RESEARCH ARTICLE OPEN ACCESS

Online City Scale Taxi Ridesharing

Mrs.C.Navamani MCA[1]., M.Phil.,M.E., M.Srimathi[2],
Assistant Professor[1], Research Scholar[2],
Department of Computer Applications,
Nandha Engineering College, Erode-52.

----------**********----------

Abstract:

The current system is design and developed a taxi-sharing system that accepts taxi passengers' real-time ride requests through smartphones and schedules proper taxis to pick up them via ridesharing, subject to time, capacity, and monetary constraints. The monetary constraints provide incentives for both passengers and taxi drivers: passengers will not pay more compared with no ridesharing and get compensated if their travel time is lengthened due to ridesharing; taxi drivers will make money for all the detour distance due to ridesharing. While such a system is of significant social and environmental benefit, e.g., saving energy consumption and satisfying people's commute, real-time taxi-sharing has not been well studied yet. To this end, we devise a mobile-cloud architecture based taxi-sharing system. Taxi riders and taxi drivers use the taxi-sharing service provided by the system via a smart phone App. The Cloud first finds candidate taxis quickly for a taxi ride request using a taxi searching algorithm supported by a spatio-temporal index. A scheduling process is then performed in the cloud to select a taxi that satisfies the request with minimum increase in travel distance. We built an experimental platform using the GPS trajectories generated by over 33,000 taxis over a period of three months. A ride request generator is developed (available at http://cs.uic.edu/_sma/ridesharing) in terms of the stochastic process modelling real ride requests learned from the data set. Tested on this platform with extensive experiments, our proposed system demonstrated its efficiency, effectiveness and scalability. For example, when the ratio of the number of ride requests to the number of taxis is 6, our proposed system serves three times as many taxi riders as that when no ridesharing is performed while saving 11 percent in total travel distance and 7 percent taxi fare per rider.

Keywords—Storage Refuge, Provable Data Possession, Interactive Protocol, Zero-knowledge, Multiple Cloud, Co-operative.

----------**********----------

1. INTRODUCTION

In recent years, cloud storage service has become a faster profit enlargement point by as long as a comparably low-cost, scalable, position-independent platform for clients' data. as cloud computing environment is construct based on open architectures plus interfaces, it has the capability to mix multiple internal and/or external cloud military jointly to give high interoperability. We call such a circulated cloud surroundings as a multi-Cloud (or hybrid cloud). frequently, by using virtual infrastructure management (VIM), a multi-cloud allows clients to easily access his/her resources distantly through interfaces such as Web services provide by Amazon EC2.

present live a variety of tools plus technologies for multi-cloud, such as Platform VM Orchestrator, VMware vSphere, plus Overt. These tools help cloud providers construct a dispersed cloud storage platform (DCSP) for organization clients' data. though, if such an significant platform is susceptible to refuge attacks, it would bring irreparable losses to the clients

Provable data possession (PDP) or (proofs of retrievability (POR) is such a probabilistic proof

technique for a storage supplier to prove the integrity and possession of clients' data with no downloading data. so, it is able to replace customary hash and signature functions in storage outsourcing. Various PDP systems have been recently proposed, such as Scalable PDP and lively PDP. However, these systems mainly focus on PDP issues at untrusted servers in a *single* cloud storage space provider and are not appropriate for a multi-cloud environment.

Motivation: To provide a low-cost, scalable, location independent stage for managing clients' data, current cloud storage systems take on several new dispersed file systems, for instance, Apache Hadoop Distribution File System (HDFS), Google File System (GFS), Amazon S3 File System, CloudStore etc. These file systems share some similar skin: a single metadata server provides central management by a global namespace; files are tear into blocks or chunk and stored on block servers; plus the systems are comprise of unified clusters of block servers. persons features enable cloud service provider to store and procedure large amount of data.

In précis, a verification system for data honesty in dispersed storage environment should have the following features:

Usability aspect: A customer should use the honesty check in the way of collaboration services. The system should hide the details of the storage to decrease the burden on clients;

Refuge aspect: The system should give adequate refuge skin to resist some obtainable attacks, such as data leak attack and tag fake attack;

Performance aspect: The system should have the lower message and calculation overheads than non-cooperative solution.

