INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Enhanced Protection Over Data Streams under Multiple Keys in Cloud

K. Dhanabakiyam
dhanakrishnan89@gmail.com
Tejaa Shakthi Institute of Technology for women, Coimbatore, Tamil Nadu

V. Divya
divyaruby28@gmail.com
Tejaa Shakthi Institute of Technology for women, Coimbatore, Tamil Nadu

R. Maheswari
maheswarir135@gmail.com
Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu

E. Sathiya Jothi
sathiya2303@gmail.com
Tejaa Shakthi Institute of Technology for women, Coimbatore, Tamil Nadu

M. S Vijay Kumar
nklvijay@gmail.com
Tejaa Shakthi Institute of Technology for women, Coimbatore, Tamil Nadu

ABSTRACT

Upload the data streams to the cloud servers for the product evaluation; it is an important to many popular stream applications. And it is applying to many organizations. But at the same time, it is difficult to verify the result on the cloud computation, it is leads to the issues of trust. Since the outsourced data streams are retrieved from various resources. In existed system, each originator has a unique secret key .But when the multiple users are tends to download the same file, it leads to some misconceptions. Hence, it extends to the multiple-key settings, where as it doesn't use any specific algorithm for encryption. In this paper, we mainly focus on the security. We propose an Advance Encryption Standard (AES) for the encryption. And also we implement an extra proxy server for the data reliability. We present a novel homomorphic verifying tag technique to verify the inner product evaluation on the dynamic data streams and then it extends to the matrix product verification. We use the random oracle access model.

Keywords: *Data Stream, Multiple keys, Public verifiability, AES.*

1. INTRODUCTION

The past few years have witnessed the rapid growth of the data stream generated by a variety of applications such as GPS, internet traffic, wireless sensors etc. Retaining a local copy of such data is prohibited for various organizations. Take into the stream-oriented application analysis, weather forecasting and traffic management where multiple sources are collect and generated data streams continuously. Example, the inner product between any two outsourced data streams for correlation analysis. The outsourcing automatically raises the issue of trust. The local proxy server may be untrusted one. It is necessary to verify the result of the computation provided by the server. And also it is difficult to verifiable computation scheme. The outsourced data streams are more sensitive. That is the given data from the different sources, the final evaluation result may be erroneous, even if the corresponding query processed correctly. Cryptography provides a method called an off-the-shelf method to solve this problem. Each data unit has its own secret key and also it attached to the 'sign' for the data distribution from which the traceability is derived. But the typical signature algorithm does not intend to perform the multi-key computation. Most of the verification schemes only focus on the single-key setting that is data are retrieved from single contributor with the same key. On the other hand, the Fully Homomorphic

Encryption (FHE) concerns on the efficiency. As a result, we can achieve the solution for the multi-key setting. In memory delegation, the outsourced stream was considered, but the size of the stream has to be priori bounded.

2. RELATED WORK

In this paper, the client is only allowed to request the query to the server for the summarization of group of data specified by the data source. Recently, the multikey setting is proposed with strong security.

There are set of machines (data sources) each has its own public key and unique private key. These machines are collecting the unbounded data and outsourced them to a third party server. And these machines are not directly connected with each other. For a new data stream $(X_{i,j})$, which is generated at time (i), by the machine (M_j) $(1 \leq j \leq l)$ computes a homomorphic and publicly verifiable tag $\sigma_{i,j}$ and outsources a tuple $\{i, X_{i,j}, \sigma_{i,j}\}$ to the server. In this scheme, the time is measured in a discrete manner. Whenever the new tuple comes the time is increased. Additionally, we assume that the clocks of the data sources, third-party server and the client are synchronized. The requirements are inherited from the various streaming applications.

A client requests the server to compute the inner product between any two machines and outsourced data streams by sending the required query. Apart from the computation result *res*, the server also provides the *proof* (π) of the computation and also provides some auxiliary information. Due to this, the client can verify the correctness of the computation result *res.*

We assume that the third-party server is untrusted because it places the outside of the trust domain of the data sources. And also the clients are also may be untrusted by the data sources, because they may be compromised or the maliciously or conclude with the server for the financial concerns in practice. Therefore, the secret keys are used by the data sources to generate the verifiable tags which are not sending to the client for the verification. Otherwise, an untrusted client with the private keys can collude with the server to modify the data and generate corresponding tags to deceive other clients. In this paper, we mainly focus on the verification of the outsourced data streams verification over the public data streams. While the sensitivity of the data protection is also of concern.

