International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 29 Issue 1 – JANUARY 2022.

TOWARDS SECURE AND PRIVACY-PRESERVING DISTRIBUTED DEEP LEARNING IN FOG-CLOUD

G.Guru Charan[#1]

[#]*Former Student, Department of Electronic and Communication Engineering, Kalasalingam Academy of Research and Education,Anand Nagar, Krishnankoil, India*

***Abstract*— The storage service is excellent without the users having to extract their sensitive data from the cloud storage server. The cloud server gains full access and controls user data once the data is extracted from the cloud. It can read or search through user data. Recently, a three-layer-based fog server development was introduced to keep it safe. In that diagram, part of the data will be stored in the cloud, fog and local user interface. Some part of the data in the cloud and the custom hash algorithm, take extra computing / storage up, In this project, we are building a cloud-based cloud storage system. In that system, the data is divided into multiple blocks using xor-combinations and combines these blocks into 2 blocks or 3 blocks using xor-operation. We therefore use this scheme, improve the efficiency of the cloud-based cloud storage service and improve the security of the cloud component of the cloud computing infrastructure and upgrade the cypto system to protect data without exposing any information to it.**

Index Terms- Random Forest, LSTM, Regression

I. INTRODUCTION

Fog-cloud computing is considered the only option available for single-cloud computing. With the provision of high-density Internet of Things (IoT) services and low latency, the fog computer uses part of the processing and storage of fog nodes, which are close to “objects”. Based on the standard computer architecture, DDL has been applied effectively to IoT, and is provided with a variety of features, such as autonomous vehicles, intelligent grid, e-health, etc. For example, a DDL-based car system can perform complex driving on time and accurately. resolutions, using channels of integrated neural networks (fog nodes) and a cloud server. Alternatively, a smart DDL-based grid can achieve electricity billing and error detection in a highly efficient way. Certainly, DDL has achieved amazing successes in IoT, and expanded IoT system scenarios. Since back in 2016, Google has officially implemented the concept of DDL (also known as integrated reading) to address the issue of data privacy with a distributed learning model. With the framework of the novel, DDL can get a training model only by exchanging parameters between cloud server and users, rather than raw data. Apart from this, DDL still has some security issues, which may prevent the use of DDL on a fog-cloud computer. Advanced research has shown that in a DDL-based framework, user confidential information may be leaked to other internal participants even with a small fraction of leaked parameters. To address this issue, a number of studies have been conducted, based primarily on the privacy of different (DP) and encryption technology. The main idea of DP-based DDL is to add sound (Laplace or Gaussian sounds), to the modified parameters to hide real information while achieving a secure integration of global weights for sensory networks. Encryption technology falls into two main categories: multi-group computerized computing (SMC) and homomorphic encryption (HE). SMC-based DDL aims to create ways for users to calculate the sum of their gradients while maintaining data privacy through Yao's garbled circuit (GC) or private key sharing (such as Shamir's private sharing). Additionally, HE-based DDL, uses full homomorphic encryption (FHE) or additive homomorphic encryption (AHE) to create privacy protections.

II. LITERATURE REVIEW

Focuses on Safe Reading and Extensive Privacy in Fog-Cloud Computing, S. Abdulrahman, H. Tout, H. Ould-Slimane, A.Mourad, C. Talhi, and M.Guizani. 2021. Integrated Reading Survey: A Journey from Intermediate to Site Distributed Reading and Above. IEEE Internet of Things Journal 8, 7 (2021). Machine learning and cloud computing are integrated into various systems to provide intelligent services. With powerful computational ability, the cloud server can use the machine learning algorithm efficiently. However, as accurate machine learning is highly dependent on model training with sufficient data. Transferring large raw data from distributed devices to the cloud leads to greater communication and privacy leaks. Distributed learning is a promising way to reduce data transfer by allowing distributed tools to participate in training local models. So the global learning work can be done in a broader way. Nearest neighbor search is a problem of finding the data points from the database such that the distances from them to the query point are the smallest. Learning to hash is one of the major solutions to this problem and has been widely studied recently. In this paper, we present a comprehensive survey of the learning to hash algorithms, categorize them according to the manners of preserving the similarities into: pairwise similarity preserving, multiwise similarity preserving, implicit similarity preserving, as well as quantization, and discuss their relations. We separate quantization from pairwise similarity preserving as the objective function is very different though quantization, as we show, can be derived from preserving the pairwise similarities. U-lite private cloud method of particle physics computing, B.

