockchain

Julando Omar[1], Samuel Hutagalung[2]
[1,2] Computer Engineering, Universitas Multimedia Nusantara, Tangerang, Indonesia
julando.omar@student.umn.ac.id
samuel.hutagalung@umn.ac.id

Accepted on mmmmm dd, yyyy
Approved on mmmmm dd, yyyy

***Abstract*—Electronic access control system is one of the popular access controls which are used today. Electronic access control system has several advantages and also have several weaknesses. These weaknesses are single point of failure and log tampering. Centralized system is the cause of these weaknesses. Based on these problems, electronic access control based on blockchain is made to solve these problems. This solution consists of three part which are Frontend, Blockchain and Node. The proposed systems can be used as an electronic access control systems which could mitigate single point of failure with high availability and log tampering. On Average, system used 35 seconds and 41 seconds to give access to users. Blockchain with PoW consensus could be used on systems that are built by a lot of nodes, and PoA consensus could be used on system that are built by several nodes.**

***Index Terms*— *Access control systems, Blockchain; Distributed systems; Electronic Access Control Systems; IPFS;***

## I. INTRODUCTION

Access control system is one of the important factors in security systems. Access control system security is given by how the system limit who could enter the room [1]. With the development of technology, physical access control - which for examples are door lock, gates, etc. – are also developing into electronic access control. Electronic access control is control systems which are the combination of cyber and physical systems [2]. Examples of electronic access control systems are smart door, fingerprint sensor, etc.

While electronic access control has some advantages than physical access control, there are some vulnerabilities affecting the electronic access control. These vulnerabilities are single point of failure and log tampering. Single point of failure is a vulnerability which if the system fails to work as intended, the whole system would be affected [3]. Log tampering is a vulnerability in which attacker would attack the logging system by injecting, manipulating, or changing the log entry, that would erase the attacker trace of attack [4].

These two vulnerabilities rise from the usage of centralized system. Centralized system is a system which a single server or computer is used as a data storage. While a centralized system offers some advantages such as the ease of use, deployment and setup, centralized system also has some vulnerabilities that was addressed in the last section Distributed system is one of the solutions that could fix these vulnerabilities.

Distributed system is a system which data is not stored in single computer or server but scattered across multiple server or computer [5]. An emerging type of distributed system that are widely used is blockchain. Blockchain is a distributed system in which data are stored in blocks that are linked to each other. These blocks than will be distributed among the computers that running as the blockchain ledger. A blockchain would fix the vulnerabilities of a centralized system, by storing data in an immutable data storage and distributed storage.

The common types of electronics access control methods of authentication are smart card, password numpad and biometrics such as fingerprint. While these authentication methods are secure enough, but there are some vulnerabilities that still exist, for example, smart cards are prone to duplication, where attacker could duplicate smart cards in matter of seconds. Multifactor authentication (MFA) could be used to combat this vulnerability. Multifactor authentication is a type of authentication that uses more than two types of authentications [6].

In this study, we introduce an electronic access control system that uses Ethereum blockchain as the system database, MFA which consist of password, QR Code and face recognition as a way to authenticate user. The system consists of the parts, Frontend, which used as an UI/UX of the system, Blockchain, which used as the database and Node, which give access to user to enter the room.

## II. RELATED WORK

Several studies have been conducted on implementing blockchain on physical and electronic access control systems. Study in [7] is one of the first

studies which uses blockchain to control door locking system, and alarm systems to detect whether intruder is present. Study [8] integrate a facial recognition technology with blockchain to control electronic access control system. Study [9] propose a management access system for accommodation using blockchain technology, in which users are given exclusive access to get access to their accommodation.

Paper [10] discussed the usage of blockchain and IPFS as a database to store immutable patient health records information. In this paper, IPFS is used as a way to store data which considered as large to be saved in blockchain. Study in paper [11] proposed blockchain and IPFS as a way to store log files, in which blockchain is used to store the log files indexes and IPFS is used to store the actual log files.

Several studies linked to QR Code usage as a method for control access systems also has been conducted. Paper [12] proposed a system that used QR code as a way to authenticate user, information on the QR Code is hashed with a SHA-2 encryption. Paper [13] proposed a system that used QR code as a way to authenticate user who wanted to access laboratories. This system, scan students QR Code and record their access log to store information about the user who access the laboratories.