3. EXISTED SYSTEM

As usual, the client requests the server to compute the inner product evaluation over the outsourced data streams. The client gives the corresponding query to the client. A client which one wants to retrieve the desired result, need to enter the keys for the authentication. Then, this server forges the query to the multiple numbers outsourced data streams $(M_{i,j})$. Now, the data sources are fetched the necessary data streams, which satisfy the user query. These data sources are the forges the data streams to the server along with the generated keys (pk_j, sk_j).

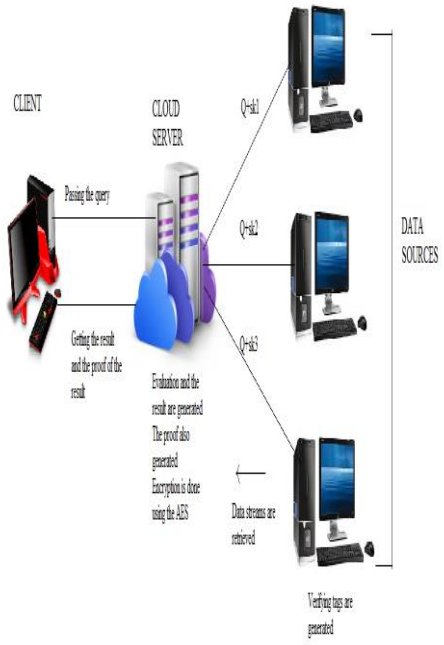
These data sources are also generating the publicly verifiable tags $(\sigma_{i,j})$ along with the time (i) which taken to generate the tags. Then the server generates the result for the query. And the result is sent to the client. After, receiving the result the client chooses the random number. The server then generates the result for the inner product evaluation based on the random number which is sent to the client. After only, the server generates the proof (π) for the inner product evaluation result. Then this proof is sent to the client. After only the result of the computation, correctness is verified.

4. INNER PRODUCT QUERY

Based on the group by sum query, we present a publicly verifiable computation scheme for the inner product query over data streams with two different keys in the subsection. Specifically, any two machines M_1 and M_2 outsource , the data stream $X_1=\{X_{1,1}, X_{1,2}, \ldots X_{1,n}\}$ and $X_2=\{X_{2,1}, X_{2,2}, \ldots X_{2,n}\}$ to the server respectively . A client requests the server to compute the inner product function F_{IP} on X_1 and X_2 i.e.,$res=F_{IP}(X_1, X_2)=X_1$ xor $X_2 = \sum_{i=1}^{n} X1, i.X2, i$ .This is known as concrea te protocol . The main idea behind this construction $res=\sum_{i=1}^{n} X1, i.X2, i$ is the sum of $X_{1,i}$. $X_{2,i}$ $(i \in [1,n])$. Then the server generate the proof $\sigma_{1,i}$ for the data $X_{1,i}$. $X_{2,I}$, and aggregates these proofs into a whole one . Thus the proof for the final result res is obtained.

5. PROPOSED SYSTEM

In the existed system we do not concern about the sensitivity and security of the data. Hence, in this paper, we propose a security technique called AES (Advance Encryption Standard). And also we use the extra back-up server for the data reliability. Whenever a client wants to retrieve a data stream it should authenticate the server by using the secret keys. Then only the client can retrieve the data streams from the cloud. This model also ensures the data reliability and consistency of the database.



Algorithm Formulation

KeyGen (1^k)

- For j=1 to l do
- Choose a random number $sk_i=s_j \in Zq *$ as the secret key
- Compute $pk_{j=} g^{sj}$
- Output ($pk_{j,}$ skj)
- End of

TagGen (sk_j,j, Xj,i)

- 1.compute $\sigma j,i = (g_1^{(h1,Mj,i)} g_2^{(h2,Mj,i)} g_3^{Xj,i})^{skj}$
- 2. Output $\sigma_{j,i}$

Evaluate ($F_{IP,}$ X1, X2)

- Compute res=X_1 xor X_2
- Output res

Advance Encryption Standard (AES)

- Substitution of bytes
- Shift of rows
- Mix of columns
- Add round key

Matrix Product Query Extension

The publicly verifiable inner product evaluation scheme to support the matrix product query. Specifically, machine M_1 (M_2) generates over a row vector.

