A Survey of Public Provable Data Possession Schemes with Batch Verification in Cloud Storage*

Sooyeon Shin and Taekyoung Kwon†
Graduate School of Information, Yonsei University, Seoul, Republic of Korea
shinsy80@gmail.com, taekyoung@yonsei.ac.kr

Abstract

Cloud storage service, which enables users to store the data in the remote cloud and to access to it over the Internet regardless of location and time, is an important development trend in information technologies. This cloud storage provides on-demand high quality storage and computing resources, but it also introduces new security challenges. Thus, an auditing service is desired to convince users that their data are correctly stored in the cloud. In this paper, we present a survey of remote auditing schemes in the literature. We explain the concept of remote data auditing, and then describe system model and taxonomy of remote data auditing schemes, especially focusing on provable data possession schemes which support public auditability and batch verification. We review seven batch auditing schemes and compare them in terms of their features and performance. Finally, we introduce some challenging issues in the design of efficient batch auditing.

Keywords: Cloud Computing, Remote Data Auditing, Provable Data Possession, Public Auditability, Batch Auditing

1 Introduction

Cloud computing is one of the emerging technologies that aims to provide on-demand use, device and location independence over the Internet, efficient share of resources, and maintenance. Cloud storage enables users to store and access the data in the remote cloud through the Internet regardless of location and time and it is an important service of cloud computing. Although this new paradigm of data hosting service promises more secure and reliable environment, it also brings new security vulnerability to the users. Users' outsourced data can be lost or corrupted due to the inside and outside attacks, or system failures [9], [12]. However, cloud service providers might be dishonest. For example, they could discard the data that has been rarely accessed for monetary reasons, or they might even hide data loss incidents in order to maintain their reputations. Therefore, users need a way to check that their outsourced data are correctly stored in the cloud. To solve the problem of data integrity checking, many remote data auditing schemes have been proposed [1], [11], [14], [15], [16], [18], [19], [22], [25], [26].

Remote data auditing refers to a sampling of the outsourced data to securely, frequently, and efficiently verify the correctness of the data in the cloud. Remote data auditing is a probabilistic approach for the data integrity because it randomly selects and checks a small portion of the whole data. The following security and performance criteria should be taken into account to design remote data auditing schemes such as (1) Efficiency: the auditing service is reasonable for the storage, computational, and communication overheads, (2) Public auditability: to reduce the computation cost over the user, the third

Journal of Internet Services and Information Security (JISIS), volume: 5, number: 3 (August 2015), pp. 37-47

*This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1008) supervised by the IITP(Institute for Information & communications Technology Promotion).

†Corresponding author: Graduate School of Information, Yonsei University, 50 Yonsei-ro, Seodamun-gu, Seoul, 120-749, Republic of Korea, Tel.: +82-2-2123-4523, Web: `http://seclab.yonsei.ac.kr/`

party auditor verifies the correctness of outsourced data underlying delegations of users, (3) Storage correctness: there exists no cheating cloud server that can pass the auditor's audit without indeed storing users' data intact, (4) Privacy-preserving: the third party auditor cannot derive users' data content during the auditing processes, and dynamic update, and (5) Dynamic update: the user can update the outsourced data through block insertion, modification, and deletion without requiring to download the data.

As cloud storage services have been widely adopted, the third party auditor may receive many requests from multi-users, even in multi-clouds. Thus, to improve the efficiency of the auditor, several batch auditing schemes have been proposed, which allow the auditor to simultaneously handle multiple auditing delegations from different users [11], [16], [22], [25]. In large-scale cloud storage systems, batch auditing for multi-users in multi-clouds is essential to improve the auditing performance. In this paper, we give an survey of batch auditing schemes with public auditability. Firstly, we describe the concept of provable data possession-based remote data auditing as well as its system model and taxonomy. Then we review seven batch auditing schemes and compare them in terms of their features and performance. Furthermore, we introduce open research challenges with regard to batch auditing.

In Section 2, we describe the concept, system model, and taxonomy of provable data possession-based remote data auditing. Next, a review and comparison of batch auditing schemes are presented in Section 3 and 4, respectively. In Section 5, open research challenges are introduced, and the conclusion is given in Section 6.

2 Provable Data Possession-based Remote Data Auditing

Remote data auditing techniques can be largely classified into two categories: provable data possession (PDP) based remote data auditing and proof of retrievability (PoR) based remote data auditing. For a user with data stored at an untrusted cloud server, PDP allows to verify the data possessed by the server without retrieving it. PDP is responsible to preserve the integrity of outsourced data, while PoR ensures both the privacy and integrity of outsourced data and data recovery by using error-correcting codes. In this section, we describe the system model and taxonomy of PDP-based remote data auditing schemes.

