EasyChair Preprint

№ 2923

# Verification Of Content Against Dithering Auditors Using CPVPA

V. Vignesh, L. Tanush Navneeth, S. Vignesh and N. Surya



March 11, 2020

# VERIFICATION OF CONTENT AGAINST DITHERING AUDITORS USING CPVPA

V Vignesh[1] IV Year UG
L Tanush Navneeth[2] IV Year UG
S Vignesh[3] IV Year UG
N.Surya[4] Assistant Professor
vignesh.v1.2016.cse@rajalakshmi.edu.in [1]
tanushnavneeth.l.2016.cse@rajalakshmi.edu.in [2]
vignesh.s.2016.cse@rajalakshmi.edu.in [3]
surya.n@rajalakshmi.edu.in [4]
Dept of CSE, Rajalakshmi Engineering College, Chennai

---

**ABSTRACT**

The deployment of cloud storage services has significant benefits in managing data for users. However, it also causes many security concerns, and one of them is data integrity. Public verification techniques can enable a user to employ a third-party auditor to verify the data integrity on behalf of her/him, whereas existing public verification schemes are vulnerable to procrastinating auditors who may not perform verifications on time. Furthermore, most of public verification schemes are constructed on the public key infrastructure (PKI), and thereby suffer from certificate management problem. In this paper, we propose the first certificate less public verification scheme against procrastinating auditors by using blockchain technology. The key idea is to require auditors to record each verification result into a blockchain as a transaction. Since transactions on the blockchain are time-sensitive, the verification can be time-stamped after the corresponding transaction is recorded into the blockchain, which enables users to check whether auditors perform the verifications at the prescribed time. Moreover, CPVPA is built on certificate less cryptography, and is free from the certificate management problem. We present rigorous security proofs to demonstrate the security of CPVPA, and conduct a comprehensive performance evaluation to show that CPVPA is efficient.

## I. INTRODUCTION

With cloud storage services, users outsource their data to cloud servers and access that data remotely over the Internet. These services provide users an efficient and flexible way to manage their data, while users are free from heavy local storage costs. Although users enjoy great benefits from these services, data outsourcing has also incurred critical security issues. One of the most important security concerns is data integrity. Unlike traditional data management paradigm, where users store their data locally, users would not physically own their data once having outsourced the data to cloud servers. Therefore, users are always worried about the data integrity, i.e., whether the outsourced data is well maintained on cloud servers. The integrity of outsourced data is being put at risk in practice. For example, the cloud servers may always conceal incidents of data corruption for good reputation, or may delete a part of data that is never accessed to reduce the storage costs. Furthermore, an external adversary may tamper with the outsourced data for financial or political reasons. Therefore, the integrity of outsourced data should be verified periodically. The verification can be performed by the users themselves. Public verification techniques enable users to outsource the data integrity verification to a dedicated third-party auditor. The auditor periodically checks the data integrity, and informs the users that the data may be corrupted

once the checking fails. In most of public verification schemes, the auditor is assumed to be honest and reliable. If the auditor is compromised, these schemes would be invalidated. For example, an irresponsible auditor may always generate a good integrity report without performing the verification to avoid the verification costs. In such a way, the auditor is virtually non-existent. Furthermore, a malicious auditor may collude with the cloud servers to generate a bias verification result to deceive the users for profits. To ensure the security in the case that the auditor is compromised, the users are required to audit the auditor's behaviors after each verification and the auditor records the information used to verify the data integrity, which enables the user to audit the validity of the auditor's behaviour.

**Java framework**

Java is a programming language originally developed by James Gosling at Sun Micro systems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications. The java framework is a new platform independent that simplifies application development internet.. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to data centre's , game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

**Java servlets**

Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server. Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server so these are safe and portable.

Unlike applets they do not require support for java in the web browser. Unlike CGI, servlets don't use multiple processes to handle separate request. Servlets can be handled by separate threads within the same process. Servlets are also portable and platform independent.

