ISSN:2231-0711

M.Thangadurai et al | IJCSET(www.ijcset.net) | September 2016 | Vol 6, Issue 9, 331-334

Secure Outsourced Data Stream under Multiple Keys Using Random Algorithm in Cloud Computing

M.Thangadurai [#1], K.Baskar [*2]

[#]*PG Scholar- CSE dept, Kongunadu College Of Engineering &Technology, Trichy, Tamilnadu,India*

[*]*Assistant professor-CSE dept,Kongunadu College Of Engineering &Technology, Trichy, Tamilnadu,India*

***Abstract*—Exchanging data streams to an advantage rich cloud server for internal thing evaluation, a basic building frustrate in various standard stream applications (e.g., quantifiable watching), is addressing various associations and individuals. On the other hand, checking the delayed consequence of the remote count accept an essential part in tending to the issue of trust. Since the outsourced data amassing likely starts from different data sources, it is needed for the structure to have the ability to pinpoint the originator of bungles by administering each data source a unique secret key, which requires the inward thing check to be performed under any two social occasions' differing keys. Regardless, the present game plans either depend on upon a lone key supposition or powerful yet practically inefficient totally homomorphic cryptosystems. In this paper, we focus on the all the more troublesome multi-key circumstance where data streams are exchanged by various data sources with unmistakable keys. We first present a novel homomorphic clear name framework to transparently affirm the outsourced inward thing count on the dynamic data streams, and after that extend it to support the check of network thing count. We exhibit the security of our arrangement in the discretionary prophet model. Furthermore, the exploratory result similarly shows the practicability of our arrangement.**

***Keywords*— Data stream, Computation outsourcing, Storage outsourcing, Multiple keys, Public verifiability**

I. INTRODUCTION

The past few years have witnessed the proliferation of streaming data generated by a variety of applications/systems, such as GPS, Internet traffic, asset tracking, wireless sensors, etc. Retaining a local copy of such exponentially-growing volume of data is becoming prohibitive for resource-constrained companies/ organizations, let alone offering efficient and reliable query services on it. Consider a stream-oriented service (e.g., market analysis, weather forecasting and traffic management), where *multiple* resource-constrained sources continuously collect or generate data streams, and outsource them to a powerful external server, e.g. cloud, for desired critical computations and storage savings. For example, using inner product computation over any two outsourced stock data streams from different sources for correlation analysis, a stock market trader is able to spot the arbitrage opportunities.

The secret keys used by data sources to make names won't be traded to clients for the result affirmation; for the most part, a malicious client with the private keys can plan with the server to change the data and make relating names to mislead distinctive clients. In this paper, we focus on the affirmation of the outsourced figuring over open data streams, while sensitive data protection is outside the degree of our work.

II. RELATED WORK

The problem of verifying the outsourced algebraic computation has attracted extensive attention in the past few years. These schemes can be divided into two categories: under single-key setting [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] and under multi-key setting [23][24].

Single-key Setting. Fully homomorphic message authenticators [6], [7], [8] 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 [9] 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 [6], [7], [8] or publicly [9] verify the computation result. Our work is also related to a line of verifiable schemes [10], [11], [12], [13], [14], where a resource-constrained data source can outsource a computationally-intensive task to a third-party server and efficiently verify computation result. Recently, several works towards public verification either for specific classes of computations [15], [16] or for arbitrary computations [17] have been proposed. However, the outsourced data [15], [16] has to be a priori fixed. Another interesting line of works [18], [19], [20] considered a different setting for verifiable computation.In their models, the client needs to know the We consider our structure plan as plot in There are a course of action of machines (data sources) each of which claims a fascinating open besides, private key pair..

