SDABS: A Secure Cloud Data Auditing Scheme Based on Blockchain and SGX

Hong Lei[1], Zijian Bao[1], Qinghao Wang[2(✉)], Yongxin Zhang[2], and Wenbo Shi[2]

[1] Oxford-Hainan Blockchain Research Institute, Chengmai 571924, China
{leihong,zijian}@oxhainan.org
[2] Department of Computer Science and Engineering, Northeastern University, Shenyang 110001, China
1286579221@qq.com, silence_yongxin@163.com, shiwb@neuq.edu.cn
https://www.oxhainan.org

Abstract. With the continuous growth of data resources, outsourcing data storage to cloud service providers is becoming the norm. Unfortunately, once data are stored on the cloud platform, they will be out of data owners' control. Thus, it is critical to guarantee the integrity of the remote data. To solve this problem, researchers have proposed many data auditing schemes, which often employ a trusted role named Third Party Auditor (TPA) to verify the integrity. However, the TPA may not be reliable as expected. For example, it may collude with cloud service providers to hide the fact of data corruption for benefits. Blockchain has the characteristics of decentralization, non-tampering, and traceability, which provides a solution to trace the malicious behaviors of the TPA. Moreover, Intel SGX, as the popular trusted computing technology, can be used to protect the correctness of the auditing operations with a slight performance cost, which excellently serves as the of the blockchain-based solution. In this paper, we propose a secure auditing scheme based on the blockchain and Intel SGX technology, termed SDABS. The scheme follows the properties of storage correctness, data-preserving, accountability, and anti-collusion. The experiment results show that our scheme is efficient.

Keywords: Blockchain · Intel SGX · Cloud storage · Data auditing · Data preserving

1 Introduction

With the development of big data, the Internet of things, 5G, and other new technologies, data have grown explosively and become a strategic resource. As the local storage and computing capacities of users are very limited, which can not meet the users' storage and computing needs, users normally store massive data in a remote cloud storage service provider with a low cost. However, cloud storage service providers are not fully trusted, and they may modify or even

Z. Zheng et al. (Eds.): BlockSys 2020, CCIS 1267, pp. 269–281, 2020.
https://doi.org/10.1007/978-981-15-9213-3_21

delete users' data for commercial interests. Besides, natural disasters (such as earthquakes and fires, etc.) may also cause damage to the data integrity. Incomplete or wrong data in the big data analysis will lead to the error of results, and even cause huge economic losses, so we need to ensure the data integrity.

The traditional data integrity auditing method needs to download the cloud data to the local for verification, which will consume a lot of bandwidth resources, and also bring huge waste of storage space and computing resources to users. A smarter approach relies on a third-party auditor (TPA) for verification (i.e., calculating the auditing task) instead of the user. But audit institutions may collude with cloud service providers to give back false results to users to obtain benefits.

Blockchain, which has the characteristics of decentralization, non-tampering, and traceability, provides a new solution to the cloud data auditing [22–24]. Some schemes [9,10] used blockchain to eliminate the TPA hypothesis and improved the reliability of data audit results. However, these schemes employed smart contracts to complete the audit of user data, which increased the computation of blockchain, with the high cost and low efficiency [25]. Other schemes [11,16] made use of the blockchain to assist in recording audit results and realized the regulatory function for malicious roles. However, it can only guarantee the responsibility after the event and cannot directly ensure the reliability of audit results.

Intel software guard extensions (SGX), as a new trusted computing technology, can provide the trusted execution environment (TEE) for applications, which protects the confidentiality and security of internal data and provides runtime security [26]. In this paper, we propose a cloud data auditing scheme based on SGX and blockchain, named SDABS. We use SGX to ensure the correct implementation of audit work, and blockchain to ensure the traceability of audit results. Finally, we analyze the security of SDABS, and implement the simulation experiment to evaluate its performance. The experiment results illustrate the effectiveness of the scheme.

