Indonesian Journal of Electrical Engineering and Computer Science
Vol. 7, No. 3, September 2017, pp. 830 ~ 838
DOI: 10.11591/ijeecs.v7.i3.pp830-838

Secure Cloud based Privacy Preserving DataMinning Platform

S Kumaraswamy*[1], Manjula S H[2], K R Venugopal[3]
Department of Computer Science and Engineering, Sri Siddhartha Institute of Technology, Tumakuru, Bangalore, India 572105
*Corresponding author, e-mail: kumarswamy.vtu@gmail.com

Abstract

The adoption of cloud environment for various application uses has led to security and privacy concern of user's data. To protect user data and privacy on such platform is an area of concern. Many cryptography strategy has been presented to provide secure sharing of resource on cloud platform. These methods tries to achieve a secure authentication strategy to realize feature such as self-blindable access tickets, group signatures, anonymous access tickets, minimal disclosure of tickets and revocation but each one varies in realization of these features. Each feature requires different cryptography mechanism for realization. Due to this it induces computation complexity which affects the deployment of these models in practical application. Most of these techniques are designed for a particular application environment and adopt public key cryptography which incurs high cost due to computation complexity. To address these issues this work present an secure and efficient privacy preserving of mining data on public cloud platform by adopting party and key based authentication strategy. The proposed SCPPDM (Secure Cloud Privacy Preserving Data Mining) is deployed on Microsoft azure cloud platform. Experiment is conducted to evaluate computation complexity. The outcome shows the proposed model achieves significant performance interm of computation overhead and cost.

***Keywords**: Cloud computing, Cryptography, Content sharing, Privacy preseving*

1. Introduction

Cloud computing is a platform that provide remote client/devices access to shared/hosted service. It offers services, resources on demand and pay as you go. The cloud offers advanced technological infrastructure service benefit to its client. It allows remote user such as small, medium to large sized company/organization to use computational infrastructure such as software, virtual machine, storage etc. The cloud services are broadly classified as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as Service (IaaS) [1]. Google, Amazon [2] and Microsoft [3] are some of the well-known public cloud platform provider. Different cloud provider offer different cloud services such as web hosting, storage, computing platform and so on.

As of today there has been increase in number of online transaction for various application uses. The application such as social networking, smart transport application, health care sector, e-commerce and so on [4-8]. The adoption of such application is expected to grow rapidly due to huge potential and backup by various government and industries such as hospital, banks, database etc. Majority of these application services adopt cloud platform for storage, processing and computation. The aim of all these services is to save money and time. The major issues with online application based service delivery are the theft of private information [9]. Therefore ensuring security and privacy is most desired.

A customary way of protecting data or reducing information leakage is by adopting cryptography mechanism i.e. by encrypting data. Nevertheless, this induces additional overhead in cloud or remote server environment. It is a challenging task to search and retrieve the encrypted resource. Recently many approaches have been proposed to search on encrypted data [10-13] by adopting different cryptography technique. These methods presented have a provable security yet they induce high computation overhead. When working with mining data, the data has huge volume and varieties as a result these methods are not suitable.

Received May 9, 2017; Revised July 16, 2017; Accepted August 2, 2017

To address, the Digital signature scheme is adopted by various methods. The Digital signature with public-key cryptography is first invented by Diffie and Hellman [14] and was first adopted by Rivest, Shamir and Adleman [15]. Digital signatures consist of following setup such as Key Generation process which produces secret/master key and public key, Signing process which generates a signature by using secret/master key for a data and Verification process checks whether some signature on data using public key. This model is not applicable in practice since the signature is valid if one-way function is exist [15] [16] i.e. the signer key need to be changed during the execution process of signing process. Therefore a random signature generation scheme is required but it is not suitable for plain text scheme [17] as a result they are not provably secured. In [18, 19] presented first signature model that support practical implementation with strong RSA assumption. The model is not evaluated under Ideal randomness assumption and designed for specific standalone application. Therefore, the key is to develop an anonymous access ticket (credential) signature generation model.

