IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A THREE LAYER PRIVACY PRESERVING STORAGE SCHEME FOR PROVIDING SECURITY

[1]Ch.Deekshitha, [2]K. Amrutha Sarvani, [3]K. Sadhana, [4]K. Chaturya, [5]Mrs.M. Mamatha Laxmi,

[1] Student, [2]Student, [3]Student, [4]Student, [5] Assistant Professor

[1,2,3,4 & 5]Department of Computer Science and Engineering,

[1,2,3,4 & 5]Vignan's Institute of Engineering for women, Visakhapatnam, India

Abstract: In current scheme, user's data is completely stored in data servers. While storing the info in main server, users lose their right of control on data and face privacy leakage risk. Traditional privacy protection schemes usually support encryption technology. But these forms of methods cannot effectively resist attack within main server. So as to unravel this problem, we propose a framework which can take full advantage of information storage and also protect the privacy of information. We use Hash-Solomon code algorithm to divide data into different parts, so that these parts of data can be stored in local machine and main server and secondary server so as to guard privacy. To support computational intelligence, this algorithm can compute the distribution stored in main server, secondary server and native machine respectively.

Index Terms **– Data storage, Privacy, Hash-Solomon code.**

I. INTRODUCTION

Since the 21st century, technology has developed rapidly with the development of network bandwidth. The degree of users is rising geometrically. User's requirement can't be satisfied by the capacity of local machine anymore. Therefore, people try and see new methods to store their data. To pursue more powerful storage capacity, a growing number of users selects the server storage. However, storing data in server leads to lot of security problems. The privacy problem is especially significant among those security issues.

User (owner of data) upload the information to the server directly. Subsequently, the server provider will now act like a user (owner) to manage the information. In consequence, users don't actually control the physical storage of their data, which ends up within the separation of ownership and management of information. The server provider can freely access and search the information stored within the server. Meanwhile, the attackers can even attack the server to get the users information. These two cases both make users fell into danger of data leakage. Traditional secure server storage solutions for the above problems are usually that specialize in access restrictions or encryption. These methods can actually eliminate most a part of these problems. However, all of those solutions cannot solve the interior attack. Well, irrespective of how the algorithm improves. Therefore, we propose a TLS (Three Layer Scheme)

which uses Hash-Solomon code. In our scheme, we split user's data into three parts and save them separately in two different servers (main server and secondary server) and user's local machine. Here main server is considered as owner1 and secondary server as owner2. So for the given data along with original owner, we have 3 different owners. Besides, counting on the property of the Hash-Solomon code, the scheme can make sure the original data can't be recovered by partial data. On the other hand, using Hash-Solomon code will produce a little of redundant data blocks which can be utilized in decoding procedure. By reasonable allocation of the information, our scheme can really protect the privacy of user's data. The Hash-Solomon code needs Computational Intelligence (CI). Paradigms of CI are successfully utilized in recent years to handle various challenges. For instance, the matter in wireless sensor networks (WSN's) field. CI provides adaptive mechanisms that exhibit intelligent behavior in complex and dynamic environments like WSN's. Thus with the profit of CI methods, our scheme can provide a better privacy protection from interior.

II. LITERATURE SURVEY

In order to resolve the privacy issue, paper [11] proposed a privacy-preserving and duplicate deterrence (CBIR) scheme using encryption and watermarking techniques. The scheme they proposed is secure and might resist possible attacks. [9] Fu et al propose a content-aware search scheme, which could make semantic search smarter. The experiment results show that their scheme is efficient.

In paper [12], Hou, Pu, and Fan consider that within the standard situations, user's data is stored through CSP (Cloud Server Providers), whether or not CSP is trustworthy, attackers can still get user's data if they control the cloud storage. To avoid this problem, they propose an encrypted index structure supported an asymmetric challenge-response authentication mechanism.

In paper [10], Wei et al., implies that most of the previous works on security focus on the storage security rather than taking the computation security into considering together. Thus, they propose a privacy cheating discouragement and secure computation auditing protocol, which may be a primary protocol bridging secure storage and secure computation auditing in cloud and achieves privacy cheating discouragement by designated verifier signature, batch verification and probabilistic sampling techniques.