2.1 System model

Figure 1 illustrates the system model of PDP-based remote data auditing. Remote data auditing schemes usually include three main components: users including data owners, cloud service provider (CSP), and Third Party Auditor (TPA). The users store their data in the remote cloud servers and rely on them for data maintenance. The CSPs with cloud servers provide the data access to users as well as powerful storage and computational resources. The TPA has expertise and capabilities to audit data storage on behalf of the user upon request. Remote data auditing services follow two phases: setup and audit. The setup phase involves a key generation step and an authenticator generation step. In the key generation step, users negotiate the cryptographic keys with CPSs and the TPA. In the authenticator generation step, users compute authenticators as data tags of their data. The audit phase is usually conducted via a challenge-response procedure, so this phase involves challenge, response, and verify steps. If users delegate auditing tasks to a TPA, the TPA performs the challenge step to check the correctness of users' data. In the challenge step, the TPA generates a challenge message which includes indexes of randomly selected data blocks and sends it to a CSP (or multiple CSPs). In the response step, upon receiving the challenge message, the CSP generates and sends a response message as a proof of possession to the TPA. The proof of possession includes both a data proof and the an authenticator proof. Upon receiving the response message from the challenging CSP, the TPA verifies the correctness of the proof in the verify step.

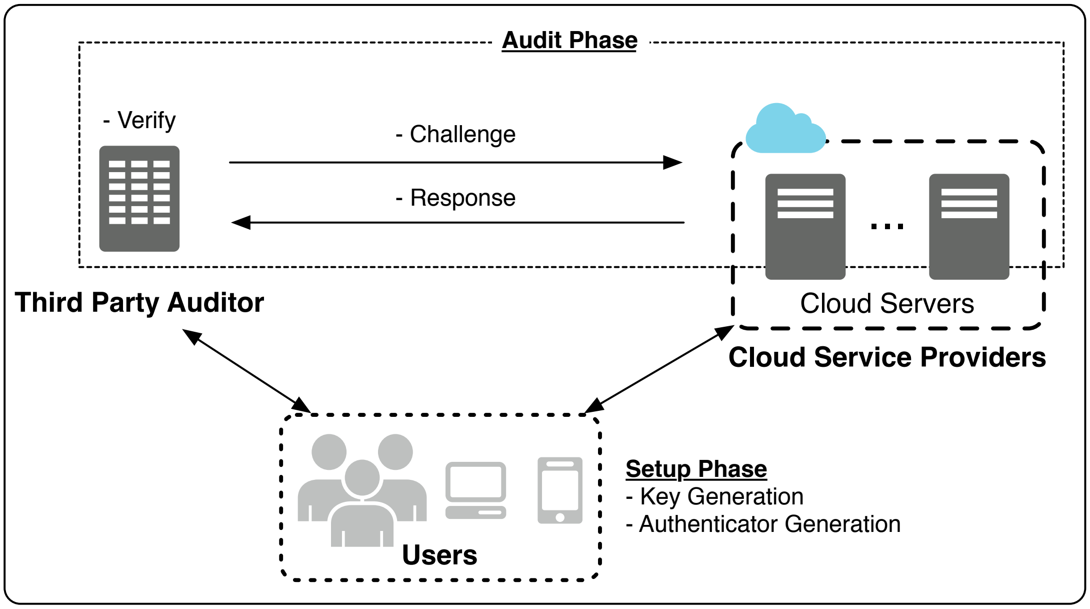


Figure 1: System model of PDP-based remote data auditing.

2.2 Taxonomy

PDP-based remote auditing schemes can be categorized based on auditing mode, security property, update mode. Auditing mode can be either private mode or public mode. In private auditing mode, only users can audit the outsource data, while in public auditing mode (public auditability), the TPA checks the integrity of the data on demand without retrieving a copy of the whole data or introducing online burden to the user. Security properties for a secure remote auditing scheme are following: Storage correctness, privacy-preserving, and batch auditing. Update mode can be either static or dynamic mode. In static mode, users should download the whole data to update, while in dynamic mode, users can perform block-level update operations (e.g., insertion, modification, deletion) on the file while maintaining the same level of data correctness assurance.

Figure 2 shows the taxonomy of PDP-based public auditing schemes in terms of update mode and one of the security properties, batch auditing. Ateniese et al. proposed PDP schemes (PDP) by firstly considering public auditability in their model and then utiilzing RSA-based homomorphic tags to achieve public auditability [1]. However, this scheme does not support dynamic updates. In further research [2], Ateniese et al. proposed a scalable PDP scheme (SPDP) based on symmetric-key cyptography by extending PDP to partially support dynamic data operations. Erway et al. firstly propose two fully dynamic PDP schemes, DPDP I and DPDP II, by using rank-based authentication skip list structure and rank-based RSA trees, respectively [7]. Esiner et al. proposed a flexible dynamic PDP scheme (F-DPDP) [8] by using FlexList (Flex Length-based Authenticated Skip List method) in order to overcome disadvantages of DPDP I and DPDP II. Wang et al. proposed a public PDP scheme (PPDP) by using Mekle Hash Tree (MHT) to authenticate block tags, which supports both public auditability and dynamic updates [18]. Zhu et al. also proposed dynamic PDP scheme (D-PPDP) which depends on fragment structure, random sampling, and index-hash table [24]. In the next section, we will describe seven batch auditing schemes in Figure 2.