Paper [14] and [15], mainly discussed about the usage of blockchain to store face recognition dataset. The paper [14] proposed to store image of user face to be broken down into several pixels that are distributed in blockchain, while paper [15] propose to store image of user face in a yml file, which then saved on the blockchain.

## III. PROPOSED SYSTEM

The proposed system will be divided into 3 sub-systems, which are Frontend, Blockchain and Node

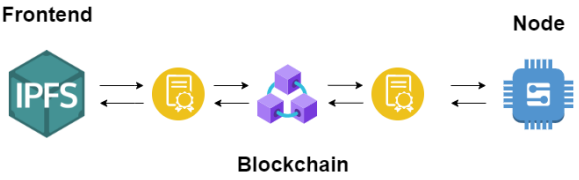


Fig. 1. Proposed System Architecture

### A. *Frontend*

Frontend is the UI/UX of the system. Frontend system is written with Javascript language. Vue framework is used to build the frontend system. Javascript libraries such as Axios and EtherJS is used to communicate with blockchain and IPFS HTTP API, while node-forge is used to create RSA encryption to generate the QR. Figure 2 shows the architecture diagram of Frontend.

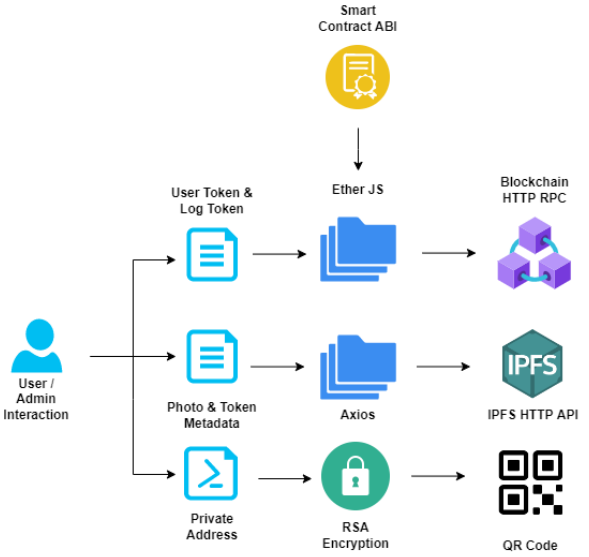


Fig. 2. Frontend System Architecture

Frontend system is divided into 2 parts, which are admin and user. Admin is used to manage users and to see all recorded logs, while user is used by the users to generate QR Code, see the current user log and to manage the user account. Smart contract is used to check if the current logged user has a admin role or not, then the system proceed to redirect the user to their respective page. Figure 3 shows the page flows of Frontend.

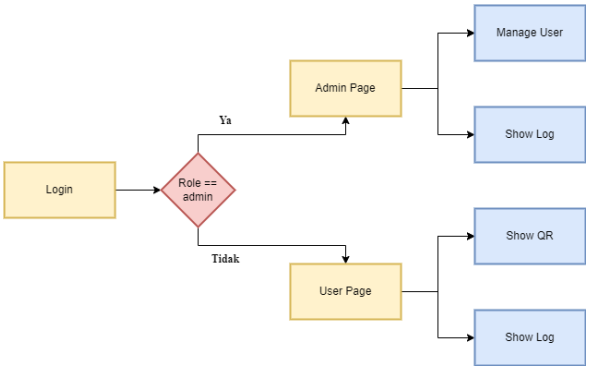


Fig. 3. Frontend System Flow

A user is represented by a user token, and a log is represented by a log token. When a new user is made, admin will get the private key of the user, which then to be used as authentication. The token will be explained on blockchain section. Etherjs is used to call smart contract to get both user token and log token, while axios is used to hit the API endpoint of IPFS to upload or download files which are stored in IPFS. RSA algorithm with 1024 bit encryption is used to encrypt data when a user QR is created.

