Searching Key Words Through Outsourcing Linear Programing in Cloud Computing

[1]D. Madhuri, [2]K. Eswar, [3]Dr. P. Harini

[1,2,3]Dept. of CSE, St. Ann's College of Engineering & Technology, Andhra Pradesh, India

Abstract

Cloud computing refers that the delivery of computation and storage capacity as a service to a heterogeneous community of end-users. The name comes from the use of clouds as an high-level abstraction for the complex structure. Cloud computing assign the responsibility for doing services with a user's data, software and computation over a network. It has considerable overlap with Software as a Service (SaaS). Treating the cloud as an intrinsically insecure computing platform from the viewpoint of the cloud customers, we must design mechanisms that not only protect sensitive information by enabling computations with encrypted data, but also protect customers from malicious behaviors by enabling the validation of the computation result. Such a mechanism of general secure computation outsourcing was recently shown to be feasible in theory, but to design mechanisms that are practically efficient remains a very challenging problem.

In this paper we focus on linear programming to Search the Keywords (SKW) for security/efficiency tradeoff via higher-level abstraction of LP computations than the general circuit representation. particular, by formulating private data owned by the customer for LP problem as a set of matrices and vectors, we are able to develop a set of efficient privacy-preserving problem transformation techniques, which allow customers to transform original LP problem into some arbitrary one while protecting sensitive input/output information. To validate the computation result, we further explore the fundamental duality theorem of LP computation and derive the necessary and sufficient conditions that correct result must satisfy. Such result verification mechanism is extremely efficient and incurs close-to-zero additional cost on both cloud server and customers. End users can access cloud based applications via web browser or a light weight desktop application or mobile application while the business software and data are stored on servers at a remote location. The Vindicator claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintainability, and enables Information Technology to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud computing dependent on sharing of resources to achieve coherence and the scale of economy is similar to a utility over a network.

Keywords

SAAS, LP, OS, SKW

I. Introduction

As Cloud Computing becomes prevalent, more and more sensitive information are being centralized into the cloud, such as emails, personal health records, government documents, etc. By storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service. However, the fact that data owners and cloud server are not in the same trusted domain may put the oursourced data at risk, as the cloud server may no longer be fully trusted. It follows that sensitive data usually should be encrypted prior to out- sourcing for data privacy and combating unsolicited accesses. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Moreover, in Cloud Computing, data owners may share their outsourced data with a large number of users. The individual users might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways is to selectively retrieve files through keyword-based search instead of retrieving all the encrypted files back which is completely impractical in cloud computing scenarios. Such keyword-based search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios, such as Google search [1]. Unfortunately, data encryption restricts user's ability to perform keyword search and thus makes the traditional plaintext search methods unsuitable for Cloud Computing. Besides this, data encryption also demands the protection of keyword privacy since keywords usually contain important information related to the data files. Although encryption of keywords can protect keyword privacy, it further renders the traditional plaintext search techniques useless in this scenario. espite the tremendous benefits, outsourcing computation to the commercial public cloud is also depriving customers' direct control over the systems that consume and produce their data during the computation, which inevitably brings in new security concerns and challenges towards this promising computing model [2]. On the one hand, the outsourced com- putation workloads often contain sensitive information, such as the business financial records, proprietary research data, or personally identifiable health information etc. To combat against unauthorized information leakage, sensitive data have to be encrypted before outsourcing [3] so as to provide end- to-end data confidentiality assurance in the cloud and beyond. However, ordinary data encryption techniques in essence pre- vent cloud from performing any meaningful operation of the underlying plaintext data [4], making the computation over encrypted data a very hard problem. On the other hand, the operational details inside the cloud are not transparent enough to customers [5].

