Hindawi Publishing Corporation
The Scientific World Journal
Volume 2016, Article ID 4913015, 8 pages
http://dx.doi.org/10.1155/2016/4913015

Research Article

An Elliptic Curve Based Schnorr Cloud Security Model in Distributed Environment

Vinothkumar Muthurajan[1] and Balaji Narayanasamy[2]

[1]*Department of CSE, University College of Engineering, Dindigul, Tamil Nadu 624622, India*
[2]*Department of Information Technology, KLN College of Engineering, Madurai, Tamil Nadu 630612, India*

Correspondence should be addressed to Vinothkumar Muthurajan; vinothkumarphd2015@hotmail.com

Received 1 December 2015; Accepted 21 January 2016

Academic Editor: Xinyi Huang

Cloud computing requires the security upgrade in data transmission approaches. In general, key-based encryption/decryption (symmetric and asymmetric) mechanisms ensure the secure data transfer between the devices. The symmetric key mechanisms (pseudorandom function) provide minimum protection level compared to asymmetric key (RSA, AES, and ECC) schemes. The presence of expired content and the irrelevant resources cause unauthorized data access adversely. This paper investigates how the integrity and secure data transfer are improved based on the Elliptic Curve based Schnorr scheme. This paper proposes a virtual machine based cloud model with Hybrid Cloud Security Algorithm (HCSA) to remove the expired content. The HCSA-based auditing improves the malicious activity prediction during the data transfer. The duplication in the cloud server degrades the performance of EC-Schnorr based encryption schemes. This paper utilizes the blooming filter concept to avoid the cloud server duplication. The combination of EC-Schnorr and blooming filter efficiently improves the security performance. The comparative analysis between proposed HCSA and the existing Distributed Hash Table (DHT) regarding execution time, computational overhead, and auditing time with auditing requests and servers confirms the effectiveness of HCSA in the cloud security model creation.

1. Introduction

Cloud computing intensifies the Information Technology (IT) architecture with the following advantages: on-demand self-service, resource elasticity, and shared pool access. The objective of cloud paradigm is to share the data computations over the scalable network nodes, namely, user computers, cloud services, and data centers. Several grades of services are available in the cloud architecture, namely, Software As A Service (SAAS), Platform As A Service (PAAS), and Infrastructure As A Service (IAAS) as shown in Figure 1. IAAS describes the consumer ability to handle the provisional processing by using conventional resources. PAAS denotes the deployment of consumer-created applications into the cloud structure. SAAS defines the running process of provider's applications on the cloud structure. The movement of data to cloud raised the integrity challenges in the auditing process. The cloud services auditing assures the remote data integrity.

The higher data burden consumes more time. Hence, new methods are required to reduce the burden of metadata in the cloud services for better security. The data storage auditing system carries three modules, namely, owner, auditor (Third Party Auditor (TPA)), and server. The TPA audits the owners and servers in the system model. Several privacy-preserving, dynamic audit service protocols govern the auditing process. The cloud services relying on the network infrastructures suffer the various attacks such as replace, forging and reply attacks that affect the security. Key management schemes such as symmetric (compact key, pseudorandom functions) and asymmetric algorithms (RSA, DES, AES, and Blowfish) are applied to a cloud computing model to ensure the security.

The key length of the cryptographic mechanism should be maintained for higher security that leads to more computational workload. The Elliptic Curve Cryptography (ECC) is applied to cloud service model to overcome the problems of computational load and key length maintenance.

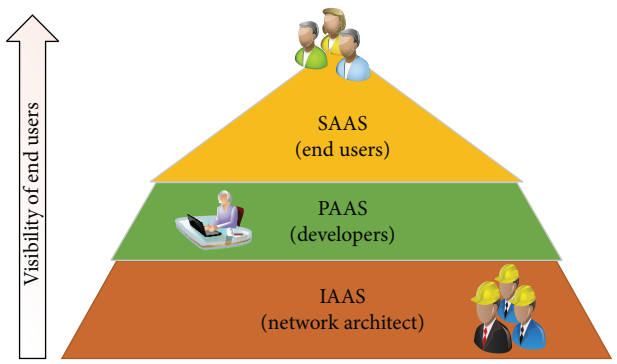

Figure 1: Cloud services.

The optimization of servers and the allocation of data centers are achieved by a virtual machine (VM) based cloud computing. The overloading effect of server moves the VM into an energy model in which temperature based resource utilization is performed to analyze the degree of overloading effect. But the secure data transfer is an investigating process in an energy model.

