IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Vol.7, Special Issue.2, Jan 2019 E-ISSN: 2347-2693

Internet of Things: Architecture, Security and Cryptdb : Monomi

G. Ambika[1*] , P.Srivaramangai[2]

[1,2]Department of Computer Science, Marudupandiyar College, Thanjavur, Tamilnadu, India

Available online at: www.ijcseonline.org

Abstract: Cloud users demands security to their data which are stored in data repositories of cloud service provider. Thus the concept of Network Security can be applied over the cloud network, where several encryption algorithms are applied to provide integrity on the data. Such algorithms include Symmetric encryptions, Asymmetric encryptions, Hashing algorithms and Digital signatures. MONOMI is a system for securely executing analytical workloads over sensitive data on an untrusted database server. MONOMI works by encrypting the entire database and running queries over the encrypted data. CryptDB is a MySQL proxy that allows SQL aware encryption inside existing database management systems. To offer the best possible protecting while enabling the greatest computational flexibility it relies on a new concept called onions, where different layers of encryption are wrapped around each other and are only revealed as necessary. While its concept to improve database security looks fresh and interesting from an academic standpoint we wanted to examine the usability in practical application to determine if a real world productive use is desirable.

Keywords: *Security, Internet of things, Crypt DB, Monomi.*

I.INTRODUCTION

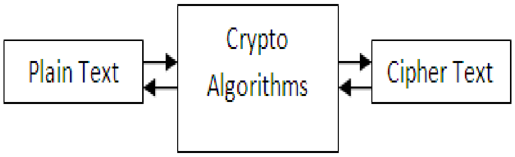

Fig. 1

The success of the Internet, computing resources have become cheaper, more powerful and more ubiquitously available than ever before. This technological trend has evolved in new network based computing called as Cloud computing.

The infrastructure providers, who manage cloud platforms and lease resources such as CPU, storage system etc., according to a usage-based pricing model, and service providers, who rent resources from one or many infrastructure providers to serve the cloud users. In order to overcome such attacks certain security prototypes have been developed which resists such ad hoc attacks, Denial-of-Service attacks, jamming attacks and other fabrication attacks. Security algorithms are based on encryption standards, where the original text is encrypted to cipher text in the sender‟s end and is transmitted through any network, upon receiving cipher text by the receiver, decrypts it and reads the original message.

Now-a-days the data is stored in a database where the DBA controls the data and access controls of that data are under the DBA, who cannot be easily trustworthy. One possible solution to this problem is to encrypt data on the client machine (assumed to be trusted) before uploading it to the server, and process queries by reading back the encrypted data from the server to the client, decrypting the data, and executing the query on the client machine. However, for database queries, and analytical workloads in particular, this requires transferring much more data than is needed, since large fractions of a database are read by the query, but the results are typically small aggregate reports or roll-ups.

All these security paradigms involve generation of keys to encrypt and decrypt the original message. Cipher text is sent through the network instead original text there is less chance of leaking the original message and other attacks are moderately prevented. Since this encryption revolves around keys for encryption and decryption, in order to decrypt the encrypted message by the receiver he must know the private/secret key which the sender has to send besides the cipher message.

CryptDB the concepts behind CryptDB, similarly to how we looked at databases and SQL in the first chapter. We start with the general setup and then go into details and explain the different encryption methods that are used and what the so called “Onion Layers” are. CryptDB is intended to work as a proxy between the application and the database. An application for example might be a website, an application on a mobile device (a so called “App”) or a classic desktop application, basically anything that connects to a database.

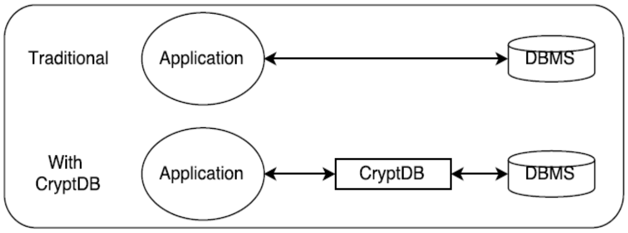

Fig. 2

The operator GROUP BY relies on equality checks concerning the encrypted data, other functions like SUM rely on the ability to perform additions of the encrypted data. CryptDB deals with these different computational aspects by clustering functions by their underlying operations, as mentioned above. At the same time the outmost layer is the one with the least functionality while the innermost one offers the greatest functionality. The transformation from one layer into another ("peeling off a layer") happens automatically when the need arises (i.e. when a query with a certain operator/function is issued). In this case CryptDB automatically reencrypts the entire column and remembers its state.