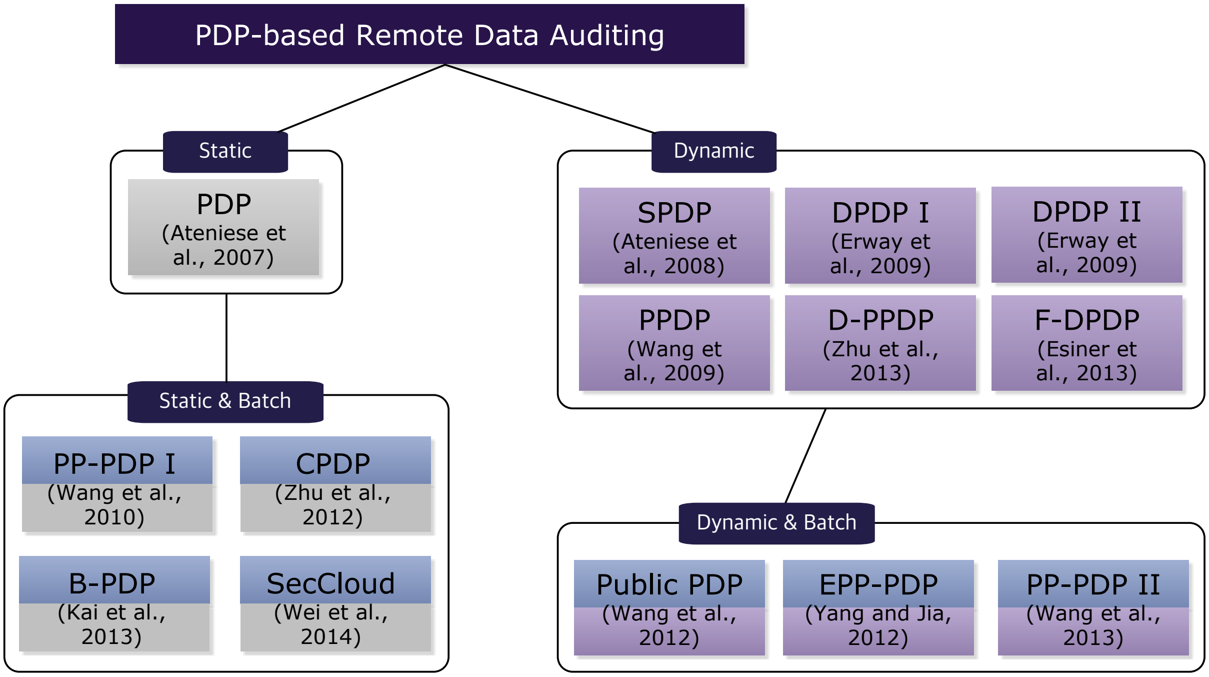

PDP-based Remote Data Auditing
Static
PDP (Ateniese et al., 2007)
Dynamic
SPDP (Ateniese et al., 2008)
DPDP I (Erway et al., 2009)
DPDP II (Erway et al., 2009)
PPDP (Wang et al., 2009)
D-PPDP (Zhu et al., 2013)
F-DPDP (Esiner et al., 2013)
Static & Batch
PP-PDP I (Wang et al., 2010)
CPDP (Zhu et al., 2012)
B-PDP (Kai et al., 2013)
SecCloud (Wei et al., 2014)
Dynamic & Batch
Public PDP (Wang et al., 2012)
EPP-PDP (Yang and Jia, 2012)
PP-PDP II (Wang et al., 2013)

Figure 2: Taxonomy of PDP-based public auditing.

3 Batch Auditing Schemes

In this section, we review PDP-based public auditing schemes which support batch verification. Figure 3 illustrates overview of batch auditing schemes based on the relationship between them. In Figure 3, '$+$' denotes improvements on the previous schemes and '$-$' denotes security flaws.

