ISSN : 0973 - 8355

www.ijmmsa.com

IJMMSA

INTERNATIONAL JOURNAL OF MATHEMATICAL, MODELLING, SIMULATIONS AND APPLICATIONS

E-MAIL
editor.ijmmsa@gmail.com
editor@ijmmsa.com

INTERNATIONAL JOURNAL OF MATHEMATICAL, MODELLING, SIMULATIONS AND APPLICATIONS
IJMMSA

ISSN 0973-8355www.ijmmsa.com
Vol 7,Issue 2.June 2019

“Protecting Cloud DWs using Secure Data Management Systems

[1]D.Wasiha Tasneem, [2]M.Mahaboob Peera ,[3]M.Prathap

ABSTRACT”

Information systems that enable data-oriented analysis and the improvement of company performance are known as decision support systems (DSS). Because of the value it may bring enterprises, cloud-based decision support systems are becoming more popular. However, due to the sensitivity of decision support data, data privacy is still a big concern. To ensure that cloud-based decision support systems are secure, this paper examines the security and cryptography protocols in place and addresses current research challenges related to them.

1.INTRODUCTION

When you utilize cloud computing, you pay only for what you use on the Internet. With lower startup costs and a way of adapting to fluctuations in system demand, the flexibility that cloud computing provides is particularly enticing to many enterprises, especially midsized and small ones. Cloud-based data outsourcing is especially intriguing. Decision-support data, such as personal, health, and business records, are more critical[1]. Analytical databases, such as data warehouses (DWs), serve as the cornerstone of decision support systems. Central repositories, such as data warehouses (DW), have traditionally housed decision support data, allowing for on-the-fly analytical processing of historical data from many sources (OLAP). Security concerns arise when cloud-based decision-support data is used outside. Traditional security techniques are no longer enough to secure data in the cloud from both internal and external threats[2].

As part of this article, we examine the various security measures that may be used to secure cloud decision-making data. “Differential privacy (Section 2), threshold cryptography (Section 3), and cryptographic systems that enable computations over encrypted data are the three categories into which we place these security techniques (Section4) [3][4] These issues and obstacles to implementation are addressed in Section 5 of our paper.” There is nothing further to say in Section 6.

2.DIFFERENTIALPRIVACY

Normally, confidential information cannot be accessed in this manner. Authorized users, on the other hand, have the ability to extrapolate sensitive data from query results by combining those results with outside information. Linkage and probabilistic assaults are two types of threats that pose a risk to the network. If you are concerned about protecting your personal information, take steps like removing names and social security numbers or using data anonymization techniques like perturbation and masking[5].

The original data is simply replaced when using data masking. [6] Some database management systems (DBMSs) already have these functionalities built in. Data may be obscured by techniques such as k-anonymity, diversity, and t-closeness, which create noise into the data. To maintain a healthy balance between privacy and usefulness, such strategies must be deployed with extreme care.

[1,2,3]Assistant Professor
[1,2,3] Department of Computer Science & Engineering,
[1,2,3]Dr.K.V.Subba Reddy College Of Engineering For Women

Privacy-preserving OLAP in a distributed setting may be done by perturbing and reconstructing tables [1]. Count aggregates may be reconstructed using algorithms.

The sum and the minimum aggregation functions are not supported. Addition of noise in response to inquiry is also employed [8], with numerous ways automatically altering scales of noise for reduction of relative mistakes while still preserving differential privacy.

Anonymizing numerical characteristics [10], such as KPIs in DWs, is made easier by (k, e)-anonymity [k, e]-anonymization. When aggregating data from huge databases, privacy is maintained by using partitioning and permutation. While a perturbation technique eliminates privacy breaches, it may also lead to inaccuracies in aggregate outcomes.

3.**THRESHOLDCRYPTOGRAPHY**

Under specific conditions, a group of authorities may be granted access to material that would otherwise be inaccessible to just one person or organization. This is known as threshold cryptography. To prevent a single point of attack, confidential data must be kept in several places.

Shamir proposed threshold cryptosystems in 1979 [11]. Each participant must reassemble the secret after it has been held at each of the n participants' locations for t minutes or longer in Shamir's version of secret sharing. It is easy to foresee cloud service providers participating in secret sharing (CSPs). ElGamal cryptography and secret sharing are combined for the first time in a threshold encryption technique [10]. With the addition of signatures, it can be verified and resilient [7].