Related Works: To check the ease of use and integrity of outsourced data in cloud storages, researchers contain future two basic approach called Provable Data Possession (PDP) [2] and Proofs of Retrievability (POR) [3]. Attendees et al. [2] first proposed the PDP model for ensuring possession of files on untrusted storages and provide an RSA-based system for a still case that achieve the (1) communication cost. They also proposed a publicly provable version, which allows anybody, not just the proprietor, to confront the server for data control.

This possessions greatly comprehensive request areas of PDP protocol due to the division of data owner and the users. though, these systems are unsure of yourself against replay attack in lively scenarios since of the dependencies on the index of blocks. furthermore, they do not fit for multi-cloud storage space space due to the loss of homomorphism property in the confirmation process.

Our Contributions:

In this scheme, we address the difficulty of provable data control in dispersed cloud environment from the following aspect: *high refuge*, *transparent verification*, and *high performance*. To attain these goals, we first suggest a corroboration framework for multi-cloud storage along with two basic techniques: hash index hierarchy (HIH) and homomorphism provable response (HVR).

2. STRUCTURE AND TECHNIQUES

In this section, we existent our verification framework for multi-cloud storage and a formal definition of CPDP. We bring in two fundamental techniques for construct our CPDP system: hash index hierarchy (HIH) on which the response of the clients' challenge computed from multiple CSPs can be joint into a single reply as the final result; plus homomorphism demonstrable response (HVR) which ropes circulated cloud storage space in a multi-cloud storage and equipment an efficient construction of crash resistant hash function, which can be view as a chance oracle model in the confirmation protocol.

A. Verification Framework for Multi-Cloud

though existing PDP systems offer a publicly easy to get to remote interface for checking plus organization the great amount of data, the majority of existing PDP systems are unable to please the inherent supplies from multiple clouds in terms of communication and calculation costs. To address this difficulty, we think a multi-cloud storage service as illustrate in Figure 1. In this structural design, a data storage space service involve three

different entities: customers who have a large amount of data to be stored in multiple clouds plus have the permissions to access and influence stored data; Cloud Service Providers (CSPs) who work together to give data storage services plus have enough storages and calculation resources; and Trusted Third Party (TTP) who is trust to store verification parameter and offer group of people query armed forces for these parameter.

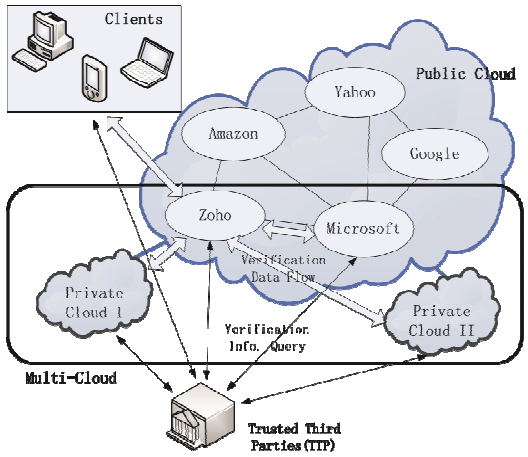

Fig. I. Verification architecture for data integrity.

In this structural design, we consider the survival of multiple CSPs to supportively store and uphold the clients' data. furthermore, a cooperative PDP is used to verify the honesty and ease of use of their stored data in all CSPs. The confirmation process is describe as follows: initially, a client (data owner) uses the clandestine key to pre-process a file which consists of a compilation of $n$blocks, generate a set of public confirmation information that is stored in TTP, transmit the file and some confirmation tags to CSPs, plus may delete its local copy; after that, by using a verification protocol, the clients can issue a confront for one CSP to check the honesty and ease of use of outsourced data with admiration to public in order stored in TTP.