As a result, there do exist various motivations for cloud server to behave unfaithfully and to return incorrect results, i.e., they may behave beyond the classical semi- honest model. For example, for the computations that require a large amount of computing resources, there are huge financial incentives for the cloud to be "lazy" if the customers cannot tell the correctness of the output. Besides, possible software bugs, hardware failures, or even outsider attacks might also affect the quality of the computed results. Thus, we argue that the cloud is intrinsically not secure from the viewpoint of customers. Without providing a mechanism for secure computation outsourcing, i.e., to protect the sensitive input and output information of the workloads and to validate the integrity of the computation result, it would be hard to expect cloud customers to turn over control of their workloads from local machines to cloud solely based on its economic savings and resource flexibility. For practical consideration, such a design should further ensure that customers perform less amount of operations following the mechanism than completing the computations by themselves directly. Otherwise, there is no point for customers to seek help from cloud. It has been widely used in various engineering disciplines that analyze and optimize real-world systems, such as packet routing, flow control, power management of data centers, etc. [6]. Because LP computations require a substantial amount of computational power and usually involve confidential data, we propose to explicitly decompose the LP computation outsourcing into public LP solvers running on the cloud and private LP parameters owned by the customer. The flexibility of such a decomposition allows us to explore higher-level abstraction of LP computations than the general circuit representation for the practical efficiency in fig 1:

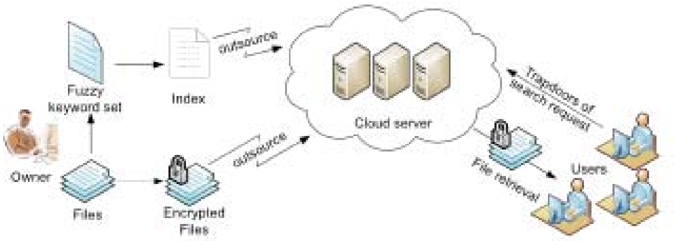

Fig. 1: Architecture of the Fuzzy Keywords Search

Specifically, we first formulate private data owned by the customer for LP problem as a set of matrices and vectors. This higher level representation allows us to apply a set of efficient privacy-preserving problem transformation techniques, including matrix multiplication and affine mapping, to transform the original LP problem into some arbitrary one while protecting the sensitive input/output information. One crucial benefit of this higher level problem transformation method is that existing algorithms and tools for LP solvers can be directly reused by the cloud server. Although the generic mechanism defined at circuit level, e.g. [7], can even allow the customer to hide the fact that the outsourced computation is LP, we believe imposing this more stringent security measure than necessary would greatly affect the efficiency. To validate the computation result, we utilize the fact that the result is from cloud server solving the transformed LP problem. In particular, we explore the fundamental duality theorem together with the piece-wise construction of auxiliary LP problem to derive a set of necessary and sufficient conditions that the correct result must satisfy. Such a method of result validation can be very efficient and incurs close-to-zero additional overhead on both customer and cloud server. With correctly verified result, customer can use the secret transformation to map back the desired solution for his original LP problem. We summarize our contributions as follows: 1) For the first time, we formalize the problem of securely outsourcing LP computations, and provide such a secure and practical mechanism design which fulfills input/output privacy, cheating resilience, and efficiency. 2) Our mechanism brings cloud customer great computation savings from secure LP outsourcing as it only incurs $O(n\rho)$ for some $2 < \rho \le 3$ local computation overhead on the customer, while solving a normal LP problem usually requires more than O(n3) time [8].3) The computations done by the cloud server shares the same time complexity of currently practical algorithms for solving the linear programming problems, which ensures that the use of cloud is economically viable. 4) The experiment evaluation further demonstrates the im-mediate practicality: our mechanism can always help customers achieve more than 30× savings when the sizes of the original LP problems are not too small, while introducing no substantial overhead on the cloud.

II. Related Work

Searchable encryption. Traditional searchable encryption has been widely studied in the context of cryptography. Among those works, most are focused on efficiency improvements and security definition formalizations. The first construction of searchable encryption was proposed by Song et al. [9], in which each word in the document is encrypted independently under a special two-layered encryption construction. Goh [10] proposed to use Bloom filters to construct the indexes for the data files. To achieve more efficient search, Chang et al and Curtmola et al. both proposed similar "index" approaches, where a single encrypted hash table index is built for the entire file collection. In the index table, each entry consists of the trapdoor of a keyword and an encrypted set of file identifiers whose corresponding data files contain the keyword. As a complementary approach, Boneh et al. [11] presented a public-key based searchable encryption scheme, with an analogous scenario Note that all these existing schemes support only exact keyword search, and thus are not suitable for Cloud Computing.