Data management context protocols utilize threshold encryption to allow parties to work together on aggregate searches without revealing intermediate results to one other. Users may also employ signatures to verify the accuracy of query results. Although this protocol is limited to a certain database with preset schemas, it does not allow for aggregations beyond sums and averages. Exact match and range searches may be performed with the use of index keys as predicates. There are, however, no plans to answer aggregate requests.

I. “COMPUTATIONOVERENCRYPTEDDATA

SuitableCryptographicSchemes”

HomomorphicCryptography:If you want to process encrypted data, you will often need to decode them first. Cloud-based data processing implies that the CSP has full access to the data, which is unwanted. Only homomorphic encryption (HE) provides for unrestricted computation without the necessity for decryption while working with encrypted material.

With completely homomorphic encryption, any operation on encrypted data may be implemented (FHE).Because of its high computational requirements, this method cannot be applied in the real world Building an effective and useful FHE remains a difficulty, despite several advancements, such as lowering the size of the encryption key or removing the bootstrapping mechanism from the protocol. The Paillier encryption system, for example, allows only a limited amount of operations to be performed on the encrypted data. SWHEs are well-represented in the HELib1 library, which offers several implementations and optimizations.

FunctionalEncryption:Recently, functional encryption (FE) has taken the cryptography world by storm. FE is a public key encryption method that allows for "partial" decryption keys to be generated. For each function f, the data owner sets a secret key that can be used to calculate f(x) over encrypted data quickly. When SKf is used to decrypt cipher c = E(pk, m) it just shows f(m) and nothing more.

However, traditional public-key encryption falls within the FE umbrella as well. SKf, for example, is a user identification or a collection of characteristics in identity or attribute-based encryption[5].While FHE generates encrypted results, FE's output is available to the user in plain text as a last point of differentiation.

With its non-interactive nature and wide range of applications, for example in spam filtering on encrypted emails, large-data mining, and distributing computation over several clients, functional evaluation is an excellent choice. Execution issues, on the other hand, must be addressed.

QueryingEncryptedData

Comparison: Order preservation encryption (OPE) or multiparty computation may be used to compare encrypted data (OPE). SMC provides a technique for comparing two private variables using the "greater than" comparison operator based on Yao's millionaire problem (8). In the meanwhile, even if SMC procedures have been shown to be safe, they cannot be used.

Range queries over encrypted data may be performed using OPE while preserving ciphertext order.

Using OPE to encrypt data exposes the data's sequence, rendering it vulnerable to plaintext attacks. Plaintext distribution patterns may be used by an attacker to map the encryption key or infer encrypted values.

Encrypting the same data several times while keeping the order of the original values, known as multivalued OPE (MVOPE), is one way to get around this problem. MV-OPE can be used to perform operations such as =,,>,,, G, min; max and count.

Search: To access data saved on untrusted CSPs, encrypted data must be accessed by searching through encrypted data. The easiest way to search through encrypted entries in a database is via Deterministic Encryption. It is possible to do a quick search on encrypted data, for example, using the Encrypt-with-Hash deterministic technique. However, the server is exposed to a little amount of information. However, the search is slower with other methodologies, which provide better security assurances.

SQLQuerying:Encrypted data may be queried using SQL queries without having to decode it. Bucketization uses the database's distributional features to partition the plain text space into buckets. IDs and maximum and minimum values for each bucket are provided. The bucket's ID is used as a tag for each piece of data. Query requests from clients are first translated into bucket-level inquiries and then submitted to the CSP. Data objects are then analyzed using simply their index tags, which include the relevant information. There are certain advantages to using bucketization as a query processing strategy. A large number of false positives are possible. A post-processing step is thus necessary. Bucketization is a method for answering multidimensional range queries on multidimensional data that computes safe indexing tags of data to avoid the server from finding individual values. In order to reduce the danger of disclosure, bucketization is treated as an optimization issue. In order to assist data owners balance disclosure risk and expense, a threshold has been established.

1) SecureDataManagementSystems

A safe database management system, although desired in our cloud DW model, is seldom discussed in academic literature. Nonetheless, there have been a few ideas put out. To yet, none of the options we have looked at in this section have addressed the most pressing issue of safe database management systems: expressly guaranteeing privacy.

Bucketization-based Using a partitioning index, we have devised new strategies that let the CSPs handle the majority of SQL execution. As a result, the remainder of the query must be decrypted by the client before it can be run. We present an algebraic framework for reducing the amount of processing the client has to do. Any SQL query, including aggregation queries, joins, and grouping, may be executed using this method. To finish an enquiry, you must speak with the customer. The CSPs execute row filtering.