The rest of the paper is organized as follows. Section 2 presents the blockchain and Intel SGX technology. Section 3 explains the related research works of the cloud data auditing. Then, we introduce the system model, the threat model, and the design goals of SDABS in Sect. 4. Our scheme is described in Sect. 5. We analyze the security of our scheme in Sect. 6, and present the simulation results in Sect. 7. We discuss the application of TEE technology in Sect. 8. Finally, we give the conclusion in Sect. 9.

2 Background

2.1 Blockchain Technology

Blockchain is the underlying technology of Bitcoin [17], which aims to provide a decentralized and tamper-proof digital ledger in an untrusted environment. Essentially, blockchain is a chain of blocks, and each block consists of many transactions. Cryptography technology is used to link the two blocks. A slight change in one block will affect blocks following it. Besides, nodes in the blockchain network jointly maintain this ledger using certain consensus algorithm (e.g., PoW [17], PBFT [18], PoS [19], etc.). Thus, it is very difficult for malicious attackers to modify the data on the blockchain, which ensures the credibility of blockchain data. One of the dilemmas of big data is how to deal with data sharing in an untrusted environment due to the doubt among different profit-driven entities [21]. In this scenario, blockchain technology can deals with this trust issue gracefully.

2.2 Intel SGX

Intel software guard extensions (SGX) is one of the most popular TEE technologies, which has been integrated into the commodity CPU of Intel [26]. It provides the new CPU instructions to help users create the secure container called enclave in an SGX-enabled platform. The confidentiality of programs in the enclave is protected by some hardware modules, even the privileged softwares (e.g., kernel, hypervisor, etc.) are malicious. Specifically, an external program cannot access the data in the enclave directly, and the data in the enclave are also encrypted until being brought to the processor.

SGX provides the remote attestation mechanism to help an enclave communicate with a remote party securely. Before information is exchanged, two parties perform the attestation protocol, which can prove that the particular code is running securely in this enclave and this enclave is on a real SGX-enabled platform. Moreover, the remote attestation mechanism can help two parties build a secure communication channel by the additional key exchange protocol. We refer the readers to read [27] for a detailed description.

3 Related Work

To ensure the integrity of data stored on an untrusted cloud server, Ateniese et al. [1] first proposed the concept of provable data possession (PDP), realizing that the user can verify the integrity of remote data without downloading. However, it made users keep online for verification, which was unfriendly to them. Then, Wang. et al. [2] introduced the concept of Third Party Auditor (TPA) into the PDP scheme, where users were liberated from the heavy burden of auditing. The introduction of TPA has brought privacy concerns, so Wang et al. [3,4] used proxy re-signature for the group users to hide the identity of the individual within the group. Wang et al. [5] adopted the homomorphic authenticable ring signature for protecting the user's privacy, but it was not suited to large-scale users due to the computation cost. Liu et al. [6] further extended the TPA hypothesis to the malicious one, pointing out that the audit scheme should be able to deal with malicious TPA. Huang et al. [7] used matrix calculation to invoke multiple TPAs for auditing, but it introduced extra and large useless calculations.

Blockchain, as the underlying technology of the digital cryptocurrency, implements the tamper-proof storage, which meets the need of auditing scheme for solving the problem of TPA. Consequently, many blockchain-based auditing schemes have emerged in recent years. Suzuki et al. [8] adopted blockchain as the information channel among users, TPA, and cloud service providers. Liu et al. [9] proposed a private IoT data auditing framework, using smart contracts to verify the data integrity. Yu et al. [10] used smart contracts instead of the TPA. Wang and Zhang [11] proposed a Blockchain and Bilinear mapping based Data Integrity Scheme (BB-DIS) for large-scale IoT data. Huang et al. [12] and Hao et al. [13] put the blockchain nodes as the auditor to verify the data integrity. Indeed, these schemes increased the computing overhead of blockchain. Zhang et al. [14] proposed the first certificateless public verification scheme against procrastinating auditors (CPVPA) by using blockchain technology to address the procrastinating attack of auditors. Xu et al. [15] proposed an arbitrable data auditing protocol, adopting the commutative hash technique, for the dishonest parties. Lu et al. [16] adopted Hyperledger Fabric [20] as a platform for the auditing and proposed two algorithms for choosing TPA. Blockchain technology is better suited as a tool for accountability in hindsight, but a more secure and reliable tool is needed at runtime when verifying the integrity.