III. Problem Formulation

In this paper, we consider a cloud data system consisting of data owner, data user and cloud server. Given a collection of n encrypted data files C = (F1,F2,...,FN) stored in the cloud server, a predefined set of distinct keywords W = {w1,w2,...,wp}, the cloud server provides the search service for the authorized users over the encrypted data C. We assume the authorization between the data owner and users is appropriately done. An authorized user types in a request to selectively retrieve data files of his/her interest. The cloud server is responsible for mapping the searching request to a set of data files, where each file is indexed by a file ID and linked to a set of keywords. The fuzzy keyword search scheme returns the search results according to the following rules: 1) if the user's searching input exactly matches the pre-set keyword, the server is expected to return the files containing the keyword1; 2) if there exist typos and/or format inconsistencies in the searching input, the server will return the closest possible results based on pre-specified similarity semantics (to be formally defined in section III-D). An architecture of fuzzy keyword search is shown in the Fig. 1.

A. Threat Model

We consider a semi-trusted server. Even though data files are encrypted, the cloud server may try to derive other sensitive information from users' search requests while performing keyword-based search over C. Thus, the search should be conducted in a secure manner that allows data files to be securely retrieved while revealing as little information as possible to the cloud server. In this paper, when designing fuzzy keyword search scheme, we will follow the security definition deployed in the traditional searchable encryption. More specifically, it is required that nothing should be leaked from the remotely stored files and index beyond the outcome and the pattern of search queries.

B. Design Goals

In this paper, we address the problem of supporting efficient yet privacy-preserving fuzzy keyword search services over encrypted cloud data. Specifically, we have the following goals: i) to explore new mechanism for constructing storage- efficient fuzzy keyword sets; ii) to design efficient and effective fuzzy search scheme based on the constructed fuzzy keyword sets; iii) to validate the security of the proposed scheme.

C. Preliminaries

Edit Distance There are several methods to quantitatively measure the string similarity. In this paper, we resort to the well-studied edit distance for our purpose. The edit distance $ed(w1,w2)$ between two words w1 and w2 is the number of operations required to transform one of them into the other. The three primitive operations are 1) Substitution: changing one character to another in a word; 2) Deletion: deleting one character from a word; 3) Insertion: inserting a single character into a word. Given a keyword w, we let Sw,d denote the set of words w?satisfying $ed(w,w?) \leq d$ for a certain integer d.

D. Fuzzy Keyword Search

Using edit distance, the definition of fuzzy keyword search can be formulated as follows: Given a collection of n encrypted data files $C = (F1,F2,...,FN)$ stored in the cloud server, a set of distinct keywords $W = \{w1,w2,...,wp\}$ with predefined edit distance d, and a searching input (w,k) with edit distance k ($k \leq d$), the execution of fuzzy keyword search returns a set of file IDs whose corresponding data files possibly contain the word w, denoted as FID w: if $w = wi \in W$, then return FID wi; otherwise, if w $? \in$ W, then return {FIDwi}, where $ed(w,wi) \leq k$. Note that the above definition is based on the assumption that $k \leq d$. In fact, d can be different for distinct keywords and the system will return {FIDwi}

III. Linear Programming Approach

An optimization problem is usually formulated as a mathematical programming problem that seeks the values for a set of decision variables to minimize (or maximize) an objective function representing the cost subject to a set of constraints. For linear programming, the objective function is an affine function of the decision variables, and the constraints are a system of linear equations and inequalities. Since a constraint in the form of a linear inequality can be expressed as a linear equation by introducing a non-negative slack variable, and a free decision variable can be expressed as the difference of two non-negative auxiliary variables, any linear programming problem can be expressed in the following standard form, minimize

cTx subject to $Ax = b, x \geq 0$ (1)

Here x is an $n\times1$ vector of decision variables, A is an $m\times n$ matrix, and both c and b are $n\times1$ vectors. It can be assumed further that $m \leq n$ and that A has full row rank; otherwise, extras rows can always be eliminated from A. In this paper, we study a more general form as follows, minimize

cTx subject to $Ax = b, Bx \geq 0$ (2)

In Eq. (2), we replace the non-negative requirements in Eq. (1) by requiring that each component of Bx to be non-negative, where B is an $n \times n$ non-singular matrix, i.e. Eq. (2) de- generates to Eq. (1) when B is the identity matrix. Thus, the LP problem can be defined via the tuple $\Phi = (A,B,b,c)$ as input, and the solution x as output.

