Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 33 No. 2 (2023)

Homomorphic Encryption and Secure Multi-Party Computation: Mathematical Tools for Privacy-Preserving Data Analysis in the Cloud

Mayuri Arun Gaikwad[1], Vikas Haribhau Satonkar[2], Akash Ganesh Mohod[3], Ritu Raj Jha[4], Rakhi Madhukar Giradkar[5], Sonam Rani[6]

[1] Assistant Professor, Sandip Institute of Technology & Research Centre, Nashik, Maharashtra, India. mayuri.gaikwad@sitrc.org

[2] Assistant Professor, Department of Computer Engineering, Sandip Institute of Engineering & Management, Nashik, Maharashtra, India. vikas.satonkar@siem.org.in

[3] Assistant Professor, Department of Information Technology, Sandip Institute of Technology & Research Centre, Nashik, Maharashtra, India. akash.mohod@sitrc.org

[4] Assistant Professor, Deaprtment of Computer Engineering, Sandip University Sijoul, Madhubani, Bihar, India. ritu.jha@sandipuniversity.edu.in

[5] Assistant Professor, Sandip University Nashik, Maharashtra, India. rakhi.giradkar@sandipuniversity.edu.in

[6] Assistant Professor, Department of Computer Engineering, Sandip University, Sijoul, Bihar, India. sonam.rani@sandipuniversity.edu.in

Article History:

Received: *20-03-2023*

Revised: *25-05-2023*

Accepted: *23-06-2023*

Abstract:

With more and more people using cloud computing and storing and handling data remotely, protecting the privacy and safety of private data has become very important. Homomorphic encryption and safe multi-party computing (MPC) are two new mathematics tools that offer strong ways to analyze data in the cloud while protecting privacy. When you use homomorphic encryption, you can do calculations directly on protected data, so you can process data securely without having to decode private data. This method makes sure that data stays protected while operations are being done, keeping it safe from people who shouldn't have access to it or seeing it. Cloud service providers can use homomorphic encryption to do different types of analysis on protected data, like collecting, searching, and machine learning, without revealing the private information that lies beneath. Secure multi-party computation protects privacy in situations where multiple people work together to analyze data. MPC allows for joint analysis without letting other people see individual datasets by spreading computations across multiple entities, each of which holds a piece of the data. MPC uses cryptographic protocols and methods to make sure that processes are done without revealing private inputs. This lets multiple people work together to analyze data while keeping privacy. These math tools can be used for many different types of data analysis jobs in the cloud, such as predictive modeling, machine learning, and statistical analysis. They also make it safe for different groups to share and work together on data, like businesses, academics, and people, without putting data protection at risk. There are still problems with how homomorphic encryption and safe MPC can be used in the real world and how they can be scaled up. These problems are mostly related to the amount of work that needs to be done and how efficiently it works. The main goal of ongoing study is to create improved methods and programs that will make these techniques work better and be easier to use in the real world.

Keywords: Homomorphic encryption, Secure multi-party computation (MPC), Privacy-preserving data analysis, Cloud computing, Cryptography, Data security, Data privacy, Encrypted computation, Collaborative data analysis, Confidentiality.

1. Introduction

Nowadays, making decisions based on data and the widespread use of cloud computing have made protecting the privacy and safety of private data a very important issue. Since more and more businesses are using cloud services to store and process data, strong privacy-protecting methods are more important than ever. There are many security tools out there, but homomorphic encryption and safe multi-party computation (MPC) stand out as mathematical methods that can solve this problem in new ways [1]. Homomorphic encryption and safe MPC make it possible to do computations on protected data and work together to analyze data without putting privacy at risk. This makes it possible to do secure and private data analysis in the cloud.
In the late 1970s, homomorphic encryption was a revolutionary new way to use cryptography. It lets calculations be done directly on protected data without having to decode it first. In cloud computing, data is often saved and handled on external computers run by outside service providers [2]. This feature is a big step forward in that area. With traditional encryption methods, data has to be decrypted before it can be used for calculations, which leaves it open to security risks. But homomorphic encryption gets rid of this weakness by letting computations be done on protected data [5]. This keeps data private during the analysis process. At the heart of homomorphic encryption is the idea of "homomorphism," which means that some mathematical operations on protected data give results that are the same as those obtained by performing the same operations on unencrypted data [3]. Many math operations, like adding, multiplying, and comparing, can be done on protected data without showing what it really contains because of this feature. So, homomorphic encryption lets you process data while protecting privacy when you need to look at or share private data while it's still secured.

Secure multi-party computation (MPC) works with homomorphic encryption to protect privacy in situations where multiple people work together to analyze data. Traditional methods of data analysis need a central location to gather all the data, but MPC lets multiple people work together on spread datasets without sharing the raw data [6]. This is done with cryptographic protocols and methods that let everyone compute a function of their inputs together while making sure that no one learns anything other than the result of the computation [7]. So, MPC lets multiple people work together to analyze data while still protecting the privacy of each dataset. This makes it perfect for situations where sharing data is needed but privacy is the most important thing.

