ANS 2

by Cheshan Jayathilaka

Submission date: 24-Apr-2021 02:08PM (UTC+0530)

Submission ID: 1373804776

File name: rabilities_in_process_of_offloading_mobile_apps_to_the_cloud.pdf (95.84K)

Word count: 4355

Character count: 24398

Vulnerabilities in the process of offloading mobile app data to the cloud and use of blockchain to mitigate vulnerabilities

Cheshan Jayathilaka
MSc in Cybersecurity
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
cheshanj95temp@gmail.com

***Abstract*—The app's processing power, storage, memory, and speed are significant concerns in the current app development process for Android and iOS or any other operating system. In the present day, mobile devices focus on multi-tasking and cannot focus on one application to finish their tasks. Also, these mobile devices cannot allow one application to run solo in the mobile device. As a solution for this issue, researchers and developers establish a subcategory of cloud computing called mobile cloud computing, also known as MCC. the primary purpose of this MCC is to get the burden of processing power, storage, ram to the cloud services. So the Mobile device only needs to store or process critical parts in it. This method is primarily increasing the productivity of any application.However, the solution of MCC is coming with several issues as well. Such as platform diversity, privacy and security concerns of the application, continuous connectivity, and the service's cost. This research paper mainly focuses on the privacy and security concerns and mitigation effective mitigation methods for identified problems. Moreover, to explore how a blockchain module uses to mitigate most concerning security areas of the MCC data offloading.**

The significant vulnerabilities that need to look in MCC are data privacy and access controls. This paper has proposed a system with blockchain integration as described above. With that integration, users can secure their offloading data from malicious access and data thefts also.

***Index Terms*—Mobile cloud computing, cloud, MCC, offloading, partitioning, security, privacy, Blockchain, Platform Authenticity**

I. Introduction

The SMD or Smart mobile devices are now getting where they can use resources outside of the devices. This process of using outside resources has bound with MCC as well. [1] Using MCC for SMDs applications is to go through areas such as processing, storage, and power. However, not all SMD applications are using MCC. [2] Applications such as cameras, calculators do not need extensive processing power or storage within the device. [3] Sometimes, these apps only need data transmission media such as WIFI, cellular, or NFC, to determine location. [3] However, when it comes to photo editing apps, document processing apps, object scanning apps, they need a large amount of processing power, power, and storage to complete tasks. [3]

Applications that use resources of MCC could depend on it partially or fully depending on the type of the application. Behavior depends on the batteries of the SMDs because the power of batteries is limited due to their maximum size. [1] SMD always needs to be portable since these characteristics came from the name itself, "Smart Mobile Devices." Research and developments of mobile device batteries are deficient. [4] The research and development rate of mobile device batteries is 5 percent annually. [5] This number is a defective percentage comparing with other development rates of SMDs. When applications and operating systems run on mobile devices become more complex, they need to run those OS and applications. However, compared with the current processing power, it can be observed as a minor risk. [5] However, even the powerful processors need lots of energy to run at their maximum to give the best output. So the main drawback of the mobile device is a power source. [5]

The main problem of even a powerful SMD is resource limitation to perform tasks. One of the best solutions for resource limitations is to provide unavailable amount resource from outside. The researcher has figured out that cloud computing services can be used to recover from that resource limitation. [6] [7] [8] [1] The main advantage of the current cloud service is that they provide a pay-as-go payment scheme. This enabled a cost-effective method of using cloud computing. Cyber forging also referred to the use of external resources that can take within MCC. [9]

As in the above context, some applications need to interact with the MCC. Using services of MCC to cover fulfillments of the app comes with several concerns. These concerns are distributed among several types of fields. Those fields can identify as follows. [10] [1] [11] [12]

- Platform diversity
- Privacy and security concerns of the application
- Keeping continuous connectivity
- Cost of the service
- Device architectural issues
- Low bandwidth
- Limited Energy life
- Low processing power
- Computational offloading