A. Enhanced Techniques via Affine Mapping

To enhance the security strength of LP outsourcing, we must be able to change the feasible region of original LP and at the same time hide output vector x during the problem input encryption. We propose to encrypt the feasible region of Φ by applying an affine mapping on the decision variables x. This design principle is based on the following observation: ideally, if we can arbitrarily transform the feasible area of problem Φ from one vector space to another and keep the mapping function as the secret key, there is no way for cloud server to learn the original feasible area information. Further, such a linear mapping also serves the important purpose of output hiding, as illustrated below.

Let M be an $n \times n$ non-singular matrix and r be an $n \times 1$ vector. The affine mapping defined by M and r transforms x into $y = M-1(x + r)$. Since this mapping is an one-to-one mapping, the LP problem Φ in Eq. (2) can be expressed as the following LP problem of the decision variables y.minimize $cTMy - cTr$ subject to $AMy = b + Ar$ $BMy \geq Br$.

Using the basic techniques, this LP problem can be further transformed to minimize $\gamma cTMy$ subject to $QAMy = Q(b + Ar)$, $BMy - \lambda QAMy \geq Br - \lambda Q(b + Ar)$.

One can denote the constraints of above LP via Eq. (3):

$A' = QAM$

$B' = (B - \lambda QA)M$

$b' = Q(b + Ar)$

$c' = \gamma MTc$ (3)

If the following conditions hold, $|B'| = 0$, $\lambda b' = Br$, and $b + Ar = 0$, (4)

then the LP problem $\Phi K = (A',B',b',c')$ can be formulated via Eq. (5), minimize $c'Ty$ subject to $A'y = b', B'y \geq 0$ (5)

B. Effective Key Word Search

The key idea behind our secure fuzzy keyword search is two-fold: 1) building up fuzzy keyword sets that incorporate not only the exact keywords but also the ones differing slightly due to minor typos, format inconsistencies, etc.; 2) designing an efficient and secure searching approach for file retrieval based on the resulted fuzzy keyword sets.

C. Advanced Technique for Constructing Fuzzy Keyword Sets

To provide more practical and effective fuzzy keyword search constructions with regard to both storage and search efficiency, we now propose an advanced technique to improve the straightforward approach for constructing the fuzzy keyword set. Without loss of generality, we will focus on the case of edit distance $d = 1$ to elaborate the proposed advanced technique. For larger values of d, the reasoning is similar. Note that the technique is carefully designed in such a way that while suppressing the fuzzy keyword set, it will not affect the search correctness. Wildcard-based Fuzzy Set Construction In the above straightforward approach, all the variants of the keywords have to be listed even if an operation is performed at the same position. Based on the above observation, we proposed to use a wildcard to denote edit operations at the same position.

The wildcard-based fuzzy set of wi with edit distance d is denoted as Swi,d={S, wi,0,S wi,1, ···, S? wi,d}, where S, wi,τ denotes the set of words w I with τ wildcards. Note each wildcard represents an edit operation on wi. For example, for the keyword CASTLE with the pre-set edit distance 1, its wildcard-based fuzzy keyword set can be constructed as SCASTLE,1 = {CASTLE, *CASTLE, *ASTLE, C*ASTLE, C*STLE, ···, CASTL*E, CASTL*, CASTLE*}. The total number of variants on CASTLE constructed in this way is only 13 + 1, instead of 13 × 26 + 1 as in the above exhaustive enumeration approach when the edit distance is set to be 1. Generally, for a given keyword wi with length, the size of Swi,1will be only 2+ 1 + 1, as compared to (2 + 1) × 26 + 1 obtained in the straightforward approach. The larger the pre-set edit distance, the more storage overhead can be reduced: with the same setting of the example in the straightforward approach, the proposed technique can help reduce the storage of the index from 30GB to approximately 40MB. In case the edit distance is set to be 2 and 3, the size of Swi,2 and Swi,3 will be C1.

IV. Security Analysis

we analyze the correctness and security of the proposed fuzzy keyword search scheme. At first, we show the correctness of the schemes in terms of two aspects, that is, completeness and soundness.