In anonymous access ticket model the Requestor access the resources by presenting an access ticket that he is authenticated. The real example of this scenario is the Passport, Driver license etc. as access ticket which permits the Requestor to prove ownership/ citizenship that authorize the Requestor to vote, drive vehicle etc. In digital environment the Requestor identity is his master key and the access ticket as a signature on this Master key. An access ticket model is said to be anonymous if it allows Requestor to get access ticket without revealing more information of its identity. By adopting zero-knowledge proofs we can achieve this but it induces additional computation which is expensive. Since it requires the Boolean circuitry to prove the statement is true. Many existing model adopted two party computation models to allow Requestor to be anonymous. This also induces computation overhead due to Boolean circuitry gat by gate verification. To address these in this work we consider that an anonymous access ticket model can allow for a setting in which Requestor can have varied scenario and varied pseudonyms. Most of the existing access ticket signature model has adopted signature generation based on public key cryptography model which induces high computation overhead. Especially when it is deployed on cloud platform it increases cost.

Here we present a key based access key verification model which allow the Ticket Provider (TP), and Ticket Checker (TC) to share admittance to the some master keys, then Elliptical Curve Cryptography (ECC) is adopted to validate that the TP and TC belong to similar group. As long as the Requestor does not required to be authenticated by other Requestor the TP can share distinct master key with all the TC and then at same time it gives access tickets to all TC. In a non-interaction or a standalone model the TP must give access ticket to all TC at same time which is not suitable when large number of TC exist in a system. We consider an online cloud based model where the Requestor wishes to get access ticket from the TP whenever he is desired. The blind issuance adopted in our model allows the Requestor to get to access ticket on same attributes without revealing it to the TP, i.e. in an anonymous way. Our model allows to translation of publically access ticket verification into an efficient keyed based access ticket verifiable with same function and attributes. The key challenges of secure resource provisioning in public cloud environment [20, 21] such as availability, data confidentiality, security, communication bottleneck, inside attack, outside attack, privacy consideration for processing of sensitive data [22, 23], [34].

1.1. Motivation and Contribution

The adoption of cloud computing environment by various applications has raised numerous security issues such as Denial of Service (DoS), malicious insiders and data loss which have been broadly analyzed in [20, 21]. These issues exist mainly due to loss of control over data, multi-tenancy and trust. The major public cloud service provider such as Microsoft Azure, Goggle clouds and Amazon simple storage services namely Amazon S3 do not assure in service level agreement specific to security and privacy as part of contractual agreements condition among cloud consumer and cloud service provider. Therefore it is critical and necessary concern in providing security and privacy when cloud platform is adopted. Therefore when designing a secure resource sharing model using cloud platform the security and privacy concern of all parties involved in the model must be taken care off.

The model presented here achieves anonymous access ticket sharing among different parties for secure access of resources stored public cloud platform. The proposed model provides party based security provisioning with reduced computation overhead by adopting key based encryption technique.

1.3. Related Work

With the rapid growth of online based service delivery such as social media, healthcare and so on. These platforms adopt cloud technology which store data of user in cloud storage for various purposes. Therefore providing security and preserving privacy of data is most desired. There have been several methodologies that have been proposed in recent time in order to improve the security and privacy of data that are stored in remotely or on public cloud environment which are surveyed below.

In [25] considered the issue pertaining to secure sharing of huge multimedia data and its privacy. The main concern of their approach is that how to preserve privacy for such huge volume and variety of data and how to enforce privacy policies to provide secure and efficient resource management. The research survey conducted by them shows that the privacy policy enforcement should be done in access control model. They presented a hybrid model namely Intelligent Privacy Manager (iPM) where the requirement of privacy is implemented in access control mechanism and presented an enforcement and composition of privacy policies framework. They considered permitting only the Requestor to provide a conflict free privacy rule and not security manage or architecture. They also considered their model to be context aware and proved privacy policy correctness. To ensure providing security of sensitive data and enforce privacy policies a verification step is performed. The implantation model did not consider the cloud platform and there is no simulation proof to show their model reduces computation cost.

In [26] they considered a privacy preserving model for encrypted data that are stored in cloud environment by the user. Since the data stored on cloud platform is of encrypted form there is a need of secure and efficient search mechanism of cipher text. The area of concerned here is how to categorize document during encryption which is a crucial factor for retrieval result accuracy. The increase in storage of huge variety and volume of data on cloud storage environment brings added challenges in providing efficient online large encrypted resource retrieval model. Here they presented an hierarchical clustering based semantic information retrieval system of large encrypted data. Their model used minimum relevance threshold to cluster the document. Their model is solving linear computation complexities issues and to authenticate search result privacy they adopted hash sub-tree. They conducted experimental study on IEEE Xplore built datasets. The outcome of their result show better rank privacy then its counter parts. They did not considered query encryption as a result the prone to chosen cipher attack.