3.1 PP-PDP I

Wang et al. addressed two fundamental requirements for introducing efficient and secure TPA: firstly, TPA should be able to verify the correctness of the cloud data without retrieving a copy of the whole data and introducing additional on-line burden to the cloud users; secondly, during the auditing process, TPA should prevent data leakage and preserve user data privacy [16]. They proposed a privacy-preserving public auditing (PP-PDP I) scheme which meets the above requirements by uniquely integrating the Homomorphic Linear Authenticator (HLA) with random masking technique. In PP-PDP I, a file of each user is divided into n blocks, so the user generates n authenticators. In PP-PDP I, a CSP masks the sample blocks with randomness when generating a data proof, thus the TPA cannot derive user's data from the data proof. PP-PDP I supports batch auditing for multi-users in a single CSP by using the bilinear aggregate signature scheme [3] which aggregates multiple signatures into a single signature by distinct signers on distinct messages. Based on the technique of signature aggregation and bilinear property, the CSP can combine k multi-users' authenticator proofs for k auditing tasks into a single one, and the TPA can simultaneously audits those multiple tasks.

Although the security of PP-PDP I was evaluated in [16] by analyzing its fulfillment of storage correctness guarantee, privacy preserving guarantee, and security guarantee, Xu et al. found that PP-PDP I is vulnerable to four types of attacks from a malicious cloud server and an outsider attacker: data modification tag forging attack, data lost auditing pass attack, data interception and modification attack, and data eavesdropping forgery [21].

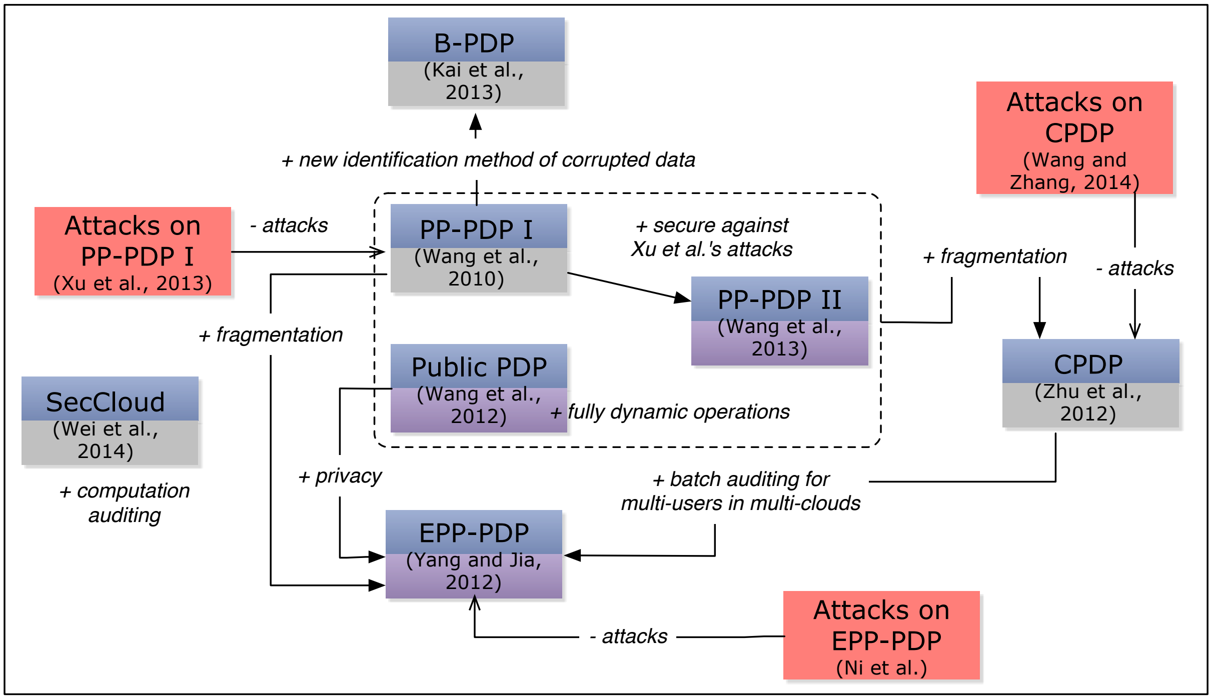


Figure 3: Overview of batch auditing schemes.

3.2 Public PDP

Wang et al. proposed a public and dynamic auditing (Public PDP) scheme that supports both public auditability and complete data dynamics including block insertion by combining the MHT and the PKC-based HLA (e.g., BLS (Boneh-Lynn-Shacham) [4] signature or RSA signature-based authenticator) [19]. The MHT can be used to efficiently and securely verify that a set of tree data elements are undamaged and unaltered. Although the MHT is commonly used to authenticate values of data blocks, Public PDP used it to authenticate both the values and the positions of data blocks. Furthermore, Public PDP solved a node balancing issue of MHT structure. In Public PDP, a MHT is used to check whether a CSP has performed update operations as required. Like PP-PDP I, Public PDP supports batch auditing for multi-users in a single CSP by using the bilinear aggregate signature scheme.

Public PDP does not support the privacy-preserving requirement, thus may leak the data content, since the TPA receives the linear combinations of data blocks from a CSP without any processing.