Even there are lots of security concerns that can identify within the offloading process and its technology. Most of the time, attackers are targeting the best possible weak point of the process. In this case, that weak point can identify as the network that mobile devices and clouds communicate, which is considered the most vulnerable spot within the whole process. There are several attempts to outcome these vulnerabilities. Most of the time, current offloading frameworks are using WIFI, cellular networks for complete offloading tasks. [13] [14] Among these kinds of conventional methods, other networks are used for mobile offloading, such as heterogeneous networks and opportunistic networks. [8] [12]

The most sophisticated and advanced network designs mainly focused on offloading codes or other information to the designated cloud services. However, almost all current systems/frameworks are not given a fine place to think about the data privacy and security transmitted to outside through a network. [15] [11] Apart from mobile OS and application vulnerabilities, data that came out from the mobile device to the outside work are in the potential to data breach any time. [11]

A scenario is like establishing a connection between mobile devices and MCC and then transferring data involving various access control approaches. In conventional access control approaches, the origin of the data offloading is to give all required permissions to their dedicated MCC. [11] In that situation, the MCC service can read, write or edit any data with that allowed perimeter. However, almost all cloud services are now tracking these permitted data streams and obtaining other personal information such as username, device types, IMEI, serial number, and OS type. [11] This can lead to privacy and security issues of that particular data and the user's personal information.

Continuous connectivity is a critical factor when devices are doing data offloading tasks along with the MCC. That is because in a situation where can be sudden network disconnection can fail the whole operation. [5] [4] [10] Some of these data offloading processes can be critical such as creating backups and uploading them into the cloud. Any disturbance to the data offloading process can b harm the quality of the experience provided by that application. [5] [4] [10] Most of the current MCC services are using centralized systems to control every aspect of it. [16] If a sudden service failure such as server downtime can fail, all the applications and mobile devices that perform app/data are offloading to fulfill their tasks. [16]

Starting with the centralized system risk, this can replace with decentralized access control systems. [17] This can be P2P networks or blockchain networks. But the P2P is not a very good solution when considering the security and privacy of that framework. [17] [18] But in the blockchain can provide the following characteristics it. [17] [18]

- Decentralized network/Framework
- Dedicated immutable ledger of transactions
- Manage, records, and log distributions to other network participants in case of illegal requests to the application
- Smart contract to perform user authentications
- Smart contract to perform user verification
- Work well with preventing complete data losses

This paper proposes a solution to mitigating network-related vulnerabilities in the mobile app offloading issues, and that solution is a blockchain-based authentication system to control access between mobile devices and the MCC while performing offloading. This system can effectively handle all access control in any network to prevent exploit network vulnerabilities or access control vulnerabilities.

II. RELATED WORK

Several solutions have been identified separately for the mobile cloud offloading and blockchain. Also, several related works combined mobile cloud offloading with blockchain implementations to secure offloading process. In [11], they have introduced a separate system to protect confidential information while mobile apps are offloading to the cloud. In this case, they have demonstrated the technique using login credentials. [11] The purpose of the system ”TinMan” is to separate confidential details from other functionalities of the protected app. More importantly, they have used a separate node for separate offloading code and personal records. [11]

In the paper [19], authors have proposed a way to identify the need for offload and offload based on remote execution rather than using computation offloading executions. The proposed system mainly focused on the factor that ”When” needs to be the offloading happens. This approach also goes the same as the [11] because they have separated confidential records while offloading. [19]

The authors of [18] have introduced a P2P network made of mobile users. This network is more of an opportunistic network from its characteristics. The network is built upon blockchain technology. This system has used a hashgraph consensus algorithm that uses the blockchain ledger to communicate between blockchain nodes and implement trust between those nodes. [18] The hashgraph uses an Asynchronous Byzantine Fault Tolerance (ABFT) security algorithm to ensure nodes are not grouping and alter the final output of the offloading process. [18] But in this research, authors of [18] have not considered the security and privacy aspects of the implemented system. The main goal here is to archive the low power consumption and better performance and practical usage of the blockchain in the system. [18]

In this paper, it is considered data privacy and security. Concerning that area of concern for mobile cloud offloading, authors of [16] have proposed a system that focuses on authorization framework that integrates with both IoT and cloud services. They have used blockchain special for this research because blockchain follows a rigorous set of rules that directly helps secure data security, privacy, and integrity. [16] Apart from data security/privacy advantages, these systems also provide low computational overhead, enhanced access control, and complete mutual authenticity. [16]