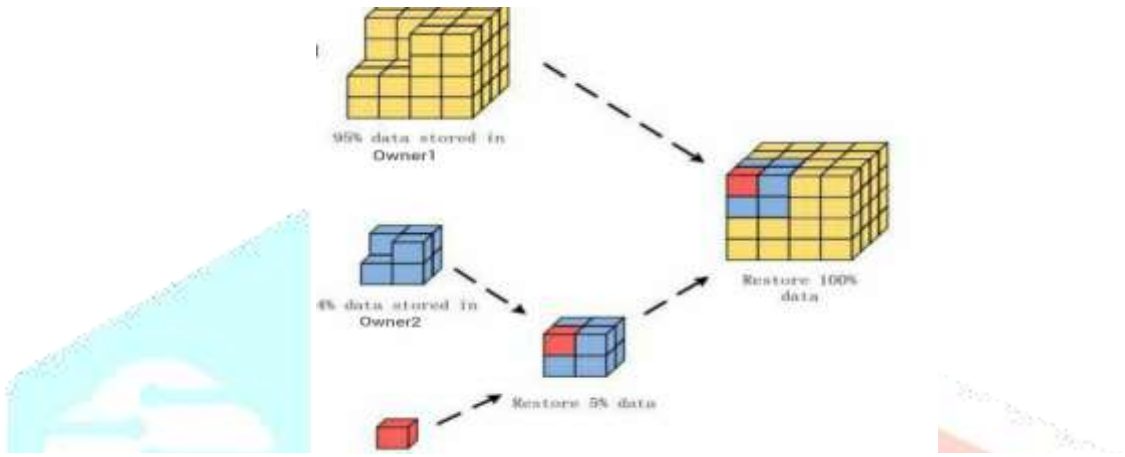
Also, in paper [2], C. Wang et al, "privacy-preserving public auditing for data storage security", users can remotely store their data in large servers so on enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources.

All these techniques discussed so far are the improvements in privacy protection in large storages in different aspects. Many of them use a diffusion of encryption policies in numerous positions. Others solve the privacy problems with the help of auditing or building their own security framework. Once the CSP is untrusted, all of these schemes are invalid. They cannot resist internal attacks or prevent the CSP from selling user's data to earn illegal profit. The private data are decoded once malicious attackers latch on regardless of how advanced the encryption technologies are because the user's data was integrally stored during an outsized server. Therefore, we propose a novel secure storage scheme in this paper. Here, the user's file is divided into 3 parts and stored at different locations to achieve a high degree of privacy protection of data.

III. PRELIMINARY

In order to protect the user's privacy, we propose a TLS framework. The TLS framework can give the user a certain power of management and effectively protect the user's privacy. In our scheme, the user's data is divided into three different size parts with encoding technology. Each of them will have a part of the key information to provide confidentiality. The three parts of data will be stored in the two servers (main server and secondary server) and the user's local machine according to the order from large to small. Here main server is considered as owner1 and secondary server as owner2. By this method, the attacker cannot recover the user's original data even if he gets all the data from a certain server.

Figure 1 diagram of division of memory blocks



The architecture includes three layers, the main server, second server, and the local machine. Each server saves a certain part of the data. For example, let 1% of encoded data be stored in the machine. Then upload the remaining 99% data to the owner2 (secondary server). Secondly, on the data which comes from the user's machine there will be about 4% of the data stored in the owner2 server and then upload the remaining data to the owner1 (main server). The above operations are based on Hash-Solomon code. After being encoded by the Hash-Solomon code, the data will be divided into 'k' parts and generates redundant data. Hash-Solomon code has such property. In these 'k+m' parts of data, if someone has at least 'k' parts, he can recover the complete data with less than k parts of data. According to this property, in our scheme we let no more thank-1 parts of data be stored in the higher server, which has larger storage capacity and lets the remaining be stored in the lower server. Thus, we can ensure the privacy of user's data.

Let us assume that, we want to save r% data on the secondary server.

- **Invalid Ratio:** The ratio of number of data blocks stored in lower server to the number of data blocks stored in the upper server.
- **Maximal invalid ratio:** The ratio of invalid data to the number of data blocks when the upper server can just recover the complete data by the data blocks stored in them. This can be expressed as $m \div k + m$

In order to avoid the upper server recovers the data, the value of k, m, and r must satisfy the relationship.

$$m \div k + m \leq (k + m \div k) \times r \quad \text{----------eq 1}$$

Though functional transformation, the relationship between k, m and r can be expressed as equation 2, we can see that if '**r**' is determined, '**k**' can be expressed by m. so, we can only consider the ratio and the number of data blocks when we use our scheme.