Demin, S. Parlati, P. F. Spinnato and S. Stalio, "U-lite a private cloud approach for particle physics computing", Int. J. Cloud Appl. Computer., Vol. 9, no. 1, pp. 1-15, 2019. The advent of app virtualization and cloud paradigm has allowed for a complete overhaul of the LNGS computer operating system and management, and the recognition of U-LITE, a secret place with many functions suitable for such hosting. a diverse ecosystem and provides LNGS science users with a standard computer interface that hides all the complexities of modern data management management. Since 2011, U-LITE has proven to be an important LNGS test tool, and provides an example of the effective use of a secret cloud computing system in real science contexts.

Looking at the green cloud computing algorithmic approach to reducing power in cloud data centers, JA Jeba, S. Roy, MO Rashid, ST Atik and M. Whaiduzzaman, "Towards green cloud computing algorithmic approach to reducing power in cloud data centers", Int. J. Cloud Appl. Computer., Vol. 9, no. 1, pp. 59-81, 2019. In this article, the authors propose algorithms that focus primarily on live VM live streaming mitigation strategies called "Power_reduction" and "VM_migration." In addition, authors use flexible server configurations based on sequential searches, random searches, and fairly high search for easy sharing and high utility resources. Authors do simulation work using CloudSim and Cloudera template to test the performance of the proposed algorithms. The results show that the proposed methods achieve energy efficiency of about 30% over existing algorithms.

Automatic Driver Safety: Threats, Defense, and Future Directions, K. Ren, Q. Wang, C. Wang, Z. Qin and X. Lin, "Automatic driving safety: Threatened protections and future directions", Proc. IEEE, vol. 108, no. 2, pp. 357-372, Feb. 2020. In addition to an in-depth review of these threats, we also summarize the corresponding defense strategies. In addition, we discuss future research guidelines on new safety threats, especially those related to in-depth learning based on self-driving vehicles. By providing safety guidelines in this early phase, we aim to promote new strategies and projects related to AVs from both academic and industry, and promote the development of safe driving.

Protecting Shared Learning: Reducing Toxic Attacks through Client Acquisition, L. Zhao et al., "Protecting collaborative learning: Reducing toxic attack through client acquisition", IEEE Trans. Depend. Secure Comput., Apr. 2020. In this article, we introduce a new defense system to get amazing updates on both IID and non-IID settings. Our main focus is to get the opposite side of the client side, where each update is evaluated with the local data of other clients. The server will adjust the weights of the updates based on the test results when performing the integration. In order to accommodate the unequal distribution of data in non-IID settings, the flexible client allocation method is designed to assign the most appropriate clients the diagnostic functions. During the acquisition process, we also protect client-level privacy to prevent malicious clients from being able to participate in other clients, by combining different privacy with our design without compromising adoption performance. Our experimental experiments on three real-world databases show that our system is extremely resilient to two toxic attacks.

In this paper, we propose to protect and maintain privacy distributed deep learning (SPDDL) for fog-cloud computing,where threshold Paillier encryption is used to encrypt local gradients, and a boundary signature is used to verify user identity. It shows that our proposal achieves the transaction between security,efficiency, as well as performance have current functions. Focussecurity, our system carries encryption tools as wellit does not need to measure accuracy and confidentiality, bothmake our system more secure than AHE-based methods and DP-based methods. At the same time, our system operates with higher accuracy than DP-based methods. As tasks such as the SMC-based approach can provide, our SDDDL provides privacy protection against conflict between multiple internal participants and tightness for users exiting during the training process. Note that, the small interaction between participants and the appropriate encryption method allows our SDDDL to operate with very minimal communication and computing. In addition, the authentication system is designed to protect external enemies from fraudulent user identity, which gives our SPDLL a high level of security. Compared to the previous version of the conference, we have improved the security level of our system

III. EXISTING SYSTEM

Cloud Service Provider (CSP) can access, search or modify its data stored in cloud storage. At the same time, the CSP may lose data unintentionally due to some technical errors. On the other hand, a hacker may violate the privacy of user data. Using other encryption methods (such as encryption, hash series), privacy or integrity can be protected. However the secret method cannot prevent internal attacks.

IV. PROPOSED SYSTEM

For confidentiality and availability (even after malicious incidents), we propose a method called Xor-combination that divides data into a few blocks, combines multiple blocks using Xor functionality and removes resulting blocks on different cloud / fog servers. We achieve two goals in our project, one is data privacy and the other is data recovery in the event of data loss. We improve the security of the fog server and cloud server. To protect data encryption, Integrity proposes a cloud-based cloud encryption scheme, integrity and availability.

