IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

Analysis of Provable Data Possession on Clustered Data

D Prasanth[1], N Sandeep Chaitanya[2], T Chaitanya Sai Kumar[3]

Dept of ECE, VNRVJIET, Telangana, India[1, 3]

Dept of CSE, VNRVJIET, Telangana, India[2]

Abstract: Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this paper, we address the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data. We present a cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, we also propose a fuzzy clustering system for analyzing the high dimensional Data bases in cloud Environments. This paper proposes novel effective fuzzy soft clustering systems with the combination of possibilistic c-means.

Keywords: Storage Security, Provable Data Possession, fuzzy clustering systems

1. INTRODUCTION

Cloud computing provides a scalability environment for growing amounts of data and processes that work on various applications and services by means of on-demand self service. One of the strength of cloud computing is that data are being centralized and outsourced in clouds. This kind of outsourced storage in clouds has become a new profit growth point by providing a comparably low-cost, scalable, location independent platform for managing clients' data.

The cloud storage service (CSS) relieves the burden for storage management and maintenance. However, if such an important service is vulnerable to attacks or failures, it would bring irretrievable losses to the clients since their data or archives are stored in an uncertain storage pool outside the enterprises. These security risks come from the following reasons: the cloud infrastructures are much more powerful and reliable than personal computing devices.

However, they are still facing all kinds of internal and external threats; for the benefits of their possession, there exist various motivations for cloud service providers (CSP) to behave unfaithfully towards the cloud users; furthermore, the dispute occasionally suffers from a lack of trust on CSP. Consequently, their behaviors may not be known by the cloud users, even if this dispute may result from the users' own improper operations. Therefore, it is necessary for cloud service providers to offer an efficient audit service to check the integrity and availability of the stored data [10]. Security audit is an important solution enabling tracking and analysis of any activities including data accesses, security breaches, application activities, and so on. Data security tracking is crucial for all organizations that must be able to comply with a range of federal laws including the Sarbanes-Oxley Act, Basel II, HIPAA and other regulations1. Furthermore, compared to the common audit, the audit service for cloud storages should provide clients with a more efficient proof of the integrity of stored data. Provable data possession (PDP) [2] (or proofs of retrievability (POR) [3]) is such a probabilistic proof technique for a storage provider to prove the integrity and ownership of clients' data without downloading data. The proof-checking without downloading makes it especially important for large-size files and folders (typically including many clients' files) to check whether these data have been tampered with or deleted without downloading the latest version of data. Thus, it is able to replace traditional hash and signature functions in storage outsourcing. Various PDP schemes have been recently proposed, such as Scalable PDP [4] and Dynamic PDP [5].

However, these schemes mainly focus on PDP issues at untrusted servers in a single cloud storage provider and are not suitable for a multi-cloud environment. To provide a low-cost, scalable, location independent platform for managing clients' data, current cloud storage systems adopt several new distributed file systems, for example, Apache Hadoop Distribution File System (HDFS), Google File System (GFS), Amazon S3 File System, Cloud Store etc. These file systems share some similar features: a single metadata server provides centralized management by a global namespace; files are split into blocks or chunks and stored on block servers; and the systems are comprised of interconnected clusters of block servers. Those features enable cloud service providers to store and process large amounts of data. However, it is crucial to offer an efficient verification on the integrity and

DOI10.17148/IJARCCE.2016.51095

availability of stored data for detecting faults and automatic recovery. Moreover, this verification is necessary to provide reliability by automatically maintaining multiple copies of data and automatically redeploying processing logic in the event of failures.

