Blockchain-Based Private Provable Data Possession

Huaqun Wang, Qihua Wang, and Debiao He

Abstract—Remote data secure storage is of crucial importance in cloud computing. In order to check remote data integrity, an important paradigm PDP (i.e., provable data possession) is proposed. All the existing PDP schemes make use of RSA or bilinear pairings. One large file has to be divided into a great many of blocks. For example, 1T (Terabit) file has to be divided into 1.0737×10^9 blocks (RSA where the length of the index is 1024 bits) or 6.8719×10^9 blocks (bilinear pairings where the order of the elliptic curve is 160 bits). The huge computation cost and communication cost incurs the inefficient PDP implementation. In other words, they are not practical. In order to solve the problem, we propose a new PDP model: blockchain-based private PDP. The new concept makes use of blockchain which is the core of cryptocurrency. For the new concept, the paper formalizes its system model and security model. Then, a concrete blockchain-based private PDP scheme is designed by making use of blockchain and RSA. The proposed blockchain-based private PDP scheme is provably secure. At the same time, we also analyze its performance from two parts: theory analysis and implementation prototype. Our analysis shows that the proposed PDP scheme is secure, efficient and practical.

Index Terms—Cloud computing, provable data possession, blockchain, RSA

1 Introduction

In these years, cloud computing has widely been used in many fields. Cloud computing regards the information processing as a service, such as outsourced storage, outsourced computing, *etc*. By making use of cloud computing, the clients are relieved of the burden for storage management, universal data access with independent geographical locations. By using the cloud computing, it is not necessary that the clients purchase the hardware, software, and personnel maintenances, *etc*. Thus, it saves more capital for them. The clients can focus on their own professional work. Thus, cloud computing attracts more personal and enterprise interests. Data outsourcing entails the data security risks, such as, confidentiality, integrity and availability of data, *etc*. Remote data integrity checking is an important primitive in cloud computing. The integrity checking scheme must be efficient in order to make it suitable for capacity-limited end devices. Generally, cloud can be categorized in three types: public cloud, private cloud, and hybrid cloud. Public cloud is an external or publicly available cloud environment that can be accessed by any user on a pay-per-use model. Public cloud services are attractive because they are inexpensive, they are easy to set up, they scale well, and they make an efficient use of resources.

- *H. Wang is with the Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and also with the Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China. E-mail: wanghuaqun@aliyun.com.*
- *Q. Wang is with the School of Medical Information Engineering, Jining Medical University, Rizhao 272067, China. E-mail: wd19791209@163.com.*
- *D. He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China. E-mail: hedebiao@163.com.*

Manuscript received 23 Apr. 2019; revised 20 Aug. 2019; accepted 22 Oct. 2019. Date of publication 28 Oct. 2019; date of current version 1 Sept. 2021.
(Corresponding author: Huaqun Wang.)
Digital Object Identifier no. 10.1109/TDSC.2019.2949809

Blockchain is the core technique of cryptocurrency. It is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a merkle tree root hash). The basic structure of blockchain is shown in Fig. 1. In the structure, Prev-Hash denotes the hash value of the previous block. Timestamp denotes the time when the block is generated. Tx-Root denotes the storage structure of the transactions. Nonce denotes a random value. Along with the rapid development of blockchain, it has been used in many other fields, such as smart grid, electronic evidence, *etc*.

1.1 Related Work

Because the malicious cloud server may corrupt the clients' data, remote data integrity checking is an important security problem since the clients' massive data is outside their control. In 2007, provable data possession (PDP) paradigm was proposed by Ateniese et al. [1]. PDP is a probabilistic checking scheme for remote data integrity. First, they designed two provably secure PDP schemes. Second, Ateniese et al. proposed the dynamic PDP scheme although it does not support insert operation [2]. In order to support the insert operation, in 2009, Erway et al. proposed a full-dynamic PDP scheme from the authenticated flip table [3]. At the same time, Sebé et al. also proposed the PDP concept [4]. Hao et al. also proposed a privacy-preserving PDP protocol with data dynamics and public verifiability [5]. PDP allows a verifier to verify the remote data integrity without retrieving or downloading the whole data. It is a probabilistic proof of possession by sampling a random set of blocks from the server. PDP drastically reduces I/O costs. The verifier only maintains small metadata to perform the remote data integrity checking. Usually, metadata is "data that

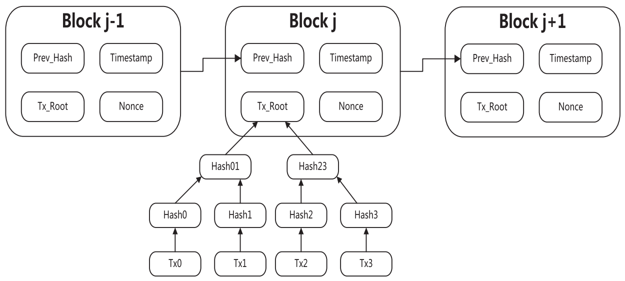


Fig. 1. The basic structure of blockchain.

provides information about other data". In this paper, metadata denotes the auxiliary data that is used to check remote data integrity. In other words, it mainly denotes the block tags besides of file properties and block properties. In 2008, Shacham presented the first proof of retrievability (POR) scheme with provable security [6]. In POR, the verifier can check the remote data integrity and retrieve the remote data at any time. Their scheme is based on the bilinear pairing which comes from the elliptic curve cryptography. Many researchers design PDP schemes in different application environments. In 2012, Wang proposed the security model and concrete scheme of proxy PDP in public clouds [16]. At the same time, Zhu *et al.* proposed the cooperative PDP in the multi-cloud storage [7]. There exist many other research results in this field [8], [9], [10], [11], [12], [13]. From the role of the PDP verifier, PDP can be classified into two categories: private PDP and public PDP. In the phase *CheckProof* of private PDP, some private information is needed. In other words, if the verifier does not have the private information, it cannot perform the *CheckProof* phase. On the contrary, private information is not needed in the phase *CheckProof* of public PDP. Private PDP is necessary in some cases [14], [15], [16]. In this paper, our proposed blockchain-based private PDP scheme pertains to private PDP.