The blooming filter process in cloud computing improves the data security. The system consists of three modules, namely, Aggregation and Distribution (AD), users, and clouds. The utilization of multiple ADs reduced the communication cost. But the retrieval of matched files further improves the reduction of cost in demand. This paper addresses the security problems in cloud computing and discusses the solution by ECC. Initially, a reconfiguration of the traditional ECC model with the new key generation mechanism to improve the security and reduce the execution time is carried out. Then, blooming filter concept application in proposed ECC-based model removes the irrelevant resources from server to improve the auditing efficiency and malicious activity prediction. The proposed blooming filter-ECC-based model analyzes the performance parameters of execution time, storage complexity, and security which confirms the effectiveness.

This paper is organized as follows: Section 2 describes the related works on cloud security model and auditing process. Section 3 discusses the proposed Hybrid Cloud Security Algorithm (HCSA) implementation. Section 4 presents the performance analysis of HCSA regarding security, execution time, and storage complexity. Finally, Section 5 presents the conclusion.

2. Related Work

This section discusses the traditional research works on the auditing process and addressed the security issues in the cloud service models. Wang et al. submitted the auditable cloud data storage to validate the data hosting on the network architecture [1]. The secure data hosting to the cloud was affected by the different identities of data owners and servers. Yang and Jia proposed an independent auditing service for host monitoring. They suggested the auditing protocol requirements and analyzed the existing auditing process on the security and performance [2]. The growth of cloud computing model depends upon the security challenges. Kulkarni et al. introduced the detailed analysis of security challenges in cloud computing system and the service delivery types [3]. The outsourcing through the untrusted cloud leads to insecure model. Zhu et al. constructed the dynamic audit service based on the index hash table and fragment structure techniques [4]. The performance of services was improved by a probabilistic query and periodic verification. The privacy vulnerabilities and online burden in fragmentation lead to the security problem. Wang et al. proposed the secure cloud storage system based on Privacy-Preserving Public Auditing (PPPA) [5] to reduce the vulnerabilities. They extended the results to offer simultaneous multiusers auditing. Yang and Jia designed an auditing framework for efficient cloud storage systems based on PPPA [6] for dynamic operations of data. The PPPA based cloud computing effectively reduced the computation cost of the audit. The auditing services address the risk issues in data access.

Li et al. performed Attributes-Based Encryption (ABE) schemes on Personal Health Record (PHR) [7] file. They focused cloud computing model on three aspects, namely, multiple data owner scenarios, a division of the multiple users into security domain, and complexity reduction in key management policies. The encrypted data suffered from the multikeyword search problem in multisecurity domain. Cao et al. solved the challenging problem of Multikeyword Ranked Search Encryption (MRSE) [8] that improved the privacy requirements. Ryan briefly analyzed the issues related to secure cloud computing model creation. The data sharing between the service providers based on symmetric key management schemes is regarded as a core scientific problem [9]. The data sharing in cloud storage depended on the factors such as security and flexibility. Chu et al. presented the aggregate cryptosystem [10] for data sharing. The aggregate key released by key holder according to flexible choice and the other keys were kept confidential.

Hwang et al. separated the encryption/decryption mechanisms by using the Customer Relationship Management (CRM) [11] service. CRM provided suggestions for multiparty Service Level Agreement (SLA). The security requirements were specified by using CRM according to privacy issues. Suo et al. discussed the processing groups of cloud service model encryption, communication security, and protection of sensor data [12]. Jager et al. extended the cloud service model by considering the unrestricted attacks [13] to sealed cloud. The data confidentiality was poor in sealed cloud, which was improved by new Cipher cloud. Kaur and Singh ensured the data confidentiality by two encryption schemes [14]. The data transfer between cloud server and client is encrypted and kept confidential in cipher cloud.

Key Distribution Centers (KDC) distributed keys to both the users and the cloud servers where a single key was replaced by the multiple keys of the owners. Ruj et al. proposed the Distributed Access Control in Cloud (DACC) algorithm [15] for KDC. They applied Attribute-Based Encryption (ABE) by a pairing of elliptic curves. The unique KDC-based cloud computing inherited the security issues. The insolubility of mathematical problems in KDC

dealt with the new cryptography scheme termed as Elliptic Curve Cryptography (ECC). Chakraborty et al. presented homomorphic encryption scheme [16] based on ECC. Fast data access was performed by using the Merkle Hash Tree (MHT) at the server. The data leakage problem was not considered in a homomorphic scheme. Lee and Chen presented the cloud aided computation with elliptic curve cryptosystems [17] to deal with the leakage problem. They also prevented the active and passive attacks such as guessing and modification attacks. Cloud Service Providers (CSP) required updating and scaling of data on remote servers. Barsoum and Hasan performed the data outsourcing from owner to CSP with mutual trust between the CSPs by using outsourcing algorithm [18]. Chen et al. proposed secure outsourcing algorithm [19] to untrusted program models and achieved the secure encryptions and signatures. The energy consumption was more in traditional cloud computing processes.

