WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Novel Model of Query Processing Database-as-A-Service (QPDAS) using Privacy Homomorphism Technique

Dr. R. Geetha[1], Mr. M. Prabakaran[2], Mr. P. Ramesh[3], Mr. S. Ravinkumar[4], Mr.M.Santhoshkumar[5]

[1] Associate Professor, Department of MCA, KSR College of Engineering, Tiruchengode, geethar@ksrce.ac.in,
[2, 3, 4, 5, 6] Student, Department of MCA, KSR College of Engineering, Tiruchengode
prabajuju767@gmail.com, rameshperiyasamy.n@gmail.com, ravinsavitha@gmail.com, samkumar12993120@gmail.com

ABSTRACT:

Almost every scientific, academic, or commercial organization faces the fundamental and critical challenge of effective data processing today. Data processing is essential for organizations to create better business strategies and increase their competitive edge. The biggest difficulty is guarding by the DAS of unauthorized both accesses preventing use of the information. Here, our attention is on the second problem. Each user's inputted data's ciphertext is combined to create a single ciphertext. Utilizing the privacy homomorphism technique, the query is run over encrypted data. As much of the query processing should be done by the user without the requirement to decrypt the data. Without decrypting the info, our mission is to retrieve the individual information from the combined outcome.The effectiveness of current security techniques is quickly increased by this strategy. We also carried out comparisons in the end to demonstrate the suggested scheme's effectiveness.

KEYWORDS: Query processing**,** Aggregation, Database-as-a-Service, Cipher Block Chaining, and Privacy Homomorphism.

1. INTRODUCTION

The usage of concealed data aggregation is the process of compiling typically [large] amounts of information from a given databases and organizing it into a more consumable and comprehensive medium. WSNSs may categorize based on their functionality, mode of operation, and kind of target applications. The unneeded traffic and energy consumption of sensor nodes to extend the network's lifespan. Data aggregation is a strategy that aims to solve the issue of localized congestion. It makes an effort to gather a variety of data from the event's sensors. Then, it reduces congestion and its related issues by sending only the end point's useful information. For energy use, cluster-based WSN have been proposed. Every sensor delivers data to the cluster head rather than immediately to the sink. Each sensor encrypts its data for security purposes before transforming it and sending it to the cluster head. Without decrypting the data, the cluster leader collects it. Aggregative procedures are often algebraic, like adding or multiplying incoming data, such as statistical finding the average, lowest, or the most extreme data collection. The coded messages from many applications are possible combined into "only" Crypt text only using CDAMA [11]. However, the primary issue with CDAMA is that the cost of retransmitting lost fragments must be factored into the ciphertext of CDAMA with a larger k. In order to recover the individual data from the aggregated cipher text when utilizing this scenario to aggregate the DAS model's ciphertext, It must be able to ciphertexts bigger than k. Even if the ciphertext is longer than necessary, without decrypting the data, the user's query is processed across the database.

2. RELATED WORK

To run queries over encrypted data, many indexing strategies have been proposed, each appropriate for a specific type of query. These methods are required to let outside servers do queries on encrypted data. Not all conditions can be met when storing encrypted data. It is specified for secrecy reasons that only the client should have access to data decryption. The creator suggested query performance strategies to choose information to be returned in response without the necessity for data self-decryption [1,2,3,6,7]. A hash-based technique for selecting queries may be found in [1, 3] under the heading "Other Query Execution Techniques." In order to enable equality and range query, the author of [3] developed the order-preserving encryption scheme (OPES).The main flaw is that it only works with integer numbers. The suggested execution of aggregating queries over encrypted data is done in [4, 5]. The limitation of the aforementioned method is that only the server side of the data is secured. Different individuals, groups of users, or programmers are subject to access restrictions enforced by access control mechanisms. Every set of tuples is encrypted using the essential assigned to the user base who may obtain it after assembling users with similar access credentials. This method was only applicable to static groupings. In this situation, every time a group's membership changes, the outsourced database must be re-encrypted. The service provider's server hosts an encrypted database.The encrypted database is supplemented with new data, enabling some query processing to take place on the server where the client retains metadata.