In [27] presented a architecture namely CypherDB to protect the data confidentiality that are store in external cloud storage environment. Their model enables an efficient and low computation model to retrieve resource from cloud database storage environment. The entire data are encrypted and are stored in cloud database which can be retrieved securely by using their CypherDB secure processor architecture. To avoid data leakage and optimizing computation efficiency their architecture data paths is tightly coupled during query execution and database access. Simulation is conducted on TPC-H database and compared with well-known SQLite database management architecture. An average reduction of 10% is achieved considering without security. They adopted symmetric approach using 128-bit security keys which does not incur much overhead at the same time security is compromised as compared with other public-key cryptography. Any changes in architecture is difficult to achieve as a result their model is not robust.

In [28] presented a framework to address the issue pertaining to data sharing on cloud based big data environment. There is been increased personal data storage in big data environment. Here they presented secure data sharing, storage, processing and destruction on cloud hosted virtual machine. They adopted proxy re-encryption technique for cipher generation by using AES symmetric algorithm on a semi trusted server. Their model preserves the privacy of data owner in protecting sensitive information. It also retains complete control of data owner data. The drawback with their model proxy key generation is done on semi trusted server which is protected by TPM key. As a result any party misbehaving will result in loss of privacy and SLA (Service level agreement).

In [29] identified the computation complexity issue pertaining in designing and comparing cryptographic mechanism to preserve privacy of data. Here they compared some of existing approach such garbled circuits, secret sharing and homomorphic approach. They evaluated the performance efficiency of each model in order to evaluate the efficiency claim by existing approaches. To evaluate a fair comparison among these models the following performance metric is considered. The parameters are bandwidth requirement, run-time and complexity. Here they considered well known two party mechanism on semi trusted environment. The performance of these metric are evaluated in following stages, initialization, online computation and pre-processing considering identical platform. The overall outcome of their model shows pros and cons of different model.

The overall survey shows that preserving privacy of data in cloud platform is most desired. The existing lacks in providing efficient privacy preserving of data owner due to computation complexity. Therefore there is a necessity to develop a zero knowledge proof based access ticket mechanism for public cloud environment and reducing the cloud expense/cost.

The paper organization is as follows: The Related work is presented in section two. The research background work is presented in section three. In section four the problem definition is identified. The proposed models are presented in Section five. The simulation results and the experimental study are presented in the section six. The concluding remark is discussed in the last section.

2. Research Method

In [30] presented multi-authority CP-ABE model for secure data access in cloud environment. Their model adopts is presented in random oracle model which is considered to be CPA- secure considering constant cipher text under q-Bidirectional Diffe-Hellman encryption (q-BDHE) assumption. To reduce the computation complexity of data owner the computation is outsources to cloud environment. Similarly in [31] presented CP-ABE access control mechanism by using homomorphic encryption for public cloud environment for secure sharing of multimedia data. Both these protocol achieves significant performance improvement over existing approaches. The main drawback of their strategy is they induce high cloud expenses since the proxy key generation is computed on cloud environment. When the number of group is increased the cloud computation cost also increases linearly.

Given a Cloud based model with N set of group/Requestors, the objectives are given as follows:

1. To develop a secure public cloud platform to store mining data using security services
2. To develop a key based access ticket verification model.
3. To evaluate the performance of proposed model interm of cloud computation cost, processing time
4. To evaluate performance by varying the attributes size.
5. To evaluate performance by varying the group/requestor size.

2.1 System Model

The proposed secure public cloud platform consists of following parties, Requestor, Knowledge Base Provider (KBP) and Privacy Preserving Data Mining (PPDM) platforms. The KBP consist of following parties Ticket Provider which act as a third party between Requestor and the PPDM cloud platform and PPDM consist of following parties Ticket Checker, Validator and Revocator. The Proposed Secure Cloud Privacy Preserving Data Mining (SCPPDM) architecture is shown in Figure 1. The Requestor, obtain access ticket from different TP and collect information of access ticket pertaining to a specific TC. The Requestor is considered to be in local system or mobile platform or on trusted cloud platform.

The TP provides access ticket to Requestor, which assures the correctness of access ticket information for a particular Requestor to the person the ticket, is given. The TP must first authenticate the Requestor before giving the access ticket. The authentication is done in presence of user physically or remotely through online by collecting information such username and password etc.

The TC provides a secure access control mechanism to access services or resources. It restrict the access tickets that Requestor must possess and also restrict the amount of information of access ticket the Requestor need to present in order to access resources. The restriction imposed of TC depends on Ticket Specifier (TS). The Requestor computes from their access ticket which consist necessary information and valid cryptography proof.