Third, some of the techniques for processing queries over encrypted data can speed up certain queries but slow down others, thus requiring careful design of the physical layout and careful planning of each query's execution, for a given database and query mix. For example, Paillier encryption can be used to sum encrypted data on the server, but the cost of decrypting Paillier ciphertexts at the client can be prohibitively high in some situations, such as when computing the sum of a handful of values. As a result, it can be more efficient to decrypt and sum individual data items at the client rather than run aggregates over encrypted data at the server. Similarly, materializing additional columns can improve some queries but slow down others due to increased table sizes.

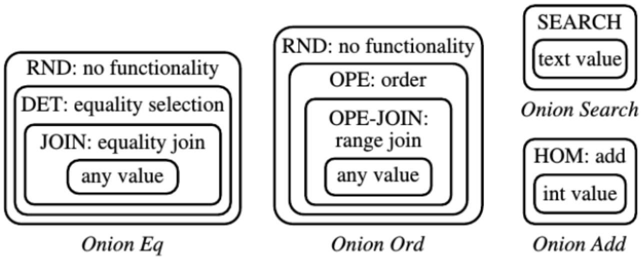

Fig. 3

II.OVERVIEW

MONOMI's design addresses these challenges in three ways. First, we introduce split client/server execution of complex queries, which executes as much of the query as is practical over encrypted data on the server, and executes the remaining components by shipping encrypted data to a trusted client, which decrypts data and processes queries normally. MONOMI-supplied UDF that computes the encrypted sum of several Paillier ciphertexts, and price_paillier and order_id_det are the Paillier and deterministically encrypted columns of the original price and order_id columns, respectively. Once MONOMI's client library receives the results, it decrypts them, and executes the HAVING total > 100 clause. Any matching results are sent to the application. §4 describes how MONOMI performs this transformation for arbitrary queries.

*Select * from AES*(*emp*);

Which requires a symmetric key , which can generated through a Key Generator(), one cannot assure that the key generated in cryptdb is same as the key generated by the DBA manually, but there is a chance of getting same key.

Security. Since MONOMI's design builds on CryptDB, it inherits similar security properties. Although the untrusted server stores only encrypted data, it can still learn information about the underlying plaintext data. Another key factor is that the split executions that MONOMIncan perform depend on the encryption schemes available at then server, and some splits perform better than others. For example, if a Paillier encryption of price was not available in the above example, MONOMI would have to download the entire price column to the client. On the other hand, if no query asked for SUM(price), storing a Paillier encryption of price wastes storage space. To choose a set of encryption schemes that maximizes query performance, MONOMI uses an optimizing designer.

Using MONOMI. Our MONOMI prototype consists of three major components, as shown in Figure 1. First, during system setup, MONOMI's designer runs on a trusted client machine and determines an efficient physical design for an untrusted server. To determine important characteristics of the workload for achieving good performance, our designer takes as input a representative subset of the queries and statistics on the data supplied by the user, such as the

TPC-H workload. As we show in §8, MONOMI's designer achieves good performance with only a small subset of queries, as long as they contain the key features of the workload.

III.ENCRYPTION TYPES

Homomorphic encryption (HOM)

The HOM onion layer provides an equally strong security assurance, as it is considered to be IND-CPA secure too [1]. It is specifically designed for columns of the data type integer and allows the database to perform operations of an additive nature. This includes of course the addition of several entries, but also operations like SUM or AVG.

Word search (SEARCH)

The SEARCH onion layer is exclusive for columns of the data type text. In the version of CryptDB that we used in this thesis (see Appendix A.1) we have been unable to successfully create such an onion. The following explanation is therefore solely of a theoretical nature and based on the paper provided by Popa et al.

Deterministic (DET)

The DET onion layer provides the second strongest security assurance: In contrary to RND this layer is deterministic, meaning that the same plaintext will be encrypted to the same ciphertext. This means that the DBMS can identify fields with equal (encrypted) content Order-preserving encryption (OPE) The OPE onion layer is significantly weaker than the DET layer as it reveals the order of the different entries. This means that the DBMS knows relations like bigger and smaller, but also equality. In regards to security it is noteworthy that this onion layer is the most revealing one: It can not fulfill the security definition of IND-CPA, as is shown by Boldyreva et al.