4 Problem Statement

In this section, we describe the system model of SDABS and then describe the relevant threat model. Finally, we indicate the design goals of our scheme.

4.1 System Model

We show the architecture of our scheme in Fig. 1, which consists of four roles: Users, Cloud Storage Service Provider (CSP), Third Party Auditor (TPA), and Blockchain (BC). Users are resource-constrained individuals, who are unable to store big data and perform onerous auditing tasks. The CSP provides storage service for uses and responds to auditing challenges of users' data. The TPA is an auditing service provider, which maintains an SGX-enabled platform to perform auditing operations. The BC is a blockchain system (e.g., Bitcoin, Ethereum, Hyperledger Fabric), which serves as the bridge between users and the TPA. To ensure the quality and integrity of the data, users synchronize auditing requests to the BC. The TPA will obtain the requests from the blockchain and send auditing challenges to the CSP to check the integrity of users' data. Finally, the auditing results will be synchronized to the blockchain network, which can be acquired by users.

4.2 Threat Model

In our assumption, the CSP is an unreliable party. It will try to remove the less-used data of users to reduce the cost of storage. More seriously, the CSP may

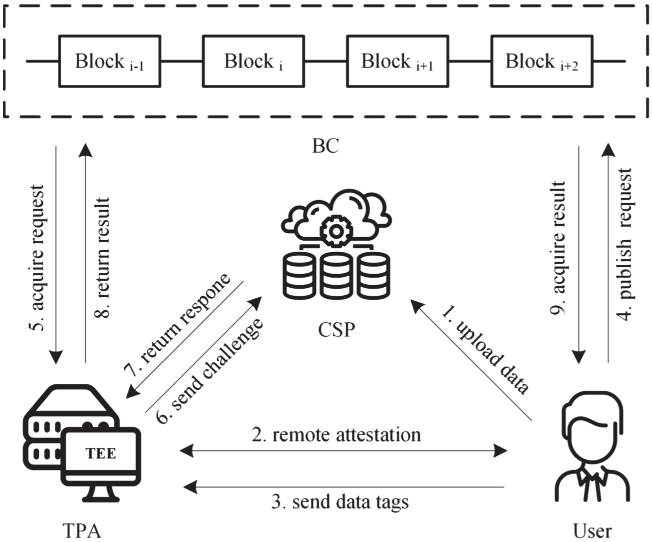

Fig. 1. The architecture diagram of SDABS.

tamper with partial data for some malicious purpose. We assume that the TPA is untrusted, and can control the communication of the enclave or terminate the enclave. Specially, the TPA may collude with the CSP to fraud users. Moreover, we assume the SGX technology is normally trustworthy. Security and confidentiality of data can be protected by the enclave and the remote attestation can prove the reliability of the remote SGX-enabled platforms. We also assume that over 50% of nodes in the BC are honest and hence the blockchain can not be arbitrary tempered.

4.3 Design Goals

As SDABS is designed to operate with untrusted entities, it is designed to achieve the following goals, including storage correctness, data-preserving, accountability, and anti-collusion.

(1) *Storage Correctness*: the data is stored by the CSP. For the data challenge proposed by the TPA, the CSP can only pass the auditing if it provides correct data proof.
(2) *Data-Preserving*: during the auditing process, based on the tag from the user and the data proof from the CSP, the TPA cannot infer the user's real data.
(3) *Accountability*: in our scheme, once an entity violates the agreement, it will be found and blamed.
(4) *Anti-Collusion*: it is difficult for the CSP and the TPA to deceive users by colluding.

5 The Proposed Scheme

5.1 Preliminaries

The bilinear pairing plays a significant role in the data auditing schemes [2–4,8]. We will introduce the properties of the bilinear pairing briefly.

Bilinear Pairing. Given two multiplicative cyclic groups of large prime order q, G_1 and G_T. Let g_1 and g_2 be the generators of G_1 and G_T, respectively. A cryptographic bilinear map is a map e: $G_1 \times G_1 \rightarrow G_T$ satisfying the three properties as follows.