$$k = (m - 2mr) + \sqrt{(2mr - m)2 - 4m2r2} \div 2r \text{----------eq2}$$

Here, k is the number of blocks after divided

m is the number of redundant blocks

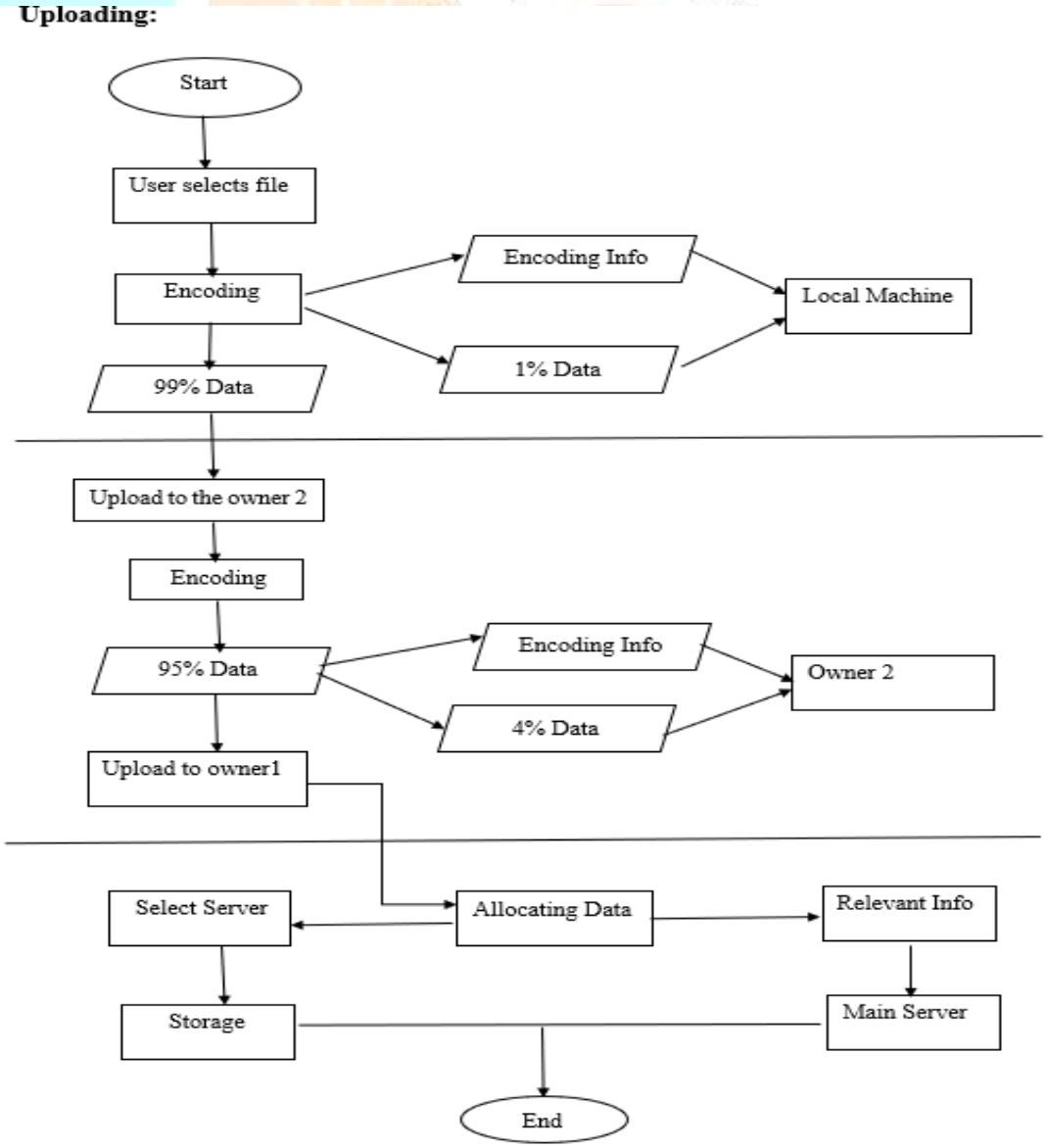
r is the storage ratio of different servers.

IV. IMPLEMENTATION

4.1 STORED PROCEDURE

Firstly, the user's file is encoded with Hash-Solomon code, then the file is divided into several data blocks and therefore the system also will feedback the encoding information simultaneously. Assuming that 1% of information blocks are stored locally. The remaining 99% of information blocks are uploaded to the owner2. Secondly, after receiving the 99% data blocks from the user's machine, these data blocks are encoded with Hash-Solomon again.

Figure 2 diagram of Stored Procedure



These data blocks are divided into smaller databases and generate new encoding information. Similarly, assuming that 4% of information blocks and encoding information are stored in the owner2 server. The remaining 95% of information blocks are uploaded to the owner1 server. Thirdly, after the owner1 server received the information blocks from the owner2 server, these data blocks are distributed by the owner1 server. Finally, the storage procedures end when all the related information is recorded in numerous servers.