3.3 PP-PDP II

Wang et al. proposed a privacy-preserving public auditing (PP-PDP II) scheme [15] which is secure against four types of attacks mentioned by Xu et al. [21] and also privacy-preserving. Since PP-PDP II improves over PP-PDP I and Public PDP, it also utillized the HLA with random masking technique, the bilinear aggregate signature scheme, and the MHT structure to achieve the privacy-preserving public auditing, the batch auditing for multi-users, and the dynamic auditing, respectively.

In many situations, a set of proofs generated by the CSP may contain invalid proofs by a malicious CSP or an accidental data corruption. Since all proofs are aggregated into a single proof in the batch auditing, the batch verification will fail if there is even one invalid proof for a single user. Namely, the batch verification only holds when all the proofs are valid. In this case, Wang et al. introduced a recursive divide-and-conquer approach (binary search) to find out whose data blocks or authenticators are corrupted [16], [15]. However, this approach may introduce communication and computation overheads because the CSP should reaggregate proofs and retransmit all proofs, thus the TPA should repeatedly perform batch auditing.

3.4 CPDP

Zhu et al. considered the existence of multiple CSPs to cooperatively store and maintain the clients' data [25]. Thus, data blocks and authenticators of a single users will be separately stored in multiple CSPs. Therefore, Zhu et al. proposed a cooperative provable data possession (CPDP) scheme based on homomorphic verifiable response (HVR) and hash index hierarchy to support dynamic scalability on multiple storage servers in the distributed multi-cloud environments. They extended homomorphic verifiable tags (HVTs), which allow a user to combine the computed tags (authenticators) for multiple blocks of each file into a single value to the concept of HVR which is used to integrate multiple proofs from the different CSPs. Unlike PP-PDP I, Public PDP, and PP-PDP II, in CPDP, a file is divided into n blocks, and each block is split into s sectors, so the file will have $n \times s$ sectors. Also it has n authenticators since each block corresponds to an authenticator. Due to this fragmentation, the storage of authenticators decreases as s increases. Additionally, in CPDP, there is another entity, an organizer, one of CPSs that directly contacts with the TPA. The organizer initiates, verifies (during auditing), organizes, and manages all CSPs.

The authors do not mention the batch auditing but CPDP supports the batch auditing for multi-clouds due to the inherence of its verification architecture. Since an organizer synthesizes a final proof from the received responses from CSPs, the TPA only needs to verify the final proof received from the organizer. However, CPDP does not provide the batch auditing for multi-users because parameters used to generate authenticators for each user are different. In other words, authenticators of distinct users cannot be aggregated into a single proof to conduct batch auditing for multi-users. Recently, Wang and Zhang showed that CPDP does not satisfy the property of knowledge soundness, thus malicious CSPs can deceive the TPA and malicious organizer can trivially deceive the TPA [17].

3.5 EPP-PDP

Yang and Jia proposed a privacy-preserving and efficient storage auditing scheme (EPP-PDP) based on HVT [22]. They pointed out the data privacy problem of Public PDP, the large number of data authenticators of PP-PDP I, and the impossibility of the batch auditing for multi-users of CPDP. To solve the data privacy problem, EPP-PDP generates an encrypted proof instead of the random masking technique by using the bilinear pairing. Due to the bilinear property of the bilinear pairing, the TPA cannot decrypt the encrypted proof but can verify the correctness of the proof. To solve the storage problem of authenticators, EPP-PDP divides a file into n data blocks and each block splits into s sectors as in CPDP.

To support the batch auditing for multi-users in multi-clouds, the TPA in EPP-PDP sends challenges to each CSP. In addition, EPP-PDP is not necessary to add any trusted organizer and to have any commitment phase in CPDP because it does not use the random masking technique. Ni et al. demonstrated that EPP-PDP is vulnerable to an active adversary [13]. They showed that the active adversary can modify the cloud data without being detected by the TPA during an auditing process.

3.6 B-PDP

Kai et al. proposed a public batch data integrity auditing protocol (B-PDP) for multi-clouds [11]. To achieve the privacy-preserving public auditing, B-PDP integrates the homomorphic authenticators used

Table 1: Comparison of batch auditing schemes on the basis of features.

Schemes	Data auditing	Privacy -preserving	Dynamic auditing	Batch auditing		Identification of Corruption
				multi-users	multi-clouds	
PP-PDP I [16]	HLA with random masking	○	×	○	×	Recursive binary search
Public PDP [19]	HLA	×	MHT	○	×	×
PP-PDP II [15]	HLA with random masking	○	×	○	×	Recursive binary search
CPDP [25]	HVR with random masking, Hash index hierarchy	○	×	×	○	×
EPP-PDP [22]	HVT, Bilinear pairing	○	Index table	○	○	×
B-PDP [11]	HLA with random masking	○	×	○	○	Recoverable coding
SecCloud [20]	CBS, Designated verifier signature	○	×	○	×	×

in PDP [1] and homomorphic ciphertext verification in which signatures and messages are encrypted with random values.