(1) *Bilinear*: for $\forall P, Q \in G_1$ and $\forall x, y \in Z_q^*$, $e(P^x, Q^y) = e(P, Q)^{xy}$;
(2) *Non-degenerate*: $\exists g_1 \in G_1$, then $e(g_1, g_1) \neq 1$;
(3) *Computable*: the map e can be computed efficiently.

5.2 The Process of SDABS

SDABS includes three phases in a typical circumstance: tag generation, storage, and auditing.

Tag Generation Phase. In order to audit data correctly, the user needs to generate the tags of the data. The user first generates the relevant parameters as follows. Two multiplicative cyclic groups are denoted as G_1 and G_T, respectively. p is the prime order of the groups and g is the generator of the group G_1. Thus, the cryptographic bilinear map e can be expressed as $G_1 \times G_1 \rightarrow G_T$. $H_1 : \{0,1\}^* \rightarrow G_1$ is the hash function that maps a string data to a point in G_1. Similarly, $H_2 : G_1 \rightarrow Z_q^*$ denotes the other hash function, which maps a point in G_1 to a point in Z_q^*. The user selects a secret key $x \in Z_q^*$ randomly and then calculates the public key $PK = g^x$. It is worth noticing that the part of parameters need to be published to all parties, including e, H_1, H_2, PK. Then the user splits the data into data blocks denoted as $D = \{d_1, d_2, \cdots, d_n\}$ and chooses a random element $r \in G_1$ to calculate the tag $\sigma_i = (H(d_i) \cdot r^{(d_i)})^x$ for each data block d_i. At this point, the user holds the data blocks $\{d_1, d_2, \cdots, d_n\}$ and the corresponding tags $\{\sigma_1, \sigma_2, \cdots, \sigma_n\}$.

Storage Phase. This phase is divided into two steps: (1) The user uploads data to the CSP and deletes the local data. (2) The user sends the corresponding tags to the TPA and removes them from the local. The details of the two steps are as follows.

Step1. The user sends a request to the CSP for storage service and then uploads the data blocks $\{d_1, d_2, \cdots, d_n\}$. After the successful upload, the user can delete the local data to save storage.

Step2. To establish trust with the enclave in the TPA platform, the user must receive assurance that the enclave is running in a real SGX-enabled platform and that the auditing codes are correctly loaded in the enclave. This is implemented through remote attestation provided by SGX. Note that the remote attestation protocol has the self-defined data field, which can be used to build a secure communication channel by the key exchange protocol (e.g., Diffie-Hellman key agreement method [33]). In this step, the user first employs the remote attestation protocol to verify the enclave in the TPA platform and to build a secure channel with it. Thus, if the user transmits the message through the secure channel, the message can only be decrypted by the enclave. Then, the user sends the tags $\{\sigma_1, \sigma_2, \cdots, \sigma_n\}$ to the TPA using the secure channel. The tags correspond to the data blocks in the Step1 of this phase. The tags will be delivered to the enclave and be stored securely. The enclave then returns the storage result to the user by the secure channel. Finally, the user removes the local tags to save storage.

Auditing Phase. In this phase, the user issues the auditing request in the BC. Then, the TPA obtains the request from the BC and constructs a challenge in the enclave to the CSP. When the CSP receives the challenge, it computes the response and sends it to the TPA. After verification, the TPA will publish the auditing results in the BC.

Step1. The user issues the auditing request in the form of a blockchain transaction and synchronizes the transaction to the blockchain network. A request includes the identifiers of the data blocks to be audited such as $\{id_2, id_9, \cdots, id_{20}\}$ (id_i is the unique identifier of d_i), the expected time T, the rewards R, and the signature of the request. Due to the user has built the secure channel through the remote attestation, the user can use the exchanged key to encrypt the request as the signature, which can only be verified by the enclave.