The Revocator is accountable for access ticket revocation. It prevents the access ticket in generating of specifier ticket. The access ticket cannot be used by any TC for any purposes, if the TP execute access ticket revocation from Revocator. Similarly the revocation is considered to be with in TC and does not affect the specifiers of other TC, if revocations are executed by the TC. The party such as the Requestor and the TC must first obtain the most recently updated revocated data from the Revocator to compute, respectively validate, ticket specifier.

The Validator is a trusted party which preserves privacy of specifier tickets under certain conditions. To assure the TC first agree in the specifier rule in which Validator should be accessible to attributes under certain conditions. Therefore the Requestor has knowledge of privacy preserving when the ticket is computed and can take part to achieve it. This helps the Requestor to make desired resolution based on their requirement and trust of Validator.

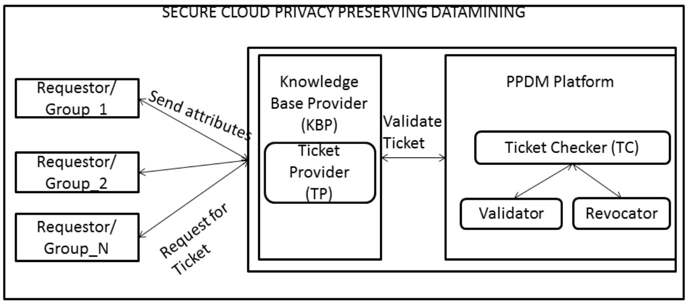

Figure 1. Proposed Secure Cloud Privacy Preserving Datamining Architecture

2.1.1 A. Keyed Verification Access Ticket

$Process(1^n)$ is a system specific parameter (p) that is available to models and guarantee all parties it is generated correctly. $AcctktKeycmpt\ (p)$ is computed by TP on given input p to compute TP parameter t and master key mk.

$BlindTicket(mk, S) \leftrightarrow BlindRetrieve\left(i, \left(x_1, \ldots x_y\right)\right)$ is an possible collaborating mechanism where Requestor can obtain access ticket on attributes $\left(x_1, \ldots x_y\right)$ form TP which gives Subset B of its attributes. $View\left(i, a, \left(x_1, \ldots x_y\right), \mu\right) \leftrightarrow ViewCheck(mk, \mu)$ is an collaborating mechanism among Requestor and a TC. View is invoked by Requestor to assure evidence of ownership of access ticket a attesting some set of attributes $\left(x_1, \ldots x_y\right)$ that fulfil a set of conditions μ under key resultant to i. The $ViewCheck()$ is invoked by the TC in ownership of mk to check that the Requestor has a access ticket for attributes fulfilling the condition μ.

2.1.2 Keyed Verification Access ticket from SCPPDM

The proposed access ticket model from $SCPPDM = (Process, Keycmpt, SCPPDM, Check)$. We now compute the following process mechanism for access ticket model.

$Process(1^n) = Output(U, z, u, j) \overset{\Delta}{\leftarrow} Process(1^n)$. $AcctktKeycmpt(p)$: Parse p as (U, z, u, j).

The $SCPPDM$ keys is computed as $\left(\vec{T}, \vec{t}\right) \overset{\Delta}{\leftarrow} KeyCmpt(p)$. Now, commit to the master key $t_0$by selecting $\bar{\bar{t}}_0 \leftarrow H_z$ and establishing commitment $K_{t_0} = u^{t_0} j^{\bar{\bar{t}}_0}$. Output $i = \left(K_{t_0}, \vec{T}\right)$ and $mk = \left(\vec{t} = \bar{\bar{t}}_0\right)$ Ticket Providence: to produce an access ticket with y attributes $\left(x_1, \ldots, x_y\right) \in H_z^y$ all of which acknowledged to the TP, the TC evaluates $(c, \bar{c}) \leftarrow SCPPDM\left(mk, \left(x_1, \ldots, x_y\right)\right)$ and retrieve α and $(c, \bar{c})$ to the Requestor, where

$$\alpha \coloneqq PK\left\{\left(t_0, t_1, \ldots, t_y, \bar{\bar{t}}_0\right): \bar{c} = c^{t_0} \prod_{l=1}^{y} (c^{t_0})^{t_y} \wedge K_{t_0} = u^{t_0} j^{\bar{\bar{t}}_0} \wedge T_l = j^{t_l} \forall l \in \{1, \ldots, y\}\right\}$$