2. COOPERATIVE PDP

In order to prove the integrity of data stored in a multi-cloud environment, we define a framework for CPDP based on interactive proof system (IPS) and multi-prover zero-knowledge proof system (MPZKPS), as follows:

Definition 1 (Cooperative-PDP): A cooperative provable data possession $\mathcal{S}= (KeyGen,\ TagGen,\ Proof)$is a collection of two algorithms ($KeyGen$, $TagGen$) and an interactive proof system $Proof$, as follows:
$K(1\kappa)$: takes a security parameter κ as input, and returns a secret key sk or a public-secret key pair(pk, sk);

***$TagGen(sk, F,\mathcal{P})$*:** takes as inputs a secret key sk, a file F, and a set of cloud storage providers $\mathcal{P}= \{Pk\}$, and returns the triples (ζ,ψ, σ), where ζ is the secret in tags, ψ = (u,H) is a set of verification parameters u and an index hierarchy H for F, σ = $\{\sigma(k)\}Pk\in\mathcal{P}$denotes a set of all tags, $\sigma(k)$ is the tag of the fraction $F(k)$ of F in Pk;
($\mathcal{P}$, V): is a protocol of proof of data possession between CSPs ($\mathcal{P}= \{Pk\}$) and a verifier (V), that is,
$\langle\Sigma Pk\in(F(k), \sigma(k)) \longleftrightarrow V\rangle(pk, \psi)=$
1 $F= \{(k)\}$ is intact
0 $F= \{(k)\}$ is changed,
Where each Pk takes as input a file $F(k)$ and a set of tags $\sigma(k)$, and a public key pk and a set of public parameters ψ are the common input between P and V . At the end of the protocol run, V returns a bit {0|1} denoting false and true. Where, $\Sigma Pk\in\mathcal{P}$denotes cooperative computing in $Pk\in\ \mathcal{P}$. A trivial way to realize the CPDP is to check the data stored in each cloud one by one, i.e.,$\wedge Pk\in\mathcal{P}\ \langle(F(k),\ \sigma(k)) \longleftrightarrow V\rangle(pk,\ \ \psi)$, where $\wedge$denotes the logical AND operations among the Boolean outputs of all protocols $\langle Pk,\ V\rangle$for all $Pk\in\mathcal{P}$. However, it would cause significant communication and computation overheads for the verifier, as well as a loss of location-transparent. Such a primitive approach obviously diminishes the advantages of cloud storage: scaling arbitrarily up and down on demand [13]. To solve this problem, we extend above definition by adding an organizer(O), which is one of CSPs that directly contacts with the verifier, as follows:

$$\langle\Sigma Pk\in((k), (k)) \longleftrightarrow O\longleftrightarrow V\rangle(pk, \psi),$$

where the action of organizer is to initiate and organize the verification process. This definition is consistent with aforementioned architecture, e.g., a client (or an authorized application) is considered as , the CSPs are as $\mathcal{P}= \{Pi\}i\in[1,c]$, and the Zoho cloud is as the organizer in Figure 1. Often, the organizer is an independent server or a certain CSP in $\mathcal{P}$. The advantage of this new multi-prover proof system is that it does not make any difference for the clients between multi-prover verification process and single prover verification process in the way of collaboration. Also, this kind of transparent verification is able to conceal the details of data storage to reduce the burden on clients.

Cooperative PDP Scheme

In this section, we propose a CPDP scheme for multi cloud system based on the above-mentioned structure and techniques. This scheme is constructed on collision-resistant hash, bilinear map group, aggregation algorithm, and homomorphic responses.

2.1 Notations and Preliminaries

Let $\mathbb{H}$ = $\{Hk\}$ be a family of hash functions : $\{0, 1\}^{n}\longrightarrow$ {0, 1}*index by $k\in\mathcal{K}$. We say that algorithm $\mathcal{A}$has advantage ϵ in breaking collision resistance of $\mathbb{H}$ if $\Pr[\mathcal{A}(k) = (m0,m1) : m0 \neq m1,Hk(m0) = Hk(m1)] \geq \epsilon$, where the probability is over the random choices of $k\in\mathcal{K}$and the random bits of $\mathcal{A}$. So that, we have the following definition.