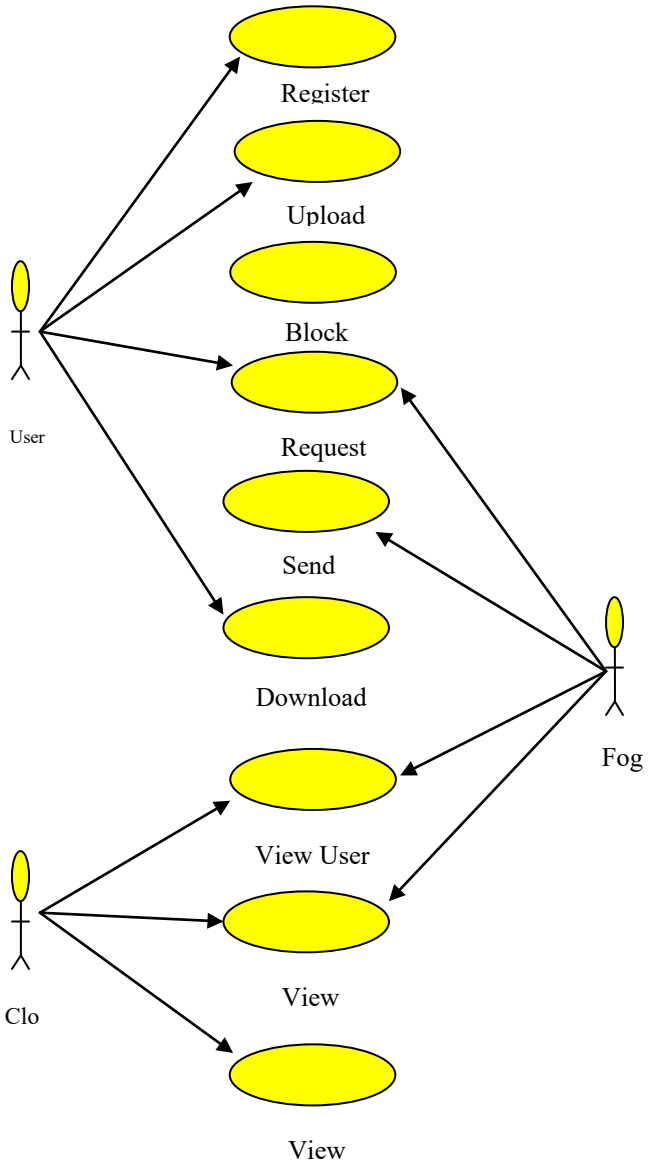
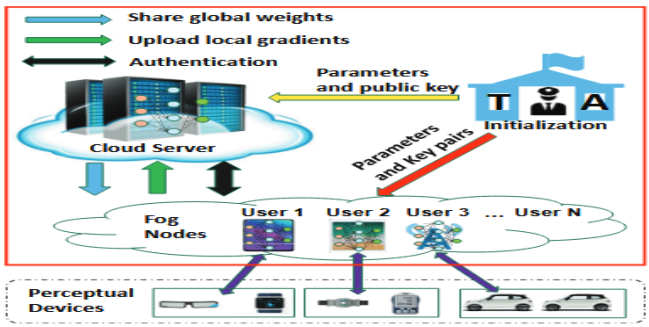


Figure 1: Proposed System

V. METHODOLOGY

Property description is the official description of a system,



organized in a way that supports thinking about system properties. It defines system components or building blocks and provides a system in which products can be purchased, as well as improved systems, that will work together to run the entire system.

Figure 2: Methodology

VI. TECHNOLOGY USED

ASP.NET

ASP.NET, the next version of ASP, is a planning framework used to create business - class web applications. Business class web applications are accessible worldwide through uploading to information management. However, the benefits offered by ASP.NET make it more than just the next version of ASP.NET.

ASP.NET is integrated with the visual studio.Net, which provides a GUI designer, a rich toolbox and a fully integrated debugger. This allows for the development of applications the way you see it is what you find (WYSIWYG) WAY.

The .NET Framework is a common site for creating, deploying, and deploying Web Services and Web Applications. The .NET Framework contains standard class libraries - such as ADO.NET, ASP.NET and Windows Forms - to provide advanced services that can be integrated into a variety of computer programs.

The .NET Framework is language-neutral. It currently supports C ++, C #, Visual Basic, JScript (Microsoft version of JavaScript) .The new Visual Studio.NET is a standard upgrade for the new .NET Framework. Provides a rich application environment, simplified development and easy integration between a number of different development languages.

Unique Nature Features .Net

1. Internal Internet
2. Ordinary language support
3. Ordinary Class Libraries
4. Working time for a common language
5. Garbage collection.
6. Cross Language Reference
7. Web Services

INTERNET INFORMATION SERVICE 6.0

IIS 6.0 has strong support for multiple server settings, to allow new Web Applications to run on any browser in any field.

• ASP.NET is a server-side scripting technology that allows scripts (embedded in web pages) to be used by an Internet server.