### B. *Blockchain*

Blockhain sub-system is used as the database of the system. Blockchain is consisted of 3 nodes of geth and IPFS node. Geth is a program that is used to run ethereum node. Bootnode is used to create a private

network of ethereum blockchain. go-ipfs is used to run IPFS server on the node. A private swarm key is needed to run IPFS as a private cluster. A node exposed port 8545, and port 8080, that are used by geth and go-ipfs

Smart contract is used as a way to communicate between sub-systems with the blockchain. The proposed system consists of 2 smart contracts which are UserToken and LogToken. These two smart contracts is derived from ERC-721 smart contract, which wildly used as a NFT token smart contract. Figure 4 shows the architecture of blockchain sub-system.

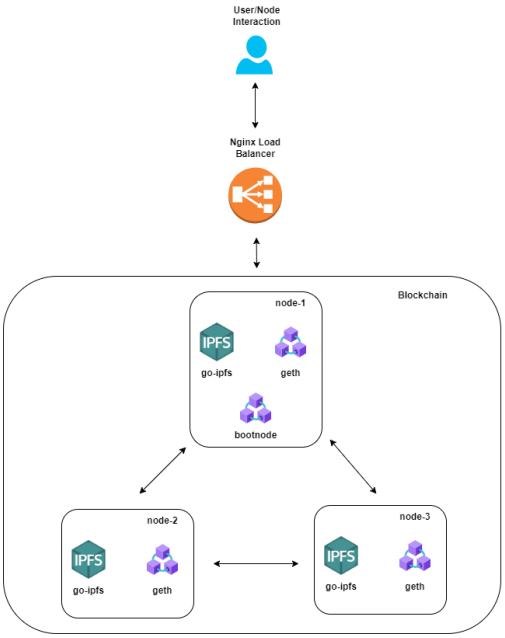


Fig. 4. Blockchain Architecture

A smart contract call is used to create or manage a user. Name, and photos are needed to create a user. a user is represented with a single token in the blockchain. The metadata of a user is stored in the IPFS and linked with it's respective token in the blockchain. A user metadata is consist of the user name, IPFS hash of the user keystore, and IPFS hash of the user photos. User photos are also linked to the user metadata.

Logs are also represented as token in the blockchain. A metadata of a log consisted of timestamp of the log, the user who requested the access, IPFS hash of photo when user requested the access and status of the log. The metadata and photo of the log are also stored in IPFS. Figure 5 shows the storage architecture of the systems.

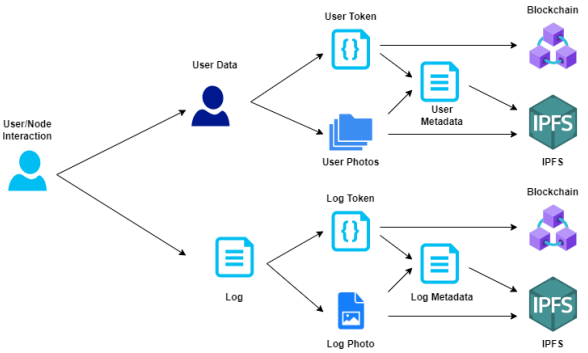


Fig. 5. Systems storage architecture

## C. *Node*

Node subsystem is used to control and give users access to the rooms which the system controls. Node sub-systems consist of a Raspberry Pi, Pi Camera, Relay and LCD Screen. Raspberry Pi is used as the brain of the Node. Raspberry pi run Python script, which used the web3.py, pyzbar, cv2 and deepface package. Pi camera is used to capture QR Code and users faces, relay is used to control physical access and LCD Screen is used as an interface to user. The architecture of node is shown in figure 6