Definition 2 (Collision-Resistant Hash): A hash family $\mathbb{H}$ is (t, ϵ)-collision-resistant if no t-time adversary has advantage at least ϵ in breaking collision resistance of $\mathbb{H}$. We set up our system using bilinear pairings proposed by Boneh and Franklin [14]. Let $\mathbb{G}$ and $\mathbb{G}T$ be two multiplicative groups using elliptic curve conventions with a large prime order p. The function e is a computable bilinear map $e : \mathbb{G}\times\mathbb{G}\longrightarrow \mathbb{G}T$ with the following properties: for any $G,H\in\ \mathbb{G}$ and all a, $b\in\ \mathbb{Z}p$,we have 1) Bilinearity: $e([a]G, [b]H) = e(G,H)ab$; 2)Non-degeneracy: $e(G,H) \neq 1$ unless G or H = 1; and3) Computability: $e(G,H)$ is efficiently computable.

Definition 3 (Bilinear Map Group System): A bilinear map group system is a tuple $\mathbb{S} = \langle p,, , e\rangle$composed of the objects.

KeyGen(1κ): Let $\mathbb{S}= (p,\mathbb{G},\mathbb{G}T, e)$ be a bilinear map group system with randomly selected generators g, $h\in\mathbb{G}$, where $\mathbb{G}$,$\mathbb{G}T$are two bilinear groups of a large prime order p, $|p|= O(\kappa)$. Makes a hash function $(\cdot)$ public. For a CSP, chooses a random number $s\in R\mathbb{Z}p$and computes $S= gs\in\mathbb{G}$. Thus, $skp= s$and $pkp= (g,)$. For a user, chooses two random numbers α, $\beta\in R\mathbb{Z}p$and sets $sku= (\alpha,\ \beta)$ and $pku= (g,h,H1 = h\alpha,H2 = h\beta)$.

Tag Gen (sk,F, $\mathcal{P}$): Splits Finto $n\times s$sectors $\{mi,j\}i\in[1,n],j\in[1,s]\ \in\mathbb{Z}n\times sp$. Chooses $s$random $\tau1$, ⋯, $\tau s\in\mathbb{Z}p$as the secret of this file and computes $ui= g\tau i\in\mathbb{G}$for $i\in[1, s]$. Constructs the index table $\chi= \{\chi i\}=1$ and fills out the record χia in $\chi$for $i\in[1, n]$, then calculates the tag for each block mias
$^{(1)}\leftarrow H\Sigma^{s}{}_{i=1}\tau(F_{n})$,
$^{(2)}k\leftarrow_{(1)}(C_{k})$,
$^{(3)}i, \leftarrow H_{\xi_{k}}{}^{(2)}(\chi i)$,

DOI10.17148/IJARCCE.2016.51095

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

$\sigma i, \leftarrow(^{(3)}{}_{,k})^{\alpha}\cdot(\Pi^{s}j=1\ u_{\mathrm{j}}^{mi\cdot j})^{\beta}$,
where $Fn$is the file name and $Ck$is the CSP name of $Pk\in\mathcal{P}$. And then stores $\psi= (u, (1), \chi)$ into TTP, and$\sigma k= \{\sigma i,\}\forall j=k$to $Pk\in\mathcal{P}$, where $u= (u1, \cdot, us)$. Finally, the data owner saves the secret $\zeta= (\tau 1, \cdots, \tau s)$.