4.2 DOWNLOAD PROCEDURE

Main server receives user's request and then integrates the data in different distributed servers. After integration, main server sends the 95% data to the owner2. The owner2 server receives the data from the owner1. Next the owner2 combines the received data from owner1 with the 4% data blocks of owner2 and the encoding information, so we can recover 99% data. Then the owner2 returns the 99% data to the user. Finally, the original owner of the data, combines the this 99% of received data with the 1% data available in the local machine, decodes it and makes its available to the requested user.

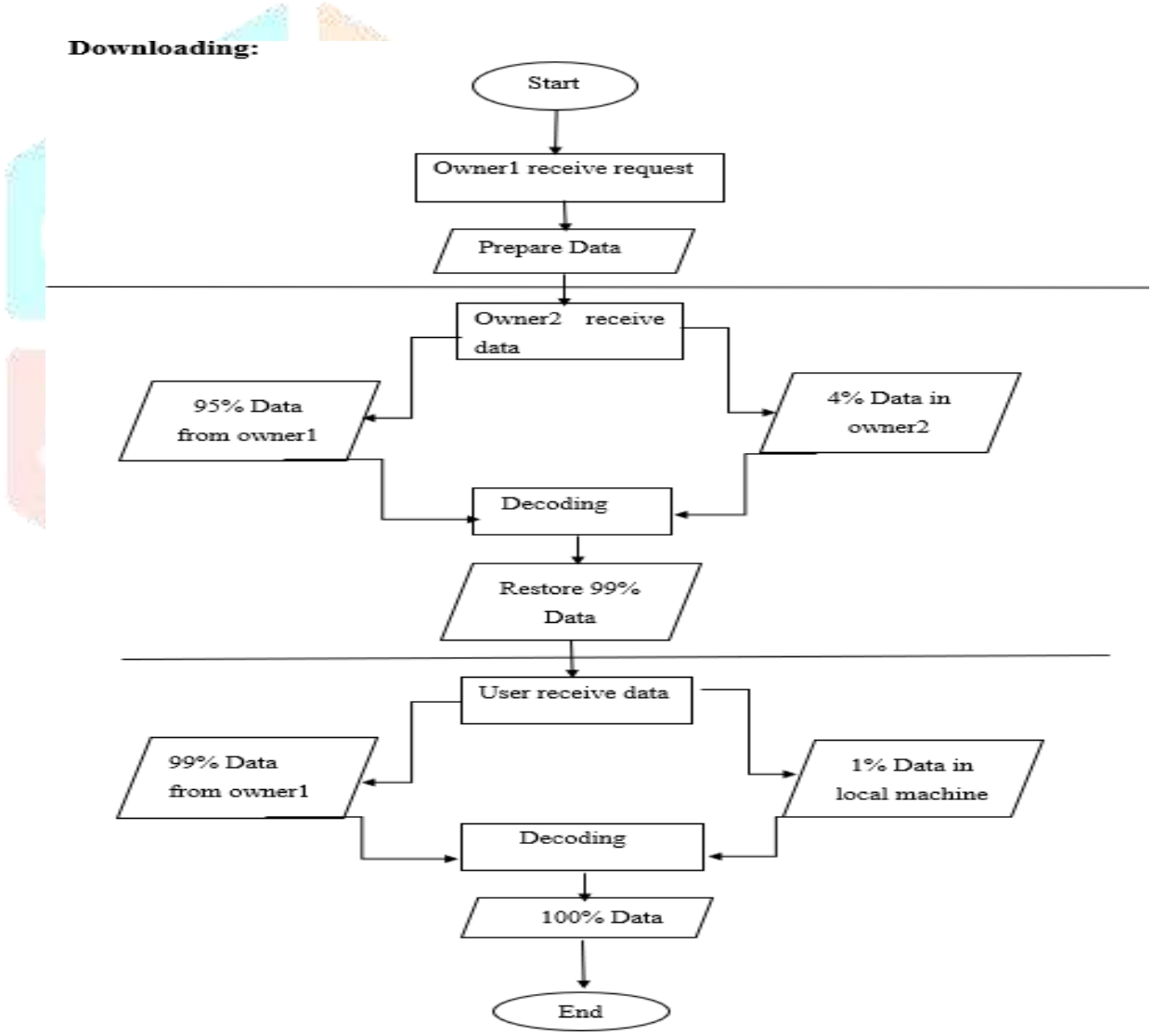


Figure 3 diagram of Download Procedure

V. THEORETICAL SAFETY ANALYSIS

This will prove that storage structure can really improve the capacity of privacy protection. Here, we are using Hash-Solomon code algorithm. The Hash-Solomon encoding process is actually a matrix operation as shown in figure 4. Firstly, we should do mapping transformation on the file which is prepared to be stored, so that each word of the file corresponds to a number in GF(2ω). After mapping transformation, we get file matrix O. Secondly, we do hash transform on matrix O and get matrix X the multiplication will generate '**k**' data blocks X1 to X6 and '**m**' redundant blocks C (**k**=6, **m**=1).