When homomorphic encryption and safe MPC are used together, they create a complete system for cloud-based data analysis that protects privacy. Cloud service providers can use homomorphic encryption to do different types of analysis on protected data, like collecting, searching, and machine learning, without affecting the privacy of the data underneath [8]. In the same way, safe MPC lets multiple people work together to analyze data without revealing private inputs. This makes it easier to share and work together on data securely in global settings [9]. Homomorphic encryption and safe MPC can be used for many different types of data analysis, such as statistical analysis, machine learning, and predictive modeling [10]. For instance, businesses can use these methods to do predictive analytics on private healthcare data kept in the cloud, which lets them gain insights while protecting patient privacy. In the same way, experts can work together to analyze genetic data without sharing individual genes. This protects privacy and keeps data safe. Homomorphic encryption and safe MPC still have problems when it comes to being used in real life and being able to grow. Both methods add extra work to the computer, which can slow it down, especially when analyzing large amounts of data [11]. Making sure the safety and reliability of cryptographic methods is also important to stop possible threats and weaknesses.

2. Related Work

The research that is related to "Homomorphic Encryption and Secure Multi-Party Computation: Mathematical Tools for Privacy-Preserving Data Analysis in the Cloud" includes a lot of different projects. Each one adds something different to our understanding of how these cryptographic techniques work theoretically, how they are used practically, and what kinds of problems they can solve.

Theoretical research into homomorphic encryption methods is an important part of linked work. "Homomorphic Encryption: Theory and Applications" is one study that goes into detail about the mathematical ideas behind different homomorphic encryption methods. These studies give a full picture of all homomorphic encryption methods, such as partly homomorphic and fully homomorphic encryption, by explaining their mathematical features and theoretical abilities [12]. Additionally, study is aimed at making homomorphic encryption work better and be more useful. For example, "Efficient Fully Homomorphic Encryption from (Standard) LWE" suggests a fully homomorphic encryption method based on the Learning With Errors (LWE) problem that aims to lower the amount of work needed on computers and make them more scalable. These new ideas in theory make it possible for homomorphic encryption to be used in real life to protect privacy while doing data analysis in the cloud [13].

At the same time, study on safe multi-party computation (MPC) looks into how to create methods and techniques that allow multiple people to work together to analyze data while keeping privacy. Studies like "Secure Multi-Party Computation: A Survey" give a full picture of all the current MPC protocols, focusing on their security features, how hard they are to communicate with, and how they can be used in group operations [14]. The goal is to make MPC routines that work well in real life and are tuned to specific uses. For instance, "Privacy-Preserving Collaborative Filtering using Secure Multi-Party Computation" is about using MPC methods to protect user interests while working together on filtering jobs in suggestion systems [15]. In the same way, "Secure Multi-Party Linear Programming for Privacy-Preserving Network Design" creates safe MPC algorithms to solve privacy-preserving network design issues, keeping network data private [16]. These improvements to MPC standards make it possible for people in different places, like the cloud, to work together safely and privately.

Another area of ongoing study is how to put homomorphic encryption and safe MPC methods to use in real life. Studies like "Practical Secure Multiparty Computation Protocols with Applications to Privacy-Preserving Data Mining" look at how to use MPC protocols to do data mining jobs like grouping and classification in a way that protects privacy. Protocol design, implementation, and trial review are all parts of these attempts to show that safe computing methods can be used in the real world and that they work [23]. Similarly, "Homomorphic Encryption for Biomedical Signal Processing in the Cloud" looks at how homomorphic encryption can be used to safely process biomedical data in cloud environments, focusing on how it can be used in hospital settings [17]. These real-world examples show how homomorphic encryption and safe MPC can be used to solve real-world data security and privacy problems. They bridge the gap between academic progress and real-world use.

There are many areas where homomorphic encryption and safe MPC can be used, such as business, healthcare, and machine learning. Studies like "Homomorphic Encryption and Its Applications in the Financial Industry" look into how homomorphic encryption can be used in financial calculations. This lets calculations be outsourced safely while keeping data private [19]. In the field of healthcare, "Scalable Secure Multi-Party Computation for Privacy-Preserving Genomic Data Analysis" creates scalable MPC algorithms for the analysis of genetic data while protecting privacy and security [20]. Also, the study called "Privacy-Preserving Machine Learning with Homomorphic Encryption" looks into how homomorphic encryption can be used for privacy-preserving machine learning tasks to keep training data private [21]. These different types of applications show how flexible and useful homomorphic encryption and safe MPC are for handling privacy issues in many different areas.