In 2008, bitcoin was invented by S. Nakamoto [30]. Bitcoin is the first cryptocurrency which takes advantage of the new technique-blockchain. Now, the market capitalization of bitcoin has reached to about $50000000000. Blockchain is the core technology behind several modern cryptocurrencies, such as bitcoin, Monero [17], [18], [19], zerocoin [20], [21], *etc*. Blockchain is a very new network paradigm which makes use of cryptography, hash function and proof of work, *etc*. It also solves the double-spending problem without trusted third party. At the outset, bitcoin is designed to realize the anonymity for the vendor and the purchaser in the transactions. Regretfully, some research results show that bitcoin transaction cannot satisfy the clients' anonymity [22], [23]. In order to get better anonymity, some new cryptocurrencies, such as Mixcoin [24], Monero, Zerocoin, *etc*, have been proposed. Based on the properties of bitcoin, bitcoin has been used in privacy preserving payment mechanism for vehicle-to-grid [26] and vehicular ecosystem [27], [28], [29]. Blockchain can also be used to many different fields, such as delegated computation [30], key agreement protocol [31], incentive mechanism [32], [33], internet of things [34], etc. In 2018, Liang et al. discussed how blockchain technology can be used to enhance the robustness and security of the power grid [35]. Aitzhan et al. studied the transaction security in decentralized smart grid energy trading without reliance on the trusted third party [36].

1.2 Motivation

The existing PDP schemes can be classified into two types:

1) PDP schemes which make use of the difficult problems of large integer factoring and KEA1-r.
2) PDP schemes which make use of bilinear pairings.

The existing PDP schemes are unpractical because of their computation and communication costs. In order to realize the remote data integrity checking, the processed file F will be divided into many blocks. At the same time, the corresponding metadata will also be generated. Let the uploaded file F be divided into n blocks $(F_1, F_2, \ldots, F_n)$. For the first type of PDP schemes, the tag T_i is calculated as $T_i = (h(W_i)g^{F_i})^x \bmod N$. In the generation of tags, h is a hash function. The symbols N, QR_N, g, x denote the big safe composite number, quadratic residue group modulo N, the generator of QR_N, and the client's private key, respectively. Since the block F_i is taken as the index of g, the time cost will increase rapidly along with the length increase of the block F_i. The up-to-date RSA makes use of the security parameter $k = 1024$ bits. When $|F_i| = 1024$ bits, 1T (Terabit) file F will be divided into 1.0737×10^9 blocks. The computation cost is $2.9042 * 10^8$ ms (millisecond) which is huge. On the other hand, the generated tags will also cost huge communication cost. For the second type of PDP schemes, the tag T_i is calculated as $T_i = x(H(W_i) + F_iP)$. The symbols H, P denote hash function and the generator of the elliptic curve group, respectively. The state-of-the-art elliptic curve cryptography (for bilinear pairings) makes use of the elliptic curve whose order is 160 bits. Thus, 1T bits file F will be divided into 6.8719×10^9 blocks. The computation cost is $1.4885 * 10^{10}$ ms which is much larger than type (1). The tags are much larger than the file F. The generated tags will also incur more critical communication cost.

Being faced with the critical computation cost and the critical communication cost, the existing PDP schemes are not practical for large file F. Usually, clients will upload many large files to cloud server. These clients include film & television corp., genome corp., large hospital, *etc*. In order to ensure the remote data integrity, PDP is necessary. Regretfully, the existing PDP schemes cannot be used for these large files.

The large number of file blocks overwhelmingly decreases the efficiency of PDP schemes. In order to improve the efficiency, we must reduce the number of blocks for large files. The existing PDP schemes (including Type (1) and type (2)) cannot solve the problem. It motivates us to write the paper. By making use of the properties of blockchain, we propose the new concept of blockchain-based private PDP. Our blockchain-based private PDP scheme is efficient and practical for large files.

1.3 Contributions

In order to make the remote data integrity scheme more efficient and practical, we make use of blockchain technology to construct blockchain-based private PDP scheme. Our contributions can be listed below:

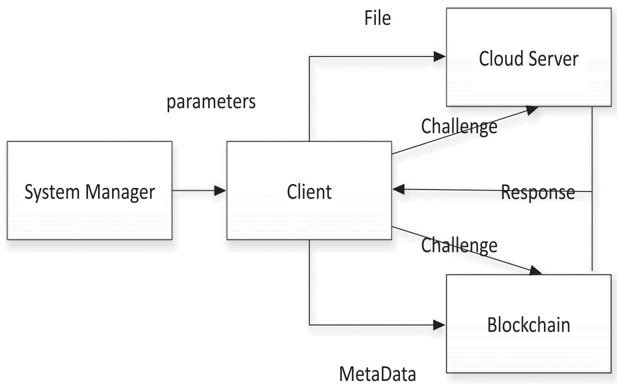


Fig. 2. The System Model of Blockchain-Based private PDP.

1) We propose the PDP study method by making use of blockchain for the first time. Namely, we propose the new concept of blockchain-based private PDP.
2) We formalize the system model and security model of blockchain-based private PDP.
3) Based on RSA, quadratic residue group modulo the big composite number N and blockchain, we design the concrete blockchain-based private PDP scheme.
4) For the proposed scheme, we give the security proof, efficiency analysis and implementation.
5) Our scheme can also realize the anonymity for the clients according to the their security requirements.

1.4 Paper Organization

The rest of the paper is organized as follows. Section 2 formalizes the system model and security model of blockchain-based private PDP. Section 3 presents our blockchain-based private PDP scheme. Section 4 evaluates the security of the proposed blockchain-based private PDP scheme. Section 5 evaluates the performance of the proposed blockchain-based private PDP scheme. Finally, Section 6 concludes the paper.

2 System Model and Security Model of Blockchain-Based Private PDP

This section presents the system model and security model of blockchain-based private PDP scheme. We also give the formal definition of blockchain-based private PDP scheme. A blockchain-based private PDP scheme comprises four different entities: *Client*, *Cloud Server*, *Blockchain* and *System Manager*. The interactions among them are illustrated in Fig. 2. They are described as follows:

1) *Client*: an entity, which has massive data which will be stored on the cloud for maintenance and computation, can be either individual consumer or corporation. It also generates the challenge and sends it to *Cloud Server* and *Blockchain*. Then, it verifies the received responses which come from *Cloud Server* and *Blockchain*. If the response can pass the verification, the remote data is thought to be integer; otherwise, the remote data is thought to be not integer. *Client* also takes the role of data verifier.
2) *Cloud Server*: an entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain *Client*'s data. Upon receiving *Client*'s challenge, *Cloud Server* generates the corresponding data and sends it to *Client*.
3) *Blockchain*: an entity, which will store some metadata for *Client*. Based on the data irreversibility on the blockchain, these metadata can be used to prove the remote data integrity. Upon receiving *Client*'s challenge, *Blockchain* generates the corresponding metadata and sends it to *Client*.
4) *System Manager*: an entity, which generates the system parameters, makes the public parameters public in the system and sends the secret parameters to the corresponding entities by the secret channel.