The attacker can hardly crack the encoding matrix but the values of m and k are usually very large, so it is impossible to crack the encoding matrix in theory. But, using encoding technology cannot ensure the privacy of each data block especially for document file. So, we add a hash transform before encoding to disrupt the sequence of original data and save the relevant hash information in the local server. The code divides a sentence into different fragments according to original sequence. However, the hash code divides the sentence into different fragments according to random sequence. Thereby, Hash-Solomon code improves the privacy protection and prevents the attacker from getting fragmentary information.

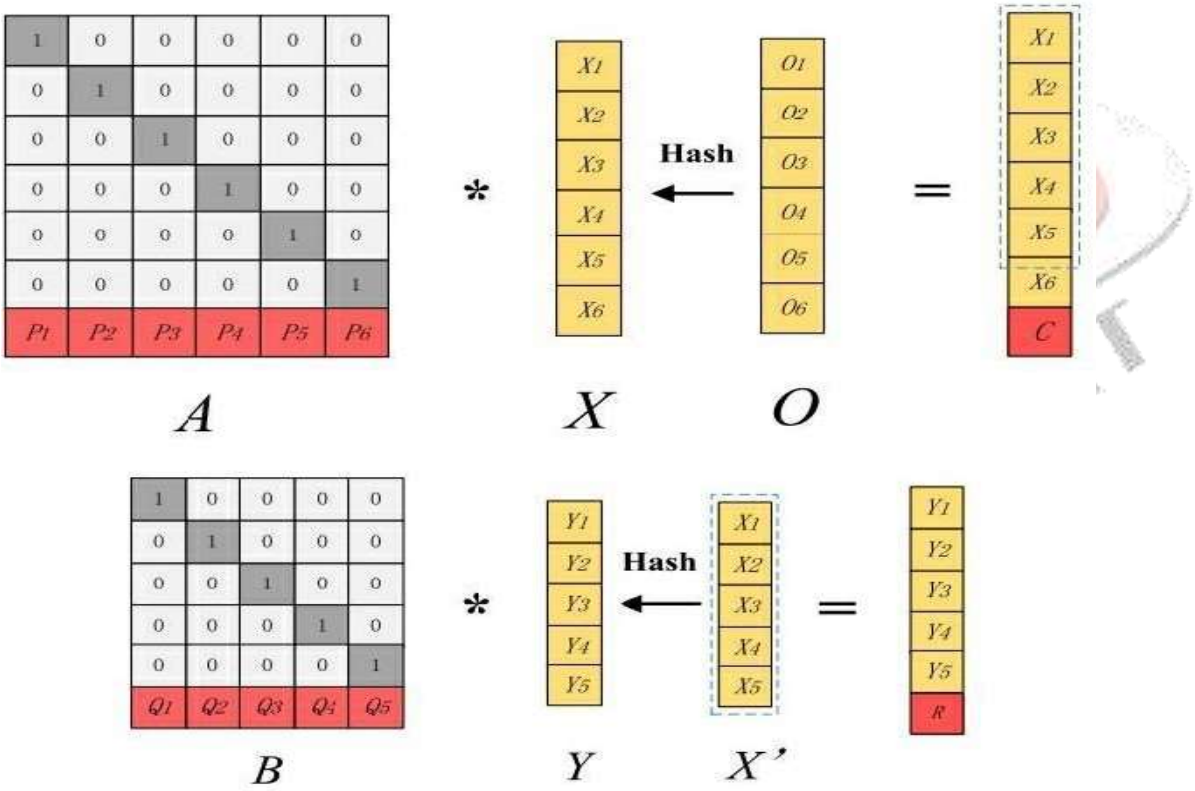


Figure 4 hash transformation

VI. RESULT

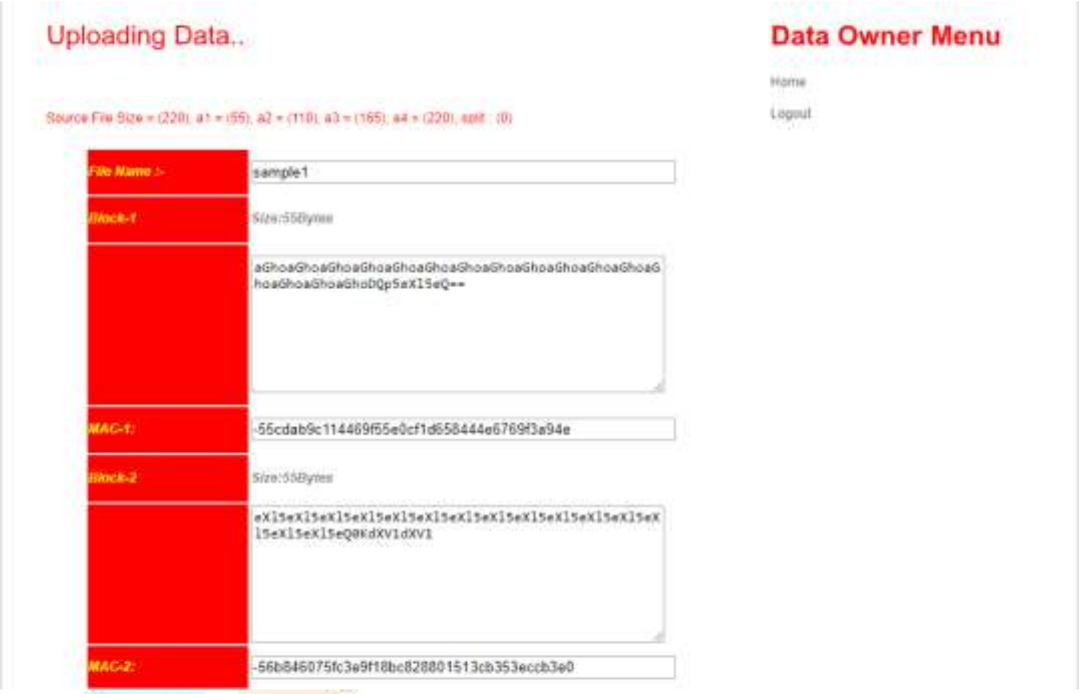


Figure 5. data uploading page

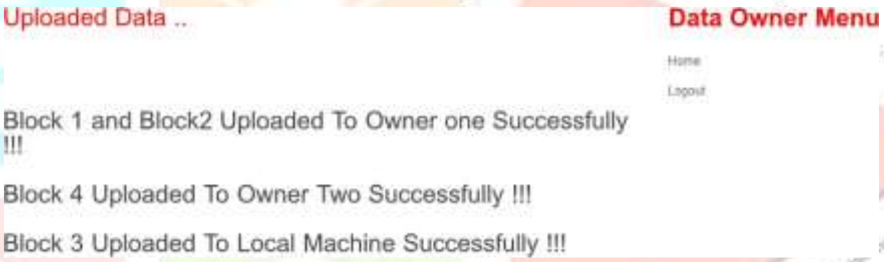


Figure 6. data uploaded successfully

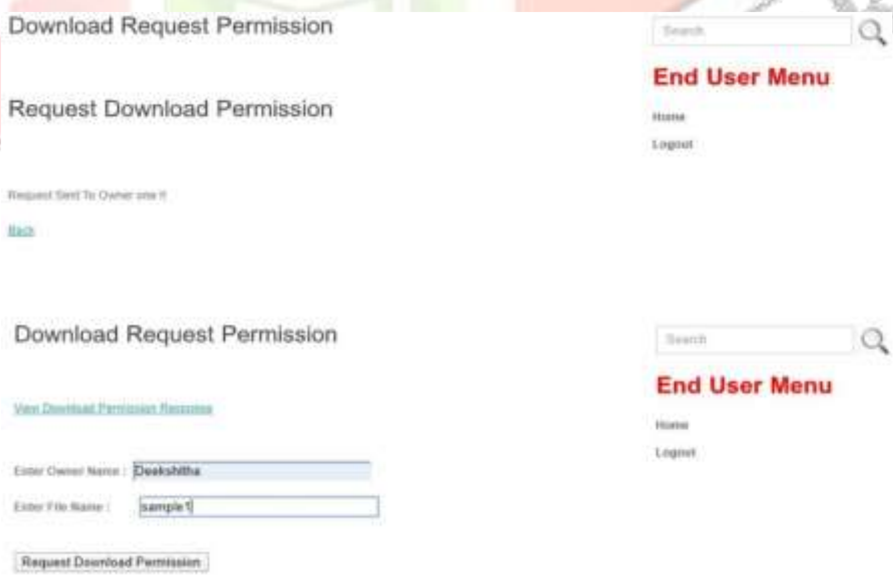


Figure 7. download request

Figure 8. authorize download request

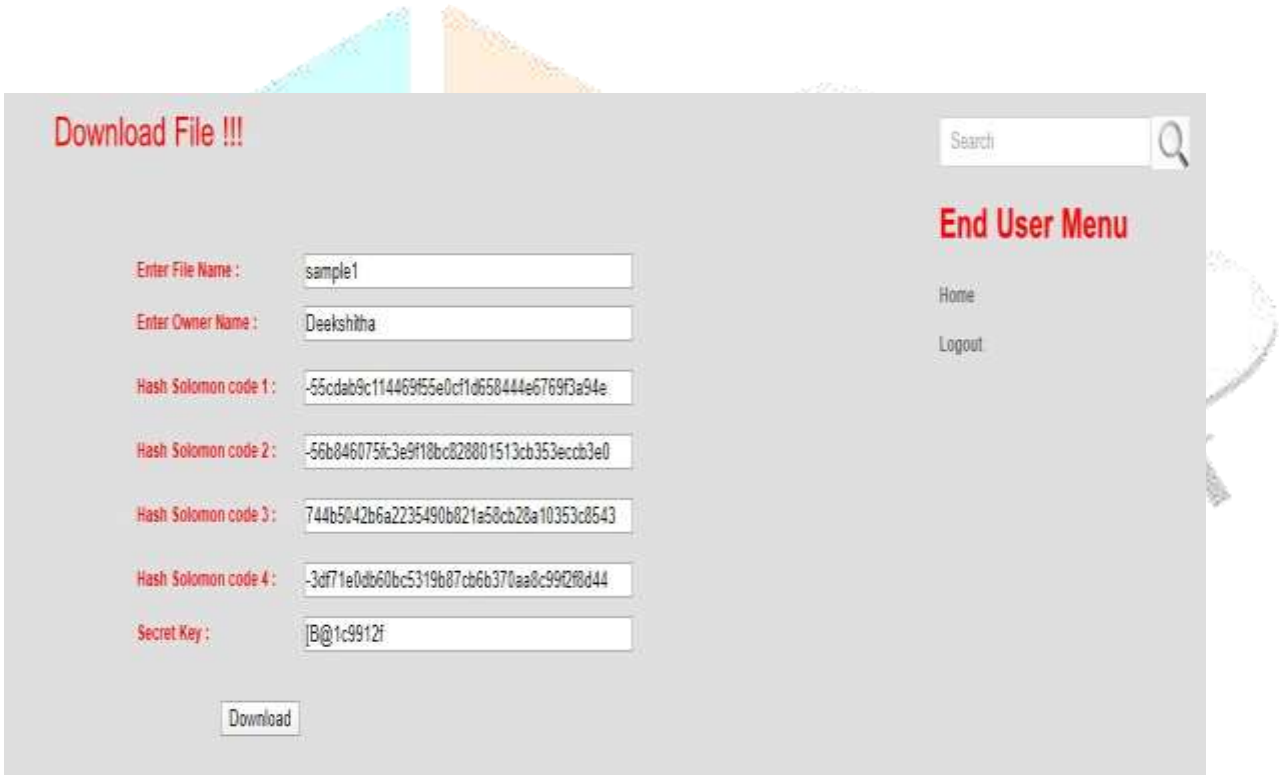


