ISSN 2350-1049

International Journal of Recent Research in Interdisciplinary Sciences (IJRRIS)
Vol. 4, Issue 1, pp: (16-20), Month: January - March 2017, Available at: www.paperpublications.org

PUBLIC INTEGRITY AUDITING FOR SHARED DYNAMIC DATA STORAGE UNDER ON-TIME GENERATED MULTIPLE KEYS

[1]C. NISHA MALAR, [2]M. S. BONSHIA BINU

[1]PG, Student, Department of CSE, Ponjesly College of Engineering, Nagercoil
[2]Assistant Professor, Department of CSE, Ponjesly College of Engineering, Nagercoil

Abstract: **Nowadays verifying the result of the remote computation plays a crucial role in addressing in issue of trust. The outsourced data collection comes for multiple data sources to diagnose the originator of errors by allotting each data sources a unique secrete key which requires the inner product conformation to be performed under any two parties different keys. The proposed methods outperform AISM technique to minimize the running time. The multi-key setting is given different secrete keys, multiple data sources can be upload their data streams along with their respective verifiable homomorphic tag. The AISM consist of three novel join techniques depending on the ADS availability: (i) Authenticated Indexed Sort Merge Join (AISM), which utilizes a single ADS on the join attribute, (ii) Authenticated Index Merge Join (AIM) that requires an ADS (on the join attribute) for both relations, and (iii) Authenticated Sort Merge Join (ASM), which does not rely on any ADS. The client should allow choosing any portion in the data streams for queries. The communication between the client and server is independent of input size. The inner product evaluation can be performed by any two sources and the result can be verified by using the particular tag.**

Keywords: **Computation of outsourcing, Data Stream, Multiple Key, Homomorphic encryption.**

I. INTRODUCTION

Data security involves encryption and key management skills for a given packet transmission in the network. As the number of users and nodes are increased in the network, the security level has tend to fall. In the recent trends there is drastically increase in the number of users, therefore the need of data security mechanism seek continues attention. This paper is based upon the utility of multiple keys in network in order to improve the security level and reduce congestion. The recent ground-breaking development of fully homomorphic encryption allows us to maintain confidentiality/privacy of outsourced data in this setting. In a homomorphic message authenticator scheme, we can authenticate some large data using the secrete key. Later anybody can homomorphically execute an arbitrary program over the authenticated data to produce a short tag which certifies the value y as the output. Volunteer computing is the server to split large computation into small units, send these units to volunteers for processing and resemble the result.

Verifiable computation scheme as a protocol between two polynomial time parties to collaborate on the computation of a function. This scheme consists of three main phase:

1. Preprocessing- This stage is performed once by the client in order to calculate some auxiliary information associated with function. Part of this information is public to be shared with the worker while the rest is private and kept with the client.

2. Input preparation- In this stage, the client calculate some auxiliary information about the input of the function. Part of this information is public to be shared with the worker while the rest is private and kept with the client. The public information is sent to the worker to compute on the input data.

3. Output computation and verification- In this stage, the worker uses the public information associated with the function and the input, which are calculated in the previous two phases to compute an encoded output of the function on the provided input.

This result is then returned to the client to verify its correctness by computing the actual value of the output by decoding the result returned by the worker using the private information calculated in the previous phase. The defined notation of verifiable computation scheme minimizes the interaction between the client and the worker into exactly two messages, where a single message sent from each party to the other party during the different phases of protocol. Homomorphic message authenticator allows the holder of the evaluation key to perform computations over previously authenticated data. More precisely, a user knowing the secrete key used to authenticate the original data, can verify that authenticates the correct of the computation.

II. RELATED WORK

In existing verifiable computation schemes only focus on the single-key setting, i.e., data and its computation are outsourced from merely one contributor or from multiple contributors but with the same key. On the other hand, we may resort to the powerful Fully Homomorphic Encryption (FHE) but are hardly willing to use it in practice due to efficiency concern. Fully homomorphic encryption technique allow the holder of a public evaluation key to perform computations on previously authenticated data, in such a way that the produced proof can be used to certify the correctness of the Computation.