First, we give the formal definition of blockchain-based private PDP scheme. Then, we present the security requirements and formal security model of blockchain-based private PDP scheme.

Definition 1 (Blockchain-Based Private PDP). *A blockchain-based private PDP scheme consists of six phases:* KeyGen, SignBlock, Challenge, GenProof, CheckProof, *and* Compensation. *They are formally described as the following.*

1) KeyGen*: Input the security parameter k, this phase outputs the system parameters which includes the public parameters and secret parameters. Public parameters are made public in the system and secret parameters are sent to the corresponding entities by using the secret channel.*
2) SignBlock*: Input the public parameters, some secret parameters, and the file which will be uploaded, this phase outputs the metadata which corresponds to the uploaded file. Then, it uploads the file to* Cloud Server *and sends the metadata to* Blockchain*, respectively.*
3) Challenge*: In order to check the remote data integrity,* Client *chooses some parameters and sends them to* Cloud Server *and* Blockchain*, respectively.*
4) GenProof*: Upon receiving the challenge from* Client*,* Cloud Server *computes some values based on the stored data of* Client*. Then,* Cloud Server *sends these values to* Client *as the response.*
5) CheckProof*: Upon receiving the response from* Cloud Server*,* Client *queries* Blockchain *and gets some metadata. By making use of the response and the metadata,* Client *checks whether the remote data integrity is verified.*
6) Compensation*: When some data are broken,* Client *interacts with* Cloud Server *and gets the compensation from* Cloud Server.

A practical blockchain-based private PDP scheme must be efficient based on the communication and computation cost. In other words, the computation cost and communication cost must be as low as the practical efficiency requirements. Besides of the efficiency requirements, a practical blockchain-based private PDP scheme must satisfy the security requirements below:

1) When the stored data is kept integer, *Cloud Server* can pass *Client*'s remote data integrity checking. In other words, it satisfies the *Correctness*.
2) *Client* can perform the blockchain-based private PDP scheme without the local copy of the file(s) to be

TABLE 1
Notations and Descriptions

Notations	Descriptions
N	The modulo of RSA which is the product of two big primes
p, q	Two safe primes where $N = p \times q$
p', q'	Two safe primes where $p = 2p' + 1$, $q = 2q' + 1$
QR_N	The set of quadratic residues modulo N which is also the unique cyclic subgroup of $\mathbb{Z}_N^*$ of order $p'q'$.
$\phi(N)$	Euler function
g	The generator of QR_N
$\mathbb{Z}_N^*$	The set of elements which have the inversions in $\mathbb{Z}_N$
F	The stored file which is divided into n blocks, i.e., $F = (F_1, F_2, \ldots, F_n)$
H	Cryptographic hash functions
f	Pseudo-random function
π	Pseudo-random permutation
(N, e)	Public key of client
d	Private key of client
f_i	$f_i = F_i (\bmod p'q')$
T_i	$T_i = g^{f_i} (\bmod N)$
σ	The signature on $(T_1, T_2, \ldots, T_n)$
chal	The challenge $chal = (c, k_1, k_2)$ where $1 \leq c \leq n, k_1, k_2 \in \mathbb{Z}_N^*$
Blockchain	Store $(T_1, \ldots, T_n)$ and σ
Cloud Server	Store $(F_1, \ldots, F_n)$

checked. At the same time, it can also perform the scheme many times.

3) If the challenged blocks are integer, *Cloud Server*'s response can not pass the verification only with negligible probability. Dishonest *Client* cannot make a false charge against honest *Cloud Server*.
4) When some data are modified or lost, *Client* can get the compensation. The others cannot get the compensation by passing themselves off as *Client*.

In order to solve the above security requirements, we give the formal security models of blockchain-based private PDP scheme below.

Definition 2 (Unforgeability). *When the challenged data are modified or lost, the probability that the (probabilistic polynomial time) adversary $\mathcal{A}$ (i.e., dishonest* Cloud Server*) can pass the verification is negligible. In this case, our blockchain-based private PDP scheme satisfies the property of unforgeability. Formally, the game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$ is given below:*

1) Setup*: The challenger $\mathcal{C}$ runs $KeyGen(1^k)$ and gets the corresponding parameters. It send the public parameters to $\mathcal{A}$ and keeps confidential the secret parameters.*
2) First-Phase Queries*: The adversary $\mathcal{A}$ adaptively makes* Hash *queries and* SignBlock *queries to the challenger $\mathcal{C}$. Correspondingly, $\mathcal{C}$ answers $\mathcal{A}$'s queries step by step.*
3) Challenge*: $\mathcal{C}$ generates a challenge which defines the challenged block-tag pairs.*
4) Second-Phase Queries*: It is similar to* First-Phase Queries*. The restriction is that there exists at least one challenged block-tag pair which has not been queried in* First-Phase Queries *and* Second-Phase Queries.
5) Forge*: For the challenge, $\mathcal{A}$ generates the response θ and sends it to $\mathcal{C}$.*

We say that $\mathcal{A}$ wins the game if the response θ can pass $\mathcal{C}$'s verification. If the probability that $\mathcal{A}$ wins the game is negligible, we say the proposed scheme satisfies the property of unforgeability.

Definition 2 states that, for the challenged blocks, the dishonest *Cloud Server* cannot produce a proof of data possession if the blocks have been modified or deleted. On the other hand, the definition does not state clearly the status of the blocks that are not challenged. In practice, for a secure blockchain-based private PDP scheme, it is necessary to ensure *Client* all of his stored data is kept intact with a high probability. In order to realize the purpose, we give the following security definition.

Definition 3 ((ρ, δ) Security). *A blockchain-based private PDP scheme is (ρ, δ) secure if* Cloud Server *corrupted ρ fraction of the stored blocks, the probability that the corrupted blocks are detected is at least δ.*

Definition 4 (Client Fraud Prevention). *When the stored data is integer, i.e., not broken by* Cloud Server*, the probability that* Client *can successfully swindle* Cloud Server *and get the compensation is negligible.*

The practical PDP scheme must be fair. It can protect honest *Cloud Server* and honest *Client*. *Definition* 2 gives the method to protect *Client*. *Definition* 4 gives the method to protect *Cloud Server*.