Figure 9. request mac

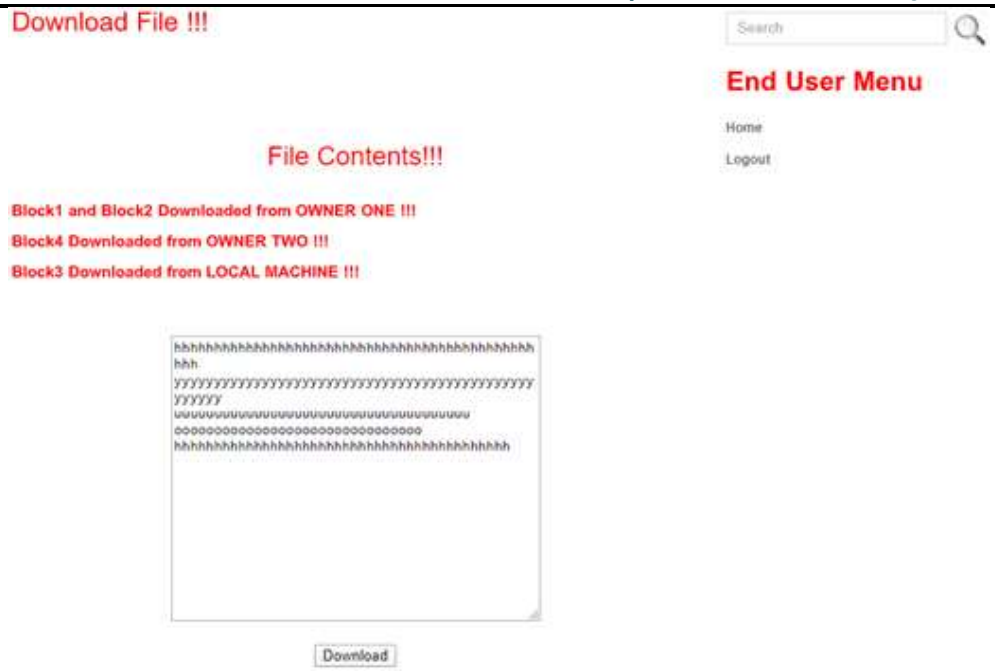


Figure 10. download file

VII. CONCLUSION

The development of data security brings us plenty of advantages. Security could also be a convenient technology which helps users to expand their storage capacity. However, network storage also causes a series of secure problems. When using network storage, users don't actually control the physical storage of their data and it results in this partition of ownership and management of data. So to unravel the matter of privacy protection, we propose a TLS framework based on three users and elegance a Hash-Solomon algorithm. Through the theoretical safety analysis, the scheme is proved to be feasible. By allocating the ratio of data blocks stored in numerous servers reasonably, we can ensure the privacy of data in each server. On another hand, cracking the encoding matrix is impossible theoretically. Besides, using hash transformation can protect the fragmentary information. Through the experiment test, this scheme can efficiently complete encoding and decoding without influence of the storage efficiency.