Strategy: Split the original query as the following condition

• A client request to post-process the outcome of the server's query;

• A similar query over an encrypted relation store on the server.

Different overheads are the amount of filtering, bandwidth use, and wasted storage caused by metadata at the client. The data will be encrypted using the PH cryptosystem, the ciphertext of each data will be combined into RCBC will allow for the retrieval of each individual piece of data from a single cipher text utilizing CBC.

3. ENCRYPTION SCHEME

The user encrypts the data due to secrecy requirements. The encryption key is an asymmetric key, which means that the encryption and decryption keys used by the sender and recipient are distinct. The collector receives the encrypted data and generates the ciphertext for each user's data. When the user sends multiple pieces of data continuously, the process of creating ciphertext continues. Each data's ciphertext is combined to create a single ciphertext. A sequence of bits is aggregated into a single unit of bits using the CBC mode of operation.

3.1 GENERATION OF SINGLE CIPHER TEXT (CIPHER BLOCK CHAINING)

Cypher block chaining (CBC) is the mode of operation for a block cypher (one in which a group of bits are encrypted as a single unit or block with a cypher key applied to the entire block). In cypher block chaining, an initialization vector (IV) of a particular length is employed. One of its distinguishing characteristics is the use of a chaining mechanism, which makes the decryption of a block of ciphertext dependent on all blocks that came before it. As a result, the falling previous blocks' validity is fully applied to the ciphertext block that comes right before it.

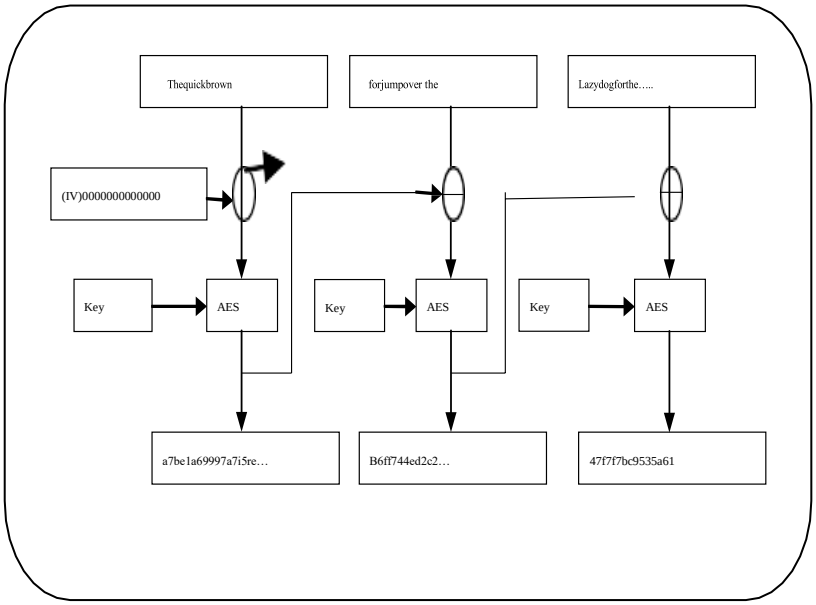

Fig.1CBC operation

Step 1: The first step with CBC is to convert the data and keys to binary.

Text in plain form: 00001101000001001010 10010100110111001111 10010001110010010001000100

Key: 11000001000000000111

Step 2: Remove any whitespace from the key and split the plaintext into various bigger (12 bit) pieces as follows: Plaintext: 100110011100 100000011010 000010010101 1001111100 001010011011

Key:11000001000000000111

Since four bits were added, the last block is going to "000000000100."Use of this additional block is made to facilitate message decoding.Without it, it is impossible to know what additional bits were added.