The [20] researchers have introduced a blockchain-based system that includes a data offloading mechanism using an ethereum blockchain framework and blockchain-based decentralized data storage to improve the functionality of the hosting system. [20] The primary purpose of using blockchain for an offloading mechanism is to ensure a reliable access control system within the application. This system has archived a secure system to manage the mobile application and the cloud using the blockchain-based mechanism. [20] In the front end of the application, they have used critical public infrastructure to start the protection from the beginning of the application data flow. [20]

Also in the [20] [21] has a system to introduce a better access control framework that can use when mobile cloud offloading happening. Generally, this framework manages the access controls between the device and the designated cloud service provider that provides mobile cloud features. The introduced framework does the identifying malicious activities such as malicious logins as well. [21]

III. MOTIVATION

After careful analyses of related works to the domain, it was clear that the security and the privacy of the data was not always the priority in new mobile cloud offloading frameworks but the performance and the usage of decentralized storage management. But the privacy and security of the data in the network when mobile clod offloading happens is critical. Because attackers might exploit vulnerabilities of the network and bypass data sending with the process of data offloading, the existing framework can always have cracked somehow and get the transferring data through the network to the cloud. On the other hand, Blockchain technology is immutable, cryptographically safe, flexible, and easy to use. [20] [16] [17] [21] This project's primary goal is to analyze current achievements in blockchain and implement them into the mobile cloud offloading process while preserving efficiency.

IV. PRELIMINARIES

This paper has used a blockchain framework to mitigate mobile cloud offloading network vulnerabilities. There are several blockchain frameworks or platforms available to develop any application, such as bitcoin and ethereum. [22] [23] In this paper, ethereum has been chosen as the blockchain platform to create the required applications. The reason to select ethereum as the blockchain for this application is to have good community support and resources to resolve any problem during developments. Also, they have updated documentation as well. [23] Another reason is to use ethereum as the base blockchain platform because this blockchain is supported by almost all add-ons and several mainstream programming languages. [23] The ethereum blockchain supports building any open-source or commercial projects that need to integrate with blockchain. [20] [23]

Blockchain frameworks are contained two critical parts that can identify a block of data transactions and smart contracts. [20] [23] In here transaction refers to cryptographically signed instructions from accounts. A block includes several of these transactions in it. Each of these blocks has a link to the previous block. [20] [24] With this characteristic, blockchain can preserve data integrity even there is a change to one block. Moreover, to protect data integrity and secure data, miners use an algorithm called ”Proof of Work” to achieve said target. [20] [25]

On the other hand, the smart contract is a specific predefined program in the blockchain. [26] This is an ethereum account in this project. The user does not control these smart contracts. But they can send traction all over the blockchain network. Using these smart contracts, managing users can define rules for different applications and run them automatically. [20] [21] [26]

V. SYSTEM ARCHITECTURE

As the system's starting point, the offloading mobile application and blockchain mobile client can be introduced. The application that performing the mobile cloud offloading has uploading requests. This unique request can assign unique IDs and those that can be used as part of the blockchain address. [20] Also, we can use data filed ID as one of the parts of the blockchain address. With both parts, blockchain can identify that address as a unique address. These address parts can declare as follows.

- Request ID as,

$$R_{ID}$$

- Data filed ID as,

$$D_{ID}$$

The reason to use data field ID instead of using full details on the data filed because it not feasible to use relatively great information in the blockchain. that will occur lots of transaction and user will cost for that. [18] If there is any time to retrieve data filed data, it is feasible to use it via a data store—; preferably the mobile cloud service. [18] [20] To secure the blockchain addresses more, we can use blockchain manager. Only the manager has access to the blockchain address, restricting access to restricted records on the mobile cloud via the blockchain. [20] Suppose any user knows the exact address of a request or data filed value, that user can access the cloud records for that particular address. The blockchain mangers duty is to control all user transactions and protect the mobile cloud from offloading from malicious transactions. [20]