Goudarzi and Pedram utilized the virtual machine [20] and server consolidation in the data center to reduce the energy consumption. The resource requirement reduced significantly by using virtual machine model. The dynamic allocation of data centers was an important et al. presented the virtualization based system for optimization [21]. They also introduced the concept of skewness to measure the dissimilar items in the multidimensional resource utilization. The deployment of virtual machine required computing resources. Shiraz et al. analyzed the effect of virtual machine deployment [22] at the execution time. The migration cost of virtual machine altered in accordance with configurations and workloads. Liu et al. predicted the performance of migration and the cost of energy quantitatively by hypervisor virtual machine model [23]. The hypervisor virtual machine model was evaluated on representative workloads. The usage of resources was efficiently enhanced by deduplication technique. But deduplication suffered from security weakness. Blasco et al. presented the solution based on bloom filter [24] for efficient deduplication. They provided the description about bloom filter and compared the solution through security analysis by using extensive benchmarking sets. The search time of text in encrypted documents was more. Pal et al. reported the novel approach for storing the data in a remote server and the searching process in constant time without degradation [25]. The cloud security model analyzed by an enhanced bloom filter with the EC-Schnorr based encryption scheme is presented in this paper.

3. Elliptic Curve Based Schnorr Model for Cloud Security Improvement Using HCSA

This section presents the detailed description of the proposed Hybrid Cloud Security Algorithm (HCSA) in the cloud security model. The flow diagram of the HCSA implementation is shown in Figure 2. The workflow comprises various processes such as system model, threat model, auditing, signature set creation/verification, and duplication removal to improve the security performance. Initially, the cloud security model is created in two stages, namely, system model and threat model. Then, the auditing process is performed

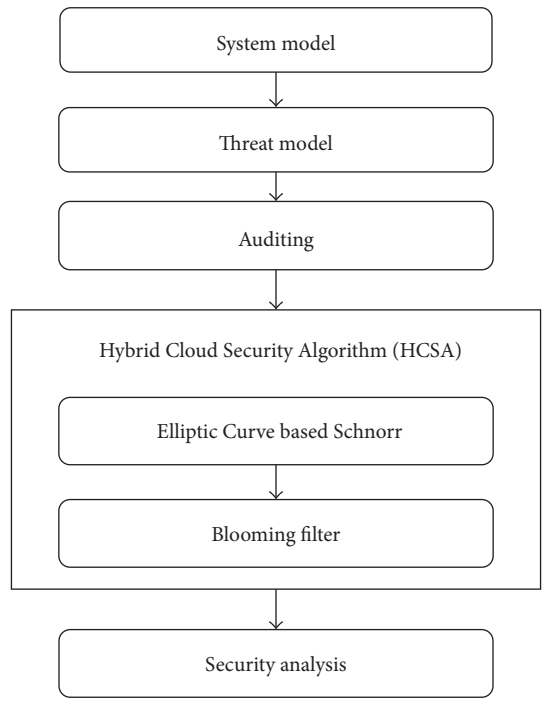

Figure 2: Flow diagram of proposed method.

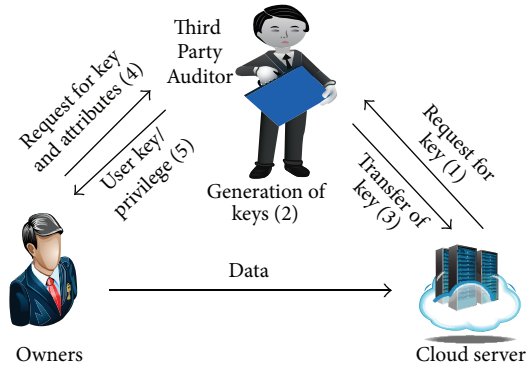

Figure 3: System initialization.

on the created models to address the various security issues and attacks. Then, an Elliptic Curve-Schnorr scheme based encryption/decryption performed on cloud security model and, finally, the application of the enhanced bloom filter concept to EC-Schnorr result in that enhanced the security performance with less overhead and execution time.