A web server is the combination of computer and the program installed on it. Web server interacts with the client through a web browser. It delivers the web pages to the client and to an application by using the web browser and the HTTP protocols respectively.

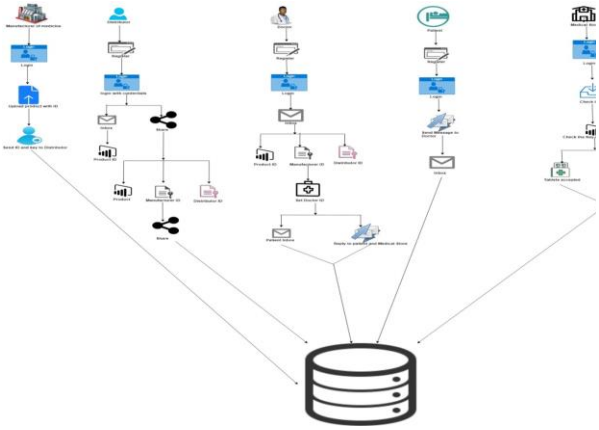
The define the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using File Transfer Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

## II. LITERATURE SURVEY

In most of existing public verification schemes, auditors are assumed to be honest and reliable. This means that the auditor would honestly follow the prescribed schemes, and performs the verification reliably. These schemes cannot resist malicious auditors. The most trivial attack a malicious auditor can perform is that it always generates a good integrity report without burden. To thwart such attacks, the user is able to audit the auditor's behaviour at the end of each epoch. The auditor colludes with the cloud server, and always generates bias challenging messages such that only

the data blocks which are well maintained are verified, this avoids revealing the data corruption. This is the existing system we have against procrastinating auditors.

## III. ARCHITECTURE DIAGRAM



## IV. METHODOLOGY

**User interface design**

This is the first module of our project. The important role for the user is to move login window to user window. This module has been created for the security purpose. In this login page we have to enter login user id and password. It will check whether username and password matches or not (valid user id and valid password). If we enter any invalid username or password we can't enter into login window to user window and it will show an error message. So we are preventing from unauthorized user entering into the login window to user window. It will provide a good security for our project. Since server contains user id and password it check the authentication of the user. It improves the security and prevents from unauthorized user entering into the network. In our project we are using JSP for creating design. Here we validate the login user and server authentication.

**Data owner request for key**

Here data owner will register and login and requests a key to upload the files. With the use of key only he can upload a file in the cloud. Data owner will request for key to the key center.

**Key center generates the key for the owner**

In this module, key center checks the data owner list or profile; if data owner is a valid person then the key center generates a key for uploading a file. Otherwise it can't generate a key.

**Data owner uploads file with that key**

In this module, data owner will login and upload some files i.e.pdf or a text file. The uploaded file gets encrypted and it is stored in the database. While uploading a file, key is also stored there.

**Send file for auditing**

Uploaded file will be sent to the auditor for checking purpose. In this module separate auditing team will be there for checking and correcting the files. All uploaded files are sent here for auditing.

**Auditor checks the file**

Auditor checks all files uploaded by all data owner with the file key. Auditor can block the file when there is an incorrect file or an incorrect user.

**Data user request for file**

Here data user will register he can login and request for some files uploaded by the data owner. Data user can view the all files uploaded in the database. Files will be viewed as a encrypted text to the data user.If they click request button it will be sent to the admin.

**Admin accepts the request**

Admin receives notification after getting logging in. here there will be set of request which is sent by other data user. If admin accept than , key will be sent for downloading the file. The key will be sent to the requested data user for downloading file with acceptance notification. Otherwise it will be rejected.

**Admin maintain the data**

Here the admin will login and he can view the files uploaded by data owner, and Admin manages all files uploaded by all data owners the file key. Admin maintains the files in the database.

**Data user download the file**

Here the response notification will be received with the key. The file key will be sent by admin in the backend for downloading the file. When he downloads the file it prompts for entering the key. If it is matches it will be downloaded otherwise key will be wrong.