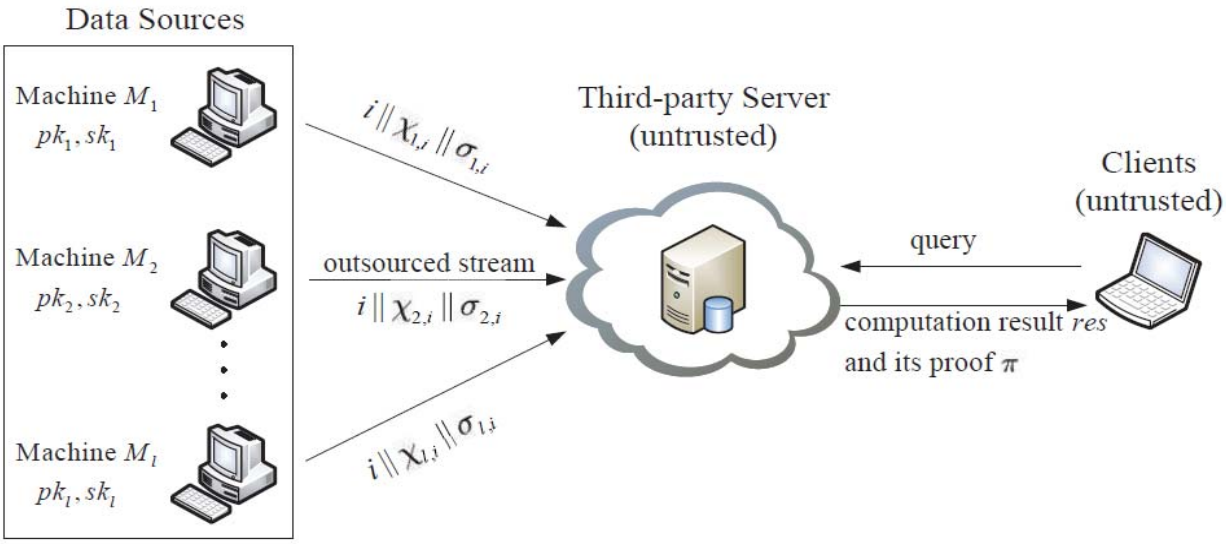

FIG.1. SYSTEM MODEL

These machines accumulate or create possibly unbounded data streams and outsource them to an outcast server. We expect that these machines are not required to clearly bestow with each other. More conclusively, for another data forms a holomorphic and transparently irrefutable tag and outsources a tuple to the server. The time measured in our arrangement is discrete and extended with the arrival of another tuple

Multi-key Setting. Recently, a multi-key noninteractive verifiable computation scheme was proposed in [23], followed by a stronger security guarantee scheme [24]. In their constructions, n computationally-weak users outsource to an untrusted server the computation of a function f over a series of joint inputs (x(i) 1 ; x(i) 2 ; :::; x(i) n) without interacting with each other, where i denotes the ith computation. In their schemes, after the generation of system parameters, data sources Pj(j ∈ [1; n]) outputs an encoded function f to the server. Then for the ith computation, Pj outsources the encoding of x(i) j to the server and computes a secret for the verification. However, these schemes may not be applied to the stream setting since sources lost data control after the outsourcing and thus cannot generate the corresponding secrets for the verification.

In this work, we 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.

III. DYNAMIC BROADCAST ENCRYPTION

Broadcast encryption enables a broadcaster to transmit encrypted data to a set of users so that only a privileged subset of users can decrypt the data. Besides the above characteristics, dynamic broadcast encryption also allows the group manager to dynamically include new members while preserving previously computed information, i.e., user decryption keys need not be recomputed, the morphology and size of ciphertexts are unchanged and the group encryption key requires no modification. The first formal definition and construction of dynamic broadcast encryption are introduced based on the bilinear pairing technique in [14], which will be used as the basis for file sharing in dynamic groups.