REFERENCES

[1] Ateniese, et al *"A practical and provably secure group signature scheme"* Proceedings of CRYPTO '00(2000), pp 255-270.

[2] C. Wang, et al *"Privacy -preserving public aid security in cloud computing"* INFOCOM, Proceedings IEEE, San Diego (2010), pp 1-9 march.

[3] z fu. F. Huang, K Ren, J. Weng and c-wang, "*Privacy - preserving smart semantic search based on conceptual graph over encrypted outsourced of data"* IEEE Trans Inf. Forensics Security, vol. 12 no. 8, pp 34 - 1884, Aug 2017.

[4] Z. Fu. K. Ren, J. Shu, X Sun and F. Huong, "*Enabling personalized search over encrypted outsourced data with efficiency improvement",* IEEE Trans: Parallel Distributed System vol. 27, no. 4, pp - 25462559 Sep 2016.

[5] Z. Xia, X. Wang, X. Sun and Q. Wang *"A secure and dynamic multi keyword ranked encrypted cloud data"*, IEEE Trans Parallel Distributed systems vol 21, 72, PP 340 - 352, Feb 2016

[6] T. Weng et al., *"Maximizing real time streaming service based on a multi- servers networking framework"*, Computer Network, vol. 93. pp. 199.-212, 2015.