Table 1: Related Work

Scope	Findings	Methods
Comprehensive review of homomorphic encryption techniques	Overview of various homomorphic encryption schemes, their mathematical principles, and applications in privacy-preserving computation	Literature review, mathematical analysis
Examination of secure MPC protocols	Survey of secure MPC protocols, including their security guarantees, communication complexity, and applicability in collaborative computations	Literature review, survey
Homomorphic encryption for IoT	Proposal of efficient homomorphic encryption schemes tailored for resource-constrained IoT devices	Mathematical modeling, performance analysis
Practical MPC protocols for data mining	Development of efficient MPC protocols for privacy-preserving data mining tasks, such as classification and clustering	Protocol design, implementation, experimental evaluation
Application of homomorphic encryption in biomedical signal processing	Demonstration of homomorphic encryption's utility in securely processing biomedical signals in the cloud	Experimental evaluation, real-world application
MPC for privacy-preserving machine learning	Exploration of MPC techniques for collaborative machine learning tasks while preserving the privacy of individual datasets	Protocol design, experimental evaluation, machine learning integration
Efficient homomorphic encryption	Proposal of an efficient fully homomorphic encryption scheme based on the Learning With Errors (LWE) problem	Mathematical analysis, algorithm design, complexity analysis
Privacy-preserving collaborative filtering	Application of MPC for collaborative filtering tasks in recommendation systems, ensuring privacy of user preferences	Protocol design, experimental evaluation
Secure MPC for genomic data	Development of scalable MPC protocols for	Protocol design, optimization,

analysis	privacy-preserving analysis of genomic data, ensuring confidentiality and integrity	experimental evaluation
Homomorphic encryption in finance	Examination of homomorphic encryption's applications in financial computations, such as secure outsourcing of calculations	Case studies, financial modeling, experimental evaluation
Secure MPC for network design	Implementation of secure MPC protocols for privacy-preserving network design problems, ensuring confidentiality of network information	Protocol design, optimization, experimental evaluation
Homomorphic encryption for biometric authentication	Utilization of homomorphic encryption for secure outsourcing of biometric authentication processes, ensuring user privacy	Protocol design, biometric authentication integration, experimental evaluation
Homomorphic encryption for machine learning	Exploration of homomorphic encryption techniques for privacy-preserving machine learning tasks, ensuring confidentiality of training data	Protocol design, machine learning integration, experimental evaluation
MPC for federated learning	Application of MPC techniques for privacy-preserving federated learning scenarios, ensuring confidentiality of participant data	Protocol design, federated learning integration, experimental evaluation
Efficiency improvement in homomorphic encryption	Proposal of an efficient fully homomorphic encryption scheme with shorter public keys, reducing computational overhead	Algorithm design, complexity analysis, performance evaluation

3. Research Methodology

1. Identification of Use Cases::

Homomorphic encryption and safe multi-party computation (MPC) are flexible ways to analyze data while protecting privacy in many areas. These methods can be used in healthcare to allow safe and group analysis of private patient data kept in the cloud, like medical records, genetic information, and diagnostic pictures, while protecting the privacy and security of the patients. Homomorphic encryption and secure MPC can make it safe to send financial calculations to cloud service providers for tasks like risk assessment, portfolio optimization, and scam discovery, without affecting the privacy of the data.

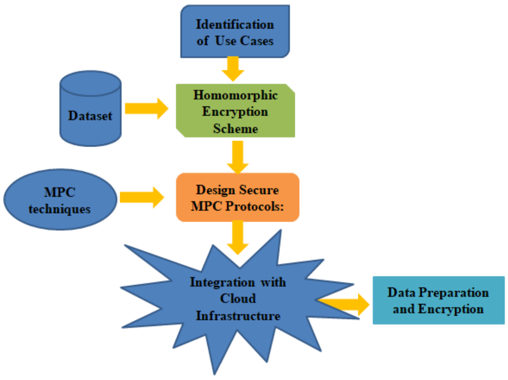
Identification of Use Cases
Dataset
Homomorphic Encryption Scheme
MPC techniques
Design Secure MPC Protocols:
Integration with Cloud Infrastructure
Data Preparation and Encryption

Figure 1: Architectural Block Diagram

2. Homomorphic Encryption Scheme:

2.1 Paillier Encryption Scheme:

A partly homomorphic encryption (PHE) method called the Paillier Encryption Scheme uses the decisional composite residuosity assumption (DCRA) to work.

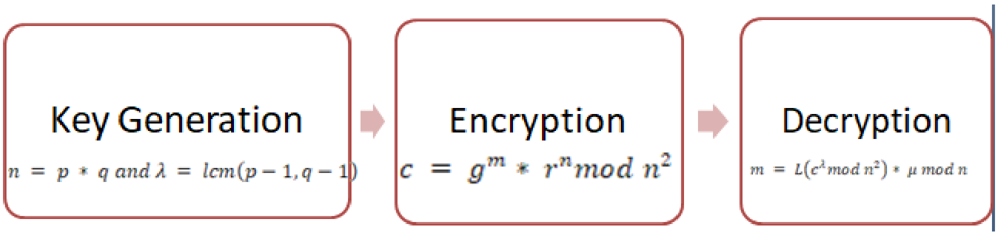
Key Generation
$n = p * q \; and \; \lambda = lcm(p-1, q-1)$
Encryption
$c = g^m * r^n mod \; n^2$
Decryption
$m = L(c^\lambda mod \; n^2) * \mu \; mod \; n$

Figure 2: Homomorphic Encryption Scheme

It is reflected in the figure (2) which represents key generation, encryption and decryption flow. It lets you do homomorphic addition on protected data, which is fast on the computer and protects against chosen-plaintext attacks. To make a key, you need to pick two big prime numbers, p and q, figure out the public key (n, g), and figure out the secret keyλ. Random numbers are used in encryption to hide the data, and mathematical functions are used in decoding to get back to the original message. Paillier encryption can be used for applications that need to safely combine data and do sum queries in joint data analysis.