Proof($\mathcal{P}$,V): This is a 5-move protocol among the Provers ($\mathcal{P}= \{Pi\}i\in[1,c]$), an organizer ($O$), and a Verifier ($V$) with the common input (pk, ψ), which is stored in TTP, as follows:

1) Commitment($O\rightarrow V$): the organizer chooses a random $\gamma\in R\mathbb{Z}_{p}$and sends $H'1 = H\gamma 1$ to the verifier;

2) Challenge1($O\leftarrow V$): the verifier chooses a set of challenge index-coefficient pairs $Q= \{(i, vi)\}i\in I$and sends$Q$to the organizer, where $I$is a set of random indexes in [1, n] and $vi$is a random integer in $\mathbb{Z}*p$;

3) Challenge2($\mathcal{P}\leftarrow O$): the organizer forwards $Qk= \{(i, vi)\}mi\in Pk\subseteq Q$to each Pkin $\mathcal{P}$;

4) Response1($\mathcal{P}\rightarrow O$): Pkchooses a random $rk\in\mathbb{Z}p$and $s$random $\lambda j,k\in\mathbb{Z}p$for $j\in[1, s]$, and calculates a response$\sigma'k\leftarrow Srk\cdot\Pi(i,vi)\in Qk\sigma vii$, $\mu j,k\leftarrow\lambda j,k+\Sigma(i,vi)\in Qkvi\cdot mi,j$, $\pi j,k\leftarrow e(u\lambda j,kj,H2)$,where $\mu k= \{\mu j,k\}j\in[1,s]$ and $\pi k=\Pi sj=1\ \pi j,k$. Let $\eta k\leftarrow grk\in\mathbb{G}$, each Pksends $\theta k= (\pi k, \sigma'k, \mu k, \eta k)$ to the organizer;

5) Response2($O\rightarrow V$): After receiving all responses from $\{Pi\}i\in[1,c]$, the organizer aggregates $\{\theta k\}Pk\in\mathcal{P}$into a final response θas$\sigma'\leftarrow(\Pi Pk\in\mathcal{P}\sigma'k\cdot\eta-sk)\gamma$, $\mu'j\leftarrow\Sigma\ Pk\in\mathcal{P}\gamma\cdot\mu j,,\ \pi'\leftarrow(\Pi Pk\in\mathcal{P}\pi k)\gamma$.
Let $\mu'= \{\mu'j\}\in[1,s]$. The organizer sends $\theta= (\pi', \sigma', \mu')$ to the verifier.

Verification: Now the verifier can check whether the response was correctly formed by checking that$\pi'\cdot e(\sigma', h)$ $?=e(\Pi(i,vi)\in QH\xi(2)k(\chi i)vi,H'1)\ \cdot e(\Pi sj=1u\mu'jj,H2)$.
For $\chi i=$ "$Bi, Vi,$" , we can set $\chi i= (Bi= i, Vi= 1, Ri\in R\{0, 1\}*)$ at initial stage of CPDP scheme.
In our scheme, the manager first runs algorithm$KeyGen$ to obtain the public/private key pairs for CSPs and users. Then, the clients generate the tags of outsourced data by using $TagGen$. Anytime, the protocol $Proof$ is performed by a 5-move interactively proof protocol between a verifier and more than one CSP, in which CSPs need not to interact with each other during the verification process, but an organizer is used to organize and manage all CSPs. This protocol can be described as follows: 1) the organizer initiates the protocol and sends a commitment to the verifier; 2) the verifier returns a challenge set of random index-coefficient pairs Q to the organizer; 3) the organizer relays them into each Pi in $\mathcal{P}$according to the exact position of each data block; 4) each Pi returns its response of challenge to the organizer; and 5) the organizer synthesizes a final response from received responses and sends it to the verifier. The above process would guarantee that the verifier accesses files without knowing on which CSPs or in what geographical locations their files reside. In contrast to a single CSP environment, our scheme differs from the common PDP scheme in two aspects: 1) Tag aggregation algorithm: In stage of commitment, the organizer generates a random $\gamma\in R\mathbb{Z}p$ and returns its commitment H' 1 to the verifier. This assures that the verifier and CSPs do not obtain the value of γ. Therefore, our approach guarantees only the organizer can compute the final σ' by using γ and $\sigma'\ k$ received from CSPs. After σ' is computed, we need to transfer it to the organizer in stage of "Response1". In order to ensure the security of transmission of data tags, our scheme employs a new method, similar to the ElGamal encryption, to encrypt the combination of tags $\Pi\ (i,vi)\in Qk\sigma vii$, that is, for $sk = s\in\mathbb{Z}p$ and $pk = (g, S = gs) \in\mathbb{G}2$, the cipher of message m is $\mathcal{C} = (\mathcal{C}1 = gr,\ \mathcal{C}2 = m\cdot Sr)$ and its decryption is performed by $m = \mathcal{C}2.\mathcal{C}-s1$. Thus, we hold the equation