As a result, it was possible to perform aggregation on encrypted data without decryption (sum, count, average, minimum, and maximum). Two major types of queries are used in SQL aggregation, which are known as "certain" and "maybe" queries. To calculate final results, certain queries may not need to be relayed back to the client, while others can aggregate data at CSPs. Unfortunately, it has been shown that this method is susceptible to simple cipher text-only assaults.

Encrypted data may be queried using CryptDB, which provides privacy protection and controls queries. On the basis of queries seen at run-time, CryptDB applies one or more specific symmetric key encryption schemes for each specific data item. For passwords, CryptDB employs a chain of encryption keys. So long as the user logs in, an attacker can not get their hands on their personal information. Multiple ciphertexts may be stored inside each other in CryptDB's onion levels of encrypting. While the outermost layers provide the greatest protection, interior layers are more practical but less secure.

It employs a trustworthy proxy server to perform queries on encrypted data before passing them to the CSP. The proxy server decrypts the results and transmits them to the user's application once they have been processed by the CSPs and delivered back to the proxy server.

Analysis queries may be made on encrypted data that has been outsourced to an external service like CryptDB. For example, MONOMI supports 19 of the 22 queries in the TPC-H decision-support test, whereas CrypDB only supported 2 and 4 queries, respectively. By separating queries, it is possible for MONOMI to deal with performance concerns caused by enormous DW volumes and delayed query processing while dealing with encrypted data.

MONIMI, on the other hand, suffers from a large communication burden between client and server. There may be various instances when intermediate results are transmitted several times to perform separate queries.

SDB:The CSP now has the capacity to handle a broad variety of sophisticated queries securely utilizing SDB, which makes use of data interoperability. Interoperability allows the output of one operator to be used as the input for another. While a secret value in SMC is kept in a single location, in SDB it is split into two shares, one for each of the CSPs and users. For example, all TPC-H queries may be accessible via SDB's SQL query support.

Column-orientedDBMS:A thorough investigation was undertaken into how data outsourcing models respond to sum and average aggregation requests. An HE system constructed in new fashion is equivalent to the performance of a typical, symmetric encryption scheme (e.g. DES) in which calculation is conducted on plaintext after decryption, according to the researchers. It is possible to manipulate many data values in big encryption blocks using the suggested HE technique since its block size is substantially greater than that of a single numerical value. For DW storage, a column-oriented database management system (DBMS) is the most suited paradigm [9]. This approach can deal with both grouping and aggregate queries. However, SQL's HAVING clauses are not yet supported.

SecureDWs

ABACUS: In order to safeguard the privacy of distributed distributed DWs, we created a mechanism based on Shamir's secret sharing. ABACUS, a middleware developed for distributed data warehouses, may be used to query them. ABACUS is able to run queries such as aggregation, join, and intersection in a secure way. Authorized users may query numerous private DWs using the middleware, which acts as a proxy. Joins and intersections are handled using one-way hash algorithms, such as SHA-1, MD4, and MD5. SUM and AVERAGE are computed using Shamir's secret sharing. Analytical studies show that ABACUS is both efficient and scalable.

MOBAT: data masking for DWs was implemented, allowing users to compromise on performance without sacrificing their personal information. Between the user and the server lies a piece of software known as MOBAT. Private keys, for example, are exposed to MOBAT, which rewrites requests and obscures data before storing it at the server, making it vulnerable to eavesdropping. For each number, three private and one public keys are used to encrypt it before saving it in a safe database at the server. Servers maintain public keys (k3,j) in addition to encrypted values for each column's rows (rows j, k3). As a result, the server also stores two private keys for each column in encrypted form. Testing shows that the suggested masking approach has lower computational and storage requirements than encryption-based alternatives.

PartitionedencryptedDWs:It has been suggested a new way for encrypting and interrogating cloud-hosted data warehouses (DWs). Multidimensional queries may be executed on the encrypted DW using this method, which creates data that can not be decoded even by experts. The DW is distributed across many DBMSs. Each partition's address is stored in a master DW. A secure host is in charge of encrypting and decrypting query parameters on behalf of the client application. After decryption, the secure host must perform sum aggregations and data groupings. Two alternative multivalued encryption algorithms help retain the order of encrypted data, but they do not enable minimum and maximum aggregation queries and have a significant cost.

fVSS:Flexible verifiable secret sharing (fVSS) was presented by [2] as a new technique to protecting cloud DWs. Sharing and encrypting information across several CSPs, fVSS enables searches without the need for re-creation of data. With this plan, participants' honesty is also checked via internal and external verification. This minimizes outsourcing expenses in a pay-per-use model by optimizing data volume and thereby reducing outsourced costs in the cloud.