Definition 5 (Compensation). *When the stored data is broken,* Client *can get the compensation from* Cloud Server. *No one can pass himself off as* Client *to get the compensation from* Cloud Server.

In order to check the remote data integrity, some excess data is necessary besides of the uploaded file itself. We make use of redundancy rate to denote the data expansion.

Definition 6 (Redundancy Rate). *Let the uploaded file be F whose length is denoted as $|F|$. In order to realize the remote data integrity checking, the excess data M is generated. M is also called the metadata in the paper. We denote M's length as $|M|$. The redundancy rate r is defined as $r = \frac{|F|+|M|}{|F|}$.*

From *Definition* 6, we know that $r = \frac{|F|+|M|}{|F|} = 1 + \frac{|M|}{|F|}$. Along with the increasing of r, the communication cost also increases. In other words, in order to decrease the communication cost, we must try to decrease the value r.

The notations and their descriptions throughout this paper are listed in Table 1.

3 The Proposed Blockchain-Based Private PDP Scheme

In this section, we present an efficient blockchain-based private PDP scheme. It is built from RSA, the corresponding difficult problems and blockchain which will be briefly reviewed below.

3.1 RSA and Difficult Problems

RSA (Rivest-Shamir-Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission.

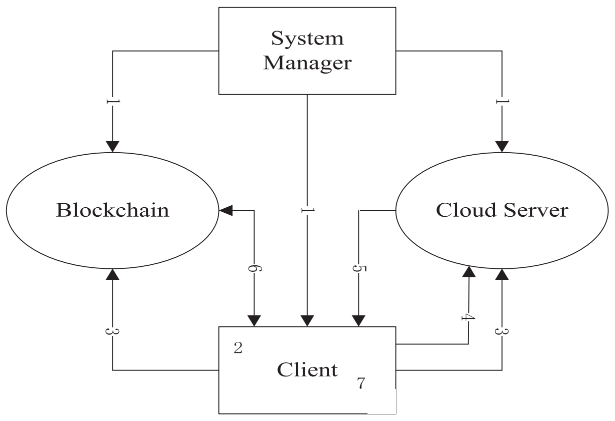


Fig. 3. Architecture of our Blockchain-Based private PDP scheme.

In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private). In RSA, this asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers, the "factoring problem".

Definition 7 (RSA Digital Signature). *Let N be a product of two random secret distinct large primes, along with integers, e and d, such that $ed = 1(\bmod \phi(N))$, where ϕ is the Euler function. The signer's public key consists of N and e, and the signer's private key is d. To sign a message m, the signer computes a signature σ, such that $\sigma = m^d(\bmod N)$. To verify the signature, the receiver checks whether $\sigma^e = m(\bmod N)$ holds. If it holds, the signature is valid; otherwise, it is invalid.*

Theorem 1 (Unique cyclic subgroup of $\mathbb{Z}_N^*$) [1]. *Let $p = 2p' + 1$ and $q = 2q' + 1$ be safe primes. Let $N = pq$ be an RSA modulus. Denote QR_N as the set of quadratic residues modulo N. Then, QR_N is the unique cyclic subgroup of $\mathbb{Z}_N^*$ of order $p'q'$.*

From the Definition 7 and Theorem 1, we know that p, q and p', q' are secret. Thus, the order of QR_N is secret. In the paper, RSA digital signature is used to ensure the integrity of the remote data and tags. Theorem 1 is used to generate the homomorphic tags. Bellare et al. [37] provided a formulation of KEA1, that we follow and adapt to work in the RSA ring:

Definition 8 (KEA1-r, i.e., Knowledge of Exponent Assumption). *Let g be a generator of QR_N. For any adversary $\mathcal{A}$ that takes input (N, g, g^s) and returns group elements (C, Y) such that $Y = C^s$, there exists an "extractor" $\bar{\mathcal{A}}$ which, given the same inputs as $\mathcal{A}$, returns x such that $C = g^x$.*

Based on *KEA1-r*, Ateniese et al. proposed the first and classic PDP scheme [1]. Our scheme is also provably secure based on Definition 8.

3.2 The Concrete Blockchain-Based Private PDP Scheme

The concrete blockchain-based private PDP scheme consists of six phases: KeyGen, SignBlock, Challenge, GenProof, CheckProof, and Compensation. We give its architecture in Fig. 3. The architecture figure is described below: 1. *System Manager* generates the system parameters. Then, it sends these parameters to *Blockchain*, *Cloud Server* and *Client*, respectively. It corresponds to the phase KeyGen. 2. *Client* divides the file. Then, it generates the metadata and signature. 3. *Client* sends the file to *Cloud Server*. At the same time, it sends metadata and signature to *Blockchain*. The above two procedures correspond to the phase *SignBlock*. 4. *Client* sends the challenge to *Cloud Server*. It corresponds to the phase *Challenge*. 5. Based on the stored data, *Cloud Server* generates the response and responds it to *Client*. It corresponds to the phase *GenProof*. 6. *Client* queries *Blockchain* for the corresponding metadata. 7. *Client* verifies whether the stored data is integer. The above two procedures correspond to the phase *CheckProof*.

Denote the stored file as F. F can be the plaintext or the ciphertext. If the stored file's content needs to be kept secret, F can be the ciphertext; otherwise, F is the plaintext. In other words, clients can choose to encrypt the file prior to out-sourcing the storage as Ateniese et al.'s scheme [1]. Without loss of generality, *Client* divides the file F into n blocks. These n blocks are denoted as $(F_1, F_2, \ldots, F_n)$, where F_i indicates the ith block. The detailed algorithms are given below:

- KeyGen: Denote $N = pq$ where p and q are safe primes. p, q can be expressed as $p = 2p' + 1, q = 2q' + 1$ where p', q' are primes. Let g be a generator of the group of quadratic residues QR_N of order $p'q'$. Let e be a large prime. Denote *Client*'s public key as $pk = (N, g, e)$ and private key as $sk = d$ such as $ed = 1(\bmod p'q')$. Let $H : \{0,1\}^* \rightarrow QR_N$ be a secure deterministic hash-and-encode function that maps strings uniformly to QR_N. In addition, let f be a pseudo-random function and π be a pseudo-random permutation. They are described as follows:

$$f : \mathbb{Z}_N^* \times \{1, 2, \ldots, n\} \rightarrow \mathbb{Z}_N^*$$
$$\pi : \mathbb{Z}_N^* \times \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}.$$