B-PDP basically provides the batch auditing for multi-users in multi-clouds with a help of an organizer, a special CSP who organizes the interaction between the TPA and CSPs. Based on a recoverable coding approach [5], B-PDP also provides quick identification of corrupted data without any repeated auditing processes when the batch auditing fails. The TPA can only contact with the organizer and not with other CSPs. The CSPs cannot communicate with each other except for the organizer. The organizer encrypts, encodes, and aggregates proofs into a response received from the distinct CSPs. For the batch auditing, the TPA firstly recovers each user's proof by decoding the aggregated proofs and then simultaneously verifies all proofs. When the batch auditing fails, the TPA identifies the corrupted data individually. The authors pointed out that a recursive binary search approach of PP-PDP I would have high communication and computation costs but B-PDP can effectively identify whose data have been corrupted without any repeated auditing processes. Although identification method in B-PDP does not require additional communication overheads, it still requires individual auditing processes.

3.7 SecCloud

Wei at al. proposed SecCloud, a privacy cheating discouragement and secure computation auditing protocol [20]. SecCloud firstly bridges data storage security and computation auditing security. Then it achieves privacy cheating discouragement to prevent an adversary from achieving a sensitive cloud data by using designated verifier signature schemes [10], [23]. To provide computation security, SecCloud employs CBC (Commitment-Based Sampling) technique introduced in the conventional grid computing [6]. To reduce computational and communication overhead, SecCloud provides the batch verification for multi-users by using identity based aggregate signatures.

Table 2: Comparison of batch auditing schemes on the basis of performance.

Schemes	Computation		Communication			Probability of Detection
	CSP	TPA	Individual auditing	Batch auditing		
				Challenge	Response	
PP-PDP I [16]	$O(t\log n)$	$O(t\log n)$	$O(t\log n)$	$O(KCst)$	$O(KCst\log n)$	$1-(1-\rho)^t$
Public PDP [19]	$O(t\log n)$	$O(t\log n)$	$O(tlogn)$	$O(KCst)$	$O(KCst\log n)$	$1-(1-\rho)^t$
PP-PDP II [15]	$O(t\log n)$	$O(t\log n)$	$O(t\log n)$	$O(KCst)$	$O(KCst\log n)$	$1-(1-\rho)^t$
CPDP [25]	$O(ts)$	$O(t+s)$	$O(t+s)$	$O(KCt)$	$O(KCs)$	$1-(1-\rho)^{ts}$
EPP-PDP [22]	$O(ts)$	$O(t)$	$O(t)$	$O(KCt)$	$O(C)$	$1-(1-\rho)^{ts}$
B-PDP [11]	$O(t\log n)$	$O(t\log n)$	$O(t\log n)$	$O(KCst)$	$O(KCst\log n)$	$1-(1-\rho)^t$

4 Comparison of Batch Auditing Schemes

In this section, we compare the batch auditing schemes in terms of features and performance. Table 1 shows comparison of the batch auditing schemes on the basis of features. In Table 1, data auditing indicates the cryptographic algorithms used to generate authenticators or to audit. HLA, HVT, and HVR belong to homomorphic encryption mechanisms that can perform computation on encrypted data without having them to decrypt the data. To ensure data privacy, PP-PDP I, PP-PDP II,CPDP, and B-PDP employed a random masking technique while SecCloud utilized a designated verifier signature scheme which is proved secure based on the assumption of BDH (Bilinear Diffie-Hellman). EPP-PDP solved a data privacy problem by using the bilinear property of the bilinear pairing. Public PDP manipulated the classic MHT construction for block tag authentication, in order to achieve efficient data dynamics. To support the data dynamic operations, EPP-PDP employed an index table that records an abstraction information of data. Most of the batch auditing schemes utilize aggregate signatures to achieve the batch auditing. EPP-PDP and B-PDP support the batch auditing for both multi-users and multi-clouds. Although PP-PDP II does not support the batch auditing for multi-clouds, it is possible to extend PP-PDP II to achieve multi-cloud batch auditing. For the batch auditing, CPDP and B-PDP require an additional organizer. Identification of corruption shown in Table 1 indicates whether the literatures mentioned a method to identify corrupted data blocks when the batch verification fails. Although Public PDP, CPDP, EPP-PDP, and SecCloud did not mentioned a method for identification of corruption, they can also utilize the recursive binary search technique of PP-PDP I and PP-PDP II.

As we mentioned in Section 1, in order to analyze the efficiency of the remote data auditing schemes, storage, computation, and communication costs should be considered. We only evaluate the computation and communication costs of the batch auditing schemes. Table 2 shows the performance comparison of the batch auditing schemes in terms of computation complexity, communication complexity, and probability of detection. In this table, n is the total number of data blocks of a file, s is the number of sectors in each data block, t is the number of challenged data blocks in the auditing phase, and ρ is the probability of block/sector corruption. Communication complexities are divided into two cases: the individual auditing and the batch auditing. To compare the communication complexity for the batch auditing, we consider the batch auditing for K users and C clouds.