3.1. System Model. The cloud security model contains three modules, namely, data owners (cloud users), cloud server, and Third Party Auditors as shown in Figure 3. The cloud users store large amount of data in the cloud. Initially, the data owners computes metadata of user data without considering cryptographic keys. Cloud server is monitored by Cloud Service Provider (CSP), which provides the data storage space and computation resources. The capability of Third Party Auditor (TPA) is to improve the reliability of cloud data

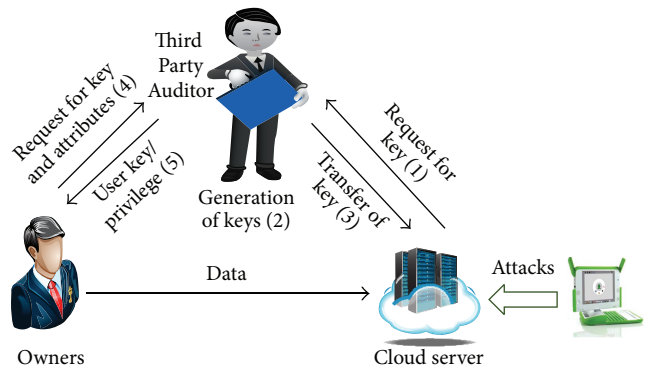

Figure 4: Threat model.

storage. The users dynamically interact with cloud server for accessing and updating stored data in various applications. The computation resources and burden are reduced by ensuring the integrity of outsourced data. The attacks introduced in cloud server significantly affect the integrity.

3.2. Threat Model. The system model creation is based on the consideration of the Third Party Auditor (TPA) to be genuine and mysterious. Hence, the privacy requirement for auditing protocol is necessary to create mysterious TPA. The assumption for creation of TPA is that none of the data are leaked out during the auditing process. But the attacks in threat model cause the data leakage. The threat model analyzes the attacks in server as shown in Figure 4.

The server in the cloud system model handles the three types of attacks, namely, replace, reply, and forge attacks.

Replace Attack. The replacement of original metadata (m_i, t_i) with the uncorrupted pair of data (m_k, t_k) denotes the replace attack.

Replay Attack. The new proof generation from the existing without referring data originality introduces an attack called replay attack.

Forge Attack. The enabling of metadata of user data misguides the auditor leads to forge attack.

3.3. Auditing. The Third Party Auditor (TPA) monitors the integrity and status of outsourced data. The assumptions for auditing process are as follows:

(1) TPA is reliable and independent.

(2) TPA evaluates and monitors the integrity and availability of delegated data on regular intervals.

(3) TPA supports the dynamic data operations.

Auditing process is grouped into three processes, namely, tag generation, periodic sampling audit, and dynamic operations. Initially, tag generation process groups the n blocks to generate the verification parameters and index hash values constituted secret key S_k. Random sampling audit process accepts the retrieval proof in response to broadcast of challenges in random sampling as shown in Figure 5.

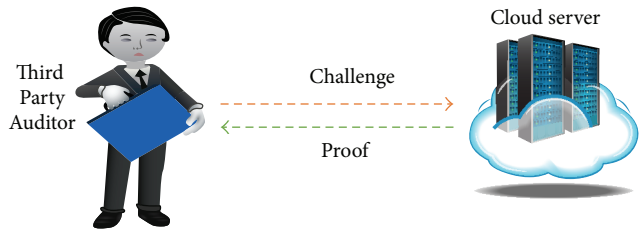

Figure 5: Auditing process.

User application contained secret key S_k derived from index hash table (IHT) update and outsourced file F manipulations. An outsource file consists of n blocks of messages $\{m_1, m_2, \ldots, m_n\}$ and each block of m_i grouped into s sectors like $\{m_{i,1}, m_{i,2}, m_{i,3}, \ldots, m_{i,s}\}$. The tag pair of messages m_i contains signatures σ_i and secrets $S = \tau_1, \tau_2, \ldots, \tau_s$. The convergence of s blocks is estimated with the help of Elliptic Curve-Schnorr algorithm.