  The public key pk, hash function H, pseudo-random function f and pseudo-random permutation π are made public in the system. The private key sk is only known to *Client*. It also implies that the order $p'q'$ of QR_N is kept secret. $p'q'$ is also known only to *Client*.
- SignBlock: For the n blocks $(F_1, F_2, \ldots, F_n)$, *Client* performs the following procedures:
  1) For $1 \leq i \leq n$, compute $f_i = F_i(\bmod p'q')$ and $T_i = g^{f_i}(\bmod N)$.
  2) Compute the hash value $F_H = H(F)$ of the file F.
  3) For the computed messages $\{T_1, T_2, \ldots, T_n\}$ and F_H, compute the RSA signature
  $$\sigma = H(T_1, T_2, \ldots, T_n, F_H)^d(\bmod N).$$
  4) Upload the n blocks and hash value $(F_1, F_2, \ldots, F_n, F_H)$ to *Cloud Server*. At the same time, upload $(T_1, T_2, \ldots, T_n, F_H)$ and σ to *Blockchain*.
  5) Upon receiving $(F_1, F_2, \ldots, F_n, F_H)$, *Cloud Server* checks whether $F_H = H(F)$ holds. At the same time, *Cloud Server* checks whether F_H has been uploaded to *Blockchain*. If both the two conditions hold, *Cloud Server* accepts the n blocks $(F_1, F_2, \ldots, F_n)$ and hash value F_H.

- Challenge: *Client* picks the challenge $chal = (c, k_1, k_2)$ where $1 \leq c \leq n, k_1, k_2 \in \mathbb{Z}_N^*$ and sends $chal$ to *Cloud Server*;
- GenProof: Upon receiving the challenge $chal = (c, k_1, k_2)$ from *Client*, *Cloud Server* performs the following procedures:
  1) For $1 \leq i \leq c$, computes $j_i = \pi_{k_1}(i), a_i = f_{k_2}(i)$.
  2) Compute $\hat{F} = \sum_{i=1}^{c} a_i F_{j_i}$.
  3) Send $\hat{F}$ to *Client*.
- CheckProof: Upon receiving $\hat{F}$, *Client* performs the following procedures:
  1) For the challenge $chal = (c, k_1, k_2)$, compute $j_i = \pi_{k_1}(i), a_i = f_{k_2}(i)$ where $1 \leq i \leq c$.
  2) Compute $\bar{F} = \hat{F}(\bmod p'q')$.
  3) Query *Blockchain* and get $\{T_{j_1}, T_{j_2}, \ldots, T_{j_c}\}$. Then, in the group QR_N, check whether the following formula holds:

$$\prod_{i=1}^{c} T_{j_i}^{a_i} \overset{?}{=} g^{\bar{F}}. \tag{1}$$

  If the formula (1) holds, output "success"; otherwise, output "failure".
- Compensation: When the remote data is modified, *Cloud Server* must compensate for *Client*. In order to verify whether the remote data belongs to *Client*, *Cloud Server* verifies the signature below:

$$\sigma^e \overset{?}{=} H(T_1 T_2, \ldots, T_n, F_H)(\bmod N). \tag{2}$$

  If the Formula (2) holds, *Cloud Server* must compensate for *Client*; otherwise, reject.

4 Security Analysis

The security of our blockchain-based private PDP scheme mainly consists of the correctness and unforgeability. Unforgeability means that *Cloud Server* can pass *Client*'s checking with negligible probability if it forges the challenged blocks. At last, we also study the stored data's whole integrity if the challenged data can pass *Client*'s checking.

Correctness. A blockchain-based private PDP scheme must be workable and correct. That is, if the *Client*, *Cloud Server* and *Blockchain* are honest and follow the specified procedures, $\hat{F}$ and $\{T_{j_1}, T_{j_2}, \ldots, T_{j_c}\}$ can pass *Client*'s checking. The correctness of CheckProof follows from below:

$$\begin{aligned}
&\prod\nolimits_{i=1}^{c} T_{j_i}^{a_i}\\
&= \prod\nolimits_{i=1}^{c} g^{a_i f_{j_i}}\\
&= g^{\sum_{i=1}^{c} a_i f_{j_i}}\\
&= g^{\sum_{i=1}^{c} a_i F_{j_i}(\bmod p'q')}\\
&= g^{\left(\sum_{i=1}^{c} a_i F_{j_i}\right)(\bmod p'q')}\\
&= g^{\bar{F}(\bmod p'q')}\\
&= g^{\bar{F}}.
\end{aligned}$$

On the other hand, the correctness of Compensation is given below:

$$\begin{aligned}
&\sigma^e\\
&=(H(T_1, T_2, \ldots, T_n, F_H)^d)^e\\
&=H(T_1, T_2, \ldots, T_n, F_H)(\bmod N).
\end{aligned}$$

Theorem 2. *Based on the non-modifiability of blockchain, the proposed blockchain-based private PDP scheme is existentially unforgeable. In other words, if* Cloud Server *modifies the challenged data,* Cloud Server*'s response can pass* Client*'s verification with the probability which is not more than $\frac{1}{p'q'}$. For the safe parameters $p = 2p' + 1$ and $q = 2q' + 1$, the probability $\frac{1}{p'q'}$ is negligible.*

Proof. Based on the unforgeability of RSA signature, *Cloud Server* cannot forge new and different $\{T_1', T_2', \ldots, T_n'\}$ which are uploaded to *blockchain*. These uploaded data and their signature need to be verified before they are accepted by *blockchain*. Thus, we prove the existential unforgeability from the following two cases:

- The existential unforgeability for single block where $c = 1$. For the challenge $chal = \{1, k_1, k_2\}$, it corresponds to single block integrity checking. In order to pass the integrity checking, the formula $T_j = g^{\hat{F}_j(\bmod p'q')}$ must hold where $j = \pi_{k_1}(1)$. Based on SignBlock, we know that $T_j = g^{f_j} = g^{F_j(\bmod p'q')}$. If $T_j = g^{\hat{F}_j(\bmod p'q')}$ holds, then $f_j = \hat{F}_j(\bmod p'q')$ must hold.