B. Definition of Cooperative PDP

In arrange to prove the honesty of data store in a multi-cloud surroundings, we define a structure for CPDP based on interactive proof system (IPS) plus multi-prover zero-knowledge evidence system (MPZKPS), as follows:

Definition 1 (Cooperative-PDP): A helpful provable data possessions a collection of two algorithms and an interactive evidence system as follows:

1**:**takes a refuge parameter κas input, and income a secret key or a public-secret key pair ;

2:takes as inputs a secret key sk, a file F, and a set of cloud storage provider and income the triples, anywhere is the secret in tags, is a set of confirmation parameter $t$and an index pecking order denotes a set of all tags, is the tag of the part of in a procedure of proof of data control between CSPs and verifier (V).A unimportant way to realize the CPDP is to check the data store in each cloud one by one.

C. Hash Index Hierarchy for CPDP

To hold up circulated cloud storage, we exemplify a resistantative architecture second-hand in our cooperative PDP system. Our structural design has a pecking order structure which resembles a natural resistantation of file storage. They are describe as follows:

1) *Express Layer*: offers an abstract resistantation of the store resources;
2) *Service Layer*: offers and manages cloud storage services; and
3) *Storage Layer*: realizes data storage on many physical devices.

In storage space layer, we describe a common fragment structure that provide probabilistic confirmation of data honesty for outsourced storage. every CSP wreckage and stores the assign data into the storage space servers in Storage Layer. This structural design also provides special function for data storage plus management.

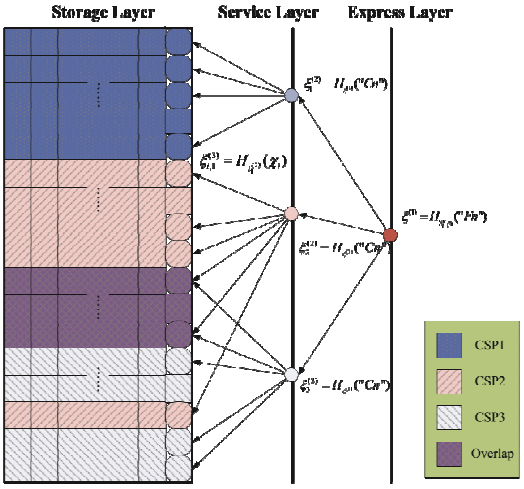

Fig.2.Index-hash hierarchy of CPDP model.

known a collision-resistant hash purpose (·), we make use of this structural design to build a Hash Index pecking order $\mathcal{H}$(viewed as a random oracle), which is used to put back the ordinary hash function in prior PDP systems.

The index-hash table consists of serial number, block number, version number, random integer, plus so on. dissimilar from the common index table, we promise that all records in our directory table differ from one one more to prevent fake of data block plus tag.

a. *Homomorphic Provable Response for CPDP*

Homomorphic provable reply is the key method of CPDP because it not only reduce the communiqué bandwidth, but also conceal the site of outsourced data in the dispersed cloud storage space surroundings.

When demonstrable data control is considered as a challenge-response procedure, we make bigger this note to the concept of Homomorphic Provable Responses (HVR), which is used to put together manifold responses as of the dissimilar CSPs in CPDP system as follows:

Definition 2 (Homomorphic Provable Response): A react is called homomorphic provable response in a PDP protocol, if given two response $\theta_i$and $\theta_j$for two challenge $Q_i$and $Q_j$from two CSPs, there exists an capable algorithm to unite them into a response θcorresponding to the sum of the challenge. Homomorphic demonstrable reply is the key technique of CPDP since it not only reduce the message bandwidth, but also conceal the location of outsourced data in the dispersed cloud storage surroundings.

D. COOPERATIVE PDP SYSTEM

In this section, we suggest a CPDP system for multicolor system base on the above-mentioned structure and technique. This system is construct on collision-resistant hash, bilinear map group, aggregation algorithm, and homomorphic response.

a. *Notations and Preliminaries*

Let $\mathbb{H}$= {H_k} be a family of hash functions: {0, 1} $^n\longrightarrow$ {0, 1} *index by $k\in\mathcal{K}$. So that, we have the following description.

Definition 3 (Collision-Resistant Hash): A hash family $\mathbb{H}$is (t,)-collision-resistant if no t-time opponent has benefit at least ϵin contravention collisionresistance of $\mathbb{H}$..