3.4. Hybrid Cloud Security Algorithm. The auditing process in this paper is based on the Hybrid Cloud Security Algorithm (HCSA) and comprises two phases, namely, Elliptic Curve based Schnorr Algorithm for signature proof creation/validation and blooming filter to avoid the duplication entry. Initially, the message field and domain parameters are applied to Elliptic Curve based cloud security model to create Schnorr signature set. Then, generation and verification of proof carried were out based on Distributed Hash Table (DHT) entries. Finally, blooming filter was applied to eliminate the multiple entries in DHT. The outsourced file F is represented by Weierstrass's equation given as follows:

$$y^2 = x^3 + ax + b, \quad a, b \in F. \tag{1}$$

The EC- Schnorr based cryptography domain consists of various parameters listed in Table 1. The proposed HCSA for cloud auditing process is as in Algorithm 1.

The algorithm accepts public and private keys and domain parameters (p, a, b, L, n, h) for key generation process. The client in cloud security model generates public P_k and private keys S_k. The base point L corresponding to the field F is chosen on elliptic curve $E(F)$. The pseudorandom

Input: Outsource field (F),
Output: Response of TPA (TRUE or FALSE) (R)
(1) Initialize domain parameters (p, a, b, L, n, h)
(2) **Begin**
(3) Choose a point in elliptic curve $E(F)$.
(4) Select pseudorandom number as a secret key S_k.
(5) Public key $P_k = S_k \cdot L$ // Key pair generation
(6) Select the random number K within the range ($1 \leq K \leq n-1$)
(7) Hash value of message blocks $e_i = H(m_i)$
(8) Random point $KL = (x_1, y_1)$
(9) Calculate $\sigma = x_1 \pmod n$
(10) Calculate $\tau = K^{-1}(e_i + S_k\sigma) \pmod n$
// Signature set creation
(17) Generate block tag for each block $T_i = (e \cdot K^{m_i})^{S_k}$
(18) Client generates Index Hash Table (IHT) with server oriented parameters T, M, and AAI (Ω_i)
(19) Generate hash table IHT
(20) Unique hash value $H(m_i)$ = record in index Hash Table (χ).
(21) S = difference between existing and new entries in IHT
(22) If ($S = 0$)
(23) Update the hash value in table
(24) Else
(25) Goto step (20)
(26) end // Proof $P(T, M, H(m_i), \Omega_i)$ generation
(27) Check $\sigma \in \{0, \ldots, 2^l - 1\}$ and $\tau \in \{1, 2, \ldots, n-1\}$
(28) if (check $\neq$ TRUE)
(29) terminate the process
(30) else
(31) $L_{\text{new}} = [\tau]L + [\sigma]P_k$
(32) If ($L_{\text{new}} = 0$)
(33) output = error
(34) else
(35) temp = OS2I($H_l(m_i) \parallel$ FE2OS(P_k))
(36) if (temp == σ)
(37) r = TRUE
(38) else
(39) r = FALSE
(40) end // Proof verification
(41) if (r = TRUE)
(42) R = blooming(message, integer, hash value)
(43) else
(44) Goto step (6)
(45) **end**

Algorithm 1: Hybrid Cloud Security Algorithm.

Table 1: Parameters.

Parameters	Description
p	Prime number
a	First coefficient in Weierstrass's equation
b	Second coefficient in Weierstrass's equation
L	Base point
n	Order of L
h	Cofactor of L

number S_k within the range ($1 \leq S_k \leq n$) is selected. The public key P_k is calculated from pseudorandom number S_k and base point L by using the following equation:

$$P_k = S_k \cdot L. \tag{2}$$

Finally, the key pair (P_k, S_k) were generated and they were regarded as an output. Then, HCSA accepts public (P_k) and private key S_k, file block F, and selected point L. Random number K is generated within the ranges of ($1 \leq K \leq n-1$). The hash value for message blocks is generated by using the following equation:

$$e_i = H(m_i). \tag{3}$$

The new point (x_1, y_1) with reference to base point (x, y) and pseudorandom number (K) is computed by using the following equation:

$$KL = (x_1, y_1). \tag{4}$$

The signature set (σ, τ) is calculated by using the following equations:

$$\sigma = x_1,$$
$$\tau = K^{-1}\left(e_i + S_k\sigma\right) \pmod{n}. \quad (5)$$

The process of extracting the signature was iteratively done until all the messages in outsourcing field were taken out.