Step2. The TPA real-time synchronizes the blockchain to the enclave. The enclave identifies the relevant transactions in the blockchain and verifies the signatures. It then acquires the identifiers of the data blocks such as $\{id_2, id_9, \cdots, id_{20}\}$, and the corresponding tags $\{\sigma_2, \sigma_9, \cdots, \sigma_{20}\}$. To construct the challenge, the also need to generate a random number for each data block, which is denoted as v_i. The challenge $C = (\{id_2, id_9, \cdots, id_{20}\}, \{\sigma_2, \sigma_9, \cdots, \sigma_{20}\}, \{v_2, v_9, \cdots, v_{20}\})$ will be deliver to the CAP and be sent to the CSP.

Step3. After receiving C, the CSP first verifies the tags $\{\sigma_2, \sigma_9, \cdots, \sigma_{20}\}$ to ensure that C is initiated by the enclave. Then the CSP computes $t_i = d_i \cdot v_i$ for each data blocks and aggregates all results as the respond $respond_{CSP}$. As the above example, $respond_{CSP} = t_2 + t_9 + \cdots + t_{20}$. The CSP final returns $respond_{CSP}$ to the TPA.

Step4. The TPA delivers $respond_{CSP}$ to the enclave, and the enclave calculates $verification_{TPA} = {\sigma_2}^{v_2} \cdot {\sigma_9}^{v_9} \cdots {\sigma_{20}}^{v_{20}}$. Then the enclave computes Formula (1) to verify $respond_{CSP}$, where I indicates the number of the audited data blocks. If the equation is true, the enclave will hold that the CSP stores the user's data correctly and completely. Otherwise, the CSP will be regarded as dishonest. After verification, the enclave returns the auditing report, which includes the auditing result and the signature of the result. The signature is generated by the exchanged key, similar to the process in the Step1 of this phase. Finally, the TPA constructs a blockchain transaction and synchronize it to publish the report.

$$e(\prod_{i \in I} {\sigma_i}^{v_i}, g) = e(\prod_{i \in I} H_1(d_i)^{v_i} \cdot r^{\sum_{i \in I} d_i \cdot v_i}, PK) \tag{1}$$

Step5. The user acquires the auditing report from the blockchain to perceive whether its data is corrupted. As the blockchain is public and tamper-proof, the TPA can provide the proof (i.e., the transaction with auditing request and the transaction with the auditing result in the blockchain) to the user and gets the rewards R. Moreover, the user can refuse to pay the rewards, if the auditing time exceeds the expected time T. It's easy to calculate the auditing time because each block in the blockchain involves the timestamp that denoting its confirmed time.

6 Security Analysis

In this section, we analyze the following security features of SDABS, namely: storage correctness, data-preserving, accountability and anti-collusion.

Storage Correctness: If all roles execute the protocol correctly and the data stored in the CSP is integrated, the storage correctness can be ensured.

Proof. We prove the correctness of our scheme as follows:

$$\begin{aligned}
&e(\prod_{i \in I} {\phi_i}^{v_i}, g) \\
&= e(\prod_{i \in I} (H(d_i) \cdot r^{d_i})^{x v_i}, g) \\
&= e(\prod_{i \in I} (H(d_i) \cdot r^{d_i})^{v_i}, g^x) \\
&= e(\prod_{i \in I} H(d_i)^{v_i} \cdot r^{\sum_{i \in I} d_i v_i}, g^x) \\
&= e(\prod_{i \in I} H(d_i)^{v_i} \cdot r^{\sum_{i \in I} d_i v_i}, PK)
\end{aligned} \tag{2}$$

Data-Preserving: The TPA cannot recover the real data from the auditing information.

Proof. It is difficult for the TPA to recover user's data by computing $\prod_{i \in I} {\phi_i}^{v_i}$ or $\sum_{i \in I} d_i v_i$, where $\phi_i = (H(d_i) \cdot r^{(d_i)})^x$. The original data d_i is protected by $H : \{0,1\}^* \rightarrow G_1$, r and the user's secret key x. Besides, in the $\sum_{i \in I} d_i v_i$,All of the d_i is blinded by v_i generated by the enclave. Based on the hardness of Computational Diffie-Hellman (CDH) problem in G_1, it is hard for probabilistic polynomial time adversary to compute d_i.