More precisely, with the knowledge of the secret key used to authenticate the original data, a client can verify the computation by checking the proof. For the asymmetric setting, Boneh and Freeman proposed a realization of homomorphic signatures for bounded constant degree polynomials based on hard problems on ideal lattices. Although not all the above schemes are explicitly presented in the context of streaming data, they can be applied there under a single-key setting. In this scenario, the data source continually generates and outsources authenticated data values to a third-party server. Given the public key, the server can compute over these data and produce a proof, which enables the client to privately or publicly verify the computation result. Recently, several works towards public verification either for specific classes of computations or for arbitrary computations have been proposed.

Considered a different setting for verifiable computation. In their models, the client needs to know the input of the outsourced computation and runs an interactive protocol with the server in order to verify the results. In memory delegation, the stream outsourcing was considered but with the restraint that the size of the steam has to be a priori bounded.

III. PROPOSED

Proposed a realization of homomorphic signatures for bounded constant degree polynomials based on hard problems on ideal lattices. Although not all the above schemes are explicitly presented in the context of streaming data, they can be applied there under a single-key setting. In this scenario, the data source continually generates and outsources authenticated data values to a third-party server. However, the outsourced data to be a priori fixed. Another interesting line of works considered a different setting for verifiable computation. Clients are only allowed to query the server for the summation of a grouped data specified by the data source. A scheme of outsourced computations including group by sum, inner product, and matrix product with private verifiability was considered. To automatically and efficiently deriving or inferring a general data from a particular behavior in social networks, this article introduces trust agents and designs their features according to group behavior features.

ISSN 2350-1049

International Journal of Recent Research in Interdisciplinary Sciences (IJRRIS)
Vol. 4, Issue 1, pp: (16-20), Month: January - March 2017, Available at: www.paperpublications.org