The algorithm generates the proof that contains a tag, Auxiliary Authentication Information (AAI), and index hash table coefficients $H(m_i)$ as a proof. The hash value is calculated by dividing the new entry by the length of the table. The remainder is the required position to insert the new item. The hash value from the distributed table is utilized to generate the proof. The HCSA verifies the generated proof with the Boolean values of TRUE and FALSE. The authenticated message, hash value, authenticated public key, and domain parameters are arranged as proof and then verify whether the generated signature is valid or not. The algorithm accepts the signature outputs (σ, τ) from SignGen. Then, the status (TRUE or FALSE) of signature set is identified by using the condition $\sigma \in \{0, \ldots, 2^l - 1\}$ and $\tau \in \{1, 2, \ldots, n-1\}$.

The false report of validation terminates the process. Otherwise, the new point is calculated according to following equation:

$$L_{\text{new}} = [\tau]\, L + [\sigma]\, P_k. \quad (6)$$

The process continued on the basis of L_{new}. The termination occurs for zero values of L_{new} and the process continued for nonzero values of L_{new}. Then, two processes such as Octet String to Integer (OS2I) and Finite Field Element to Octet Series (FE2OS) are involved in the verification process. The OS2I and FE2OS of hash value and public key (P_k) are stored in the temporary variable *temp*. Finally, the comparison between *temp* and signature σ provides the status (TRUE or FALSE) of proof.

3.5. Blooming Filter. The probabilistic data structure to predict the member in a set with minimum false positive rates is referred to as blooming filter. The use of large bit array in blooming filter concept efficiently reduced the false positive probability. The bloom filter contains the positions of bit corresponding to existing entries. The logger generates the K number of bit positions for each entry by hashing of n times and updates the data entry with new Accumulator Entry (AE). The outsourced message field $F = \{m_1, m_2, \ldots, m_n\}$ contained n set of entries arranged into a membership function (M) of bit vector B length n. The hash functions $H = \{h_1, h_2, \ldots, h_n\}$ with h_i: $x \rightarrow \{1, \ldots, n\}$ were computed initially. Then, the filter coefficients are computed by allocation of n bits to zero. The algorithm for filter coefficients prediction is shown in Algorithm 2.

The testing of an element in the membership function returns the TRUE for the presence of an element and FALSE for absence of element in membership. The number of hashing functions applied to determine the status of filter in testing phase significantly reduces the storage complexity and execution time.

Input: Message Field F, integer n, Hash function $H(m_i)$
Output: TPA response
(1) f = allocate n bits to zero
(2) for each m_i in M
(3) calculate hash value h_i
(4) for each h_i
(5) Compute $f(h_i(m_i))$
(6) if $(f(h_i(m_i)) \neq 1)$
(7) response from TPA = FALSE
(8) else
(9) response from TPA = TRUE

Algorithm 2: Filter coefficient prediction.

4. Performance Analysis

This section presents the performance analysis of the proposed Hybrid Cloud Security Algorithm (HCSA) regarding execution time, storage complexity, and security. The Elliptic Curve based Schnorr signature generation and blooming filter prediction enhanced the security performance compared to Distributed Hash Table (DHT).

4.1. Storage Complexity. The storage complexity of Cloud Service Provider (CSP) depends on outsourced data file F and hash value. CSP contains the outsourced data file $F = \{m_1, m_2, \ldots, m_n\}$, data block tags T, and random chosen point K, which are used to compute the digital signature denoted as hash value $H(m_i)$. The cost of storage C_{CSP} and the storage complexity S_{CSP} are calculated by

$$C_{\text{CSP}} = |F| + |T| + |K| + \left|H\left(m_i\right)\right|,$$
$$S_{\text{CSP}} = S_{\text{CSP}}\left(n\right). \quad (7)$$

Figure 6 depicts the auditing time variation with the number of auditing requests. The total number of auditing requests in our proposed model is 100. The DHT and proposed HCSA consume 1156 and 1090 ms for minimum auditing requests handling. Also, they consume 958 and 850 ms for maximum auditing requests. The comparison shows the proposed HCSA algorithm offer 5.71 and 11.27% reduction for minimum and maximum requests compared to existing DHT due to the duplication elimination.