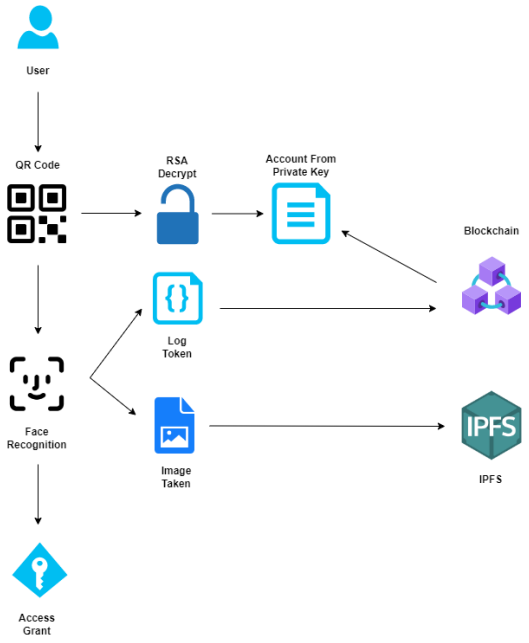


Fig. 6. Systems storage architecture

Web3.py package is used to communicate with blockchain via smart contracts, pyzbar and cv2 is used to recognize QR Code, decode QR Code and to recognize faces. DeepFace is a package which used as face recognition. DeepFace face recognition uses Siamese neural networks with VCGFace model to recognize people faces.

Tkinter library is used to build the UI for the Node. the Node UI consist of 5 Pages, which are enter page, which is used to begin the grant access page; QR page,

which is used to scan and give feedback on user QR code scanning process; face page, which is used to scan and show users faces and recognition process, access page, which is used to show user that they have been granted access; and loading page, which show user that Node in loading mode. Figure 7 shows the Node casing

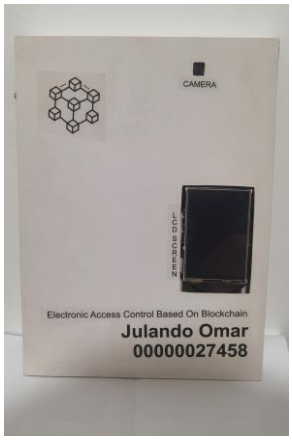


Fig. 7. Node sub-system asing

Hardhat is used to deploy smart contract, Frontend

and public key that is used in on RSA encryption on blockchain and IPFS. Hardhat is a Javascript library package that handles deployment of smart contract to Blockchain. To deploy, hardhat uses scripts that are written in Javascript. Figure

Fig. 8. Deployment Flow

## IV. Result and Discussion

A proof-of-concept of the proposed system was implemented.

### A. Experimental Enviroment

The Blockchain and Frontend sub-system was deployed into a Laptop with Intel i7-8750H processor, 16GB RAM and 512GB SSD. Blockchain and IPFS nodes was deployed on 3 Virtual Machines, running Lubuntu 20.04 OS with 2GB RAM and 2 Processor Cores. Geth and go-ipfs was used on these nodes to run a private Blockchain and IPFS network. Blockchain network was running with proof-of-work and proof-of-authority consensus.

The Node sub-system was deployed with Raspberry Pi 4B with 4GB RAM, Pi camera 2, 3.5 Inch LCD and 8 Channel Relay. To power the system, 12V DC converter was used. The Raspberry Pi was running Raspberry Pi OS 64-bit with Python 3.9.

A load balancer written in Lua and with Nginx framework was used as a load balancer between the nodes. The load balancer was made into docker images and running into the laptop.

### B. Performance Evaluation

We evaluated the performance of the system by measuring time taken from deploying the smart contract to user successfully accessing the room. Measurement taken with both Proof-of-Work and Proof-of-Authority consensus algorithm. The time measured are between block number 0 – 500, 500-1000, 1000-1500. Table 1 and Table 2 show the result of the performance evaluation

TABLE I. Performance Evaluation on PoW Consensus

| Iteration | Contract Deployment | Frontend Deployment | New User Creation | QR Code Generation |
|---|---|---|---|---|
| 1 | 23 seconds | 51 seconds | 40 seconds | 9 seconds |
| 2 | 28 seconds | 37 seconds | 54 seconds | 9 seconds |
| 3 | 29 seconds | 35 seconds | 56 seconds | 9 seconds |
| **New User on Node** | **QR Code Scan** | **Face Recognition** | **Log Creation** | **Total time** |
| 43 seconds | 3 seconds | 18 seconds | 4 seconds | 3 minutes 11 seconds |
| 48 seconds | 5 seconds | 19 seconds | 5 seconds | 3 minutes 28 seconds |
| 41 seconds | 4 seconds | 18 seconds | 4 seconds | 3 minutes 16 seconds |