Definition 4 (Bilinear Map Group System): A bilinear map group system is a tuple $\mathbb{S}$= $\langle p,,,\rangle$calm of the objects as describe above.

b. *Our CPDP System*

In our system the boss first runs algorithm $KeyGen$to get the public/private key pairs for CSPs and users. after that, the clients make the tags of outsourced data by using $TagGen$. Anytime, the procedure $Proof$is perform by a 5-move interactive.

This procedure can be described as follows: 1) the manager initiate the protocol plus sends a promise to the verifier; 2) the verifier returns a challenge set of random index-coefficient pairs $Q$to the organizer; 3) the manager relays them into each $P_i$in $\mathcal{P}$according to the exact place of each data block; 4) each $P_i$returns its reply of challenge to the organizer; and 5) the manager synthesizes a final response as of received response and sends it to the verifier.

In difference to a single CSP environment, our system differ from the common PDP system in two aspects:

1) Tag aggregation algorithm: In stage of commitment, the manager generates a chance

$\gamma \in_R \mathbb{Z}_p$ and income its commitment H_1' to the verifier. This assure that the verifier and CSPs do not get the value of γ. so, our approach guarantee only the organizer can calculate the final σ' by using γ and σ_k' conventional from CSPs.

2) Homomorphic responses: since of the homomorphic property, the responses compute from CSPs in a multi-cloud can be joint into a single final response as follows: given a set of $\theta_k = (\pi_k, \sigma_k', \mu_k, \eta_k)$

4 REFUGE ANALYSIS

We give a short refuge analysis of our CPDP construction. This building is straight derived from multi-prover zero-knowledge proof system (MPZKPS), which satisfy following property for a given assertion L:

1) *Completeness*: when $x \in L$, there exists a strategy for the provers that induce the verifier that this is the case;

2) *Soundness*: when $x \notin L$, whatever plan the proves employ, they will not induce the verifier that $x \in L$;

3) *Zero-knowledge*: no cheating verifier can learn whatever thing other than the reality of the statement.

A. Collision resistant for index-hash hierarchy

In our CPDP system, the crash resistant of index hash hierarchy is the basis and precondition for the safe haven of entire system, which is describe as being safe in the *random oracle model*. A winning hash crash can still be used to produce a forged tag while the same hash value is reused multiple times, e.g., a lawful client modify the data or repeat to insert plus delete data blocks of outsourced data.

Theorem 1 (Collision Resistant): The index-hash pecking order in CPDP system is collision unwilling, even if the client generates $\sqrt{2p \cdot \ln \frac{1}{1-\varepsilon}}$ documents by the same file first name and cloud name, plus the client repeats $\sqrt{2^{L+1} \cdot \ln \frac{1}{1-\varepsilon}}$ times to modify, insert and erase data blocks, where the collision likelihood is at least ε, $\tau_i \in \mathbb{Z}_p$, and $|R_i| = L$ for $R_i \in \chi_i$.

B. Completeness property of verification

In our system, the wholeness property implies public verifiability possessions, which allows anybody, not just the customer (data owner), to challenge the cloud wine head waiter for *data integrity* and *data ownership* without the need for any secret in order. First, for every available data-tag pair $(F,) \in Ta(sk, F)$ and a chance challenge $Q = (i, v_i)_{i \in I}$, the confirmation protocol should be completed with victory likelihood according to the Equation,

$$\left[\left\langle \sum_{P_k \in \mathcal{P}} P_k(F^{(k)}, \sigma^{(k)}) \leftrightarrow O \leftrightarrow V \right\rangle (pk, \psi) = 1\right] = 1$$

In this procedure, anyone can get the owner's public key $pk = (g, h, H_1 = h^{\alpha}, H_2 = h^{\beta})$ and the corresponding file limit $\psi = (u, \xi^{(1)}, \chi)$ from TTP to execute the confirmation protocol, hence this is a public provable protocol. furthermore, for different owner, the secrets α and β hidden in their public key pk are also different, formative that a success confirmation can only be implement by the real owner's public key.