Here α demonstrates that $(c, \bar{c})$ is a usable SCPPDM with respect to access ticket model and the TP param. The Requestor accepts the output $(c, \bar{c})$ if its proof is verified or else rejects with output β. An alternative way to hide some subset of $J \subseteq \{1, \ldots, y\}$ attributes from TP, the following procedure is followed

The user generates the Elliptical curve cryptography (ECC) key pair $(m, \omega: u^{m})$, then for each attribute x_l generates an encryption u^{x_l} as $G_l = (u^{d_l}, u^{x_l}, \omega^{x_l}) \forall l \in J$, using $d_l \in_R H_z$. Then the Requestor transmits the proof of knowledge $\{d_l, x_l\}_{l \in J}$ and ciphertext to the TP. The TP selects $n \in_D H_z$ and then evaluates $c = u^{n}$, process encryption $G_{\bar{c}}$ of $\bar{c} = u^{n t_l} \prod_{1}^{y} (u^{x_l})^{n t_0}$ using homomorphic properties and the encryption is randomized by multiplying the encryption of 0 using randomness $\bar{d} \in_R H_z$ to get $\bar{G}_{\bar{c}}$. Then it transmit $c, \bar{G}_{\bar{c}}$ to the Requestor and provides a proof (the randomizing factor $\bar{d}$ and proof of knowledge of correctness $\{t_l\}_{0}^{y}, \bar{\bar{t}}_0, n)$ that param computed is correct with respect to $\left(K_{t_0}, \{T_l\}_{1}^{y}\right)$. The Requestor decrypts $\bar{G}_{\bar{c}}$ to obtain $\bar{g}$, and output $(c, \bar{c})$ else output with β if it fails to verify the proof.

2.1.3 Access Ticket Translation

In spite of proving that the cipher data G_l are well formed, the requestor can add proof of attributes that the cipher data encodes. As a result the Requestor can prove that some of attributes x_l are similar in another access tickets which aid in reducing computation overhead resulting in increase in cost.

Access Ticket Specifier: Here the computation of View and ViewCheck is presented. Here we present how the access ticket attributes is certified for the given set of assurance. The Requestor can prove large type of condition about assured param once assurance has been established. If any attributes is exposed to the TC, the Requestor sends x_l in position of K_{x_l} and the TC directly evaluate $g^{t_l x_l}$. To reduce the computation cost of Requestor the proof of knowledge of $K_{x_l}$is not considered here in our model. The TC can re-compute the SCPPDM and compare it with assured param by using homomorphic properties of the assurance. Though there is an added criterion in SCPPDM the TC evaluates, due to randomness in assurance. To evaluate precise param required to omit these criteria and to prove the generated param is correct, the Requestor will use $\vec{T}$ from i. $View\left(p, i, \mu, a, \{x_l\}_{l}^{y}\right)$: The prover selects $d, q_1, \ldots, q_y \epsilon_D H_z$ and parses $Acctkt = (c, \bar{c})$. It then evaluates $\left\{K_{x_l} = c^{x_l} j^{q_l}\right\}_{l=1}^{y}$ and $K_{c_l} := \bar{c} u^{d}$ and transmits proof of knowledge α and $\delta = \left(c, \left\{K_{x_l}\right\}_{l}^{y}, K_{\bar{c}}\right)$. The α is evaluated as follows:

$$\alpha = \; PK\left\{(\vec{x}, \vec{q}, -d): \mu\left(x_1, \ldots, x_y\right) = 1 \wedge K_{x_l} = c^{x_l} j^{q_l} \forall_l \in \{1, \ldots, y\} \wedge W = u^{-d} \prod_{l=1}^{y} T_{l^{q_l}}\right\}.$$

$ViewCheck(p, i, \mu, \{t_l\}_{l}^{y}, q, \delta, \alpha)$: The TC parses $\delta = \left(c, \left\{K_{x_l}\right\}_{l}^{y}, K_{\bar{c}}\right)$, and evaluates M as follows:

$$W = {}^{c^{t_0} \prod_{l=1}^{y} K_{x_l}^{t_l}} \Big/ {}_{K_{\bar{c}}}$$

and check α using w. If α is not usable it outputs with β else outputs$\left(K_{x_{1, \ldots, K_{x_y}}}\right)$. The security of SCPPDM model works well when $y = 1$ and both party are considered to be honest and the TC evaluates:

$$W = {}^{K_{x_1}^{t_1} c^{t_0}} \Big/ {}_{K_{\bar{c}}} = {}^{c^{x_1 t_1} j^{t_1 q_1} c^{t_0}} \Big/ {}_{c^{x_1 t_1 + t_0} u^{d}} = j^{t_1 q_1} u^{-d} = T_1^{q_1} u^{-d},$$

which matches the condition in α. The presented model is evaluated interm of computation overhead efficiency which is presented in next section.