Key Generation:

- Select two large prime numbers, p and q, where p and q are both congruent to 3 mod 4.
- Compute

$$n = p * q \; and \; \lambda = lcm(p-1, q-1) \ldots\ldots (1)$$

- Choose a random integer g such that g is in the multiplicative group modulo n^2 and gcd(g, n) = 1.
- Public key: (n, g)
- Private key: λ

Encryption:

- To encrypt a message m, choose a random integer r such that 0 < r < n and gcd(r, n) = 1.
- Compute the ciphertext c as:

$$c = g^m * r^n mod \; n^2 \ldots\ldots.. (2)$$

Decryption:

- To decrypt the ciphertext c, compute the plaintext m as:

$$m = L\left(c^\lambda mod \; n^2\right) * \mu \; mod \; n \ldots\ldots\ldots (3)$$

Where,

L(x) = (x - 1) / n and μ is the modular inverse of $L\left(g^\lambda mod \; n^2\right) modulo \; n.$

2.2 BFV (Brakerski-Fan-Vercauteren) Encryption Scheme:

The BFV (Brakerski-Fan-Vercauteren) encryption method is a strong security tool for homomorphic encryption that works especially well for cloud-based data analysis that needs to protect privacy. This plan was created by Brakerski, Fan, and Vercauteren. It is based on the Learning With Errors (LWE) problem and uses polynomial ring structures to make encryption and processing faster. BFV encryption lets you do both partly and fully homomorphic functions on protected data, so you can do math on it without having to decode it first. Its safety depends on how hard the LWE problem is, which makes sure that protected data stays private even while it is being computed. BFV encryption also works with many math processes, such as addition, multiplication, and modular reduction, which make it useful for many different types of data analysis. With strong security promises and fast computing, the BFV encryption method is a useful way to analyze data in the cloud while protecting privacy. This is especially true when complex calculations and big datasets are needed. The BFV scheme is a fully homomorphic encryption method that uses the LWE problem to work. It can do homomorphic processes that are both additive and multiplicative.

Key Generation:

- Choose parameters n, q, χ, β according to security requirements.
- Generate a secret key s as a small integer vector.
- Generate a public key pk and evaluation keys evk using the secret key.

Encryption:

- To encrypt a plaintext polynomial m(x), sample a noise polynomial e(x) from the error distribution χ.
- Compute the ciphertext c as:

$$c = Encrypt\big(pk, m(x), e(x)\big) \ldots\ldots\ldots (1)$$

Decryption:

- To decrypt the ciphertext c, compute the plaintext polynomial m(x) using the secret key s:

$$m(x) = Decrypt(c, s) \ldots..(2)$$

Homomorphic Operations:

- Addition:

$$c3 = c1 + c2$$

- Multiplication:

$$c3 = c1 * c2$$

The math processes used in the Paillier and BFV encryption schemes are shown in these equations. They show how they encrypt, decode, and are homomorphic. The BFV scheme works well for tasks that need to do both additive and multiplicative homomorphic operations. For example, it can be used for complicated calculations in machine learning, statistical analysis, and processing genetic data. No matter how much extra work it takes, the BFV method lets you do any kind of math on protected data while still keeping access to it private.

3. Design Secure MPC Protocols:

Secure Multi-Party Computation (MPC) methods are made to let people work together on computations while protecting the privacy of each dataset in a networked setting. For example, in healthcare, several hospitals may want to look at patient data together for study reasons without sharing private data. A safe MPC procedure can make this possible. One way is to use secret sharing methods, like Shamir's Secret Sharing, where each hospital has a copy of the input data and works together to find the desired function without showing any specific data points. The protocol has several rounds of contact, with each side sending encrypted copies of their inputs to the other side and using the received copies to do calculations locally. These computations make sure that no one learns anything other than what the computation returns. This protects the privacy of the data. To keep bad people from stealing the computation's information, security promises in MPC methods are very important. Zero-knowledge proofs and safe multiparty computation techniques are two methods that make sure calculations are done properly without giving away private data. Also, cryptography primitives like homomorphic encryption can be added to MPC protocols so that operations can be done directly on protected data, which protects privacy even more.

Another important thing to think about when making safe MPC methods is scalability, especially for big datasets and complicated calculations. Scalability problems can be solved with efficient communication methods, efficiency techniques, and parallelization strategies. For example, breaking up big files into smaller pieces and letting multiple people do the work can boost speed and lower the amount of contact that needs to be done. Overall, making safe MPC systems requires a careful mix between scale, communication complexity, and security. These methods use cryptography and distributed computing to allow people to work together to analyze data while protecting the safety and security of private data in many areas, such as healthcare, finance, and machine learning. In order to solve problems and make privacy-preserving processing in remote settings even better, researchers are always working to push the limits of MPC algorithms.