A client requests the server to compute the matrix product F_{IP}=A*B. In the above equation, a and b vectors denote the inner product vectors. To provide a proof of the matrix product computation, a possible approach is to directly extend the inner product verification algorithm. Let the result represented as (i^{th}, j^{th}) entry of the matrix AXB. The server can generate a proof $\pi_{i,j}$ for res[i][j] and then send all the proofs $\pi_{i,j}(1 \leq i \leq n, 1 \leq j \leq n)$ to the client. However, that naïve solution may be prohibitive as the proof size is $O(n^2)$.

Advantages

In this project, we focus on the inner product evaluation and also we ensure the integrity of the data. It is the multi-key encryption technique. That is when the client wants to download the data the retrieval of the query is only done by the multi-key verification. We also use the additional server for the data availability. If the data in the local server are lost or any misconception occurred, the proxy server downloads the data from the original server. It ensures the data availability.

6. CONCLUSION

In this paper, we introduce a security mechanism for the data integrity. The novel homomorphic verifiable tag technique is also used. And we design the efficient and publicly verifiable inner product computation scheme on the dynamic outsourced data streams under multiple-keys. We also extend the inner product to the matrix product evaluation. It allows multiple data sources with different secret keys to upload their endless data streams and delegate the corresponding computations to a third-party server. Experimental results demonstrate that our protocol is practically efficient in both communication and computation cost.

7. REFERENCES

[1] Y. Zhu and D. Shasha, "Statstream: Statistical monitoring of thousands of data streams in real time," in *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 358–369.
[2] W. Sun, X. Liu, W. Lou, Y. T. Hou, and H. Li, "Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2110–2118.
[3] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: secure multiowner data sharing for dynamic groups in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182–1191, 2013.
[4] S. Nath and R. Venkatesan, "Publicly verifiable grouped aggregation queries on outsourced data streams," in *International Conference on Data Engineering*. IEEE, 2013, pp. 517–528.
[5] D. Catalano and D. Fiore, "Practical homomorphic macs for arithmetic circuits," in *Advances in Cryptology–EUROCRYPT*. Springer, 2013, pp. 336–352.
[6] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators," in *Advances in Cryptology-ASIACRYPT*. Springer, 2013, pp. 301–320.
[7] 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.
[8] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *Advances in Cryptology– EUROCRYPT*. Springer, 2011, pp. 149–168.
[9] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in Cryptology–CRYPTO*. Springer, 2010, pp. 483–501.
[10] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Advances in Cryptology–CRYPTO*. Springer, 2010, pp.
465–482.
[11] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," in *ACM symposium on Theory of computing*. ACM, 2008, pp. 113–122.
[12] J. R. Thaler, "Practical verified computation with streaming interactive proofs," Ph.D. dissertation, Harvard University, 2013.
[13] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Advances in Cryptology– CRYPTO*. Springer, 2011, pp. 111–131.
[14] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *ACM conference on Computer and communications security*. ACM, 2012, pp. 501–512.
[15] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute-based encryption," in *Theory of Cryptography*. Springer, 2012, pp. 422–439.
[16] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 238–252.
[17] V. Vu, S. Setty, A. J. Blumberg, and M. Walfish, "A hybrid architecture for interactive verifiable computation," in *IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 223–237.
[18] S. T. Setty, V. Vu, N. Panpalia, B. Braun, A. J. Blumberg, and M. Walfish, "Taking proof-based verified computation a few steps closer to practicality." in *USENIX Security Symposium*, 2012, pp. 253–268.
[19] S. T. Setty, R. McPherson, A. J. Blumberg, and M. Walfish, "Making argument systems for outsourced computation practical (Sometimes)." In *NDSS*, 2012.
[20] K.-M. Chung, Y. T. Kalai, F.-H. Liu, and R. Raz, "Memory Delegation," in *Advances in Cryptology–CRYPTO*. Springer, 2011, pp. 151–168.