IV. SYSTEM MODEL AND DESIGN GOALS

A. System Model

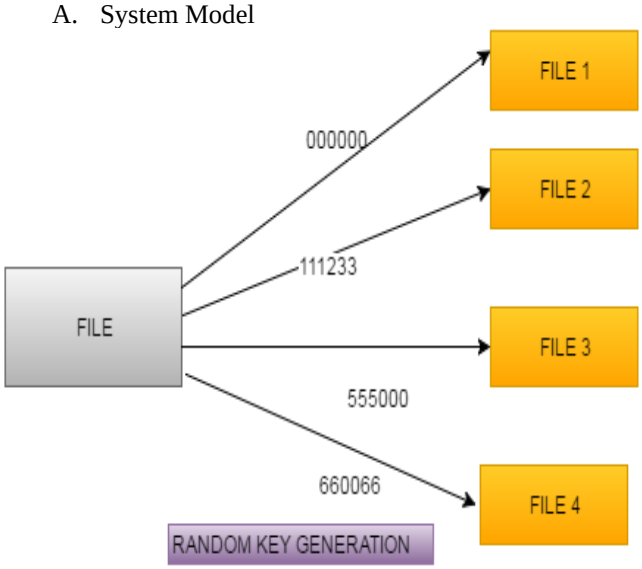

Fig 3.Random Key Generation

We consider a cloud computing architecture by combiningwith an example that a company uses a cloud to enable its staffs in the same group or department to share files. The system model consists of three different entities: the cloud, a group manager (i.e., the company manager), and a large number of group members (i.e., the staffs) as illustrated . Cloud is operated by CSPs and provides priced bundant storage services. However, the cloud is not fully trusted byusers since the CSPs are very likely to be outside

of thecloud users' trusted domain. Similar to [3], [7], we assumethat the cloud server is honest but curious. That is, the cloudserver will not maliciously delete or modify user data dueto the protection of data auditing schemes [17], [18], but willtry to learn the content of the stored data and the identitiesof cloud users.Group manager takes charge of system parametersgeneration, user registration, user revocation, and revealing the real identity of a dispute data owner. In the givenexample, the group manager is acted by the administrator of the company. Therefore, we assume that the group manager is fully trusted by the other parties.

Group members are a set of registered users that will store their private data into the cloud server and share them with others in the group. In our example, the staffs play the role of group members. Note that, the group membership is dynamically changed, due to the staff resignation and new employee participation in the company.

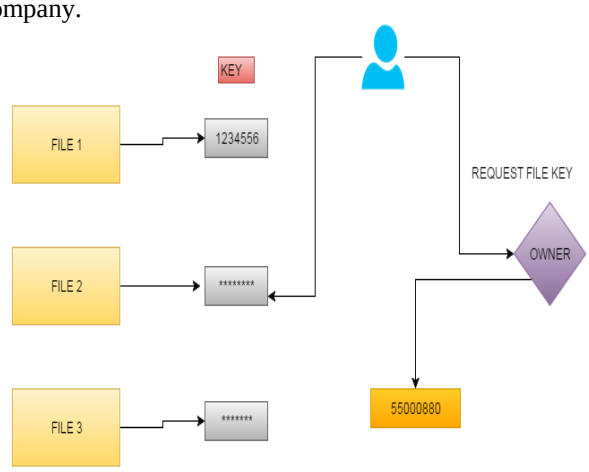

Fig 4.Multi Key Generation

V. Design Goals

In this section, we describe the main design goals of the proposed scheme including access control, data confidentiality, anonymity and traceability, and efficiency as follows:

Access control: The requirement of access control is twofold. First, group members are able to use the cloud resource for data operations. Second, unauthorized users cannot access the cloud resource at any time, and revoked users will be incapable of using the cloud again once they are revoked.

Data confidentiality: Data confidentiality requires that learning the content of the stored data. An important and challenging issue for data confidentiality is to maintain its availability for dynamic groups.

Multi-key setting: 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 his/her 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.

Query flexibility: The client should be free to choose any portion of the data streams as the input of the queried computation.

Public verifiability: All the participants involved in the protocol should be able to publicly verify the outsourced computation results without sharing secret keys with data sources.