Next, the smart contracts are defining all access controls that need to operate within the blockchain. For that, it will use the contract address and the Application Binary Interface. Smart contracts can then identify all transactions coming through and give or deny access to requesting users to perform offload requests. [20] [21] [26] Any implemented smart contract has access to any blockchain component such as transactions, minors, and nodes. Also, these smart contracts are supposed to be the core of the blockchain-based mobile cloud offloading client. [20] [21]

In Transaction records, it is a combination of Request ID and Data field ID as previously introduced. To make a data request, the requester needs to provide that R$_{ID}$ and D$_{ID}$ to the transaction. Otherwise, blockchain won't create a transaction for that request. [20] Before sending this transaction record through the blockchain network, it is also signed with the requester's private key. When this data request is decrypting by the cloud endpoints, it will use the public provided by the user.

The nodes, also called blocks have the following metadata fields in them. The hash value of the block, Hash value from the previous block, Merkle root (This is the storing structure for all nodes in the blockchain), Nonce; this is provided by the previously mentioned Proof of Work algorithm and the timestamp that the node created in the blockchain. [20] [21]

Proposing data uploading process is as follows. The beginning of the uploading process is the mobile client that specially builds to interact with the blockchain network. For a user to use the data uploading feature, that user needs to have an ethereum account and logged in to the account through the mobile client. Then the mobile client can handle the upload request that is coming through other applications. The first step is that the creation of the data uploading transaction. This transaction is responsible for carrying out has values, requester ID, sender ID, data ID digital signature, and the timestamp. This transaction needs to be routed through the manager that was previously described. The manager validates the requester ID, recipient ID, and digital signature to allow the transaction to transfer through the blockchain network. Then it is sent over the external network to the cloud. While performing described steps by the manager [20], it also sends a separate message to implemented smart contracts, and these smart contracts are watching for the validation of the transaction and deliver the output of the confirmation to the mobile client. After delivering the message from the smart contract, the mobile client can initialize the data uploading to the cloud. [26]

Once the transaction(upload request) is approved and smart contracts review the transaction validations [20] [26], the mobile app can then initialize the data uploading process. In this process, sending data blocks are first being encrypted with the public of the manager. After that, the data block will send to the cloud using the user index. [23] (User index needs to identify the correct cloud location and store the data in separate areas for separate users.) As a suggestion to the application process verification, we can also add a hash function to the cloud endpoint. [17] After being upload to the cloud, a hash can be generated from that particular data file and store in a different location. That way, it added a secondary layer of data protection to detect data integrity. [17]

This implantation aims to achieve data security in mobile cloud offloading and the efficiency of the process; furthermore, the following areas are also identified as the secondary goals that need to accomplish with the proposed system.

- The application should be able to identify and authenticate authorized users.
- The application should be able to communicate with secondary applications with high efficiency to handle their uploading requests.

- The application should be lightweight with minimum impact on the mobile device performance.
- The application should achieve maximum data privacy and integrity since it works with secondary applications and their sensitive information.

VI. DISCUSSION

In the proposed mobile cloud offloading solution, the primary goal is to protect data integrity and data privacy. Under data privacy, several areas have been concerned. Those areas include security vulnerabilities that can exploit in the network, cloud, and blockchain. As the mitigation plan in this solution, the blockchain network has been used. Smart contracts and the manager are continually on alert about all malicious activities and all types of access that happen within the blockchain network. Also, all blockchain nodes are connected. [20] [22] [23] So each of these depends on another. that makes change a node and try to violate data privacy or try to gain access to the network is impossible because every node is checking the digital signature of the other node. If there any mismatching between the original value and the present one, that node will be eliminated. [20] [22] [23] They can be missing one node without causing the total data loss since every node holds a copy of the data being transferred.

Also, in the point of data integrity, only managers can control every transaction. The users who are using the mobile client are not authorized to perform data edits nor transaction edits. Another point is that every one of these transactions is digitally signed and encrypted. Which will also work toward the protect data integrity.