TABLE II. Performance Evaluation on PoA Consensus

| Iteration | Contract Deployment | Frontend Deployment | New User Creation | QR Code Generation |
|---|---|---|---|---|
| 1 | 28 seconds | 29 seconds | 55 seconds | 10 seconds |
| 2 | 27 seconds | 32 seconds | 56 seconds | 9 seconds |
| 3 | 29 seconds | 41 seconds | 58 seconds | 16 seconds |
| **New User on Node** | **QR Code Scan** | **Face Recognition** | **Log Creation** | **Total time** |
| 43 seconds | 4 seconds | 18 seconds | 8 seconds | 3 minutes 15 seconds |
| 42 seconds | 3 seconds | 18 seconds | 5 seconds | 3 minutes 12 seconds |
| 42 seconds | 4 seconds | 20 seconds | 8 seconds | 3 minutes 38 seconds |

From the result of the performance evaluation, we conclude that both Proof-of-Work and Proof-of-Authority consensus have a similar time consumption. For both consensus, 35 seconds and 41 seconds needed for user to access the room on PoW and PoA consensus. The highest time consumption is on new user creation and getting new user on Node in both consensuses.

Block count slightly affect time on PoW consensus, while PoA consensus is not affected by block count. This is caused by the increasing difficulty rate of block found on PoW consensus, while PoA block difficulty is remain the same. PoA Block generation time is fixed as configured in the genesis file of the blockchain.

Face recognition using DeepFace library is proven to consume a small amount of time with high accuracy. This is caused by the Siamese neural network architecture which could achieve high accuracy on small dataset. VCGFace model also contribute on high accuracy and fast recognition time.

### *C. Log-tampering Evaluation*

We evaluate log-tampering by using 51% attack. 51% attack is a Blockchain attack, which if attacker has 51% hashing power, attacker could rewrite the whole blockchain which resulted in log tampering. We evaluate 51% attack on both consensuses, with each consensus, 51% attack is performed with one node and two attacker nodes. Figure 9 and 10 shows log-tampering evaluation scenarios flow.

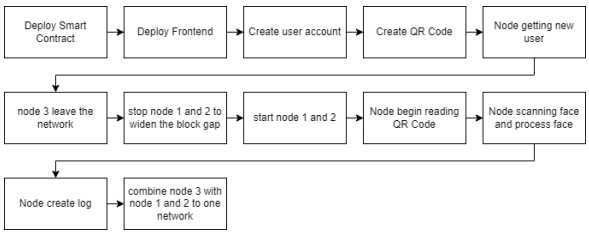


Fig. 9. Log-tampering scneario flow with one node

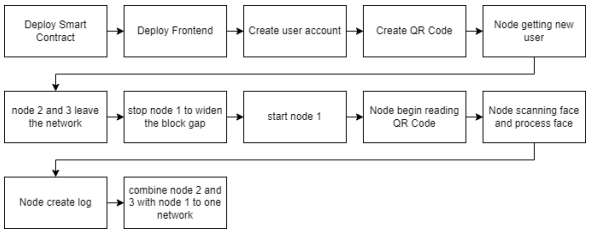


Fig. 10. Log-tampering scneario flow with one node

The experiment with one node with Proof-of-Work consensus, resulted in failed in 51% attack. On geth program log shows that when node 3 rejoined with node 1 and 2, geth detected ghost side-chain attack, and cancel the merge between node 1 and 2 with node 3. The experiment with two nodes with Proof-of-Work consensus, also resulted in failed in 51% attack. When node 1 merged with node 2 and 3, previous transactions that are stored in node 1 was bundled in one block of transaction, thus making the blockchain network still recognize all the transaction that created by node 1.

The experiment with one node with Proof-of-authority also resulted in failure. Proof-of-Authority consensus don't allow only a single node to mine the entire network. Proof-of-authority consensus needed (n/2)+1 nodes to mine the network, if only one node present, block won't be mined. The same goes with two nodes with proof-of-authority consensus, in which the single node could not mine the transaction which was given to the network.