• ASP.NET is Microsoft Technology

• ASP.NET stands for Active Server Pages

• ASP.NET is a program running within IIS

• IIS stands for Internet Information Services

• IIS comes as a free component with Windows 2008

• IIS is also part of the Windows NT 4.0 Option Pack

• The Option Pack can be downloaded from Microsoft PWS a smaller - but fully functional - version of the IIS PWS can be found on your Windows 95/98 CD.

• ASP.NET 3.0 is the latest version of ASP.NET, but never will be the ASP.NET 4.0 version.

• ASP.NET is the next generation of ASP.NET, but it is not an improved version of ASP.NET. ASP.NET is a completely new ASP.NET script paradigm next to the server

• ASP.NET is part of the new .NET (dotnet) Framework. Microsoft spent three years rewriting ASP.NET from the ground up, and ASP.NET is not fully compatible with ASP.NET 3.0.

• ASP.NET has better language support, a larger set of new controls and XML-based components, and better user authentication.

• ASP.NET provides additional functionality through integrated code.

• ASP.NET code is not fully compatible with ASP.NET.

• ASP.NET is the programming language on the server side.

• ASP.NET is an object-oriented programming language.

• Active Server Pages - ASP.NET

• ASP.NET is the latest version of ASP.NET. Includes Web Services to link applications, services and devices that use HTTP, HTML, XML and SOAP.

NEW ON ASP.NET

1. New Language Support

2. Systematic Controls

3. Event Driven Programming

4. XML Based Components

5. User authentication

6. User Accounts and Roles

7. High Scalability

8. Combined Code

9. Easy Configuration

10. Easy Shipping

11. Includes ADO .NET

WEBSITE SERVICES

• Web services are sub-units of code

• Web services designed to manage a limited set of tasks

• Web services that use XML-based communication protocols

• Independent web services for operating systems

• Independent web services for programming languages.

• .NET Web Services

• Web services are sub-units of code designed to manage limited functionality.

• Small Code Units

• Web services small code units designed to manage a limited set of tasks.

XML SUPPORTED WEBSITE PROTOCOLS

Web services that use standard web protocols are HTTP, XML, SOAP, WSDL, and UDDI.

HTTP

HTTP (Hypertext Transfer Protocol) is the Global Standard for Web communication.

XML

XML (Markup Extensible Language) is a well-known standard for storing, transporting, and exchanging data.

SQL SERVER

The archive component of Microsoft® SQL Server ™ 2008 is the Query Language (SQL) –based, scalable, relation database and integrated Extensible Markup Language (XML) support for online applications. Each of the following terms describes an integral part of the SQL Server 2008 site architecture:

DATABASE

A website is similar to a data file in that it is a data repository. Like a data file, a website does not deliver information directly to the user; the user runs an application that accesses the data from the website and presents it to the user in an understandable format.

Web systems are much more powerful than data files in that the data is more organized. On a well-designed website, there are no duplicate pieces of data a user or application

should update at once. Related pieces of data are grouped together into a single structure or record, and relationships can be defined between these structures and records.

When working with data files, the application must be coded to work with the specific structure of each data file. In contrast, a website contains a catalog used by applications to determine how data is organized. Standard web applications can use the catalog to deliver users data from different websites in a dynamic way, without being tied to a specific data format.

A website usually has two main components: first, the files that host the actual site and secondly, the database management system (DBMS) software used by applications to access data. DBMS is responsible for enforcing database formation, which includes:

☐ Maintaining data relationships on the website.

☐ Ensuring that data is stored properly, and that th

RELATIONAL DATABASE

Although there are different ways to organize data on a website, a related website is one of the most effective. Related website programs are the application of mathematical set theory to the problem of successfully organizing data. In a relationship site, data is collected in tables (called relationships in relationship theory).

The table represents a specific category of priorities in the organization. For example, a company might have a website with a staff table, another customer table, and another store. Each table is made up of columns and rows (called attributes and tuples in relation theory). Each column represents a specific element of a table. For example, an employee table will usually have columns for attributes such as first name, surname, employee ID, department, paid range, and job title. Each line represents an example of an object represented by a table. For example, one line in an employee's table represents an employee with 12345 employee ID. When organizing data by tables, you can usually find many different ways to define tables. Related website theory defines a process called normalization, which ensures that the set of tables you define will organize your data effectively.

SCALABLE

SQL Server 2008 supports having multiple users accessing one at a time. An example of SQL Server 2008 includes files that form a data set and a copy of the DBMS software. Apps running on multiple computers use the communication component of SQL Server 2008 to transmit instructions over the network to the SQL Server 2008 model. The communications component also allows connection between the SQL Server 2008 event and the same operating system. You can use multiple SQL Server 2008 instances on one computer.

SQL Server 2008 is designed to support the fullness of large websites or business data processing systems. SQL Server 2008 scenarios working on large servers, multiprocessors are able to support communication with thousands of users at once.