Accountability: The party who breaks the protocol can be blamed.

Proof. In SDABS, the results of auditing will be sent to the blockchain by the SGX. SGX provides a secure runtime environment that makes it difficult for an adversary to tamper with its contents, making data transmitted to the blockchain believable. Blockchain is a kind of decentralized database, which uses cryptography technology to realize the non-tampering characteristics, which ensures the credibility of the data in the blockchain. Both off-chain and on-chain ensure that the data is credible, which ensures the credibility of the audit results kept on the blockchain. Any breach of the agreement by either party can be determined by the audit results. In this way, SDABS realizes the accountability.

Anti-Collusion: The TPA with SGX can hardly collude with the CSP to fraud users.

Proof. If the TPA wants to collude with the CSP to defraud the user, either the CSP falsifies the data or the TPA falsifies the results. For the former, CSP needs to construct data d'_i without the user's private key, so as to satisfy $e(\prod_{i\in I} H(d'_i)^{v_i} \cdot r^{\sum_{i\in I} d'_i v_i}, PK) = e(\prod_{i\in I} {\phi_i}^{v_i}, g)$. Similarly, based on the hardness of Computational Diffie-Hellman (CDH) problem in G_1 and the collision resistance of hash function, it is hard for probabilistic polynomial time adversary to realize this equation. Hence, the CSP is incapable of forge the results. For the latter, the auditing process is carried in the enclave, including the storage of data tags, the generation of random numbers, and the verification of proof, which are not controlled by the TPA. Therefore, the TPA cannot falsify the results.

7 Performance Evaluation

We implement the simulation experiment to evaluate the performance of our scheme. The three roles (i.e., the user, the CSP, and the TPA) run on the same PC with Ubuntu 16.04 LTS operating system, a 2.40 GHz Intel(R) Core(TM) CPU i7-8700T, and 16GB RAM. Specially, the program of the TPA is loaded in an enclave on the PC to perform, and the programs of the user and the CSP are performed in the normal memory.

Figure 2 illustrates the computation time of the three roles with different numbers of audited data blocks. To show the results clearly, we fix the size of the data block to 1 KB. We can see that the computation time of all roles is increased linearly with the numbers of audited data blocks. Specially, the user has a higher computation time overhead than other roles, because the user needs to generate the tags for each data block, which contains map-to-point hash functions involving expensive computation. The tags of data blocks are only calculated once in the whole scheme, thus the time overhead is acceptable.

We also test the performance overhead caused by the use of SGX. Specifically, we compare the computation time of the auditing operations executed in the enclave with the time of the same operations performed out of the enclave.

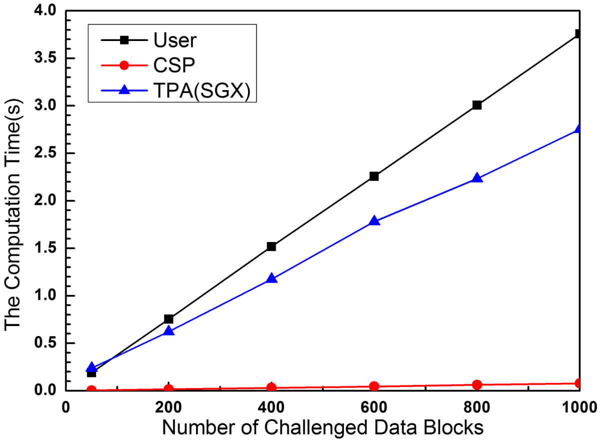

Fig. 2. The computation time of the different roles.