  Based on the assumption of our concrete scheme, d is *Client*'s private key. Then, these primes p, q, p', q' are also secret. The four primes are known only to *Client*. So, *Cloud Server* does not know the modulo $p'q'$. The probability that modified $\hat{F}_j$ satisfies $f_j = \hat{F}_j(\bmod p'q')$ is not more than $\frac{1}{p'q'}$.
- The existential unforgeability for more challenged blocks where $c > 1$. For the challenge $chal = \{c, k_1, k_2\}$ where $c > 1$, it corresponds to more blocks integrity checking. In order to pass the integrity checking, the formula $\prod_{i=1}^{c} T_{j_i}^{a_i} = g^{\bar{F}}$ must hold where $j_i = \pi_{k_1}(i)$. Thus, $\sum_{i=1}^{c} a_i f_{j_i} = \bar{F}(\bmod p'q')$ holds. When $\bar{F}$ is not the original data, since $a_i = f_{k_2}(i)$ where f is pseudo-random function, i.e., $f : \mathbb{Z}_N^* \times \{1, 2, \ldots, n\} \rightarrow \mathbb{Z}_N^*$, we can get $\sum_{i=1}^{c} a_i f_{j_i} = \bar{F}(\bmod p'q')$ holds with the probability $\frac{N^{c-1}}{N^c} = \frac{1}{N}$.

Since both $\frac{1}{p'q'}$ and $\frac{1}{N}$ are negligible, the proposed blockchain-based private PDP scheme is existentially unforgeable. □

Theorem 3. *When* Cloud Server *modifies the stored data, no one can pass himself off as* Client *to get the compensation from* Cloud Server. *At the same time, for the dishonest client* Client, *he cannot make a false charge against* Cloud Server.

Proof. We prove the theorem from the following two cases:

- Based on the unforgeability of RSA signature, no one can pass himself off as *Client* to create the signature. Thus, when some fault happens, *Client* can get the compensation from *Cloud Server* while others cannot get.

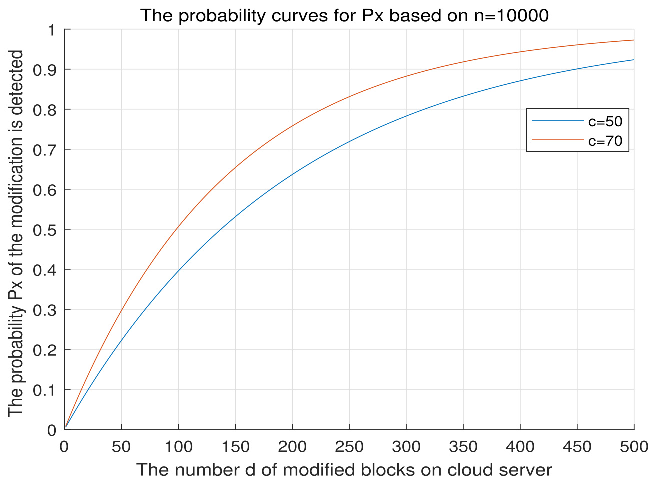


Fig. 4. The probability curve P_X of detecting the corruption.

- When dishonest *Client* uploads false data and swindles *Cloud Server*'s compensation, the following procedures are performed:
  1) *Cloud Server* sends the questioned blocks $\{F_i, i \in \mathbb{I}\}$ to the thrusted third party *TTP*.
  2) *Client* sends $p'q'$ to *TTP*
  3) For $i \in \mathbb{I}$, *TTP* computes $\bar{f}_i = F_i(\bmod p'q')$ and verifies whether $T_i = g^{\bar{f}_i}$.
  4) Based on the non-modifiability of blockchain, if they hold, *Client* is honest; otherwise, *Client* is dishonest.

When the length of block is small, *Cloud Server* verifies whether $T_i = g^{F_i}, i \in \mathbb{I}$ hold. If they hold, *Client* is honest; otherwise, *Client* is dishonest. We can also make use of the other method to finish the task: *Cloud Server* can prove it is honest by showing $F_H = H(F)$ holds where F_H has been stored on the *Blockchain*. Based on the non-modifiability of blockchain, the hash value F_H cannot be modified. If F_H is determined, its pre-image cannot be modified with non negligible probability. Of course, this method will incur huge communication cost. □

In the following, we give the probability of the stored data's whole integrity if the checked data can pass *Client*'s checking in the phase CheckProof.

Theorem 4. *Suppose* Client *uploads n blocks to* Cloud Server. *At the same time, it also uploads the corresponding signature and T_i to* Blockchain. *If d blocks are modified and c blocks are challenged, the modified blocks can be found with the probability*

$$P_X = 1 - \frac{(n-c)!(n-d)!}{n!(n-c-d)!}.$$

Proof. Denote the whole uploaded block set as $\mathcal{S}$ and the modified block set as $\mathcal{M}$. Let the random variable X be $X = |\mathcal{S} \bigcap \mathcal{M}|$. Then, we can get

$$\begin{aligned} P_X &= P\{X \geq 1\} \\ &= 1 - P\{X = 0\} \\ &= 1 - \frac{n-d}{n}\frac{n-1-d}{n-1}\cdot\cdots\cdot\frac{n-c+1-d}{n-c+1} \\ &= 1 - \frac{(n-c)!(n-d)!}{n!(n-c-d)!}. \end{aligned}$$

□

In order to show the intuition of P_X, we give Fig. 4. In this figure, based on the stored block number $n = 10000$, we give the probability curve for $c = 50$ and $c = 70$, respectively. X-label denotes the number of modified blocks. Y-label denotes the probability of the modified blocks can be detected. Based on $c = 50$, when $d = 300$, $P_X = 0.7828$; when $d = 450$, $P_X = 0.9005$; when $d = 500$, $P_X = 0.9236$. So, when some blocks are modified, they can be detected with very high probability.

Anonymity Discussion. When *Client* hopes to keep anonymous to *Cloud Server*, *Client* chooses its private key and computes its public key. The public key corresponds to the address on *Blockchain*. *Client* signs $T_i, F_H, 1 \leq i \leq n$ and gets σ by using the chosen private key. Then, it uploads $T_i, F_H, 1 \leq i \leq n$ and σ to the address which corresponds to its public key on *Blockchain*. Since *Blockchain* has no trusted third party and realizes unconditionally anonymity for its clients, it can also help *Client* keep anonymous to *Cloud Server*.

5 Performance Analysis

First, we analyze the performance of our proposed blockchain-based private PDP scheme based on the computation cost and communication cost. We also compare our blockchain-based private PDP scheme with the other existing PDP schemes. Second, we give the implementation prototypal of the proposed blockchain-based private PDP scheme. In the implementation, we also compare our scheme with the other existing PDP schemes.