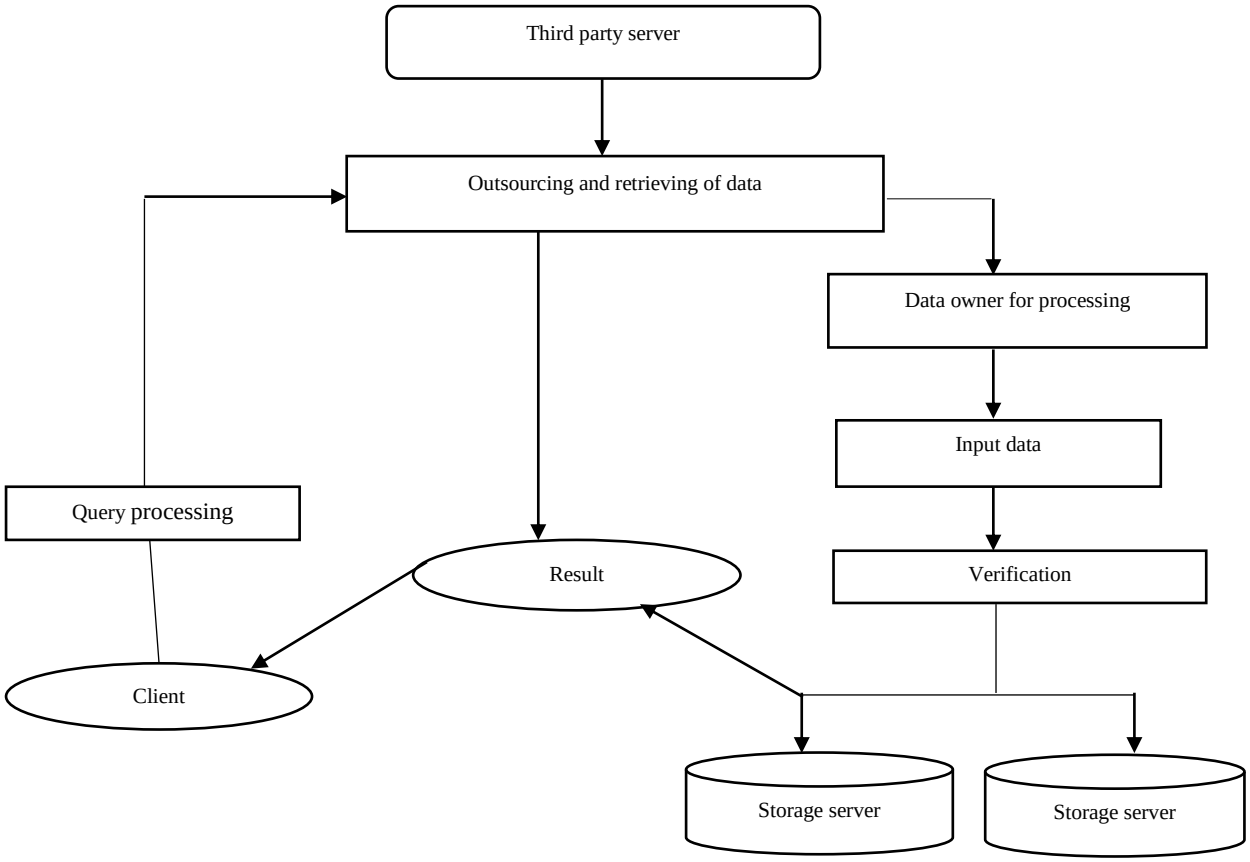

Figure.1. System Architecture

Data Storage:

In this process, clients are only allowed to query the server for the summation of a grouped data specified by the data source. A scheme of outsourced computations including group by sum, inner product, matrix product with private verifiability was considered. Data sources store their public/private keys and system parameters locally while outsourcing all the data along with the corresponding tags to a third-party server. The client should be free to be choose any portion of the data streams as the input of the queried computation. The group-by sum query described above, we present a publicly verifiable computation scheme for the inner product query over data streams with two different keys in this subsection.

Data storage involves encryption and key management skills for a given packet transmission in a network. As the number of users and nodes are increased in the network; the security level has tended to fall. In the recent trends; there is drastically increase in the number of user, therefore, the need of data security mechanisms seek continues attention. Cryptography is the best technique used to avoid unauthorized access of data. It involves homomorphic encryption algorithm and the keys which are being used by the users. Key management is very essential part of cryptography.

Multi-key Setting:

A multi-key verifiable computation scheme was proposed and followed by a stronger security guarantee scheme. After the generation of system parameters, data sources outputs an encoded function to the server. Consider publicly verifiable delegation of inner product computation over dynamic data streams under the multi-key setting. The proposed scheme is extremely lightweight for both data sources and clients. Given different secret keys, multiple data sources can upload their data streams along with the respective verifiable homomorphic tags generated by the corresponding secret keys to the cloud. As such, no source can deny the user contribution to the outsourced computations. In addition, the inner product evaluation can be performed over any two sources outsourced streams, and the result can be verified using the associated tags.

Multiple key also consumes more power and may cause congestion; therefore, it is not worthy to support all the nodes with multiple keys. Only such nodes having less security level needs to be supported with multiple keys. The performance analysis of multiple keys in a network needs to done in order to evaluate the security level of all the nodes. It is a technique used to avoid unauthorized access of data. The encryption process consists of single or multiple keys to hide the data from the hackers. The original text before the encryption process is known as Plaintext. The text obtain after encoding the data with the help of a key is known as cipher text. In the encryption process single or multiple keys can be used for the encryption of data.

Key needs to be transported over a secured channel from transmitter to receiver via various nodes. In both the cases fixed as well as variable lengths for the keys can be considered. The single key having fixed length cannot offer much resistance to the hacker. In order to increase the security level the key length has to be increased which creates more overheads. When multiple keys having same failure rate have been used to encrypt the data then the actual performance of the system has been observed. It does not include any booting time, and the time taken by the system is only concerned with encryption process.