IV. CONCLUSIONS

Analysis done over cryptdb concludes that the cryptdb provides confidentiality to the user‟s private data through encryption schemes. It is based on relational databases and supports SQL queries, but not upto full extent. MONOMI uses split client/server query execution to perform queries that cannot be efficiently computed over encrypted data alone. MONOMI introduces per-row precomputation, space-efficient encryption, grouped homomorphic addition, and pre-filtering to improve performance of analytical queries over encrypted data.

REFERENCES

[1]. D. J. Abadi, S. R. Madden, and N. Hachem. Column-stores vs. rowstores: how different are they really? In Proc. of SIGMOD, pages 967–980, Vancouver, Canada, June 2008.

[2]. S. Agrawal, S. Chaudhuri, and V. R. Narasayya. Automated selection of materialized views and indexes in SQL databases. In Proc. of the 26th VLDB, pages 496–505, Cairo, Egypt, Sept. 2000.

[3]. A. Arasu, S. Blanas, K. Eguro, R. Kaushik, D. Kossmann, R. Ramamurthy, and R. Venkatesan. Orthogonal security with Cipherbase. In Proc. of the 6th CIDR, Asilomar, CA, Jan. 2013.

[4]. S. Bajaj and R. Sion, "TrustedDB: A Trusted Hardware Based Database with Privacy and Data Confidentiality," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '11), pp. 205-216, 2011.

[5]. S. Bajaj and R. Sion, "TrustedDB: A Trusted Hardware Based Outsourced Database Engine," Proc. Int'l Conf. Very Large DataBases (VLDB), 2011.

[6]. D. Bogdanov et al. A universal toolkit for cryptographically secure privacy-preserving data mining. In PAISI, 2012.

[7]. F. Emek_ci and D. Agrawal et al. Privacy preserving query processing using third parties. In ICDE, 2006.

[8]. C. Gentry. Fully homomorphic encryption using ideal lattices. In STOC, 2009.

[9]. C. Gentry et al. Fully homomorphic encryption with polylog overhead. In EUROCRYPT, 2012.

[10]. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In STOC, 1987.

[11]. Stephen Tu, M. Frans Kaashoek, Samuel Madden, and Nickolai Zeldovich. 2013. Processing Analytical Queries Over Encrypted Data. In Proceedings of the Conference on Very Large Data Bases (VLDB).

[12]. Vlado Altmann, Jan Skodzik, Frank Golatowski, and Dirk Timmermann. "Investigation of the use of embedded Web Services in smart metering applications". In: *Conference on IEEE Industrial Electronics Society*. IECON. Oct. 2012, pp. 6172–6177. DOI: 10.1109/IECON.2012.6389071.

[13]. Sven Bendel et al. "A service infrastructure for the Internet of Things based on XMPP". In: *Int. Conference on Pervasive Computing and Communications Workshops*. PERCOM Workshops. Mar. 2013, pp. 385–388. DOI: 10.1109/PerComW.2013.6529522.

[14]. SHEN changxiang, ZHANG Huanguo and FENG Dengguo, "Literature Review of Information Security" Science in China (Series E: Information Sciences), vol.37, no.2, 2007, pp.129-150 WU chuankun, "A Preliminary Investigation on the Security Architecture of the Internet of Things," Bulletin of Chinese Academy of Sciences, vol 25,no. 4, 2010, pp 411-419.

ABOUT THE AUTHORS:

G.Ambika received her M.Phil Degree from Tamil University, Thanjavur in the year 2013. She has received her M.C.A Degree from Bharathidasan University, Trichy in the year 2010. She is pursuing her Ph.D (Full-Time) Degree at Marudupandiyar College of Arts & Science, Thanjavur, Tamilnadu, India. She has published 5 papers in International Journals. Her areas are Internet of Things, Cloud Computing and Mobile Computing.

Dr.P.Srivaramangai received her Ph.D Degree from Mother Teresa University, Kodaikanal in the year 2012. She received her M.Phil Degree from Manonmaniam University, Tirunelveli in the year 2003. She received his M.C.A Degree from Bharathidasan University, Trichy in the year 1996. She is working as Associate-Professor, PG and Research Department of Computer Science, Marudupandiyar College of Arts & Science, Thanjavur, Tamilnadu, India. She has above 30 years of experience in academic field. She published 25 papers in National & International journals so far. Her areas of interest include Computer Networks, Internet of Thing, Grid Computing, Cloud Computing and Mobile Computing.