5 Open Research Challenges

In this section, we present some issues and challenges related to the batch auditing as the future research directions. The batch auditing has several advantages in terms of computation and communication overheads compared with the individual auditing. Batch auditing enables the TPA to simultaneously verify multiple auditing proofs for different users, even in multi-cloud. Similarly, in some multi-user batch auditing approaches, a cloud server can simultaneously compute both data and authenticator proofs for multiple users . Namely, multiple proofs on distinct data of different users are aggregated into a single proof. The cloud server only needs to send one data proof and one authenticator proof to the TPA. Therefore, the batch auditing can reduce the computation complexity for the auditing on both the TPA and the cloud server thus save the communication bandwidth.

Wang et al. showed that using the binary search approach, the batch auditing still performs faster than individual verification even if up to 18% of responses are invalid [15]. However, if a single data block or authenticator has been corrupted or discarded, the batch auditing would fail, so the benefit of the batch auditing could be canceled out. As illustrated in Table 1, PP-PDP I and PP-PDP II provided a recursive binary search approach in order to identify the corrupted data. However, to perform the recursive binary search, the TPA should divide the collection of proofs into two halves and recursively verify halves. The recursive binary search incurs additional communication cost since a cloud server should send the individual version of the specific value needed for the batch auditing. In addition, in the case that multiple proofs are aggregated into a single proof for batch auditing (e.g., EPP-PDP), the TPA may repeatedly require a cloud server to send halves of the collection of responses and the cloud server may recursively re-computes halves until the proofs of corrupted data blocks are found.

To reduce communication overhead when the batch auditing fails, B-PDP provided a recoverable coding approach. In the encoding and decoding approach, the TPA can recover individual proofs from the aggregated proof by decoding it. For this purpose, an additional organizer is required to encode the proofs for each user and to aggregate these encoded proofs into one. However, such additional organizer is not practical in cloud computing systems. Although the authors mentioned B-PDP provides quick identification of corrupted data without any repeated auditing processes but the TPA should identify the corrupted data individually after decoding, which is basically similar to repeated auditing processes. In conclusion, an efficient identification protocol of corrupted data is a necessary feature of the batch auditing approaches, since it has an effect on computation and communication costs over both the TPA and the cloud server.

6 Conclusion

In cloud storage services, users can outsource their data to remote cloud servers and access their data via networks at anytime and from anywhere. However, cloud storage services introduce many challenges due to various security threats toward users' outsourced data. Thus, it is important to check the integrity of the outsourced data. In this paper, we explained the concept of remote data auditing and presented a survey of batch auditing schemes, and then compared them in terms of features and performance. Finally, we discussed some challenging issues in the design of batch auditing schemes.

References

[1] G. Ateniece, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *Proc. of the 14th ACM Conference on Computer and Communications Security (CCS'07), Alexandria, Virginia, USA*, pages 598–609. ACM, October 2007.

[2] G. Ateniece, L. V. M. R.D. Pietro, and G. Tsudik. Scalable and efficient provable data possession. In *Proc. of the 4th International Conference on Security and privacy in communication networks (SecureComm'08), Istanbul, Turkey*, pages 1–10, 2008.

[3] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Proc. of the 22nd International Conference on Theory and Applications of Cryptographic Techniques (Eurocrypt'03), Warsaw, Poland, LNCS*, volume 2656, pages 416–432. Springer-Verlag, May 2003.

[4] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.

[5] C. Chen, Y. Lin, Y. Lin, and H. Sun. RCDA: Recoverable concealed data aggregation for data integrity in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(4):727–734, April 2012.

[6] W. Du, M. Murugesan, and J. Jia. Uncheatable grid computing. In *Proc. of the 24th International Conference on Distributed Computing Systems*, pages 4–11. IEEE, 2004.

[7] C. Erway, A. Kupclu, C. Papamanthou, and R. Tamassia. Dynamic provable data possession. In *Proc. of the 16th ACM Conference on Computer and Communications Security (CCS'09), Chicago, Illinois, USA*, pages 213–222. ACM, 2009.

[8] E. Esiner, A. Kachkeev, and O. Ozkasap. FlexList: Optimized skip list for secure cloud storage. Technical report, Ko University, 2013.

[9] G. R. Goodson, J. J. Wylie, G. R. Ganger, and M. K. Reiter. Efficient Byzantine-tolerant erasure-coded storage. In *Proc. of 2004 International Conference on Dependable Systems and Networks*, pages 135–144. IEEE, June-July 2004.