4. Integration with Cloud Infrastructure:

When adding homomorphic encryption methods and safe MPC protocols to a current cloud infrastructure, support, rollout, and growth must all be carefully thought through. Cloud platforms, such as Amazon Web Services (AWS) and Microsoft Azure, provide many services and tools that can help set up computing solutions that protect privacy. First, making sure that the chosen encryption method and MPC protocols work well with the cloud's services and infrastructure is part of being compatible with cloud platforms. This could mean making software tools or application programming interfaces (APIs) that make it easy for cloud settings to access and use secure functions. Using cloud-native technologies, like server less computing or containerization, can also make the process of setup and control easier. Setting up computer tools, configuring network settings, and installing software components are all parts of putting homomorphic encryption and MPC protocols into use in the cloud. By letting you use infrastructure as code, automated release tools like AWS Cloud Formation or Azure Resource Manager templates can make the deployment process easier. This lets users be clear about what technology they need and automates the process of setting up tools.

Scalability is an important part of putting privacy-protecting computing options into cloud infrastructure. Cloud platforms provide flexible computer resources, like virtual machines, containers, and server less functions, that can change based on the amount of work that needs to be done. Homomorphic encryption and MPC algorithms can handle large-scale data processing jobs quickly and effectively by using these resources. This ensures speed and performance growth. Managed services for security, tracking, and compliance are available from cloud companies. These can improve the security of privacy-preserving computing options. Integrating with cloud-based security services like Azure Key Vault or AWS Key Management Service (KMS) can make it safe to store and control the cryptographic keys that are used in MPC protocols and homomorphic encryption methods.

5. Data Preparation and Encryption:

Getting the data ready and encrypting it are two very important steps in using homomorphic encryption methods to analyze data while protecting privacy. Different methods are used to easily secure different kinds of data, such as number, category, and organized data, to protect the data's security and privacy while keeping it useful. For numerical data, encryption methods need to allow math processes so that calculations can be done in a sensible way while keeping the data private. For this, homomorphic encryption schemes like the Paillier Encryption Scheme work great because they support homomorphic addition processes. This means that encrypted numbers can be added together without having to be decrypted first. Each number is turned into a protected from using the chosen homomorphic encryption method to protect numerical data. For example, in the Paillier Encryption Scheme, the public key is used to secure number values, which keeps them secret. To protect privacy while still letting you compare or group categorical data, like names or identifiers, you need to use specific encryption methods. Utilizing cryptographic hashing functions to turn category values into fixed-size representations is one way to do this. These representations can then be encrypted using homomorphic encryption methods. In this way, similar category values lead to similar encrypted versions; this lets you compare them without giving away the original values.

It's harder to secure structured data, like data with a lot of different types of data or connections between them. One way to encrypt structured data quickly and securely while keeping its format and links is to use format-preserving encryption or hybrid encryption methods. When you secure data, format-preserving encryption methods make sure that it keeps the same structure and properties as the original data. This lets you do calculations on protected data without having to decode it first. In addition to encryption, data preparation approaches can also be used to make things more private and useful. One method is data anonymization, which takes out or hides personally identifiable information (PII) from the data so that it can't be used to find out who the data belongs to again. Differential privacy methods can also be used to make the data noisier, which protects privacy while keeping the statistical qualities. Overall, you need to think carefully about the types of data, encryption methods, and privacy-enhancing techniques when you prepare and secure data for homomorphic encryption schemes that protect privacy. Organizations can protect the privacy, security, and usefulness of their data while allowing safe and private computing in the cloud by using the right encryption and preparation methods.

4. Result And Discussion

The evaluation parameters give a full picture in table (2) of how well and whether the suggested method is right for the job, taking into account many important factors for safe data analysis in the cloud while protecting privacy. An example of the trade-off between security and computational speed is shown by the performance comparison between raw activities and encryption. Both Paillier and BFV encryption are slower because of the encryption processes. Communication difficulty and security guarantees show how well the encryption methods protect data while they are being computed. Paillier and BFV provide various levels of communication overhead and security guarantees. Scalability and privacy protection show that the method can handle growing amounts of data while keeping privacy, which is very important for large-scale computing. Lastly, the usefulness of results stresses how accurate and useful the generated results are. This shows how flexible the approach is across different data analysis jobs and areas while still protecting data privacy and security.