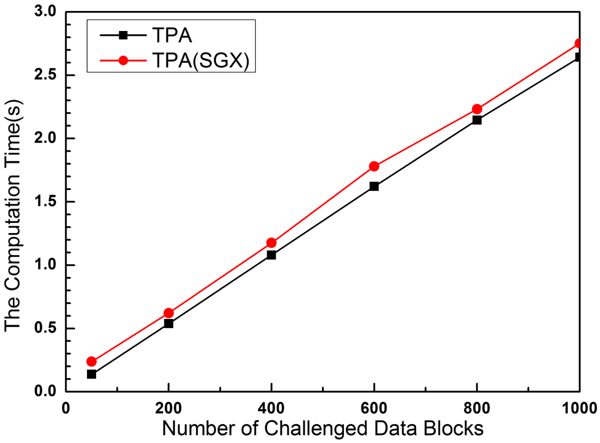

Fig. 3. The performance of the auditing operations performed in and out of the SGX enclave.

Figure 3 shows that the former is slightly more than the latter, which is because the programs in the enclave need to execute the additional encryption, decryption, and scheduling operations. However, all operations are limited on time scales of seconds, which is applicable to real scenarios. Thus, it is an efficient way to protect the audit process using the SGX technology.

8 Discussion

SGX can be applied into many fields, smart grid, blockchain, etc. Li et al. [30] leverages the SGX to protect the privacy of users when grid utilities are executing rich functionalities on customers' private data. Lind et al. [31] adopt TEE as a safe treasury to execute off-chain transactions asynchronously, prevent misbehavior of parties and maintain collateral funds. Bentov et al. [32] design a kind of exchange that offers real-time cross-chain cryptocurrency trades and secure tokenization of assets based on the SGX. And many application can be ported to the other TEE technologies if conditions permit. Arm TrustZone technology is also a famous TEE technology, offering an efficient, system-wide approach to security with hardware-enforced isolation built into the CPU [28]. However, TrustZone is focused on the mobile side [29], not the server side. TrustZone is a good choice in the future if edge computing can be used to offload the data integrity auditing to the mobile side.

9 Conclusion

The quality and reliability of data are greatly significant in the age of big data. In this paper, we propose SDABS, a new cloud data auditing scheme, based on the SGX and the blockchain technology. The scheme employs SGX to improve the reliability and stability of the auditing process and to eliminate the trust to the TPA. By introducing the blockchain, SDABS implements the accountability, which can trace the inappropriate behavior of any entities. We analyze the security of SDABS, which demonstrates that the proposed scheme has the features of storage correctness, data-preserving, accountability, and anti-collusion. The performance evaluation shows that our scheme is feasible and efficient.

Acknowledgements. This study is supported by Oxford-Hainan Blockchain Research Institute, the National Science Foundation of China (No. 61472074, U1708262) and the Fundamental Research Funds for the Central Universities (No. N172304023).

References

1. Ateniese, G., et al.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security 2007, pp. 598–609. ACM (2007)
2. Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling public verifiability and data dynamics for storage security in cloud computing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 355–370. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04444-1_22
3. Wang, B., Li, H., Li, M.: Privacy-preserving public auditing for shared cloud data supporting group dynamics. In: 2013 IEEE International Conference on Communications (ICC), pp. 1946–1950. IEEE (2013)
4. Wang, B., Li, B., Li, H.: Panda: public auditing for shared data with efficient user revocation in the cloud. IEEE Trans. Serv. Comput. **8**(1), 92–106 (2015)
5. Wang, B., Li, B., Li, H.: Oruta: privacy-preserving public auditing for shared data in the cloud. IEEE Trans. Cloud Comput. **2**(1), 43–56 (2014)
6. Liu, C., et al.: Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates. IEEE Trans. Parallel Distrib. Syst. **25**(9), 2234–2244 (2014)