C. Zero-knowledge property of verification

CPDP building is in essence a Multi-Prover Zero-knowledge Proof (MP-ZKP) system [11], which can be careful as an extension of the idea of an interactive proof system (IPS). about speaking, in the situation of MP-ZKP, a polynomial-time bounded verifier interact with several provers whose computational power are limitless. According to a *Simulator* replica, in which every dishonest verifier has a simulator that can create a transcript that "looks like" an communication between a honest prover and a dishonest verifier, we can prove our CPDP building has Zero-knowledge property.

Theorem 2 (Zero-Knowledge Property): The confirmation protocol $Proof(\mathcal{P}, V)$ in CPDP system is a computational zero-knowledge scheme under a simulator model, that is, for every probabilistic polynomial-time interactive mechanism V^*, there exists a probabilistic polynomial-time algorithm $\mathcal{S}^*$ such that the ensembles $View(\langle_{Pk \in \mathcal{P}} P_k(F^{(k)}, \sigma^{(k)}) \leftrightarrow O \leftrightarrow V^* \rangle (pk, \psi))$ and $\mathcal{S}^*(pk, \psi \sum)$ are computationally identical.

Theorem 3 (Knowledge Soundness Property): Our system has (t,ϵ') knowledge soundness in chance oracle and rewind able knowledge extractor model presumptuous the computational Diffie-Hellman (CDH) supposition holds in the group $\mathbb{G}$for $\epsilon' \geq \epsilon$.

5 PERFORMANCE EVALUATION

In this section, to detect irregularity in a low overhead and timely manner, we analyze and optimize the appearance of CPDP system based on the above system as of two aspects: assessment of probabilistic queries and optimization of length of blocks. To authenticate the effects of system, we introduce a prototype of CPDP-based review system and existent the new results.

A. *Performance Analysis for CPDP System*

We use $[E]$ to indicate the calculation cost of an exponent process in $\mathbb{G}$, namely, g^x, anywhere $x$is a positive integer in $\mathbb{Z}_p$and $g \in \mathbb{G}$or $\mathbb{G}_T$. We neglect the calculation cost of algebraic operations. The most multifaceted operation is the calculation of a bilinear map $(\cdot,\cdot)$ between two elliptic point.

TABLE 1
Comparison of computation overheads between our CPDP system and non-cooperative (trivial) system.

	CPDP System	Trivial System
KeyGen	$3[E]$	$2[E]$
TagGen	$(2n+s)[E]$	$(2n+s)[E]$
Proof($\mathcal{P}$)	$c[E] + (t+cs+1)[E]$	$c[E] + (t+cs-c)[E]$
Proof(V)	$3[E] + (t+s)[E]$	$3c[E] + (t+cs)[E]$

we examine the storage and communication costs of our scheme. We describe the bilinear combination take the form $e : E(\mathbb{F}_{p^m}) \times E(\mathbb{F}_{p^{km}}) \rightarrow \mathbb{F}^*_{p^{km}}$, anywhere $p$is a prime, $m$is a positive figure, and $k$is the embed degree (or refuge multiplier). In this case, we use an asymmetric combination : $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$to put back the symmetric combination in the original systems.

Further, our system has better presentation compare with non-cooperative move toward due to the total of calculation expenses decrease $3(c-1)$ times bilinear map operation, where $c$is the figure of clouds in a multicolor. The cause is that, before the response are sent to the verifier as of $c$clouds, the organizer has aggregate these response into a reply by using aggregation algorithm, so the verifier only need to verify this reply once to obtain the final result.

TABLE 2
Comparison of communication overheads between our CPDP and non-cooperative (trivial) system.