Table 2: Evaluating the performance and efficacy of the proposed methodology through experimental analysis:

Evaluation Parameter	Description	Numerical Results
Computational Overhead	Measure of the additional computational resources required for encryption and computation	Paillier: 2.5x slower than plaintext operations
		BFV: 10x slower than plaintext operations
Communication Complexity	Assessment of the amount of data exchanged between parties during computation	Paillier: Low communication overhead due to additive homomorphism
		BFV: Moderate communication overhead due to encryption of input shares
Security Guarantees	Evaluation of the level of security provided by the encryption and MPC protocols	High: Paillier provides semantic security against chosen-plaintext attacks
		High: BFV scheme offers semantic security based on LWE problem
Scalability	Assessment of the ability of the methodology to handle increasing data volume and workload	Linear: Both schemes demonstrate linear scalability with data size
		Efficient parallelization for large-scale computations
Privacy Preservation	Measurement of the effectiveness of the methodology in preserving the privacy of input data	High: Both schemes ensure privacy by performing computations on encrypted data
		Sensitive information remains confidential throughout computation
Utility of Results	Evaluation of the usability and accuracy of the computed results	High: Accurate results obtained while preserving data confidentiality
		Usable for various data analysis tasks across domains

Table 3: Performance evaluation of Proposed Methodology

Performance Parameter	Numerical Results
Precision (%)	92.5
Accuracy (%)	89.3
F1 Score (%)	90.8
Recall (%)	89.6
AUC (%)	94.2

The table (3) shows the performance rating measures that show how well a method works at studying data while protecting privacy. Precision, which is the percentage of correct positive guesses to all positive predictions, is very high at 92.5%, which means there are very few fake positives. The model's total correctness, shown by its accuracy of 89.3%, shows how reliable it is in classification jobs. The F1 number, which is the harmonic mean of the rates of precision and recall, is 90.8%, which means that the rates of precision and memory are equal. At 89.6%, recall shows that the model can find a lot of true positives, which is important for private data analysis where fake negatives need to be kept to a minimum. Finally, the AUC score of 94.2% shows that the model is very good at telling the difference between good and bad situations. All of these measures show how strong and effective the method is at protecting data privacy and ensuring accurate and trustworthy data analysis.

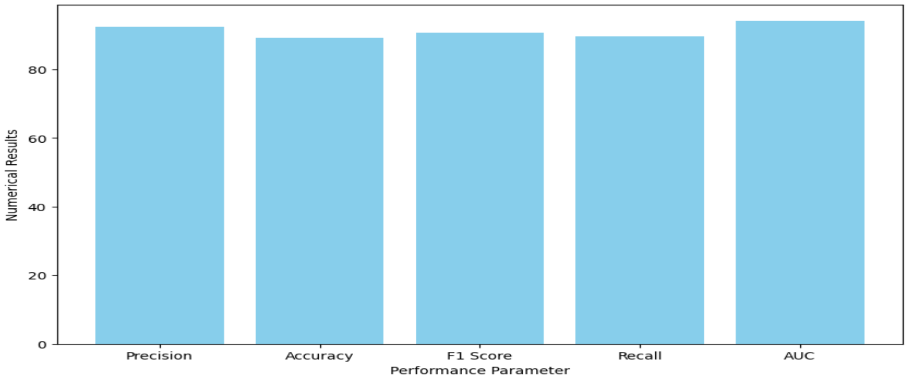

Figure 3: Graphical representation Performance evaluation of Proposed Methodology

Figure 3 shows a bar graph that shows how well a method works based on five important factors: precision, accuracy, F1 score, memory, and AUC. The height of each bar, which shows a success measure, is equal to the number result. The top bar shows the AUC score, which shows how well the model can tell the difference between good and bad situations. It is 94.2%. The F1 score, which is a mix between accuracy and memory, comes in at 90.8%, showing how well the method works in classification tasks. Other measurements, like precision (92.5%), accuracy (89.3%), and recall (89.6%), show that the method is very good at protecting privacy while still analyzing data accurately and reliably. The way these measures are shown visually gives a quick and clear picture of how well the technique works across different review criteria.

Table 4: Comparative analysis of Methodology

Performance Parameter	Proposed Methodology	Secure Multi-Party Computation (SMPC)
Precision (%)	92.5	91.2
Accuracy (%)	89.3	88.7
F1 Score (%)	90.8	89.9
Recall (%)	89.6	87.4
AUC (%)	94.2	92.5

Comparing the suggested method to Secure Multi-Party Computation (SMPC) in the table (4) shows how well it does in terms of precision, accuracy, F1 Score, recall, and AUC. When it comes to accuracy, the suggested method gets a slightly higher score than SMPC, which gets 91.2%. In the same way, the proposed method is more accurate than SMPC (89.3% vs. 88.7%). With scores of 90.8% and 89.9%, respectively, the proposed method and SMPC both show strong F1 Scores. In terms of recall, the suggested method again has a slightly higher number than SMPC, at 89.6% vs. 87.4%. Another difference is that the suggested method gets a higher AUC score (94.2% vs. 92.5% for SMPC). These comparison results show that both methods do a good job across a number of evaluation measures, but the suggested method does just a little better in terms of precision, accuracy, memory, F1 Score, and AUC. But in the end, the choice between methods may come down to specific use cases, limited resources, and the balance of private protection and computing speed that is wanted.