RESEARCHCHALLENGESANDISSUES

Significant information may be gleaned through less secure methods of encryption, such as OPE. Thus, any encryption approach that does not fulfill stringent cryptography-based security criteria should be utilized with caution. Encryption of multidimensional schemas should be increased to what level?? Attempting to do joins will be hindered if all data is encrypted, including keys. Does the fact that the primary and foreign keys are not encrypted tell anything about the data they are protecting? OLAP queries often need the aggregate of many measurements. Because of this, HE seems like a good option for encryption. It is possible to sum encrypted data using Paillier encryption, although the client's decryption costs may remain considerable in certain cases. As a consequence, decrypting data at the client's end rather than the CSP's may be more cost-effective in the long run. As a result, encrypted data cannot be maintained in a logical sequence. To avoid performance and cost bottlenecks caused by the storage overhead caused by order-preserving encryption methods in OLAP, it is necessary to use these schemes when sorting, grouping, and range operations are required.

Indexing, splitting, and view materialization may all be used to encrypted data for performance improvement. However, while certain inquiries are sped up, others are slowed down. It is thus critical to pick an encryption technology which complies with all use restrictions. Another important consideration is that a balance must be established between maintaining the desired amount of privacy and keeping the effect on system performance to a minimum. Increases in the number of buckets may negatively impact performance, while a decrease in bucket count might raise the danger of data leakage.Finally, the primary problem in the timely administration of secure cloud DW may be summarized as follows.

“How to choose and implement security mechanismsthatovercomecomputational andstorageoverheads,whileguaranteeingdatasecurityincl oudDWs?”

CONCLUSION

Using cloud decision-support data, we examined the various security measures that are presently available. For our research, we focused on cryptographic techniques and systems that allow searches over encrypted data without decryption. Using cloud computing as an example, this study shows the advantages and drawbacks of current solutions and offers some ideas for more practical ones in the future.

REFERENCES

1. DanBoneh,AmitSahai,andBrentWaters.Functional encryption: a new vision for public-key cryptography. Communications of the ACM,55(11):56-64,2012.
2. OracleCorporation.Datamaskingbestpractices.OracleWhitePaper,2010.
3. Ronald Cramery, Rosario Gennaroz, and BerrySchoenmakersx.Asecureandoptimallycientmulti-authorityelection scheme.1997.
4. IvanDamgard,MartinGeisler,andMikkelKroigaard. Efficient and secure comparison foron-lineauctions.InJosefPieprzyk,HosseinGhodosi, andEd Dawson,editors,InformationSecurity andPrivacy,volume4586ofLectureNotesinComputerScience,pages416-430.Springer Berlin Heidelberg, 2007.
5. EinarMykletun and Gene Tsudik. Aggregationqueriesinthedatabase-as-a-servicemodel.InDataandApplicationsSecurityXX,pages89-Springer, 2006.
6. ChristianNeuhausandAndreasPolze.Cloudsecuritymechanisms.
7. Adam O'Neill. Definitional issues in functionalencryption.IACRCryptologyePrintArchive,2010:556,2010.
8. Pascal Paillier. Public-key cryptosystems basedoncompositedegreeresiduosityclasses.InAdvances in cryptology EUROCRYPT'99, pages223-238.Springer, 1999.
9. Raluca Ada Popa, Catherine Redfield, NickolaiZeldovich,andHariBalakrishnan.Cryptdb:protectingconfidentialitywithencryptedqueryprocessing. In Proceedings of the Twenty-ThirdACMSymposiumonOperatingSystemsPrinciples,pages 85-100. ACM, 2011.
10. ShafiGoldwasser, S Dov Gordon, Vipul Goyal,AbhishekJain,JonathanKatz,Feng-HaoLiu,Amit Sahai, Elaine Shi, and Hong-Sheng Zhou.Multi-inputfunctionalencryption.InAdvancesinCryptology-EUROCRYPT2014,pages578-602.Springer, 2014.Hakan
11. Hacigumu¸s,BalaIyer,ChenLi,andSharad Mehrotra. Executing sql over encrypteddata in the database-service-provider model. InProceedingsofthe2002ACMSIGMOD