$$\sigma'=(\Pi P_{k}\in\mathcal{P}\sigma'k/\eta sk)^{\gamma}=(\Pi Pk\in\mathcal{P}Srk.\Pi(i,vi)\in Qk\sigma^{vi}{}_{i}/\eta^{s}{}_{k)}{}^{\gamma}$$
$$=(\Pi Pk\in\mathcal{P}\Pi(i,vi)\in Qk\sigma vi)^{\gamma}$$
$$=\Pi(i,vi)\in Q\sigma^{vi\cdot\gamma}{}_{i}.$$

2) Homomorphic responses: Because of the homomorphic property, the responses computed from CSPs in a multi-cloud can be combined into a single final response as follows: given a set of $\theta k = (\pi k, \sigma'k, \mu k, \eta k)$received from Pk, let $\lambda j = \Sigma Pk\in\mathcal{P}\lambda j,$, the organizer can compute

$$\mu'j=\Sigma Pk\in\mathcal{P}\gamma\cdot\mu j,\ =\Sigma Pk\in(\lambda j,k$$
$$+\Sigma(i,vi)\in Qkvi\cdot mi,j)$$
$$=\Sigma Pk\in\mathcal{P}\gamma\cdot\lambda j,\ +\ \gamma\cdot\Sigma Pk\in\mathcal{P}\Sigma(i,vi)\in Qkvi\cdot mi,j$$
$$=\ \gamma\cdot\Sigma Pk\in\mathcal{P}\lambda j,\ +\ \gamma\cdot\Sigma(i,vi)\in Q\ vi\cdot mi,j$$
$$=\ \gamma\cdot\lambda j+\ \gamma\cdot\Sigma(i,vi)\in Qvi\cdot mi,j.$$

The commitment of $\lambda j$is also computed by

$$\pi'=\ (\Pi Pk\in\mathcal{P}\pi k)=\ (\Pi Pk\in\mathcal{P}\Pi sj=1\pi j,)$$
$$=\Pi sj=1\Pi Pk\in(u\lambda j,kj,H2)\gamma$$
$$=\Pi sj=1(u\Sigma Pk\in\mathcal{P}\lambda j,,2)$$
$$=\Pi sj=1(u,'2).$$

It is obvious that the final response θ received by the verifiers from multiple CSPs is same as that in one simple CSP. This means that our CPDP scheme is able to provide a transparent verification for the verifiers. Two response algorithms, Response1 and Response2, comprise an HVR: Given two responses θi and θj for two challenges Qi and Qj from two CSPs, i.e., $\theta i = Response1(Qi, \{mk\}k\in Ii\ , \{\sigma k\}k\in Ii\ )$, there exists an efficient algorithm to combine them into a final response θ corresponding to the sum of the challenges$Qi\cup Qj$, that is, $\theta= Response1\ (\ Qi\cup Qj, \{mk\}k\in Ii\cup Ij, \{\sigma k\}k\in Ii\cup Ij) = Response2(\theta i,\ \theta j\ )$.For multiple CSPs, the above equation can be extended to $\theta= Response2(\{\theta k\}Pk\in\mathcal{P})$. More importantly, the HVR is a pair of values $\theta = (\pi, \sigma, \mu)$, which has a constant-size even for different challenges.