A decentralized system to control the mobile offloading process in the device will protect users from a single point of failure. [20] All used decentralized technologies like ethereum networks are operating as P2P networks. That's the reason every user needs to have an account to participate in blockchain-based mobile cloud offloading. If there any failures, other peers on the network can manage to complete the given task. But this only applies to losses occurring on the uploading application of the mobile client. Device-specific problems cannot be tolerated.

VII. CONCLUSION

This paper proposes a new solution to mitigate network vulnerabilities in mobile cloud offloading by interacting the service with blockchain technology. The paper also mentioned identified critical vulnerabilities while offloading data to the mobile cloud service, such as network exploitation and data theft, data integrity issues, and unauthorized access to applications. The primary purpose of this paper is to introduce an all-in-one solution to cover all aspects of said issues. The result is the decentralized blockchain network-enabled mobile cloud data offloading environment that works with other mobile applications.

This solution specially used smart contracts to manage access controls. Also have used a separate manager role to oversee the access controls to support the process. Then there are custom-designed transactions that enable more security through encryption, digital signatures, and has values. These transactions also allow lightweight data packs that directly impact the blockchain and ultimately reduce server request costs.

Also, enabling an extra layer of security, the application uses a hash calculating after data uploading to the cloud. It helps when doing a data integrity checkup. To help the very purpose, blockchain also provides data integrity features and backup features to the proposed application. The combination of blockchain technology and mobile cloud offloading is can successfully integrate, and the proof of that is the proposed system in this paper. Future research is required to conduct more studies on efficient communications between blockchain clients and secondary apps.

REFERENCES

[1] K. Akherfi, M. Gerndt, and H. Harroud, “Mobile cloud computing for computation offloading: Issues and challenges,” Appl. Comput. Informatics, vol. 14, no. 1, pp. 1–16, 2018, doi: 10.1016/j.aci.2016.11.002.

[2] L. J. Yao, E. Ahmed, M. Shiraz, and A. Gani, “Application Partitioning Approaches for Mobile Cloud Computing: Review , Issues and Challenges,” pp. 1–38, 2015.

[3] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang, “Enabling automatic offloading of resource-intensive smartphone applications,” Tech. report. United States Purdue Univ., 2011.

[4] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, “A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing,” IEEE Commun. Surv. Tutorials, vol. 15, no. 3, pp. 1294–1313, 2013, doi: 10.1109/SURV.2012.111412.00045.

[5] A. Aliyu et al., “Mobile Cloud Computing: Taxonomy and Challenges,” J. Comput. Networks Commun., vol. 2020, 2020, doi: 10.1155/2020/2547921.

[6] B. K. Bhargava, “Introduction to Mobile-Cloud Computing.” Department of Computer Sciences,Purdue University, 2013, [Online]. Available: https://www.cs.purdue.edu/homes/bb/cloud/MCC.pptx.

[7] M. Alizadeh and W. H. Hassan, “Challenges and opportunities of mobile cloud computing,” 2013 9th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2013, pp. 660–666, 2013, doi: 10.1109/IWCMC.2013.6583636.

[8] H. Zhou, H. Wang, X. Li, and V. C. M. Leung, “A Survey on Mobile Data Offloading Technologies,” IEEE Access, vol. 6, pp. 5101–5111, 2018, doi: 10.1109/ACCESS.2018.2799546.

[9] H. M. Kyi, “Overview of Mobile Computation Offloading in Mobile Cloud Computing Environment.”

[10] A. S. Al-Ahmad, H. Kahtan, F. Hujainah, and H. A. Jalab, “Systematic Literature Review on Penetration Testing for Mobile Cloud Computing Applications,” IEEE Access, vol. 7, pp. 173524–173540, 2019, doi: 10.1109/ACCESS.2019.2956770.

[11] Y. Xia et al., “Tinman: Eliminating confidential mobile data exposure with security oriented offloading,” Proc. 10th Eur. Conf. Comput. Syst. EuroSys 2015, 2015, doi: 10.1145/2741948.2741977.

[12] X. Gu, K. Nahrstedt, A. Messer, I. Greenberg, and D. Milojicic, “Adaptive offloading inference for delivering applications in pervasive computing environments,” Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2003, pp. 107–114, 2003, doi: 10.1109/percom.2003.1192732.