4.2. Execution Time. The execution time is the time required from challenge creation to proof verification. The execution time t_{exec} of TPA depends upon various parameters, namely, time for challenge creation (t_{chall}), K pseudorandom permutations $(K \times t_{\text{PSP}})$, K pseudorandom functions $(K \times t_{\text{PSF}})$, time for proof creation (t_{pr}), time for signature verification (t_{very}), and comparison time of proof (t_{comp}) in cloud server defined by

$$t_{\text{exec}} = t_{\text{chall}} + K \times t_{\text{PSP}} + K \times t_{\text{PSF}} + t_{\text{pr}} + t_{\text{very}} + t_{\text{comp}}. \quad (8)$$

Figure 7 depicts the execution time variation with number of servers. The total number of servers in our proposed

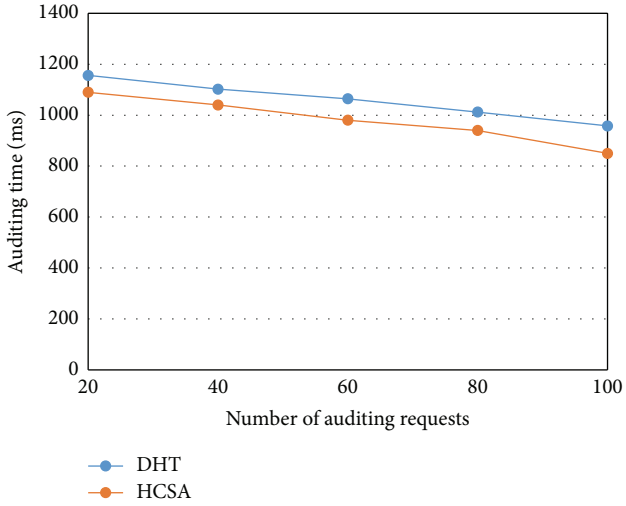

Figure 6: Auditing time versus number of auditing requests.

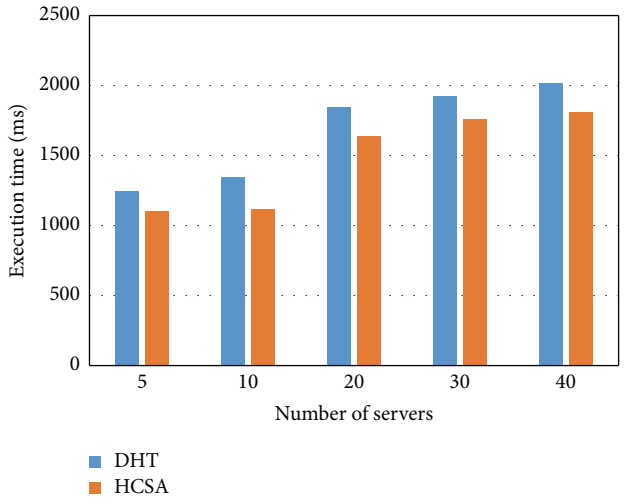

Figure 7: Execution time versus number of servers.

model is 40. The DHT and proposed HCSA consume 1245 and 1100 ms for minimum servers. Also, they consume 2014 and 1814 ms for maximum servers. The comparison shows the proposed HCSA algorithm offers 11.65 and 9.93% reduction for minimum and maximum requests compared to existing DHT due to the ECC-based signature creation with the optimized steps.

4.3. Computational Overhead. The computation overhead is lesser than the DHT models. Figure 8 depicts the computation overhead with respect to the number of servers. The total number of servers used in our proposed model is 40. For each server, the computation overhead with HCSA is lower than the DHT model.

The increase in number of servers gradually increases the computational overhead generally. But the optimization and duplication removal by using the proposed algorithm provide lesser computational overhead for minimum (47.5%) and maximum servers (23.69%) compared to existing DHT.

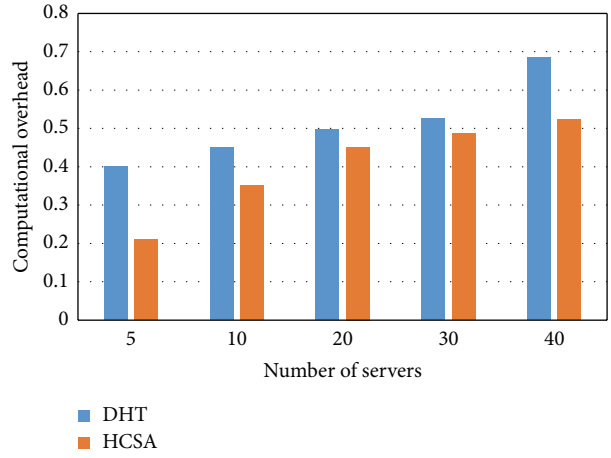

Figure 8: Computational overhead versus number of servers.