[7] X. Hou, Y. Wu, W. Zheng and G. Yang, *"A method on protection of user data privacy in cloud storage platform",* J. Comput. Res Develop, vol. 48, no 7, pp. 1146-1154,2011.

[8] M.EA Bhuiyan, T. Wong, T: Hayajneh, and G.M Weiss, *"Maintaining the balance between and data integrity in internet of things"*, in Proc. Int. Conf. Manage, Eng., Softw, Eng., Serv. Sci., 2017, pp 177-182.

[9] J. Shan, D. Liu, J. Shen, Q. Liu and X. sun, *"A secure cloud - assisted urban data sharing framework for ubiquitous – cities"*, Pervasive Mobile Computing., Vol. 41, pp. 219-230,2017.

[10] L. Wei et al., *"Security and privacy for storage and computation in cloud computing*", Inf.Sci., vol. 258, pp. 371-386, 2014.

[11] Z. Xia, X. Wong, L. Zhang, Z. Qin, X. Sun, K. Ren, *"A privacy-preserving and copy-deterrence content - based image retrieval scheme in cloud computing"*, IEEE Trans, Inf Forensics Security, vol II, no. II, Pp- 2594-2608, Nov 2016.

[12] J. Hou, C- Piao and T.fan, *"Privacy prevention cloud storage architecture research",* J. Hebei Accad, Sci, Vol 30, no. 2, PP 45-48, 2013.