	CPDP System	Trivial System
Commitment	l_2	cl_2
Challenge1	$2tl_0$	$2tl_0$
Challenge2	$\frac{2tl_0}{c}$	
Response1	$0 + 2\ 1 + _T$	$(sl_0 + l_1 + l_T)c$
Response2	$sl0 + l1 + l_T$	

with no loss of generality, let the safe haven parameter κbe 80 bits, we need the elliptic curve domain parameter over $\mathbb{F}_p$with $|p| = 160$ bits and $m = 1$ in our experiment. This means that the length of integer is $l_0 = 2\kappa$in $\mathbb{Z}_p$. likewise, we have $l_1 = 4\kappa$in $\mathbb{G}_1$, $l_2 = 24\kappa$in $\mathbb{G}_2$, and $l_T = 24\kappa$in $\mathbb{G}_\mathbb{T}$for the embedding degree $k = 6$. The storage space overhead of a file with $(f) = 1M$-bytes is $stor(f) = n \cdot s \cdot l_0 + n \cdot l_1 = 1.04M$-bytes for $n = 10^3$ and $s = 50$. The storage in the clouds of its index table $\chi$is $n \cdot l_0 = 20K$-bytes.

B. *Probabilistic Verification*

We recall the probabilistic corroboration of common PDP system (which only involves one CSP), in which the verification process achieves the detection of CSP server disobedience in a possibility example mode in arrange to decrease the workload on the server. take for granted the CSP

modify $e$blocks out of the n-block file, that is, the likelihood of disrupt blocks is$\rho_b = \frac{e}{n}$. Let $t$be the number of query blocks for a confront in the verification protocol.

one more advantage of probabilistic confirmation based on random example is that it is

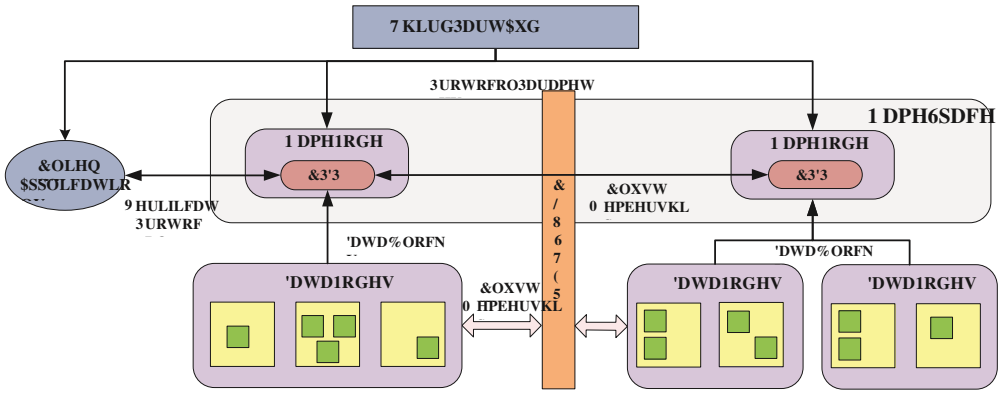

Fig. 3. Applying CPDP system in Hadoop circulated file system (HDFS).

easy to identify the tamper or forging data blocks or tags. The identification purpose is clear: when the verification fails, we can choose the partial set of confront indexes as a new confront set, and continue to execute theverificationprotocol. The Above search process can be frequently executed awaiting the bad block is establish. The difficulty of such a search procedure is (log$n$).

C. Parameter Optimization

In the fragment structure, the number of sector per block $s$is an significant parameter to affect the presentation of storage armed armed forces and audit services. therefore, we suggest an optimization algorithm for the worth of s in this section. Our consequences show that the best value can not only reduce the calculation and message overheads, but also reduce the size of extra storage space, which is necessary to store the confirmation tags in CSPs.

We decide the optimal value of $s$on the foundation of sensible settings and system demand. For NTFS format, we suggest that the value of $s$is 200 plus the size of block is 4KBytes, which is the same as the non-payment size of cluster at what occasion the file size is less than 16TB in NTFS. In this case, the value of $s$ensures that the additional storage space doesn't go beyond 1% in storage servers.To validate the efficiency plus efficiency of our future approach for audit armed armed forces, we have implement a example of an review system. The elliptic curve utilize in the trial is a MNT arc, with based field dimension of 160 bits and the embed degree 6.