### *D. Availbility Evaluation*

We evaluate the availability of the system by measuring time taken from deploying contract to giving access to user, with 2 active nodes and one active node. we measured the time taken with both consensuses, proof-of-work, and proof-of-authority. Table III and Table IV shows the result of availability evaluation with two active nodes on both consensuses.

TABLE III. AVAILBILITY EVALUATION WITH TWO NODES ON POW CONSENSUS

| **Pairs** | **Contract Deployment** | **Frontend Deployment** | **New User Creation** | **QR Code Generation** |
|---|---|---|---|---|
| 1-2 | 38 seconds | 28 seconds | 45 seconds | 10 seconds |
| 1-3 | 20 seconds | 29 seconds | 52 seconds | 10 seconds |
| 2-3 | 1 minutes 39 seconds | 19 seconds | 1 minutes 2 seconds | 10 seconds |
| **New User on Node** | **QR Code Scan** | **Face Recognition** | **Log Creation** | **Total time** |
| 40 seconds | 4 seconds | 19 seconds | 5 seconds | 3 minutes 9 seconds |
| 44 seconds | 3 seconds | 19 seconds | 11 seconds | 3 minutes 8 seconds |
| 2 minutes 16 seconds | 4 seconds | 19 seconds | 5 seconds | 5 minutes 44 seconds |

TABLE IV. AVAILBILITY EVALUATION WITH TWO NODES ON POA CONSENSUS

| **Pairs** | **Contract Deployment** | **Frontend Deployment** | **New User Creation** | **QR Code Generation** |
|---|---|---|---|---|
| 1-2 | 31 seconds | 35 seconds | 43 seconds | 10 seconds |
| 1-3 | 29 seconds | 27 seconds | 47 seconds | 9 seconds |
| 2-3 | - | - | - | - |
| **New User on Node** | **QR Code Scan** | **Face Recognition** | **Log Creation** | **Total time** |
| 1 minutes 2 seconds | 3 seconds | 20 seconds | 8 seconds | 3 minutes 22 seconds |
| 1 menit 6 seconds | 3 seconds | 18 seconds | 4 seconds | 3 minutes 18 seconds |
| - | - | - | - | - |

From both measurements, the time taken on proof-of-work consensus when node 1 was active is slightly faster than when node 1 is offline. It's caused by the

usage of nginx load balancing, which configured both node 2 and node 3 as a backup, which needed 10-20 seconds for the system to know if node 1 in offline state. For proof-of-authority measurement, when node 1 was active, systems was able to run, in which time taken was similar with the measurement taken on performance evaluation. System couldn't run when node 1 was offline. This was caused by the proof-of-authority process, which needed node 1 to sign and validate the first block signing.

TABLE V. AVAILBILITY EVALUATION WITH ONE NODE ON POW CONSENSUS

| **Pairs** | **Contract Deployment** | **Frontend Deployment** | **New User Creation** | **QR Code Generation** |
|---|---|---|---|---|
| 1 | 30 seconds | 38 seconds | 52 seconds | 10 seconds |
| 2 | 1 minute 14 seconds | 29 seconds | 1 minute 31 seconds | 9 seconds |
| 3 | - | - | - | - |
| **New User on Node** | **QR Code Scan** | **Face Recognition** | **Log Creation** | **Total time** |
| 1 minute 17 seconds | 7 seconds | 19 seconds | 15 seconds | 4 minutes 8 seconds |
| 1 minute 5 seconds | 3 seconds | 18 seconds | 3 seconds | 4 minutes 42 seconds |
| - | - | - | - | - |

Table V show the measurement of availability of the system with one active node on proof-of-work consensus. From the measurement it's shown that there're slight differences in time taken between node 1, and node 2. This is caused by the load balancer configuration which only detected node 1 offline after 10-20 seconds. This configuration also resulted in node 3 inability to process the transaction, which time exceed the timeout time, thus making the transaction declined by the node.

Measurement with one node on proof-of-authority consensus could not be done. This was caused by the proof-of-authority consensus (n/2) + 1 rule, in which a single node could not mine or seal the transaction. This result was also observed on log-tampering evaluation, which give us a same inability to mine and process the transaction given to the network.