3. FUZZY CLUSTERING SYSTEMS

Hence the aim of this paper is to find suitable clustering techniques using fuzzy clustering analysis to find the subgroups of samples sharing similar expression patterns, Fuzzy clustering has been implemented effectively in analyzing the medical database for assisting physicians to have further treatment plan. Due to random selection of

DOI10.17148/IJARCCE.2016.51095

initial centers of fuzzy c-means the algorithm takes more number of iteration to converge the termination condition, and sometimes leads improper clustering results. Hence, in order to cluster effectively the objects have similar expression Patterns of users required data is retrieved by effective Kernel based Fuzzy clustering algorithms in the combination of both fuzzy membership function and typicality of possibilistic C-means.

The combination of possibilistic with fuzzy clustering has been successfully implemented to cluster the unlabeled data of real life problems by many researchers . Here the typicality values are constrained and the sum of the overall data points of typicalities to a cluster is one.

The proposed objective function is enhanced by introducing new kernel induced distance called hypertangent kernel Bray Curtis distance to evaluate the relations between cluster prototypes and data objects.

The kernel induced distance helps to have higher dimensional feature space from original patterns pace in order to obtain strong membership for a cluster. The new novel approach offers expected resulted subtypes of cancers from Lung compared with the previous algorithms.

In order to enhance the quality of the clustering results in clustering the subtypes of similar gene expression of Lung cancer database the following objective function of fuzzy c-means called tangent fuzzy possibilistic C-means is introduced by incorporating fuzziness weighting exponent, and the expression of possibilistic typical weighting exponent:

$$J(U,V) = 2\gamma \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu^2{}_{ik} + \tau^n{}_{ik})(1 - T_B(x_k, v_i))$$

Where $T_B(x_k, v_i) = 1 - \tanh(\frac{-Bx_k, v_i}{\sigma^2})$ and the distance has b

$$B(x_k, v_i) = \sum_{i=1}^{q} \frac{x_{kt} - v_{it}}{x_{kt} + v_{it}}$$

Here the parameters $m\&\eta$ are weighting exponents on each fuzzy membership and typicalities respectively. The parameters calculate the amount of fuzziness of the resulting classification and to obtain proper center by reducing the noise effect of undesirable effect of similar gene expression.γ is the resolution parameter.

The common ground of Tangent Kernel based Fuzzy possibilistic C-means is to first map the input data element into a feature space with higher dimension via an on linear transformation and then perform HKFPCM$_{bc}$ in that feature space. In equation (1), taking the derivative of objective function with respect to$\mu_{ik}\&\tau_{ik}$,, we have

$$\mu_{ik} = \frac{\left(\frac{1}{1 - T_B(x_k, v_i)}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c}\left(\frac{1}{1 - T_B(x_k, v_j)}\right)^{\frac{1}{m-1}}}$$

$$\tau_{ik} = \frac{\left(\frac{1}{(1 - T_B(x_k, v_i))}\right)^{\frac{1}{\eta-1}}}{\sum_{l=1}^{n}\left(\frac{1}{(1 - T_B(x_l, v_i))}\right)^{\frac{1}{\eta-1}}}$$

The general equation is used to obtain membership grades for objects in data for finding meaningful groups. The precision of clustering results mainly depends on the cluster centers. Now minimizing the following objective function, this paper obtains the equations for updating the prototypes of our TFPCM

$$v_i^t = \frac{\sum_{k=1}^{n}\left(\mu_{ik}^m + \tau_{ik}^\eta\right) T_B(x_k, v_i) T_B^{'}(x_k, v_i) \psi(x_k, v_i) x_k}{\sum_{k=1}^{n}\sum_{i=1}^{c}\left(\mu_{ik}^m + \tau_{ik}^\eta\right) T_B(x_k, v_i) T_B^{'}(x_k, v_i) \psi'(x_k, v_i)}$$