Although SQL Server 2008 is designed to serve as a data storage engine for thousands of simultaneous users who connect to a network, it is also capable of operating as a standalone server directly on the same computer and operating system. The growth and ease of use of SQL Server 2008 allows it to run smoothly on a single computer without having to use multiple resources or require administrative work done by an independent user. Similar features allow SQL Server 2008 to provide the necessary resources to support thousands of users, while minimizing site management and tuning. The related SQL Server 2008 database engine automatically adapts to locate or release the appropriate computer resources needed to support the diverse load of users accessing the SQL Server 2008 event at any given time. The SQL Server 2008 related web engine has logical problem-solving features that occur when a user tries to read or modify data currently in use by others.

STRUCTURED QUERY LANGUAGE

To process data on a website, you must use a set of instructions and statements (language) defined by the DBMS software. Several different languages can be used with related websites; The most common is SQL. The American National Standards Institute (ANSI) and the International Standards Organization (ISO) define software standards, including SQL language standards. SQL Server 2008 supports SQL-92 Login Level, the SQL standard published by ANSI and ISO in 1992. The local SQL language supported by Microsoft SQL Server is called Transact-SQL (T-SQL). T-SQL is the main language used by Microsoft SQL Server applications.

EXTENSIBLE MARKUP LANGUAGE

XML is an emerging Internet standard for data. XML is a collection of tags that can be used to define high-document document formatting. XML texts can be easily processed by Hypertext Markup Language, which is the most important language in Web pages.

Although many SQL statements return their results to a set of related results, or a table, the database for SQL Server 2008 supports the FOR XML clause that returns results as XML text. SQL Server 2008 also supports XPath query

from online and intranet applications. XML documents can be added to the SQL Server website, and the OPENXML clause can be used to display data from the XML document as a set of relationship results.

V. EXPERIMENTAL RESULTS AND DISCUSSION

1. Storing Procedure

The last process takes the file to be uploaded to the cloud server securely. It has a few steps and the most important steps take place on the fog server. When a user intends to upload a data file, he or she sends the file to a fog server via a specific secure channel. Then, the fog server starts processing the file.

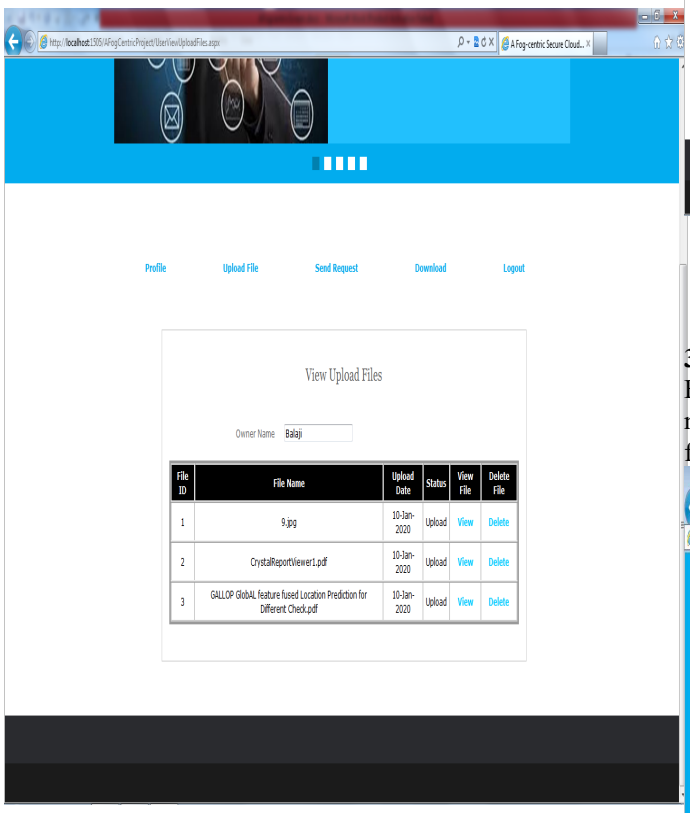


Figure 3: Storing Procedure

2. Splitting File

The fog server adjusts the file according to requirements according to system policy. The fog server then splits the file into several fixed-length blocks and merges them. At the end of this step, we find two sets of 2-block-combinations and 3-blockcombinations together known as combined blocks.