A. CPDP for Integrity Audit Services

base on our CPDP system, we bring in an audit system architecture for outsourced data in manifold clouds by replace the TTP with a third party auditor (TPA). In this structural design, this structural design can be construct into a dream infrastructure of cloud-based storage space service [1]. We show an instance of applying our CPDP system in Hadoop dispersed file system (HDFS), which a dispersed, scalable, and moveable file system. HDFS' structural design is calm of Name Node plus Data Node, anywhere Name Node maps a file name to a set of index of block plus Data Node indeed stores data blocks. To hold up our CPDP system, the index-hash pecking order plus the metadata of Name Node should be included jointly to provide an enquiry service for the hash value or index-hash evidence.

initially, we count the presentation of our audit system under different parameter, such as file size example ratio w, sector figure per block s, and so on. Our analysis show that the value of be supposed to produce with the add to of order to reduce computation and message costs. Thus, our experiment were approved out as follow: the store records were chosen from 10KB to 10MB; the

sector information were distorted from 20 to 250 in terms of file sizes; and the sampling ratios were changed from 10% to 50%. These results dictate that the calculation and message costs (including I/O costs) grow with the add to of file size plus example ratio.

6 CONCLUSIONS

In this paper, we exsistanted the building of an well-organized PDP system for circulated cloud storage.base on homomorphic demonstrable reply and hash As part of prospect work, we would make bigger our work to travel around more effectual CPDP construction. First, as of our experiments we establish that the performance of CPDP system, especially for big documents, is affected by the bilinear mapping operations due to its difficulty. To solve this problem, RSA based construction may be a improved choice, but this is still a challenging task since the obtainable RSA base system have too many limits on the performance and refuge [2]. Next, as of a practical point of view, we still require addressing some issues about integrate our CPDP system smoothly with existing systems, for instance, how to match index hash pecking order with HDFS's two-layer name space, how to competition index arrangement with cluster-network model, and how to animatedly update the CPDP parameter according to HDFS' specific requirements. Finally, it is still a challenging problem for the age group of tags with the length immaterial to the size of information blocks. We would travel around such a subject to provide the support of variable-length block verification.

ACKNOWLEDGMENTS

The work of Y. Zhu and M. Yu was supported by the nationwide Natural Science Foundation of China (Project No.61170264 and No.10990011). This work of Gail-J.Ahn and Hongxin Hu was partially support by the grants from US National Science Foundation (NSFIIS-0900970 and NSF-CNS-0831360) and Department of Energy (DE-SC0004308).

REFERENCES

[1] *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.*

[2] *G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. X. Song, "Provable data possession at untrusted stores," in ACM Conference on Computer and Communications Refuge, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 598–609.k*

[3] *A. Juels and B. S. K. Jr., "Pors: proofs of retrievability for large documents," in ACM Conference on Computer and Communications Refuge, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 584–597.*

[4] *G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th international conference on Refuge and privacy in communication netowrks, SecureComm, 2008, pp. 1–10.*

[5] *C. C. Erway, A. Ku¨pc¸u¨, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in ACM Conference on Computer and Communications Refuge, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 213–222.*

[6] *H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT, ser. Lecture Notes in Computer Science, J. Pieprzyk, Ed., vol. 5350. Springer, 2008, pp. 90–107.*

[7] *Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage refuge in cloud computing," in ESORICS, ser. Lecture Notes in Computer Science, M. Backes and P. Ning, Eds., vol. 5789. Springer, 2009, pp. 355–370.*

[8] *Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in SAC, W. C. Chu, W. E. Wong, M. J. Palakal, and C.-C. Hung, Eds. ACM, 2011, pp. 1550–1557.*

[9] *K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in ACM Conference on Computer and Communications Refuge, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 187–198.*

[10] *Y. Dodis, S. P. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in TCC, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 109–127.*

[11] *L. Fortnow, J. Rompel, and M. Sipser, "On the power of multiprover interactive protocols," in Theoretical Computer Science, 1988, pp. 156–161.*

[12] *Y. Zhu, H. Hu, G.-J. Ahn, Y. Han, and S. Chen, "Collaborative integrity verification in hybrid clouds," in IEEE Conference on the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, Collaboration, Orlando, Florida, USA, October 15-18, 2011, pp. 197–206.*