Theorem 1: The wildcard-based scheme satisfies both completeness and soundness. Specifically, upon receiving the request of w, all of the keywords {wi} will be returned if and only if ed(w,wi) ≤ k.

The proof of this Theorem can be reduced to the following Lemma:

Lemma 1: The intersection of the fuzzy sets Swi,d and Sw,kfor wiand w is not empty if and only if ed(w,wi) ≤ k. Proof: First, we show that Swi,d∩Sw,kis not empty when ed(w,wi) ≤ k. To prove this, it is enough to find an element in Swi,d∩Sw,k. Let w = a1a2···as and wi= b1b2···bt, where all these aiand bjare single characters. After ed(w,wi) edit operations, w can be changed to wi according to the definition of edit distance. Let w*= a* 1a*2···a* m, where a* i= aj or a*i=* if any operation is performed at this position. Since the editoperation is inverted, from wi, the same positions containing wildcard at w*will be performed. Because ed(w,wi) ≤ k, w*is included in both Swi,d and Sw,k, we get the result that Sw i,d ∩ Sw,k is not empty. Next, we prove that Swi,d∩Sw,kis empty if ed(w,wi) > k. The proof is given by reduction. Assume there exists an w* belonging to Swi,d∩Sw,k. We will show that ed(w,wi) ≤ k, which reaches a contradiction. First, from the assumption that w*∈ Swi,d∩ Sw,k, we can get the number of wildcard in w*, which is denoted by n*, is not greater than k. Next, we prove that ed(w,wi) ≤ n*. We will prove the inequality with induction method. First, we prove it holds when n*= 1. There are nine cases should be considered: If w*is derived from the operation of deletion from both wiand w, then, ed(wi,w) ≤ 1 because the other characters are the same except the character at the same position. If the operation is deletion from wi and substitution from w, we have ed(wi,w) ≤ 1 because they will be the same after at most one substitution from wi. The other cases can be analyzed in a similar way and are omitted. Now, assuming that it holds when n*= γ, we need to prove it also holds when n*= γ + 1. If ˆ w*= a* 1a* 2···a* n* Swi,d ∩ Sw,k, where a* i= aj or a* i= *. For a wildcard at position t, cancel the underlying operations and revert it to the original characters in wi and w at this position. Assume two new elements w*I and w*are derived from them respectively. Thenperform one operation at position t of w*I to make the character of wi at this position be the same with w, which is denoted by w i. After this operation, w* I will be changed to w*, which has only k wildcards. Therefore, we have ed(wi,w) ≤ γ from theassumption. We know that ed(wi,w) ≤ γ and ed(wi,wi) = 1, based on which we know that ed(wi,w) ≤ γ + 1. Thus,we can get ed(w,wi) ≤ n*. It renders the contradiction ed(w,wi) ≤ k because n*≤ k. Therefore, Swi,d∩ Sw,k is empty if ed(w,wi) > k.

Theorem 2: The fuzzy keyword search scheme is secure regarding the search privacy.

Proof: In the wildcard-based scheme, the computation of index and request of the same keyword is identical. Therefore, we only need to prove the index privacy by using reduction. Suppose the searchable encryption scheme fails to achieve the index privacy against the in distinguishability under the chosen keyword attack, which means there exists an algorithm A who can get the underlying information of keyword from the index. Then, we build an algorithm A that utilizes A to determine whether some function f(·) is a pseudo-random function such that f?(·) is equal to f(sk,·) or a random function. A has an access to an oracle Of (·)that takes as input secret value x and returns f?(x). Upon receiving any request of the index computation, A answers it with request to the oracle Of (·). After making these trapdoor queries, the adversary outputs two challenge keywords w* 0and w* 1with the same length and edit distance, which can be relaxed by adding some redundant trapdoors. A picks one random b * {0,1} and sends w*b to the challenger. Then, A is given a challenge value y, which is either computed from a pseudo-random function f(sk,·) or a random function. A sends y back to A, who answers with b * {0,1}. Suppose A guesses b correctly with non- negligible probability, which indicates that the value is not randomly computed. Then, A makes a decision that f?(·) is a pseudo-random function. As a result, based on the assumption of the indistinguishability of the pseudo-random function from some real random function, A at most guesses b correctly with approximate probability 1/2. Thus, the search privacy is obtained.