4. CONCLUSION

In this paper, we presented the construction of an efficient PDP scheme for distributed cloud storage. Based on homomorphic verifiable response and hash index hierarchy, we have proposed a cooperative PDP scheme to support dynamic scalability on multiple storage servers. We also showed that our scheme provided all security properties required by zero knowledge interactive proof system, so that it can resist various attacks even if it is deployed as a public audit service in clouds. Furthermore, we optimized the probabilistic query and periodic verification to improve the audit performance. Our experiments clearly demonstrated that our approaches only introduce a small amount of computation and communication overheads. Therefore, our solution can be treated as a new candidate for data integrity verification in outsourcing data storage systems. We also proposed Fuzzy clustering systems for analyzing the high dimensionality databases in cloud environments

REFERENCES

[1] P. Ammannand S.Jajodia,“Distributed Timestamp Generation inPlanarLatticeNetworks,”ACMTrans.ComputerSystems,vol.11,pp. 205-225,Aug. 1993.

[2] G. Ateniese, R. Burns, R. Curtmola,J.Herring,L. Kissner, Z. Peterson,and D.Song, “ProvableData PossessionatUntrusted Stores,” Proc.ACMConf.ComputerandComm. Security,pp. 598-609,2007.

DOI10.17148/IJARCCE.2016.51095

[3] E. BarkaandA.Lakas,“IntegratingUsageControl withSIP-Based Communications,” J. Computer Systems, Networks, and Comm., vol.2008, pp. 1-8,2008.

[4] D.Bonehand M.K.Franklin,“Identity-BasedEncryptionfromthe Weil Pairing,”Proc.Int’lCryptologyConf.AdvancesinCryptology, pp. 213-229,2001.

[5] R. Bose and J. Frew, “Lineage Retrieval forScientific Data Processing: A Survey,”ACMComputing Surveys,vol.37,pp. 1-28,Mar.2005.

[6] P.Buneman,A.Chapman,and J.Cheney,“ProvenanceManage-mentin CuratedDatabases,”Proc. ACM SIGMOD Int’l Conf. ManagementofData(SIGMOD’06),pp. 539-550,2006.

[7] B. Chun and A.C.Bavier,“Decentralized Trust Management and Accountability in Federated Systems,” Proc.Ann.HawaiiInt’lConf.SystemSciences(HICSS),2004.

[8] OASIS Security ServicesTechnical Committee,“Security Assertion Markup Language (saml) 2.0,”http://www.oasis-open.org/committees/tchome.php?wgabbrev=security,2012.

[9] Corin, S.Etalle, J.I.den Hartog,G.Lenzini,andI.Staicu, “ALogic for AuditingAccountabilityin Decentralized Systems,” Proc.IFIPTC1WG1.7WorkshopFormalAspectsinSecurityandTrust,pp. 187-201,2005.

[10] B.Crispoand G.Ruffo,“ReasoningaboutAccountabilitywithin Delegation,”Proc.ThirdInt’lConf. InformationandComm.Security (ICICS),pp. 251-260,2001.

[11] Y. Chen et al., “Oblivious Hashing:AStealthy Software IntegrityVerificationPrimitive,”Proc.Int’lWorkshopInformationHiding, F.Petitcolas,ed., pp. 400-414,2003.

[12] S. Etalleand W.H.Winsborough,“APosterioriCompliance Control,” SACMAT ’07:Proc. 12th ACM Symp.AccessControl ModelsandTechnologies,pp. 11-20,2007.

[13] X.Feng, Z.Ni, Z.Shao, and Y.Guo, “An OpenFrameworkforFoundationalProof-CarryingCode,”Proc. ACMSIGPLANInt’lWorkshopTypesinLanguagesDesignandImplementation,pp. 67-78, 2007.

[14] R. Hasan,R. Sion,and M.Winslett, “TheCaseoftheFakePicasso: Preventing History Forgery withSecure Provenance,” Proc. SeventhConf.FileandStorageTechnologies,pp. 1-14,2009.