## V. CONCLUSION

In this paper, we have proposed a certificate less public verification scheme against the procrastinating auditor, namely CPVPA. CPVPA utilizes the on-chain currencies, where each verification performed by the auditor is integrated into a transaction on the blockchain of on-chain currencies. Furthermore, CPVPA is free from the certificate management problem. The security analysis demonstrates that CPVPA provides the strongest security guarantee compared with existing schemes. We have also conducted a comprehensive performance analysis, which demonstrates that CPVPA has constant communication overhead and is efficient in terms of computation overhead.

## VI. FUTURE ENHANCEMENT

For the future work, we will examine how to develop CPVPA on other blockchain frameworks. Since the principle downside of verifications of work (PoW) is the vitality utilization, developing CPVPA on other blockchain frameworks (e.g., proofs-of-stake-based blockchain frameworks) can spare vitality. Be that as it may, it requires an explained structure to accomplish the same security ensure while guaranteeing the high productivity. These remaining parts an open research issue that ought to be further investigated. We will likewise examine how to use blockchain innovation to improve distributed storage frameworks as far as security, execution, and usefulness.

## REFERENCES

[1] J. Yu, K. Wang, D. Zeng, C. Zhu, and S. Guo, “Privacy-preserving data aggregation computing in cyber-physical social systems,” ACM Transactions on Cyber-Physical Systems, vol. 3, no. 1, p. 8, 2018.

[2] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, “Querying in internet of things with privacy preserving: Challenges, solutions and opportunities,” IEEE Network, vol. 32, no. 6, pp. 144–151, 2018.

[3] J. Li, H. Ye, W. Wang, W. Lou, Y. T. Hou, J. Liu, and R. Lu, “Efficient and secure outsourcing of differentially private data publication,” in Proc. ESORICS, 2018, pp. 187–206.

[4] L. Zhong, Q. Wu, J. Xie, J. Li, and B. Qin, “A secure versatile light payment system based on blockchain,” Future Generation Computer Systems, vol. 93, pp. 327–337, 2019.

[5] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, “Enabling efficient and geometric range query with access control over encrypted spatial data,” IEEE Trans. Information Forensics and Security, vol. 14, no. 4, pp. 870–885, 2019.

[6] K. Yang, K. Zhang, X. Jia, M. A. Hasan, and X. Shen, “Privacypreserving attribute-keyword based data publish-subscribe service on cloud platforms,” Information Sciences, vol. 387, pp. 116– 131, 2017.

[7] W. Shen, B. Yin, X. Cao, Y. Cheng, and X. Shen, “A distributed secure outsourcing scheme for solving linear algebraic equations in ad hoc

clouds," IEEE Trans. Cloud Computing, to appear, doi: 10.1109/TCC.2016.2647718.

[8] H. Yang, X. Wang, C. Yang, X. Cong, and Y. Zhang, "Securing content-centric networks with content-based encryption," Journal of Network and Computer Applications, vol. 128, pp. 21–32, 2019.

[9] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS, 2009, pp. 355–370.

[10] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," Information Sciences, vol. 472, pp. 223–234, 2018.

[11] K. Wang, J. Yu, X. Liu, and S. Guo, "A pre-authentication approach to proxy re-encryption in big data context," IEEE Transactions on Big Data, 2017, to appear, doi. 10.1109/TBDATA.2017.2702176.

[12] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," IEEE Transactions on Dependable and Secure Computing, to appear, doi. 10.1109/TDSC.2018.2791432.

[13] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, and X. Zhang, "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," IEEE Trans. Information Forensics and Security, vol. 12, no. 3, pp. 676–688, 2017.

[14] H. Shacham and B. Waters, "Compact proofs of retrievability," Journal of Cryptology, vol. 26, no. 3, pp. 442–483, 2013.

[15] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 9, pp. 1717–1726, 2013.

[16] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of ACM CCS, 2007, pp. 598–609.