5. Conclusion

The proposed Hybrid Cloud Security Algorithm (HCSA) presented the solution to the problems in secure cloud data storage system modelling based on the combination of Elliptic Curve based Schnorr (EC-Schnorr) scheme and blooming filter. The efficiency of the system improved and the storage complexity is reduced by removal of nonrelated contents and duplication. The malicious activity prediction was improved by using the proposed trust evaluation model. Moreover, blooming filter concept applied to the security model to avoid the cloud server. The optimization in the computational steps by ECC signature set and the duplication removal by blooming filter in the proposed Hybrid Cloud Security Algorithm (HCSA) effectively reduced the execution time, computational overhead, and auditing time with the number of auditing requests and servers. The comparative analysis between the HCSA-based model and Distributed Hash Table (DHT) model confirmed the effectiveness of proposed hybrid method of encryption schemes in cloud security model creation.

Conflict of Interests

The authors proclaim that there is no conflict of interests concerning the publication of this paper.

References

[1] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *IEEE Network*, vol. 24, no. 4, pp. 19–24, 2010.

[2] K. Yang and X. Jia, "Data storage auditing service in cloud computing: challenges, methods and opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[3] G. Kulkarni, J. Gambhir, T. Patil, and A. Dongare, "A security aspects in cloud computing," in *Proceedings of the 3rd International Conference on Software Engineering and Service Science (ICSESS '12)*, pp. 547–550, IEEE, Beijing, China, June 2012.

[4] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds,"

IEEE Transactions on Services Computing, vol. 6, no. 2, pp. 227–238, 2013.

[5] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[6] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2013.

[7] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[8] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 222–233, 2014.

[9] M. D. Ryan, "Cloud computing security: the scientific challenge, and a survey of solutions," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2263–2268, 2013.

[10] C.-K. Chu, S. S. M. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 468–477, 2014.

[11] J.-J. Hwang, H.-K. Chuang, C.-H. Wu, and Y.-C. Hsu, "A business model for cloud computing based on a separate encryption and decryption service," in *Proceedings of the International Conference on Information Science and Applications (ICISA '11)*, pp. 1–7, IEEE, Jeju Island, South Korean, April 2011.

[12] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: a review," in *Proceedings of the International Conference on Computer Science and Electronics Engineering (ICCSEE '12)*, pp. 648–651, Hangzhou, China, March 2012.

[13] H. A. Jager, A. Monitzer, R. Rieken, E. Ernst, and K. D. Nguyen, "Sealed cloud—a novel approach to safeguard against insider attacks," in *Trusted Cloud Computing*, pp. 15–34, Springer International, 2014.

[14] M. Kaur and R. Singh, "Implementing encryption algorithms to enhance data security of cloud in cloud computing," *International Journal of Computer Applications*, vol. 70, no. 18, pp. 16–21, 2013.

[15] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: distributed access control in clouds," in *Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '11)*, pp. 91–98, Changsha, China, November 2011.

[16] T. K. Chakraborty, A. Dhami, P. Bansal, and T. Singh, "Enhanced public auditability & secure data storage in cloud computing," in *Proceedings of the 3rd IEEE International Advance Computing Conference (IACC '13)*, pp. 101–105, IEEE, Ghaziabad, India, February 2013.

[17] N.-Y. Lee and Z.-L. Chen, "Cloud server aided computation for elgamal elliptic curve cryptosystem," in *Proceedings of the IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW '13)*, pp. 11–15, Tokyo, Japan, July 2013.

[18] A. F. Barsoum and A. Hasan, "Enabling dynamic data and indirect mutual trust for cloud computing storage systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2375–2385, 2013.

[19] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2386–2396, 2014.

[20] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD '12)*, pp. 750–757, IEEE, Honolulu, Hawaii, USA, June 2012.

[21] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107–1117, 2013.

[22] M. Shiraz, S. Abolfazli, Z. Sanaei, and A. Gani, "A study on virtual machine deployment for application outsourcing in mobile cloud computing," *The Journal of Supercomputing*, vol. 63, no. 3, pp. 946–964, 2013.

[23] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster Computing*, vol. 16, no. 2, pp. 249–264, 2013.

[24] J. Blasco, R. Di Pietro, A. Orfila, and A. Sorniotti, "A tunable proof of ownership scheme for deduplication using bloom filters," in *Proceedings of the IEEE Conference on Communications and Network Security (CNS '14)*, pp. 481–489, San Francisco, Calif, USA, October 2014.

[25] S. K. Pal, P. Sardana, and A. Sardana, "Efficient search on encrypted data using bloom filter," in *Proceedings of the International Conference on Computing for Sustainable Global Development (INDIACom '14)*, pp. 412–416, IEEE, New Delhi, India, March 2014.