3. Results and Analysis

To evaluate the performance of proposed SCPPDM model the experiment is conducted on following environment. The model is implemented on windows 10 enterprises operating system, I-5 class quad core processor with 8GB of RAM. The SCPPDM model is developed using C# and Microsoft Dot net Framework 4.5 and above, The SCPPDM model is deployed on D2 instance Microsoft Azure public cloud storage platform. The D2 instance charges its user 16.82 rupees (₹)/hour. The experiment is conducted to evaluate computation overhead by varying ticket and attributes size.

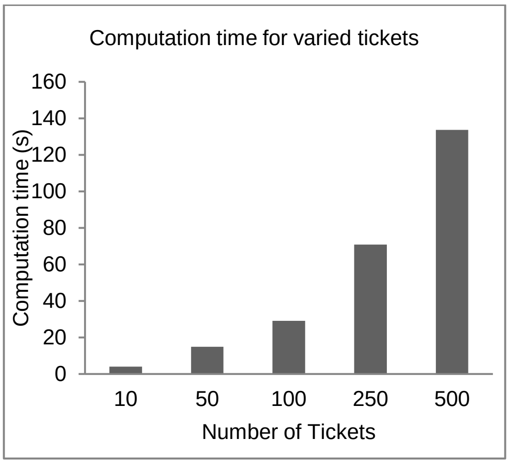

Figure 2. Computation time for varied tickets

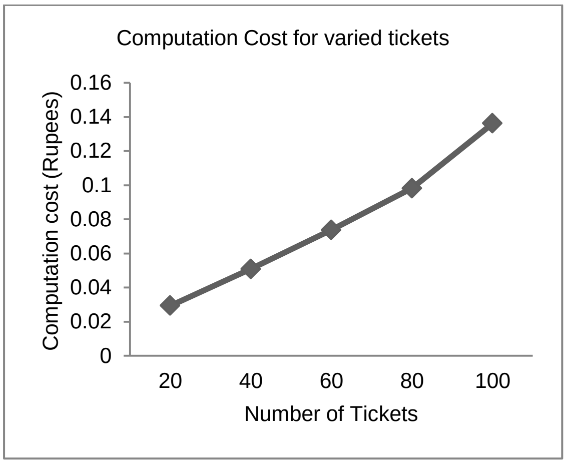

Figure 3. Computation cost for varied ticket

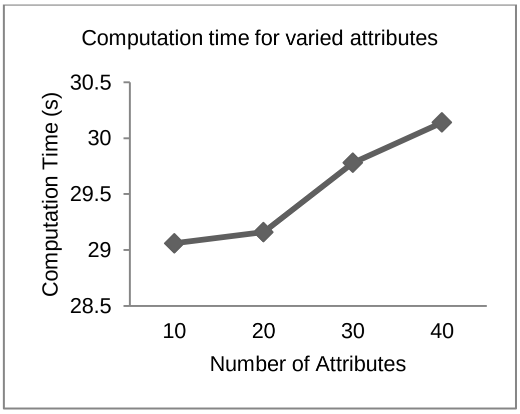

Figure 4. Computation time for varied attributes

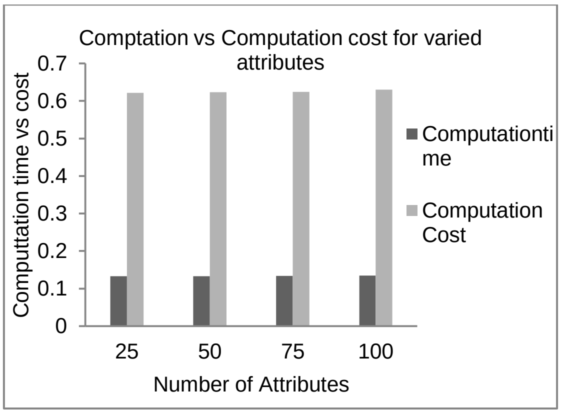

Figure 5. Computation time vs Computation cost for varied Attributes

In Figure 2, the computation time of proposed model to generate ticket and issues ticket to Requestor for varied tickets is shown. The ticket size is varied form 10, 50, 100, 250 and 500. The computation time for 10 ticket is 4.01 seconds, for 50 is 14.92 seconds, for 100 is 29.16 seconds, for 250 is 70.92 seconds and for 500 is 133.78 seconds. The experimental outcomes show when the ticket size is increased the computation time is also increase. The outcome show the proposed model is scalable irrespective of application since it can generate and issue large number of ticket.