[15] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. T. Foster, “Virtual infrastructure management in private and hybrid clouds,” IEEE Internet Computing, vol. 13, no. 5, pp. 14–22, 2009.

[16]N Sandeep Chaitanya & S Ramachandram “Raid Technology for Secured Grid Computing Environments” in IEEE NCC 2012 at IIT Karagpur Print ISBN: 978-1-4673-0815-1 INSPEC Accession Number: 12654144 Digital Object Identifier : 10.1109/NCC.2012.6176738 IEEE Catalog Number: CFP1242J-ART,

[17] N Sandeep Chaitanya & S Ramachandram “Data Privacy for Grid Systems” “Springer” Ist International Conference on Advances in Computing & Communications(ACC-11) with A. Abraham et al. (Eds.): ACC 2011, Part IV, CCIS 193, pp. 70–78, 2011. © Springer-Verlag Berlin Heidelberg 2011

[18] N Sandeep Chaitanya & S Ramachandram “Authentication & Key Establishment in Grid Computing Environments Using GDC” in (IJESAT) International Journal of Engineering Science & Advanced Technology Vol-03, Iss-01 jan 2013 page no: 39-43

[19] N Sandeep Chaitanya & S Ramachandram “ CBP based Bandwidth reduction in Secured Clouds” in International Conference(ICICSIT-15) organized by MGIT page no:203-208, ISSN 0973-4562 Vol. 10 No.81 (2015) © Research India Publications; http://www.ripublication.com/ijaer.htm ISSN 0973-4562 Vol. 10 No.81 .

[20] N Sandeep Chaitanya & S Ramachandram “Raid Technology for Secured Grid Computing Environments” in IEEE NCC 2012 at IIT Karagpur On page(s): 1 Conference Location : Kharagpur Print ISBN: 978-1-4673-0815-1 INSPEC Accession Number: 12654144 Digital Object Identifier : 10.1109/NCC.2012.6176738 IEEE Catalog Number: CFP1242J-ART, Date of Current Version :03 April 2012 Issue Date : 3-5 Feb. 2012

[21] N Sandeep Chaitanya & S Ramachandram “Authentication, Key Establishment & Cooperative Cache maintenance in Wireless P2P Environments Using GDC & Greedy Algorithm” in (IJESIT) International Journal of Engineering Science & Innovative Technology **(Volume 2 Issue 1 ,January 2013)** ISSN No: 2319-5967 (ISO 9001:2008 Certified)page no: 499-508

[22] N Sandeep Chaitanya & S Ramachandram “Secure Communication Using GDC in Cloud Computing” in International Research Journal of Computer Science Engineering & Applications(IJRCSEA) Volume 1, Issue 3,Dec 2012 page no 176-179 in Dec 2012 http://irjcsea.org/vol1issue3.html

[23] N Sandeep Chaitanya & S Ramachandram “Data Caching in wireless p2p networks Using Greedy Algorithm”in **N**ational Conference on Recent Advances on Soft Computing and Knowledge Discovery (SCKD2k12) at SreeVenkateshwara University, **Tirupati**, during January 19-21, 2012 and the same is published in International Journal “**Asia Pacific Journal of Computer Sciences (APJCS)” Volume 1, Issue 1 in** Jan - June 2012 edition.

[24] N Sandeep Chaitanya & S Ramachandram “Preemptive Routing & Intrusion detection in MANET's” in International Journal of Computer & Communication Technology(IJCCT) ISSN(ONLINE):2231-0371 Volume 3, Issue 1, 2012 page no 87-92 in Jan 2012

[25] N Sandeep Chaitanya & S Ramachandram **“**Architecture and Algorithm for an Cooperative Cache wireless p2p Networks” in IJECCE, **Vol.3,Issue(1) ,Page No.:144-151,** ISSN 2249 –071X **in Jan 2012**

DOI10.17148/IJARCCE.2016.51095