V. Performance Analysis

A. Theoretic Analysis

1. Customer Side Overhead

According to our mechanism, customer side computation overhead consists of key generation, problem encryption operation, and result verification, which corresponds to the three algorithms KeyGen, ProbEnc, and Result Dec, respectively. Because KeyGen and Result- Dec only require a set of random matrix generation as well as vector-vector and matrix-vector multiplication, the computation complexity of these two algorithms are upper bounded via O(n2). Thus, it is straight-forward that the most time consuming operations are the matrix-matrix multiplications in problem encryption algorithm Prob Enc. Since m ≤ n, the time complexity for the customer local computation is thus asymptotically the same as matrix-matrix multiplication, i.e., O(nρ) for some 2 < ρ ≤ 3. In our experiment, the matrix multiplication is implemented via standard cubic-time method, thus the overall computation overhead is O(n3). However, other more efficient matrix multiplication algorithms can also be adopted, such as the Strassen's algorithm with time complexity O(n2.81) [18] or the Coppersmith-Winograd algorithm in O(n2.376). In either case, the over all customer side efficiency can be further improved.

2. Server Side Overhead

For cloud server, its only computation overhead is to solve the encrypted LP problem ΦK as well as generating the result proof Γ, both of which correspond to the algorithm Proof Gen. If the encrypted LP problem ΦK belongs to normal case, cloud server just solves it with the dual optimal solution as the result proof Γ, which is usually readily available in the current LP solving algorithms and incurs no additional cost for cloud (see Section III-D). If the encrypted problem ΦK does not have an optimal solution, additional auxiliary LP problems can be solved to provide a proof. Because for general LP solvers, phase I method (solving the auxiliary LP) is always executed at first to determine the initial feasible solution, proving the auxiliary LP with optimal solutions also introduces little additional overhead. Thus, in all the cases, the computation complexity of the cloud server is asymptotically the same as to solve a normal LP problem, which usually requires more than O(n3) time.

Obviously, the customer will not spend more time to encrypt the problem and solve the problem in the cloud than to solve the problem on his own. Therefore, in theory, the proposed mechanism would allow the customer to outsource their LP problems to the cloud and gain great computation savings.

B. Experiment Results

We now assess the practical efficiency of the proposed secure and verifiable LP outsourcing scheme with experiments. We implement the proposed mechanism including both the customer and the cloud side processes in Matlab and utilize the MOSEK optimization through its Matlab interface to solve the original LP problem Φ and encrypted LP problem ΦK. Both customer and cloud server computations in our experiment are conducted on the same workstation with an Intel Core 2 Duo processor running at 1.86 GHz with 4 GB RAM. In this way, the practical efficiency of the proposed mechanism can be assessed without a real cloud environment. We also ignore the communication latency between the customers and the cloud for this application since the computation dominates the running time as evidenced by our experiments.

VI. Conclusion

we formalize the problem of securely outsourcing LP computations in cloud computing, and provide such a practical mechanism design which fulfills input/output privacy, cheating resilience, and efficiency. By explicitly decomposing LP computation outsourcing into public LP solvers and private data, our mechanism design is able to explore appropriate security/efficiency tradeoffs via higher level LP computation than the general circuit representation. We develop problem transformation techniques that enable customers to secretly transform the original LP into some arbitrary one while protecting sensitive input/output information. We also investigate duality theorem and derive a set of necessary and sufficient condition for result verification. Such a cheating resilience design can be bundled in the over all mechanism with close-to-zero additional overhead. Both security analysis and experiment results demonstrates the immediate practicality of the proposed mechanism.

The problem of supporting efficient yet privacy-preserving fuzzy search for achieving effective utilization of remotely stored encrypted data in Cloud Computing. We design an advanced technique (i.e., wildcard-based technique) to construct the storage-efficient keyword sets by exploiting a significant observation on the similarity metric of edit distance. Based on the constructed fuzzy keyword sets, an efficient keyword search scheme. Through rigorous security analysis, we show that our proposed solution is secure and privacy-preserving, while correctly realizing the goal of keyword search.