In Figure 3, the computation cost of proposed model to generate ticket and issues ticket to Requestor for varied tickets is shown. The ticket size is varied form 20, 40, 60, 80 and 100. The computation cost accessing resource on Microsoft azure D2 instance charges 16.82 rupees (₹) per hour usage. The computation cost for 10 ticket is 0.026 ₹, for 40 is 0.0509₹, for 60 is

0.0738 ₹, for 80 is 0.098 ₹ and for 100 is 0.13 ₹. The experimental outcomes show when the ticket size is increased the computation cost is also increase. The outcome show the proposed model is cost effective considering varied ticket size.

In Figure 4, the computation time of proposed model for varied attributes is shown. The attributes size is varied form 10, 20, 30 and 40 and ticket size is kept constant with 100 tickets. The computation time for 10 ticket is 29.06 seconds, for 20 is 29.16 seconds, for 30 is 29.78 seconds and for 40 is 30.14 seconds. The experimental outcomes show the proposed model computation time is not affected by attributes size since the computation time is in range of 29.06 seconds to 30.14 seconds.

In Figure 5, the computation time and computation cost of proposed model for varied attributes is shown. The attributes size is varied form 25, 50, 75 and 100 and ticket size is kept constant with 500 tickets. The computation time and cost for 25 tickets is 133.02 seconds and 0.62₹, for 50 is 133.336 seconds and 0.622₹, for 75 is 133.67 seconds and 0.624₹ and for 100 is 134.78 seconds and 0.629₹. The experimental outcomes show the proposed model cost of computation depends on amount of time it takes to compute on azure cloud.

4. Conclusion

Providing security and preserving privacy of user access to data stored on public cloud platform is most desire. The existing models presented by C. Yanli et al. and K. Yang et al. reduces the computation time but the cost of computation for user on cloud platform increases and the authentication model presented by them is specific to particular application as result it is not applicable for secure authentication for variety of applications service that exists currently. To address this here we presented a secure and efficient privacy preserving of mining data on public cloud platform by adopting party and key based authentication strategy. The experiment is conducted to evaluate computation time and computation cost by varying number of ticket and attributes. The outcomes show the proposed model reduces computation time and cost consider azure public cloud platform. The proposed model is not affected by the attributes size which shows the robustness, scalabity and computation efficiency of our SCPPDM model.

References

[1] J Carolan, S Gaede, J Baty, G Brunette, A Licht, J Remmell, L Tucker, J Weise. Introduction to cloud computing architecture. White Paper, 1st edn. Sun Micro Systems Inc (2009).

[2] M Brantner, D Florescu, D Graf, D Kossmann, T Kraska. *Building a database on S3*. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, 2008: 251-264.

[3] A Fox, R Griffith, A Joseph, R Katz, A Konwinski, G Lee, D Patterson, A Rabkin, I Stoica. Above the clouds: A Berkeley view of cloud computing. Dept. Electrical Eng. and Computer. Sciences, University of California, Berkeley, Rep. UCB/EECS 28 (2009): 13.

[4] W Zhu, C Luo, J Wang, S Li. Multimedia cloud computing. *IEEE Signal Process*. Mag., 2011; 28(3): 59–69.

[5] Rajashree Sasamal, Rabindra Kumar Shial. Performance Analysis of Granular Computing Model on the Basis of Software Engineering and Data Mining. *IAES International Journal of Artificial Intelligence (IJ-AI)*. Dec. 2012; 1(4): 182~192.

[6] W Raghupathi, V Raghupathi. Big data analytics in healthcare: Promise and potential. *Health Inf. Sci. Syst*., 2014; 2(1): 3.

[7] P Weill, M Vitale. Place to Space: Migrating to eBusiness Models. Cambridge, MA, USA: Harvard Univ. Press, 2013.

[8] K Zhang, X Liang, X Shen, R. Lu. Exploiting multimedia services in mobile social networks from security and privacy perspectives. *IEEE Commun*. Mag., 2014; 52(3): 58–65.

[9] Privacy Rights Clearinghouse, San Diego, CA, USA. A chronology of data breaches. 2005 [Online]. Available: http://www.privacyrights.org/data-breach.

[10] G Craig. Fully homomorphic encryption using ideal lattices. STOC. Vol. 9. 2009.