Computation Cost. Suppose the file F is divided into n equal length blocks $\{F_1, F_2, \ldots, F_n\}$ which will be uploaded to *Cloud Server*. Denote the length of F as $|F|$ and the length of block as $|F_i|$ where $|F_1| = |F_2| = \cdots = |F_n|$. Then, $n = \frac{|F|}{|F_i|}$. We denote the computation cost of exponentiation modulo N as C_{exp}. The computation cost of modulo is denoted as C_{mod}. We omit the computation cost of hash function H because its cost is very low. The challenged block number is c. In the phase SignBlock, the computation cost is $(n+1)C_{exp} + nC_{mod}$. In the phases Challenge and GenProof, the computation cost is very low because there does not exist exponentiation cost modulo N. In the phase CheckProof, the computation cost is $(c+1)C_{exp} + 1C_{mod}$. In the phase Compensation, the computation cost is C_{exp}. In the proposed blockchain-based private PDP scheme, the block length $|F_i|$ is arbitrary.

In order to show the computation advantages of our scheme, we also give the computation cost of two classic PDP schemes [1], [6]. Ref. [1] makes use of RSA. For the up-to-date RSA, the length of the security parameter is 1024 bits. We also use the same file F as the example. Usually, the exponent length is less than the length of N. Thus, F is divided into $n' = \frac{|F|}{1024}$ blocks. In the phase TagBlock, the computation cost is $2n'C_{exp}$. In the phase *GenProof*, the computation cost $(c+1)C_{exp}$. In the phase CheckProof, the computation cost is $3C_{exp} + 1C_{mod}$. Ref. [6] makes use of the bilinear pairings. The length of the security parameter is 160 bit. Denote the computation cost of bilinear pairing as C_{pair}, the computation cost of scalar multiplication as C_{Smul} and the computation cost of point addition as C_{add}. In this case, F is divided into $n'' = \frac{|F|}{160}$ blocks. In order to realize the fair

TABLE 2
Comparison of Computation Cost

Schemes	SignBlock (TagBlock, Pub.St)	GenProof (Pub.P)	CheckProof (Pub.V)
Ateniese [1]	$2n'C_{exp}$	$(c+1)C_{exp}$	$(c+2)C_{exp}+1C_{mod}$
Shacham [6]	$2n''C_{Smul}$	$cC_{Smul}+(c-1)C_{add}+1C_{mod}$	$2C_{pair}+(c+1)C_{Smul}+cC_{add}$
Our Scheme	$(n+1)C_{exp}+nC_{mod}$	very low	$(c+1)C_{exp}+1C_{mod}$

Notes: $n=\frac{|F|}{|F_i|}$, $n'=\frac{|F|}{1024}$, $n''=\frac{|F|}{160}$.

TABLE 3
Comparison of Communication Cost (bits)

Schemes	SignBlock (TagBlock, Pub.St)	GenProof (Pub.St)	CheckProof (Pub.P)	Challenge	$\bar{r}$
Ateniese [1]	$1024*n'+\|F\|$	$1024+\|\rho\|$	*	$2048+\|n'\|$	1+1024*$\frac{n'}{\|F\|}$
Shacham [6]	$1024*n''+\|F\|$	$1343+c$	*	$2048+\|n''\|$	1+1024*$\frac{n''}{\|F\|}$
Our Scheme	$1024(n+1)+\|F\|+2\|F_H\|$	$1023+c+\|F_i\|$	$1024*c$	$2048+\|n\|$	1+1024*$\frac{n}{\|F\|}$

Notes: $n=\frac{|F|}{|F_i|}$, $n'=\frac{|F|}{1024}$, $n''=\frac{|F|}{160}$.

comparison, we set $s=1$ for Ref. [6]'s scheme. In the phase Pub.St, the computation cost is $2n''C_{Smul}$. In the phase Pub.P, the computation cost is $cC_{Smul}+(c-1)C_{add}+1C_{mod}$. In the phase *Pub.V*, the computation cost is $2C_{pair}+(c+1)C_{Smul}+cC_{add}$.

In order to show the concrete comparison, we construct the Table 2. In this table, the three parameter n, n', n'' are defined below: $n=\frac{|F|}{|F_i|}$, $n'=\frac{|F|}{1024}$, $n'=\frac{|F|}{160}$. Since the block length $|F_i|$ is arbitrary for our scheme, it may be larger than 1024 bits. In this case, $n''>n'>n$ holds. Thus, our scheme's computation cost is lower than the existing PDP schemes.

Communication Cost. National Bureau of Standards and ANSI X9 have determined the shortest key length requirements: RSA and DSA is 1024 bits, ECC is 160 bits [38]. Based on the requirements, we give our blockchain-based private PDP scheme's communication cost. In the phase SignBlock, *Client* sends $|F|+|F_H|$ bits to *Cloud Server*. At the same time, *Client* sends $1024n+|F_H|$ bits to *Blockchain*. In the phase *Challenge*, *Client* sends $2048+|n|$ bits to *Cloud Server*. In the phase *GenProof*, *Cloud Server* sends $1023+c+|F_i|$ bits to *Client*. In the phase *CheckProof*, *Blockchain* sneds $1024c$ bits to *Client*.

In order to show the communication advantages of our scheme, we also give the commutation cost of two classic PDP schemes [1], [6]. For Ref. [1], F is divided into $n'=\frac{|F|}{1024}$ blocks. In the phase TagBlock, *Client* sends $1024*n'+|F|$ bits to *Cloud Server*. In the phase *Challenge*, *Client* sends $2048+|n'|$ bits to *Cloud Server*. In the phase *GenProof*, *Cloud Server* sends $1024+|\rho|$ to *Client*. For Ref. [6], F is divided into $n''=\frac{|F|}{160}$ blocks. In the phase Pub.St, *Client* sends $1024*n''+|F|$ bits to *Cloud Server*. In the phase *Pub.P*, *Cloud Server* sends $1343+c$ bits to *Client*.