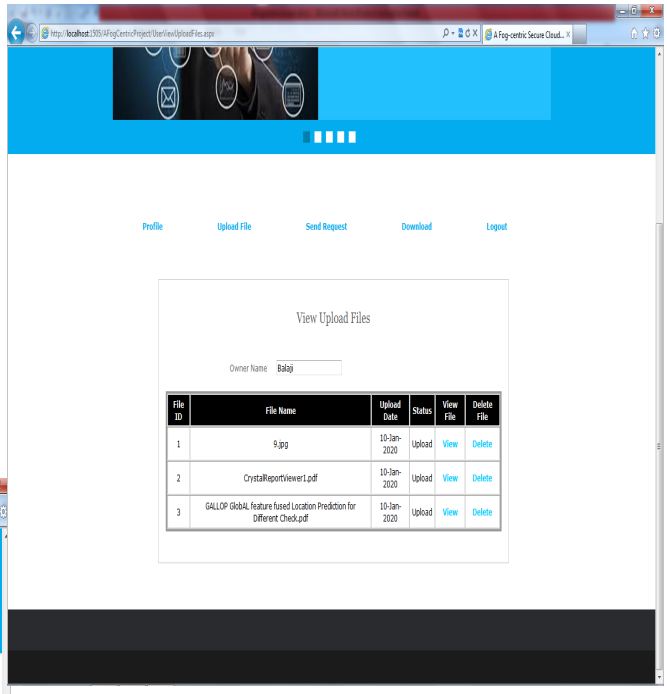


Figure 4: Splitting File

3. Integrity Processing

For each integrated block, the fog server generates a random number, file key and stores this information on a fog site for future integrity testing.

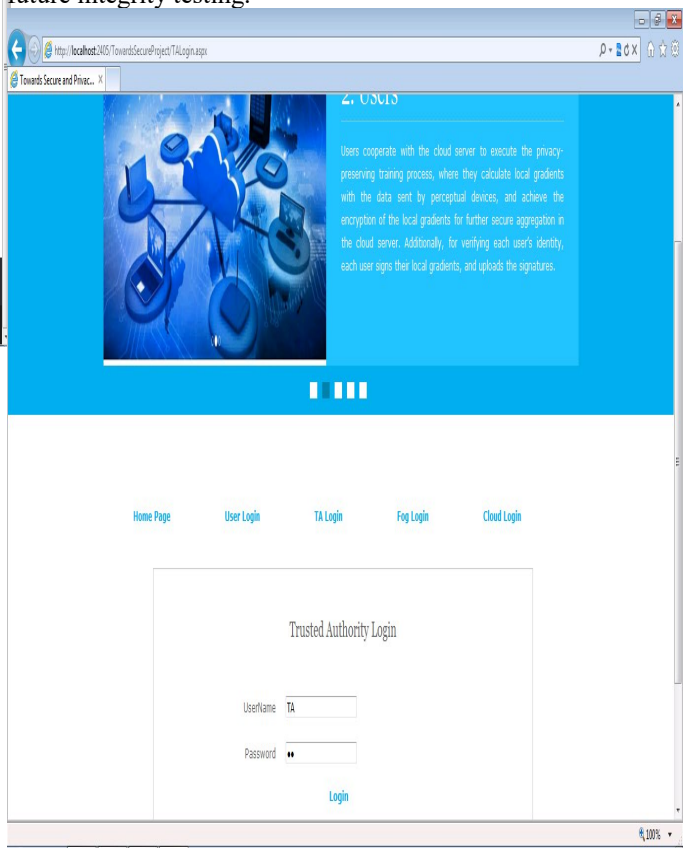


Figure 5: Integrity Processing-1

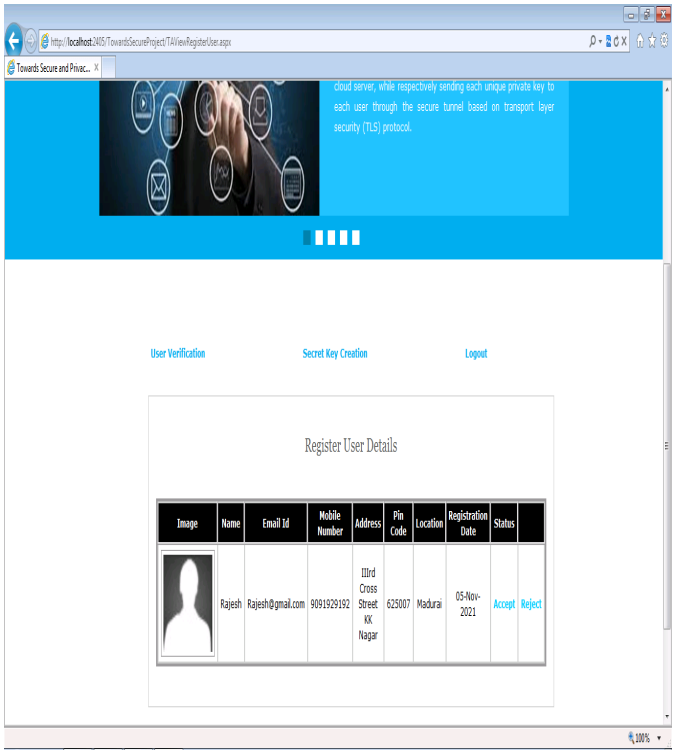


Figure 6: Integrity Processing-2

4. Block Management

In this module, the fog server determines which block to be stored on the cloud server using the block management method and stores this metadata on the fog site and sends the blocks to the appropriate cloud servers.