[13] D. Wasil, O. Nakhila, S. S. Bacanli, C. Zou, and D. Turgut, “Exposing vulnerabilities in mobile networks: A mobile data consumption attack,” arXiv, pp. 2–6, 2018.

[14] D. Liu, L. Khoukhi, and A. Hafid, “Data offloading in mobile cloud computing: A Markov Decision Process approach,” IEEE Int. Conf. Commun., pp. 0–5, 2017, doi: 10.1109/ICC.2017.7997070.

[15] Y. Duan, M. Zhang, H. Yin, and Y. Tang, “Privacy-preserving offloading of mobile app to the public cloud,” 7th USENIX Work. Hot Top. Cloud Comput. HotCloud 2015, 2015.

[16] M. Tahir, M. Sardaraz, S. Muhammad, and M. S. Khan, “A lightweight authentication and authorization framework for blockchain-enabled IoT network in health-informatics,” Sustain., vol. 12, no. 17, 2020, doi: 10.3390/SU12176960.

[17] T. T. Kuo, H. E. Kim, and L. Ohno-Machado, “Blockchain distributed ledger technologies for biomedical and health care applications,” J. Am. Med. Informatics Assoc., vol. 24, no. 6, pp. 1211–1220, 2017, doi: 10.1093/jamia/ocx068.

[18] V. Hassija, V. Chamola, V. Gupta, and G. S. S. Chalapathi, “A Blockchain based Framework for Secure Data Offloading in Tactile Internet Environment,” 2020 Int. Wirel. Commun. Mob. Comput. IWCMC 2020, no. June, pp. 1836–1841, 2020, doi: 10.1109/IWCMC48107.2020.9148559.

[19] I. Giurgiu et al., “Dynamic Software Deployment from Clouds to Mobile Devices To cite this version: HAL Id: hal-01555562,” pp. 0–20, 2017.

[20] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, “Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems,” IEEE Access, vol. 7, pp. 66792–66806, 2019, doi: 10.1109/ACCESS.2019.2917555.

[21] Q. XIA, E. B. SIFAH, J. G. KWAME OMONO ASAMOAH, X. DU, and M. GUIZANI, “MeDShare: Trust-less Medical Data Sharing Among,” IEEE Access, vol. 5, pp. 1–10, 2017.

[22] Bitcoin, “Bitcoin,” Bitcoin, 2021. https://bitcoin.org/en/ (accessed Apr. 23, 2021).

[23] Ethereum, “Ethereum,” Ethereum, 2021. https://ethereum.org/en/ (accessed Apr. 23, 2021).

[24] Ethereum, “TRANSACTIONS,” Ethereum, 2021. https://ethereum.org/en/developers/docs/transactions/ (accessed Apr. 23, 2021).

[25] R. Wu, G. J. Ahn, and H. Hu, “Secure sharing of electronic health records in clouds,” Collab. 2012 - Proc. 8th Int. Conf. Collab. Comput. Networking, Appl. Work., pp. 711–718, 2012, doi: 10.4108/icst.collaboratecom.2012.250497.

[26] Ethereum, “INTRODUCTION TO SMART CONTRACTS,” Ethereum, 2021. https://ethereum.org/en/developers/docs/smart-contracts/ (accessed Apr. 23, 2021).

ANS 2

ORIGINALITY REPORT

2%	1%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Dinh C. Nguyen, Pubudu N. Pathirana, Ming Ding, Aruna Seneviratne. "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems", IEEE Access, 2019 Publication	1%
2	ir.kdu.ac.lk Internet Source	<1%
3	"A Compression based Energy Efficient Protocol for Mobile Cloud Computing", International Journal of Engineering and Advanced Technology, 2019 Publication	<1%
4	101blockchains.com Internet Source	<1%
5	Folino, G., and F.S. Pisani. "Automatic offloading of mobile applications into the cloud by means of genetic programming", Applied Soft Computing, 2014. Publication	<1%
6	www.bth.se Internet Source	<1%

7	arxiv.org Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		