## V. CONCLUSION

From the evaluation that was conducted. The proposed system was able to be implemented as a electronic access control system which free from log-tampering and has a high availability which could mitigate single point of failure vulnerability. The performance evaluation shown that 35 seconds and 41 seconds needed to give access to the user from the moment they generate their QR Code to the Node giving access to the user. The usage of MFA authentication such as password to generate QR, the QR Code and Face recognition also contributed in a more secure way of authentication. The usage of RSA on QR Code also contribute on security of the system. The evaluation of both proof-of-work and proof-of-authority consensus also shown that both consensuses are suitable for use for the proposed system. Proof-of-work consensus suitable for the proposed system where more node is involved in blockchain network while proof-of-authority consensus is suitable for the proposed system where less node in involved in blockchain network.

## REFERENCES

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].


[1] L. Collins, "Chapter 12 - Assessments and Audits," in Cyber Security and IT Infrastructure Protection, J. R. Vacca, Ed. Boston: Syngress, 2014, pp. 281–294. doi: https://doi.org/10.1016/B978-0-12-416681-3.00012-4.

[2] R. Crowder, "11 - Cyber Physical systems and security," in Electric Drives and Electromechanical Systems (Second Edition), Second Edition., R. Crowder, Ed. Butterworth-Heinemann, 2020, pp. 271–289. doi: https://doi.org/10.1016/B978-0-08-102884-1.00011-X.

[3] D. Kevin, Designing Large Scale Lans: Help for Network Designers, 1st ed. O'Reilly Media, 2001.

[4] "CAPEC - CAPEC-93: Log Injection-Tampering-Forging (Version 3.7)."https://capec.mitre.org/data/definitions/93.html (accessed Apr. 06, 2022).

[5] M. van Steen and Andrew S. Tanenbaum, Distributed Systems, 3rd ed. Leiden: Maarten van Steen, 2017.

[6] "Multi-Factor Authentication | NIST." https://www.nist.gov/itl/smallbusinesscyber/guidance-topic/multi-factorauthentication (accessed Jun. 23, 2022).

[7] D. Han, H. Kim, and J. Jang, "Blockchain based smart door lock system," in 2017 International Conference on Information and Communication Technology Convergence (ICTC), Oct. 2017, pp. 1165–1167. doi: 10.1109/ICTC.2017.8190886.

[8] U. Nadiya, M. I. Rizqyawan, and O. Mahnedra, "Blockchain-based SecureData Storage for Door Lock System," in 2019 4th International Conference on Information Technology,

Information Systems and Electrical Engineering (ICITISEE), Nov. 2019, pp. 140–144. doi:10.1109/ICITISEE48480.2019.9003904.

[9] L. de Camargo Silva, M. Samaniego, and R. Deters, "IoT and Blockchain for Smart Locks," in 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Oct. 2019, pp. 0262–0269. doi: 10.1109/IEMCON.2019.8936140.

[10] A. Shahnaz, U. Qamar, and A. Khalid, "Using Blockchain for Electronic Health Records," IEEE Access, vol. 7, pp. 147782–147795, 2019, doi:10.1109/ACCESS.2019.2946373

[11] W. Huang, "A Blockchain-Based Framework for Secure Log Storage," in 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), Aug. 2019, pp. 96–100. doi: 10.1109/CCET48361.2019.8989093.

[12] P. Satanasaowapak, W. Kawseewai, S. Promlee, and A. Vilamat, "Residential access control system using QR code and the IoT," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 4, p. 3267, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3267-3274.

[13] A. F. M. Fauzi, N. N. Mohamed, H. Hashim, and M. A. Saleh, "Development of Web-Based Smart Security Door Using QR Code System," in 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Jun. 2020, pp. 13–17. doi:10.1109/I2CACIS49202.2020.9140200.

[14] S. Shankar, J. Madarkar, and P. Sharma, "Securing Face Recognition System Using Blockchain Technology," 2020, pp. 449–460. doi: 10.1007/978-981-15-6318-8_37.

[15] A. Ismatov, V. G. Enriquez, and M. Singh, "FaceHub: Facial Recognition Data Management in Blockchain," 2021, pp. 135–153. doi: 10.1007/978-981-33-4122-7_7.