Refrences

[1] P. Mell, T. Grance,"Draft nist working definition of cloud computing", Referenced on Jan. 23rd, 2010, [Online] Available: http://www.csrc.nist.gov/groups/SNS/cloud computing/index.html, 2010.

[2] Cloud Security Alliance,"Security guidance for critical areas of focus in cloud computing", 2009, Online Available: http://www.cloudsecurityalliance.org.

[3] C. Gentry,"Computing arbitrary functions of encrypted data", Commun. ACM, Vol. 53, No. 3, pp. 97–105, 2010.

[4] Sun Microsystems, Inc.,"Building customer trust in cloud computing with transparent security", 2009, online Available: https://www.sun.com/offers/ details/sun transparency.xml.

[5] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, E. H. Spafford,"Secure outsourcing of scientific computations", Advances in Computers, Vol. 54, pp. 216–272, 2001.

[6] S. Hohenberger, A. Lysyanskaya,"How to securely outsource cryptographic computations", in Proc. of TCC, 2005, pp. 264–282.

[7] M. J. Atallah, J. Li,"Secure outsourcing of sequence comparisons", Int. J. Inf. Sec., Vol. 4, No. 4, pp. 277–287, 2005.

[8] D. Benjamin, M. J. Atallah,"Private and cheating-free outsourcing of algebraic computations", in Proc. of 6th Conf. on Privacy, Security, and Trust (PST), 2008, pp. 240–245.

[9] D. Boneh, B. Waters,"Conjunctive, subset, and range queries on encrypted data", in Proc. of TCC'07, 2007, pp. 535–554.

[10] F. Bao, R. Deng, X. Ding, Y. Yang,"Private query on encrypted data in multi-user settings", in Proc. of ISPEC'08, 2008.

[11] C. Li, J. Lu, Y. Lu,"Efficient merging and filtering algorithms for approximate string searches", in Proc. of ICDE'08, 2008.

[12] A. Behm, S. Ji, C. Li, J. Lu,"Space-constrained gram-based indexing for efficient approximate string search", in Proc. of ICDE'09.

[13] S. Ji, G. Li, C. Li, J. Feng,"Efficient interactive fuzzy keyword search", in Proc. of WWW'09, 2009.

[14] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, R. N. Wright,"Secure multiparty computation of approximations", Proc. Of ICALP'01.

[15] R. Ostrovsky,"Software protection and simulations on oblivious rams", Ph.D dissertation, Massachusetts Institute of Technology, 1992.

[16] V. Levenshtein,"Binary codes capable of correcting spurious insertions and deletions of ones", Problems of Information Transmission, Vol. 1, No. 1, pp. 8–17, 1965.

D. Madhuri received B.tech from Chirala engineering college in Information Technology in 2008. She is currently pursuing M.tech in computer science & engineering at St.Anns college of Engg. & technology which is affiliated under the JNTU, Kakinada.She has total of 1 year of experience in Teaching field. She is a Life time Member of ISTE .

K.Eswar received the B.tech from Bapatla engineering college in computer science & engineering in 2002. He received his M.tech from Andhra university in computer science & engineering in 2004. Currently, He is working as a Assoc.Prof, in St.Anns college of engineering & Technology which is affiliated under JNTU Kakinada which is permanent NBA accreated Institue. He is member of CSI .He has total of 10 years of experience. His research interest includes Data

Mining, Cloud omputing, Computer Architecture and .He is a Sr. member of ISTE, SIEEE.

Dr.P.Harini received B.E. degree in Electronics and Communications Engg. form University of Madras, Chennai, in1993, received M.Tech. degree in Remote Sensing from JNTU, Hyderabad, in 1997, received M.Tech. degree in Computer Science and Engineering from JNTU, Hyderabad,in 2003 and received Ph.D. in Computer Science and Engineering from JNTU, Anantapur, in 2011. She has 16 Years of Experience in which 1 year of Industrial, 1 year of Research & over 14 years of rich Teaching Experience in reputed Engineering Colleges & She is currently working as Professor & HOD in Computer Science & Engineering department in St.Ann's College of Engineering & Technology, Chirala. She Published 21 Research papers in various International Journals & Conferences. Guided many UG & PG students for projects & Life time Member of ISTE & CSI. Conducted successfully many Workshops, Seminars, conferences, FDPs and many National Level Technical Symposiums.