[10] Q. Huang, G. Yang, D. Wong, and W. Susilo. Efficient strong designated verifier signature schemes without random oracle or with non-delegatability. *International Journal of Information Security*, 10(6):373–385, November 2011.

[11] H. Kai, H. Chuanhe, W. Jinhai, Z. Hao, C. Xi, L. Yilong, Z. Lianzhen, and W. Bin. An efficient public batch auditing protocol for data security in multi-cloud storage. In *Proc. of the 2013 8th ChinaGrid Annual Conference (ChinaGrid'13), Changchun, China*, pages 51–56. IEEE, August 2013.

[12] V. Kher and Y. Kim. Securing distributed storage: Challenges, techniques, and systems. In *Proc. of the 2005 ACM Workshop on Storage Security and Survivability (StorageSS'05), Fairfax, Virginia, USA*, pages 9–25. ACM, 2005.

[13] J. Ni, Y. Yu, Y. Mu, and Q. Xia. On the security of an efficient dynamic auditing protocol in cloud storage. *IEEE Transactions on Parallel and Distributed Systems*, (1):1, PrePrints, doi:10.1109/TPDS.2013.199.

[14] H. Shacham and B. Waters. Compact proofs of retrievability. In *Proc. of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'08), Melbourne, Australia, LNCS*, volume 5350, pages 90–107. Springer-Verlag, 2008.

[15] C. Wang, S. S.-M. Chow, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for secure cloud storage. *IEEE Transactions on Computers*, 62(2):362–375, February 2013.

[16] C. Wang, Q. Wang, K. Ren, and L. Lou. Privacy-preserving public auditing for data storage security in cloud computing. In *Proc. of The 29th IEEE Conference on Computer Communications (INFOCOM'10), San Diego, California, USA*, pages 525–533. IEEE, March 2010.

[17] H. Wang and Y. Zhang. On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):264–267, January 2014.

[18] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou. Enabling public verifiability and data dynamics for storage security in cloud computing. In *Proc. of the 14th European Conference on Research in Computer Security (ESORICS'09), Saint-Malo, France, LNCS*, volume 5789, pages 355–370. Springer-Verlag, 2009.

[19] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li. Enabling public auditability and data dynamics for storage security in cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(5):847–859, May 2011.

[20] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos. Security and privacy for storage and computation in cloud computing. *Information Sciences*, 258:371–386, 2014.

[21] C. Xu, X. He, and D. Abraha-Weldemariam. Cryptoanalysis of wang's auditing protocol for data storage security in cloud computing. In *Proc. of the 3rd International Conference on Information Computing and Applications (ICICA'12), Part II, Chengde, China, CCIS*, volume 308, pages 422–428. Springer-Verlag, September 2012.

[22] 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*, 24(9):1717–1726, September 2013.

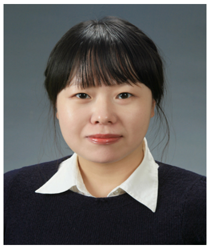
[23] J. Zhang and J. Mao. A novel ID-based designated verifier signature sheme. *Information Sciences*, 178(3):766–773, February 2008.

[24] Y. Zhu, G., H. Hu, S. S. Yau, H. G. An, and C. Hu. Dynamic audit services for outsourced storages in clouds. *IEEE Transactions on Services Computing*, 6(2):227–238, 2013.

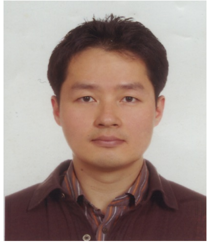
[25] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu. Cooperative provable data possession for integrity verification in multicloud storage. *IEEE Transactions on Parallel and Distributed Systems*, 23(12):2231–2244, December 2012.

[26] 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 *Proc. of the 2011 ACM Symposium on Applied Computing (SAC'11), TaiChung, Taiwan*, pages 1550–1557. ACM, 2011.

Author Biography



Sooyeon Shin received her B.S., M.S., and Ph.D. degrees in computer science and engineering from Sejong University, Seoul, Korea, in 2004, 2006, and 2012, respectively. From 2012 to 2013, she was a post-doctoral researcher at Sejong University. In 2013, she joined Yonsei University, Seoul, Korea, to continue her post-doctoral research. Her current research interests include cryptography, privacy preservation, computer network security, wireless sensor network security, usable security, and HCI.



Taekyoung Kwon received his B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, Seoul, Korea, in 1992, 1995, and 1999, respectively. He is currently an Associate Professor of information at Yonsei University, Seoul, Korea. From 1999 to 2000, he was a Post-Doc Researcher at the University of California, Berkeley, CA, USA. From 2001 to 2013, he was a professor of computer engineering at Sejong University, Seoul, Korea. In 2013, he returned to Yonsei University, Seoul, Korea. His current research interests include applied cryptography, cryptographic protocol, network protocol, usable security, and human-computer interactions.