[11] D Cash, S Jarecki, C Jutla, et al. Highly-scalable searchable symmetric encryption with support for Boolean queries [M].Advances in CryptologyCRYPTO 2013. *Springer Berlin Heidelberg*, 2013: 353-373.

[12] S Kamara, C Papamanthou, T Roeder. *Dynamic searchable symmetric encryption [C]*. Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012: 965-976.

[13] Cash, D, Jaeger, J, Jarecki, S, Jutla, C, Krawczyk, H, Rosu, MC, Steiner, M (2014). *Dynamic searchable encryption in very large databases: Data structures and implementation*. In Proc. of NDSS (Vol. 14).

[14] W Diffie, ME Hellman. New directions. New direction in cryptography. *IEEE Transaction on Information Theory*, 1976; IT-22(6): 944-654.

[15] Hoda Waguih. A Data Mining Approach for the Detection of Denial of Service Attack. *IAES International Journal of Artificial Intelligence (IJ-AI),* June 2013, 1976; 2(2): 99~106.

[16] J Rompel. *One-way functions are necessary and sufficient for secure signature*. In Procedding 22nd Annual ACM Symposium on Theory of Computing (STOC), pages 387-394, Baltimore, Maryland, 1990. ACM.

[17] R Canetti, O Goldreich, S Halevi. The random oracle methodology, revisited. In Proceeding 3oth Annual ACM Symposium on Theory of Computing (STOC), 1998: 209-218.

[18] R Cramer, V Shoup. *Signature Scheme based on the strong RSA assumption*. In Proceeding 6th ACM Conference on Computer and communications Security, pages 46-52. ACM press, 1999.

[19] R Gennaro, S Halevi, T Rabin. Secure hash-and-sign signatures without the random oracle. In J. Stern, editor, Advances in Cryptology EUROCRYPT 99, volume 1592 of Lecture Notes in Computers Science, pages 123-139. Springer Verlag, 1999.

[20] Cloud Security Alliance (CSA). Security Guidance for Critical Areas of Focus in Cloud Computing" version 3, 2011. Available at: htpps://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf.

[21] Cloud Security Alliance (CSA). The Notorious Nine: Cloud Computing Top Threats in 2013". Available at: https://cloudsecurityalliance.org.

[22] United Nations, the Universal Declaration of Human Rights. http://www.un.org/en/documents/udhr/index.shtml, 1948. Retrieved August 2015.

[23] S Pearson. Privacy, security and trust in cloud computing. in Privacy and Security for Cloud Computing (S. Pearson and G. Yee, eds.), Computer Communications and Networks, 2013: 3–42, *Springer* London.

[24] A Cavoukian. The Security-Privacy Paradox: Issues, misconceptions, and Strategies. https://www.ipc.on.ca/images/Resou rces/sec-priv.pdf, Retrieved November 2015.

[25] A Samuel, MI Sarfraz, H Haseeb, S Basalamah, A Ghafoor. A Framework for Composition and Enforcement of Privacy-Aware and Context-Driven Authorization Mechanism for Multimedia Big Data. *IEEE Transactions on Multimedia*, 2015; 17(9): 1484-1494.

[26] An Efficient Privacy-Preserving Ranked Keyword Search Method. IEEE Transactions on Parallel and Distributed Systems, 2016; 27(4): 951-963.

[27] BHK Chen, PYS Cheung, PYK Cheung, YK Kwok. CypherDB: A Novel Architecture for Outsourcing Secure Database Processing. *IEEE Transactions on Cloud Computing*, vol.PP, no.99, pp.1-1.

[28] Xinhua Dong, Ruixuan Li, Heng He, Wanwan Zhou, Zhengyuan Xue and Hao Wu. Secure sensitive data sharing on a big data platform. Tsinghua Science and Technology, 2015; 20(1): 72-80.

[29] T Veugen, F Blom, SJA de Hoogh, Z Erkin. Secure Comparison Protocols in the Semi-Honest Model. *IEEE Journal of Selected Topics in Signal Processing*, 2015; 9(7): 1217-1228.

[30] C Yanli, S Lingling, Y Geng. Attribute-based access control for multi-authority systems with constant size ciphertext in cloud computing. *China Communications*, 2016; 13(2): 146-162.

[31] K Yang, Z Liu, X Jia, XS Shen. Time-Domain Attribute-Based Access Control for Cloud-Based Video Content Sharing: A Cryptographic Approach. *IEEE Transactions on Multimedia*, 2016; 18(5): 940-950.