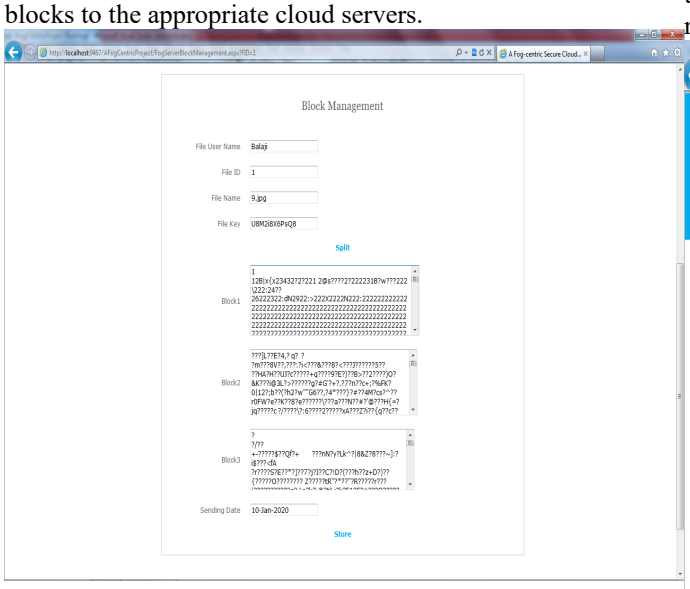


Figure 7: Block Management

5. Cloud Storage

The cloud server receives and stores blocks and metadata in its storage.

Figure 8: Cloud Storage-1

Figure 9: Cloud Storage-2

6. Retrieval Procedure

The retrieval process takes a file request, collects the required blocks of various cloud servers, and tests their integrity. If the integrity check fails then it asks for error blocks on other cloud servers. When all the required integrated blocks pass the integrity check, the fog server rebuilds the entire file and returns it to the user.

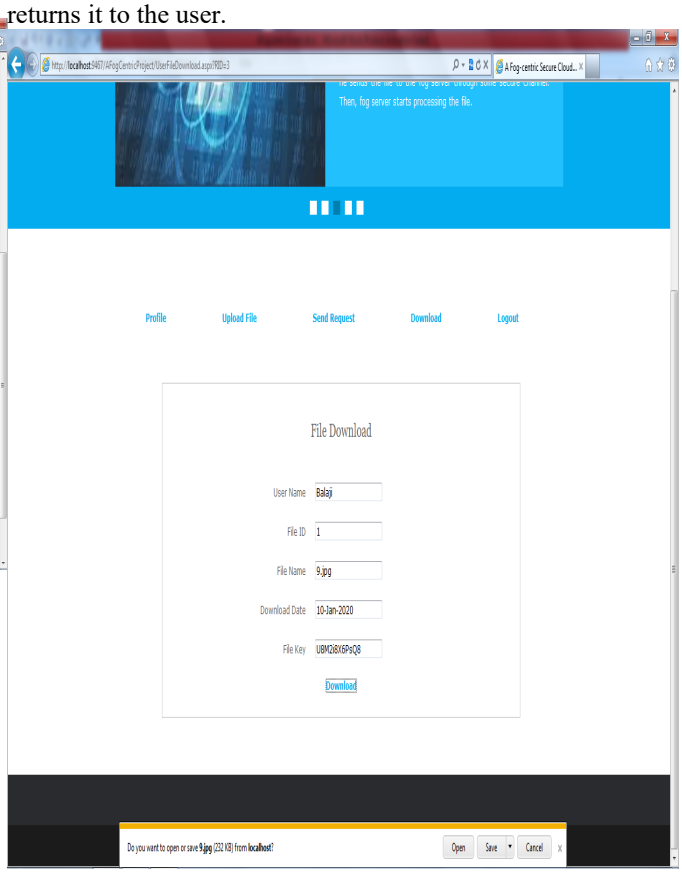


Figure 10: Retrieval Procedure

VI. CONCLUSION

A three-layer fog-based structure suits a secure cloud storage solution against cyber threats. This project has proposed a scheme that performs cloud-based blockchain operations and puts the actual data in curly format on multiple cloud servers. We have improved the efficiency of the cloud-based cloud storage service. We improve fog server security with robust cloud computing infrastructure.