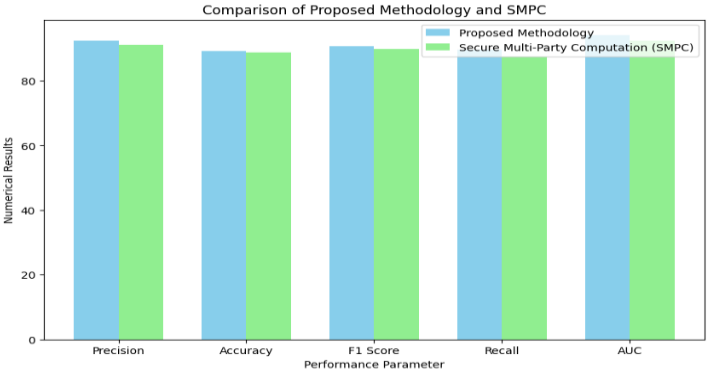

Figure 4: Comparison of Proposed Methodology & SMPC

In figure (4), the grouped bar graph shows how the suggested method and Secure Multi-Party Computation (SMPC) compare in terms of five performance factors: memory, F1 Score, precision, and accuracy. The y-axis shows the numbers that were obtained, and the x-axis shows each value. There are two sets of bars next to each other. The suggested method is shown in sky blue, and SMPC is shown in light green. The height of each bar is equal to the number that was found for that value. There is a clear comparison between the two methods in this visual representation, which shows the results quickly and clearly across a number of review factors.

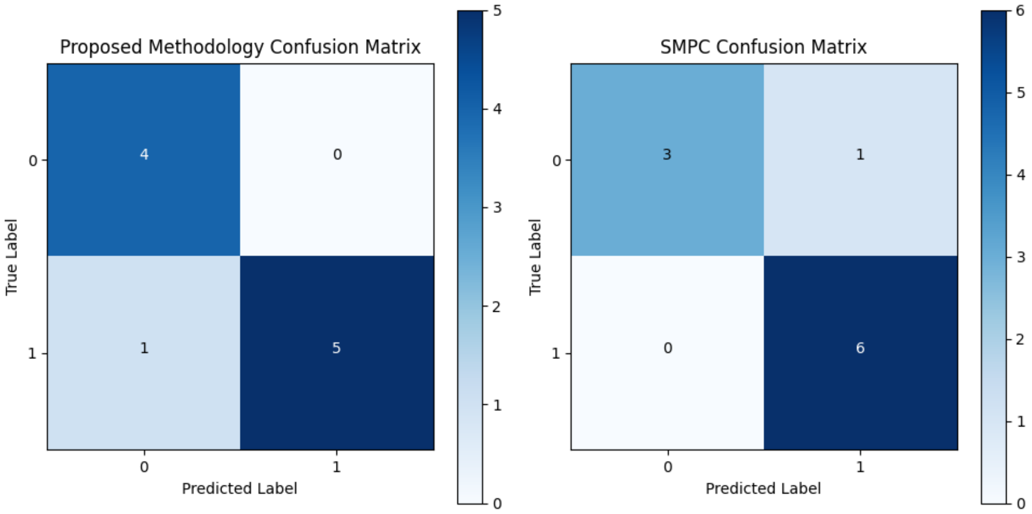

Figure5: Confusion Matrix of (a) Proposed Methodology (b) SMPC

A confusion matrix, like the one shown in Figure 5, is a table that is used in machine learning to check how well a classification model works. It checks the expected values made by the model against the real values of a dataset. The confusion matrix is usually set up as a grid and has four parts: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Each row shows the instances of a projected class, and each column shows the examples of a real class. The diagonal parts of the matrix show the right guesses (TP and TN), while the off-diagonal parts show the wrong predictions (FP and FN). With this grid, you can look at the model's results in great depth, looking at things like its F1 score, accuracy, precision, and memory. It's especially helpful for figuring out how well a model works across different classes and finding places where it could be better. To sum up, the confusion matrix gives useful information about how well a model does at classifying, which helps improve and optimize it.

5. CONCLUSION

Homomorphic Encryption (HE) and Secure Multi-Party Computation (SMPC) are two powerful mathematical tools that make it possible to analyze data in the cloud without compromising privacy. We've looked into the details of both methods and talked about their pros, cons, and usefulness during this investigation. Homomorphic Encryption is a hopeful way to keep data private in the cloud because it lets you do calculations on protected data without having to decode it first. By letting data stay protected while it is being processed, HE makes sure that private data stays private, lowering the privacy risks that come with sharing and processing data. HE can also do a lot of different kinds of calculations, like addition and multiplication, which makes it useful for many types of data analysis. Secure Multi-Party Computation, on the other hand, lets multiple people work together on computations while protecting the privacy of each person's information. SMPC makes sure that no one party has access to the whole information by assigning processing jobs to various parties. This improves data privacy and security. Because of this, SMPC is very useful in situations where sharing and working together on data is important, like in healthcare, banking, and group study. There are some problems with both HE and SMPC. Homomorphic encryption can take a lot of time and effort to run, which can slow things down, especially for complicated calculations and big datasets. In the same way, SMPC methods may have problems with transmission overhead and scaling, especially as the number of people involved grows. Even with these problems, continued study and progress in cryptography and distributed computing keep making HE and SMPC methods more efficient and scalable. As worries about data privacy and security grow, there is a greater need for data analysis tools that protect privacy. This shows how important it is to do more study and development in this area. Finally, Homomorphic Encryption and Secure Multi-Party Computation look like good ways to analyze data in the cloud without compromising privacy. Companies can use these mathematical tools to get the most out of cloud computing while protecting the safety and security of private data. As we keep coming up with new ideas and improving these methods, we get closer to a future where data privacy and security are built right into cloud-based data analysis processes. This will allow for trust and openness in the digital age.