Efficiency: More precisely, we expect that 1) the communication overhead between a client and the server is constant, i.e., independent of its input size of the queried computation, and that 2) verification overhead on the client side should be smaller than performing the outsourced computation by the client.

$\mathrm{KeyGen}(1^\kappa)$:
1. for $j = 1$ to l do
2. choose a random number $sk_j = s_j \in Z_q^*$ as the secret key
3. compute $pk_j = g^{s_j}$
4. output (pk_j, sk_j)
5. end for

$\mathrm{TagGen}(sk_j, i, \mathcal{X}_{j,i})$:
1. compute $\sigma_{j,i} = (g_1^{h_1(M_j,i)} g_2^{h_2(M_j,i)} g_3^{\mathcal{X}_{j,i}})^{sk_j}$
2. output $\sigma_{j,i}$

$\mathrm{Evaluate}(\mathcal{F}_{\mathcal{GS}}, \mathcal{X}_j)$:
1. compute $res = \sum_{i \in \Delta} \mathcal{X}_{j,i}$
2. output res

$\mathrm{GenProof}(\mathcal{F}_{\mathcal{GS}}, \sigma_j, \mathcal{X}_j)$:
1. compute $\pi = \prod_{i \in \Delta} \sigma_{j,i}$
2. output π

$\mathrm{CheckProof}(\mathcal{F}_{\mathcal{GS}}, pk_j, res, \pi)$:
1. set $S_\Delta = (S_1, S_2)$
2. compute $S_1 = \sum_{i \in \Delta} h_1(M_j, i)$ and $S_2 = \sum_{i \in \Delta} h_2(M_j, i)$
3. if $(e(\pi, g) = e(g_1^{S_1} g_2^{S_2} g_3^{res}, pk_j))$ then
4. output 1
5. else
6. output 0
7. end if

VI. Conclusions

In this paper, we introduce a novel holomorphic verifiable tag technique, and design an efficient and publicly verifiable inner product computation scheme on the dynamic outsourced data streams under multiple keys. We also extend the inner product scheme to support matrix product. Compared with the existing works under the single-key setting, our scheme aims at the more challenging *multi-key* scenario, i.e., it allows multiple data sources with different secret key sto upload their *endless data streams* and delegate the corresponding computations to a third party server, 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 cost.

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] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos, "Security and privacy for storage and computation in cloud computing," *Information Sciences*, vol. 258, pp. 371– 386, 2014.

[5] 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.

[6] D. Catalano and D. Fiore, "Practical homomorphic macs for arithmetic circuits," in *Advances in Cryptology–EUROCRYPT*. Springer, 2013, pp. 336–352.

[7] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators," in *Advances in Cryptology-ASIACRYPT*. Springer, 2013, pp. 301–320.

[8] 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.

[9] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *Advances in Cryptology– EUROCRYPT*. Springer, 2011, pp. 149–168.

[10] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegationof computation using fully homomorphic encryption," in *Advancesin Cryptology–CRYPTO*. Springer, 2010, pp. 483–501.

[11] 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.

[12] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," in *ACM symposiumon Theory of computing*. ACM, 2008, pp. 113–122.

[13] J. R. Thaler, "Practical verified computation with streaming interactive proofs," Ph.D. dissertation, Harvard University, 2013.

[14] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Advances in Cryptology–CRYPTO*. Springer, 2011, pp. 111–131.

[15] 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.

[16] B. Parno, M. Raykova, and V. Vaikuntanathan, "Howto delegate and verify in public: Verifiable computation from attribute-based encryption," in *Theory of Cryptography*. Springer, 2012, pp. 422–439.

[17] 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.

[18] 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.

[19] 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.

[20] S. T. Setty, R. McPherson, A. J. Blumberg, and M. Walfish, "Making argument systems for outsourced computation practical (sometimes)." in NDSS, 2012.