7. Huang, K., Xian, M., Fu, S., Liu, J.: Securing the cloud storage audit service: defending against frame and collude attacks of third party auditor. IET Commun. **8**(12), 2106–2113 (2014)
8. Suzuki, S., Murai, J.: Blockchain as an audit-able communication channel. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 516–522. IEEE (2017)
9. Liu, B., Yu, X., Chen, S., Xu, X., Zhu, L.: Blockchain based data integrity service framework for IoT data. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 468–475. IEEE (2017)
10. Yu, H., Yang, Z., Sinnott, R.: Decentralized big data auditing for smart city environments leveraging blockchain technology. IEEE Access **7**, 6288–6296 (2019)
11. Wang, H., Zhang, J.: Blockchain based data integrity verification for large-scale IoT data. IEEE Access **7**, 164996–165006 (2019)
12. Huang, P., Fan, K., Yang, H., Zhang, K., Li, H., Yang, Y.: A collaborative auditing blockchain for trustworthy data integrity in cloud storage system. IEEE Access **8**, 94780–94794 (2020)
13. Hao, K., Xin, J., Wang, Z., Wang, G.: Outsourced data integrity verification based on blockchain in untrusted environment. World Wide Web **23**(4), 2215–2238 (2020). https://doi.org/10.1007/s11280-019-00761-2
14. Zhang, Y., Xu, C., Lin, X., Shen, X.S.: Blockchain-based public integrity verification for cloud storage against procrastinating auditors. IEEE Trans. Cloud Comput. (to be published). https://doi.org/10.1109/TCC.2019.2908400
15. Xu, Y., Ren, J., Zhang, Y., Zhang, C., Shen, B., Zhang, Y.: Blockchain empowered arbitrable data auditing scheme for network storage as a service. IEEE Trans. Serv. Comput. **13**(2), 289–300 (2020)
16. Lu, N., Zhang, Y., Shi, W., Kumari, S., Choo, K.: A secure and scalable data integrity auditing scheme based on hyperledger fabric. Comput. Secur. **92**, 101741 (2020)
17. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
18. Castro, M., Liskov, B.: Practical byzantine fault tolerance. In: The Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), USA, vol. 99, pp. 173–186 (1999)
19. Seijas, P.L., Thompson, S.J., McAdams, D.: Scripting smart contracts for distributed ledger technology. IACR Cryptology ePrint Archive (2016)
20. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of ACM 13th EuroSys Conference (EuroSys), USA (2018)
21. Xu, C., Wang, K., Li, P., Guo, S., Luo, J., Ye, B., Guo, M.: Making big data open in edges: a resource-efficient blockchain-based approach. IEEE Trans. Parallel Distrib. Syst. **30**(4), 870–882 (2019)
22. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **14**(4), 352–375 (2018)
23. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: architecture, consensus, and future trends. In: International Congress on Big Data (2017), pp. 557–564 (2017)
24. Dai, H., Zheng, Z., Zhang, Y.: Blockchain for internet of things: a survey. IEEE Internet Things J. **6**(5), 8076–8094 (2019)
25. Huang, Y., Kong, Q., Jia, N., Chen, X., Zheng, Z.: Recommending differentiated code to support smart contract update. In: International Conference on Program Comprehension (2019), pp. 260–270 (2019)

26. Intel. 2017. Software Guard Extensions (Intel SGX) (2017). https://software.intel.com/en-us/sgx
27. Costan, V., Devadas, S.: Intel SGX Explained. IACR Cryptology ePrint Archive (2016)
28. ARM. Arm TrustZone Technology. https://developer.arm.com/ip-products/security-ip/trustzone
29. Kwon, D., Seo, J., Cho, Y., Lee, B., Paek, Y.: PrOS: light-weight privatized secure OSes in ARM TrustZone. IEEE Trans. Mob. Comput. **19**(6), 1434–1447 (2020)
30. Li, S., Xue, K., David, W., Yue, H., Yu, N., Hong, P.: SecGrid: a secure and efficient SGX-enabled smart grid system with rich functionalities. IEEE Trans. Inf. Forensics Secur. **15**, 1318–1330 (2020)
31. Lind, J., Naor, O., Eyal, I., Florian Kelbert, F., et al.: Teechain: a secure payment network with asynchronous blockchain access. In: Proceedings of the 27th ACM Symposium on Operating Systems Principles(SOSP), pp. 63–79 (2019)
32. Bentov, I., Ji, Y., Zhang, F., Breidenbach, L., Daian, P., Juels, A.: Tesseract: real-time cryptocurrency exchange using trusted hardware. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 1521–1538 (2019)
33. Rescorla, E..: Diffie-Hellman key agreement method. RFC 2631 (1999)