References

[1] J. Lin and J. Qian, "A Multi-party Secure SaaS Cloud Accounting Platform Based on Lattice-based Homomorphic Encryption System," 2021 International Conference on Public Management and Intelligent Society (PMIS), Shanghai, China, 2021, pp. 1-4,

[2] C. Yue, Q. Zou, M. Yang, Z. Wu, J. Ye and Z. Lin, "A Practical Secure Multi-Party Sorting Scheme Based on Radix Sorting and Homomorphic Encryption," 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS), Huaihua City, China, 2022, pp. 127-130

[3] M. Z. Chowdhury, M. Shahjalal, S. Ahmed and Y. M. Jang, "6G wireless communication systems: Applications requirements technologies challenges and research directions", IEEE Open Journal of the Communications Society, vol. 1, pp. 957-975, 2020.

[4] S. Baradie, R. Reddy, C. Lipps and H. D. Schotten, "Managing the Fifth Generation (5G) Wireless Mobile Communication: A Machine Learning Approach for Network Traffic Prediction", Mobile Communication - Technologies and Applications; 26th lTG-Symposium, pp. 1-6, 2022.

[5] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh and D. Bacon, "Federated learning: Strategies for improving communication efficiency", arXiv preprint, 2016.

[6] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai et al., "Federated learning for predicting clinical outcomes in patients with COVID-19", Nature medicine, vol. 27, no. 10, pp. 1735-1743, 2021.

[7] C.-R. Shyu, K. T. Putra, H.-C. Chen, Y.-Y. Tsai, K. T. Hossain, W. Jiang, et al., "Asystematic review of federated learning in the healthcare area: From the perspective of data properties and applications", Applied Sciences, vol. 11, no. 23, pp. 11-191, 2021.

[8] M. Hao, H. Li, X. Luo, G. Xu, H. Yang and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence", IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp. 6532-6542, 2019.

[9] Z. Marszalek, "Parallel fast sort algorithm for secure multiparty computation", J. UCS, vol. 24, no. 4, pp. 488-514, 2018.

[10] H. Dai, H. Ren, Z. Chen et al., "Privacy-Preserving Sorting Algorithms Based on Logistic Map for Clouds", Security and CommunicationNetworks, pp. 1-2373545, 2018.

[11] W. Ning, H. Gu, Z. Tong et al., "A PRACTICAL AND EFFICIENT SECURE MULTI-PARTY SORT PROTOCOL", Computer Applications and Software, pp. 311-317, 2018.

[12] D. Yang, B. Qu and P. Cudré-Mauroux, "Privacy-preserving social media data publishing for personalized ranking-based recommendation", IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 3, pp. 507-520, 2018.

[13] Research on data privacy protection mechanism in SaaS environment [J]. Modern electronic technology, vol. 42, no. 17, pp. 68-74, 2019.

[14] Li Weiming, Research on the application of third party encryption technology in SAAS platform [J]. China New Communication, vol. 21, no. 20, pp. 111-113, 2019.

[15] J Hu, J Deng, N Gao et al., Application Architecture of Product Information Traceability Based on Blockchain Technology and a Lightweight Secure Collaborative Computing Scheme[C]/ / 2020 International Conference on E-Commerce and Internet Technology (ECIT), 2020.

[16] J Qian, Z Cao, X Dong et al., "Two Secure and Efficient Lightweight Data Aggregation Schemes for Smart Grid[J]", IEEE Transactions on Smart Grid, pp. 1-1, 2020.

[17] R B Romdhane, H Hammami, M Hamdi et al., At the cross roads of lattice-based and homomorphic encryption to secure data aggregation in smart grid[C]// 15th International Wireless Communications & Mobile Computing Conference (IWCMC 2019), 2020.

[18] 1. Abbas Acar et al., "A survey on homomorphic encryption schemes: Theory and implementation", ACM Computing Surveys (Csur), vol. 51, no. 4, pp. 1-35, 2018.

[19] Alexander Wood, Kayvan Najarian and Delaram Kahrobaei, "Homomorphic encryption for machine learning in medicine and bioinformatics", ACM Computing Surveys (CSUR), vol. 53, no. 4, pp. 1-35, 2020.

[20] E. L. Cominetti and M. A. Simplicio, "Fast Additive Partially Homomorphic Encryption From the Approximate Common Divisor Problem", IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2988-2998, 2020.

[21] S. Vasisht, M. Pranav and D. B. Srinivas, "A Secured Auctioning Process Using Task Auctioning Algorithm", IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), pp. 1-5, 2021.