VII. FUTURE ENHANCEMENT

- Improve the efficiency of cloud-based cloud storage.
- Improving the security of the cloud server with a strong cloud computing infrastructure.
- Allowing the cloud server to compile confidential data without exposing any information to it.

REFERENCES

[1] B. Demin, S. Parlati, P. F. Spinnato, and S. Stalio, “Lite, a private cloud approach to particle physics computing,” International Journal of Cloud Applications and Computing (IJCAC), vol. 9, no. 1, pages 1–15, 2019.

[2] JA Jeba, S. Roy, MO Rashid, ST Atik, and M. Whaiduzzaman, “Focuses on green cloud computing an algorithmic approach to reducing power in cloud data centers,” International Journal of Cloud Applications and Computing (IJCAC), vol. 9, no. 1, pages 59–81, 2019.

[3] M. M. Hussain and M. S. Beg, “Using vehicles as fog transport infrastructure for cyber-physical systems (t-cps): The computer cloud of automotive networks,” International Journal of Software Science and Computational Intelligence (IJSSCI), vol. 11, no. 1, pages 47–69, 2019.

[4] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, “Automatic driving safety: Threats, protections, and future directions,” Proceedings of the IEEE, vol. 108, no. 2, pages 357–372, 2019.

[5] L. Zhao, S. Hu, Q. Wang, J. Jiang, S. Chao, X. Luo, no-P. Hu, “Protecting collaborative learning: Reducing toxic attacks through client detection,” IEEE Transactions on Reliable and Secure Computer, 2020, doi: 10.1109 / TDSC.2020.2986205.

[6] N. Saxena, S. Grijalva, V. Chukwuka, and A. V. Vasilakos, “Network security and privacy challenges in smart car-to-grid,” IEEE Wireless Communications, vol. 24, no. 4, pages 88–98, 2019.

[7] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, “Healthdep: An effective and secure deployment scheme for cloud-assisted health systems,” IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pages 4101–4112, 2019.

[8] J. Ni, K. Zhang, X. Lin, and X. S. Shen, “Computer Fog Protection for Internet of Things objects: Challenges and Solutions,” IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pages 601–628, 2019.

[9] R. Shokri, M. Stronati, C. Song, no-V. Shmatikov, “The Thoughtful Attack on Membership Against Machine Learning Models,” at the 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019, pages 3-18.

[10] B. Jayaraman and D. Evans, "Testing the learning of a secret machine separately from performance," in Proceedings of USENIX Security. USENIX Association, 2019, pages 1895-1912.

[11] S. Chang and C. Li, "Privacy in neural network learning: Threats and countermeasures," IEEE Network, vol. 32, no. 4, pages 61–67, 2019.

[12] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, HB Mcmahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “A secure and efficient integration of confidential machine learning,” in ACM CCS Procedures. ACM, 2019, pages 1175–1191.

[13] H. Bae, J. Jang, D. Jung, H. Jang, H. Ha, and S. Yoon, “Security and privacy issues for in-depth learning,” arXiv preprint arXiv: 1807.11655, 2019.

[14] X. Liu, H. Li, G. Xu, S. Liu, Z. Liu, and R. Lu, “Padl: Confidential information and in-depth learning of iot applications,” IEEE Internet of Things Journal, 2020, doi: 10.1109 / JIOT.2020.2981379.

[15] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. Deng, “Confidentiality-deep learning retention integrated with unfamiliar users,” IEEE Transactions on Dependable and Secure Computing, 2020, doi: 10.1109 / TD-SC.2020.3005909.

[16] A. Fiat and A. Shamir, “How to Identify Yourself: Practical Solutions for Identifying and Signing Problems,” at the Conference on Theory and Practice of Cryptographic Methods. Springer, 1986, pp. 186-194.

[17] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, “Enabling segmented search for more keywords over mobile cloud data encrypted with blind storage,” IEEE Transactions on Emerging Topics in Computing, vol. 3, no. 1, pages 127–138, 2015.

[18] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, “Effective acquisition of truth and confidentiality in public hearing systems,” IEEE Transactions on Vehicular Technology, vol. 68, when. 4, pages 3854–3865, 2019.

[19] P. Paillier, "Cryptosystems of public keys based on classes of composite degree residuosity," at an international conference on theory and the use of writing techniques. Springer, 1999, pp. 223-238.

[20] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, “Enables effective and geometrical query by controlling access to encrypted data,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 4, pages 870–885, 2019.

[21] P. Mohassel and Y. Zhang, "Secureml: An unparalleled record keeping system that keeps machine learning," in the pursuit of IEEE S&P. IEEE, 2017, pages 19-38.

[22] R. Shokri and V. Shmatikov, “Secrets that keep deep learning,” in Proceedings of ACM CCS. ACM, 2015, pages 1310-1321.