The following padding in a string: Plaintext: 100110011100 100000011010 000010010101 100111110010 0010010010 001001001010 000000000100

Step 3: One by one, each block is encrypted. For this example, our fundamental encryption technique entails flipping the block, then XORing each bit with the corresponding bit of the key.The first block is encrypted as follows:Plaintext:100110011100

Reversed:001110011001 First12bitsofkey:110000010000

Ciphertext:001110011001([XOR])

Step 4: In the standard method of cypher block chaining, the ciphertext of one block is utilized to aid in encrypting the next block. Before using the standard encryption method on the plaintext, To do this, the plaintext from the following block is XORed with the ciphertext from the preceding block.Block 2's plaintext and Block 1's ciphertext would be XORed as follows:

Ciphertext(Block1):111110001001

Plaintext(Block2):100000011010Exclusiveor(XOR):011110010011

Step 5: Now that the exclusive XOR result has been obtained, it is thought of as plaintext and is encrypted regularly as previously stated.Reversed:110010011110

First12bitsofkey:110000010000

Ciphertext(byXOR):000010001110

Step 6: Thisprocessrepeatsuntilthelast blockisencrypted.The outcome of encrypting all 8 blocks in this manner is as follows:

Ciphertext:111110001001 000010001110 000110010000 000100011100 101101100001 000011100001 000101000100 110000111000.

Since each of these blocks (aside from the first one) was encrypted using the key and the ciphertext of the previous block, it becomes incredibly difficult to decrypt.

3.2 PHCRYPTOSYSTEM

The user submits the query to access the data after the creation of a single ciphertext. The query gets data that has been overly encrypted. Hence the privacy homomorphism method is employed. A homomorphism encryption system is called privacy homomorphic encryption (PH).It enables the query to be run on encrypted data.According to the homomorphic property, it is possible to perform algebraic operations on plaintexts by changing the associated ciphertexts. DK(EK(m1) (EK(m2) =m1 m2, for instance, stands for operations on plaintexts and ciphertexts, respectively.

EXAMPLE:

Assume that A represents the set of unencrypted values that k represents the encryption function that uses key k, and that Dk represents the corresponding decryption function.

ZnDecryption-Dk (a) = d1qq-1 + d2pp-1 (mod n) (d1= a mod p & d2= a mod q), where a = (a mod p, a mod q)

Think about p = 5, q = 7, and n = 35.

a1 = 5 & a2 = 6

ε(a1) = (0, 5) Calculate (a1 + a2) by using the server's storage of (a2) = (1, 6).

Component-wise, the server calculates (a1) + (a2) as (0+1, 5+6) = (1, 11).

Client decrypts (1, 11) as (1.7.3 + 11.5.3) (mod 35) = 186 mod 35 = 11 as d1qq-1 + d2pp-1 (mod n)

3.2 REVERSE CIPHER BLOCK

It is determined whether or not the user is permitted to view the data after the single ciphertext has been decrypted.If the user is allowed, the user query is used to access encrypted data and retrieve the required information.The user receives the required data, which he or she can then decode on the customer side.

Step 1: After that, we'll start the first block:

First block ciphertext: 111110001001

Exclusive or (XOR): 001110011001 for the key's first 12 bits: 110000010000

Reversed (block 1): 100110011100.

Step 2: The ciphertext of block 1 is XOR'd with the key, inverted, and the second block:

2nd block ciphertext: 000010001110

Exclusive or (XOR): 110010011110 for the key's first 12 bits: 110000010000

011110010011 in reverse

Ciphertext(1stblock):111110001001

Exclusive or (XOR): block2 100000011010

2nd block ciphertext: 000010001110

Exclusive or (XOR): 110010011110 for the key's first 12 bits: 110000010000

011110010011 in reverse

Exclusive or (XOR): 100000011010. Ciphertext (1st block): 111110001001.

Step 3: This approach may now be used to decode each block. The padding can be readily eliminated after the entire plaintext has been located.

4. RESULTS

Here, the proposed method is tested using the ratio between the number of packets received by the destination and the number of packets broadcast by the source. Additionally, fig. 2 demonstrates that the proposed system has lower latency than the existing one.