AISM:

To automatically and efficiently deriving or inferring a general data from a particular behavior in social networks, this article introduces trust agents and designs their features according to group behavior features. A dynamics control mechanism can be generated to coordinate participant behaviors in social networks to avoid a specific restricted negative group behavior. The AISM consist of three novel join algorithms depending on the ADS availability: (i) Authenticated Indexed Sort Merge Join (AISM), which utilizes a single ADS on the join attribute, (ii) Authenticated Index Merge Join (AIM) that requires an ADS (on the join attribute) for both relations, and (iii) Authenticated Sort Merge Join (ASM), which does not rely on any ADS.

Traditional database query processing relies on three types of algorithms for join and for grouping operations. For joins, index nested loops join exploits an index on its inner input, merge join exploits sorted inputs, and hash join exploits differences in the sizes of the join inputs. Our goal is to end mistaken choices among join algorithms and among grouping algorithms by replacing the three traditional types of algorithms with a single one. Like merge join, this new join algorithm exploits sorted inputs. Like hash join, it exploits different input sizes for unsorted inputs. In fact, for unsorted inputs, the cost functions for recursive hash join and for hybrid hash join have guided our search for the new join algorithm. In consequence, the new join algorithm can replace both merge join and hash join in a database management system.

Public Verification:

All the participants involved in the protocol should be able to publicly verify the outsourced computation results without sharing secret keys with data sources. This process involves Key Generation, Tag Generation, Evaluation, Generating Proof and verification of Proof. Evaluate and Generation of Proof can be combined together in our verifiable non-interactive inner product computation scheme. Based on the group-by sum query described above, we present a publicly verifiable computation scheme for the inner product query over data streams with two different keys in this subsection. The auxiliary information can be pre-Computed to accelerate the verification process. The results demonstrate that our protocol is practically efficient in terms of both communication and computation cost. Outsourcing data and computation to third parties in the above setting raises issues of trust.

IV. CONCLUSION

In this paper, we introduce a novel homomorphic verifiable tag technique, and design an efficient and publicly verifiable inner product computation scheme on the dynamic outsourced data streams under multiple keys. For grouping, an index-based algorithm has been used in the past whereas today sort- and hash-based algorithms prevail. Cost-based query optimization chooses the most appropriate algorithm for each query and for each operation. Unfortunately, mistaken algorithm choices during compile-time query optimization are common yet expensive to investigate and to resolve. While the traceability can still be provided on demand. Furthermore, any keyless client is able to publicly verify the validity of the returned computation result. Security analysis shows that our scheme is provable secure under the CDH assumption in the random oracle model. Experimental results demonstrate that our protocol is practically efficient in terms of both communication and computation costs.

REFERENCES

[1] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in Advances in Cryptology, EUROCRYPT Springer, 2011.

[2] M. Backes, D. Fiore, and R. M. Reischuk, "Verifiable delegation of computation on outsourced data," in ACM conference on Computer and communications security. ACM, 2013, pp. 863–874.

[3] D. Catalano and D. Fiore, "Practical homomorphic macs for arithmetic circuits," in Advances in Cryptology, EUROCRYPT Springer, 2013.

[4] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in Advances in Cryptology–CRYPTO. Springer, 2010.

[5] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in Advances in Cryptology–CRYPTO. Springer, 2010.

[6] R.Gennaro and D. Wichs, "Fully homomorphic message authenticators," in Advances in Cryptology-ASIACRYPT Springer, 2013.

[7] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," in ACM symposium on Theory of computing.ACM, 2008.

[8] S. Nath and R. Venkatesan, "Publicly verifiable grouped aggregation queries on outsourced data streams," in International Conference on Data Engineering. IEEE, 2014.

[9] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in IEEE Symposium on Security and Privacy. IEEE, 2014, pp. 238–252.

[10] Vu, S. Setty, A. J. Blumberg, and M. Walfish, "A hybrid architecture for interactive verifiable computation," in IEEE Symposium on Security and Privacy. IEEE, 2015, pp. 223–237.