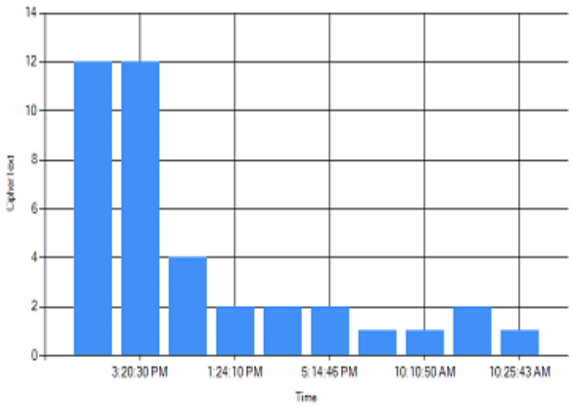

Fig.2Aggregationdelay

5. CONCLUSION

Data security and privacy are thus of utmost importance for service provider sites. In this study, we tackled a particular problem with data privacy. Our approach is to combine each ciphertext into a single cipher text. We have designed a method of query execution that the provider may use without having to decrypt the aggregated result. The aggregated result may be used to get the individual data. This makes the current security systems stronger.

6. REFERENCES

[1] PierangelaSamarati, S. DeCapitanidiVimercati, Sara Foresti, SusilJajodia, Stefano ParaboschiKey Management for Multi-User Encrypted Databases is discussed in the November 2005 Proceedings of the ACM Workshop on Storage Security and Survivability.

[2].ExecutingSQLoverencrypteddatainthedatabase-service-providermodel.Madison, Wisconsin, USA: In Proc.of the ACMSIGMOD'2002, June 2002.]

[3] E. Damiani, S. De CapitanidiVimercati, S. Jajodia, S. Paraboschi, and P. Samarati are listed as sources in item number three. juggling efficiency and secrecy in relational DBMSs that aren't reliable. Washington, DC, USA, October 27–31, 2003: In Proc. of the 10th ACMConference on Computer and Communications Security.

[4].Y. Xu, J. Kierman, R. Srikant, and R. Agrawal.Numeric data order maintaining encryption.InProc.ofACMSIGMOD2004, June 2004, Paris, France.

[5] .S.AklandP.Taylor.Cryptographicsolutiontoaproblemofaccesscontrolinahierarchy. August 1983, ACM Transactions on Computer Systems, 1(3):239–244.

[6].D.Boneh,G.D.Crescenzo,R.Ostrovsky, andG.Persiano.includes keyword search and public-key encryption.InProc.ofEurocrypt2004, May 2004, Interlaken, Switzerland.

[7] .H.Hacig¨um¨us,B.Iyer,andS.Mehrotra. Providingdatabaseasaservice.18th International Conference on Data Engineering Proceedings, San Jose, California, USA, February 2002.

[8].H.Hacig¨um¨us,B.Iyer,andS.Mehrotra. Ensuringtheintegrityofencrypteddatabases in thedatabase-as-a-servicemodel.InDBSec, 2003, pp 61–74.

[9] Executing SQL over Encrypted Data in the Database-Service-Provider Model, B. Iyer, C. Li, and S. Mehrotra, Proc. ACMSIGMOD Int'lConf.ManagementofData, pp. 216-227, 2002.

[10].H.Hacigumus, "EfficientExecution of AggregationQueriesoverEncryptedRelationalDatabases," Proc. Ninth International Conference on Database Systems for Advanced Applications (DASFAA'04), vol. 9, p. 125, 2004.

[11] S.DeCapitanidiVimercati, M. Finetti, S.Paraboschi, P. Samarati, and S. Jajodia.ImplementationofastoragemechanismforuntrustedDBMSs.Second International IEEESecurity in Storage Workshop Proceedings, Washington, DC, USA, May 2003.

[12] DAMA:Concealed Data Aggregation Scheme for Multiple Applications in Wireless Sensor Networks IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 7, July 2013, Yue-Hsun Lin, shih-yingchang, and hung-min sun.