In order to show the concrete comparison, we construct the Table 3. In this table, the three parameter n, n', n'' are defined below: $n=\frac{|F|}{|F_i|}$, $n'=\frac{|F|}{1024}$, $n'=\frac{|F|}{160}$. Since the block length $|F_i|$ is arbitrary, it may be larger than 1024 bits. In this case, $n''>n'>n$ holds. We define the redundancy rate as $r=\frac{|F|+|MetaData|}{|F|}$. 'MetaData' denotes the data which is used to check remote data integrity except of the stored file F. When r becomes greater, the communication cost also becomes greater. From the case $n''>n'>n$, our scheme's communication cost is the lowest among them. In other words, our scheme's communication efficiency is the best. When $|F|=1024*1024$ bits, we also give Fig. 5 to show the comparison of redundancy rate. X-label denotes the length of the block for our scheme. Y-label denotes the redundancy rate. For Ateniese scheme [1] and Shacham scheme [6], the block length is constant. For every block, the corresponding tag's length is also constant. Thus, the redundancy rate is constant. For our proposed scheme, the block length is variable. Our scheme's redundancy rate rapidly decreases along with the increasing of the block length.

Implementation. We give the implementation prototypal of our blockchain-based private PDP scheme by using C programming language with GMP Library (GMP-5.0.5) [39], Miracl Library [40] and PBC Library (pbc-0.5.13) [41]. *Cloud Server* and *Blockchain* work on an ASUS S56C Laptop with the following settings:

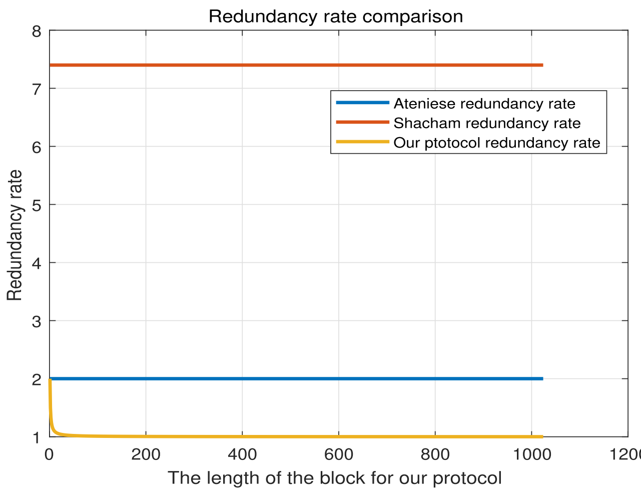


Fig. 5. The redundancy rate comparison.

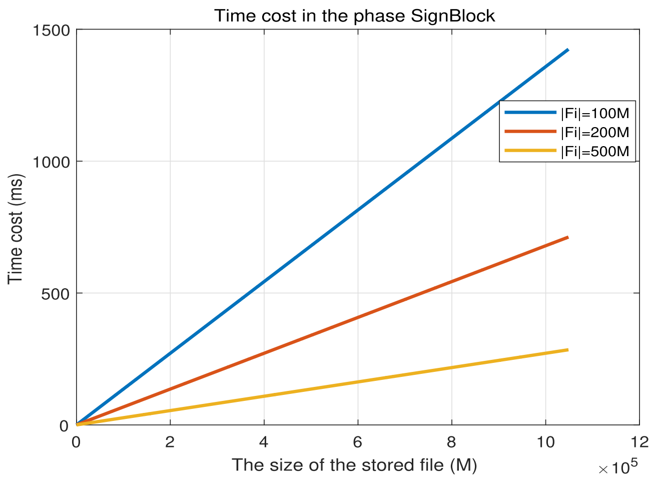


Fig. 6. Time cost of different block size in the phase SignBlock.

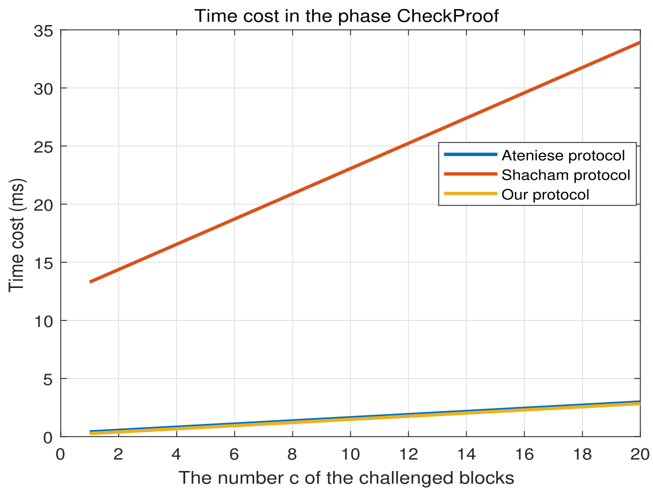


Fig. 7. Time cost in the phase CheckProof.

- CPU: Intel Core i7-3517U @ 1.90 GHz
- Physical Memory: 4 GB DDR3 1600 MHz
- OS: Ubuntu 13.04 Linux 3.8.0-19-generic SMP i686

Client works on an PC Laptop with the following settings:

- CPU: CPU I PDC E6700 3.2 GHz
- Physical Memory: DDR3 2 G
- OS: Ubuntu 11.10 over VMware-workstation-full-8.0.0

In the implementation prototypal, our proposed scheme and Ateniese [1]'s scheme works on 1024-bit RSA. Shacham [6] works on an elliptic curve with 160-bit group order. For a point of the elliptic curve, its X-coordinate and Y-coordinate are elements of the field whose length is 512-bit. They have almost the same security level.

Fig. 6 shows the concrete time cost of the phase SignBlock for our blockchain-based private PDP scheme. X-label denotes the size of the stored file (M, *i.e.*, 1 Mbit = 1024 kbit). Y-label denotes the time cost in the phase SignBlock. The three different lines correspond to three different block sizes, *i.e.*, 100 M, 200 M, 500 M, respectively. When the block size becomes larger, the computation becomes more efficient. Especially, the efficiency growth is very obvious. Thus, our scheme can be used for big data. In order to show our proposed blobkchain-based private PDP scheme's computation advantage, we compare our scheme with Ateniese's scheme [1] and Shacham's scheme [6]. Let the size of the stored file F be $|F| = 1T$ bits. Table 4 gives the comparison. Table 4 shows that our scheme's computation cost is obviously lower than Ateniese's scheme [1] and Shacham's scheme [6].

From the Table 2, in the phase GenProof, our scheme's time cost is lower than Ateniese's scheme [1] and Shacham's scheme [6]. We omit them in the paper. We further study the time cost of the phase CheckProof. X-label denotes the number c of the challenged blocks. Y-label denotes the time cost in the phase CheckProof. Fig. 7 shows that Shacham's scheme [6] has more time cost than Ateniese's scheme [1] and our scheme. At the same time, it also shows that Ateniese's scheme [1] has almost the same time cost as our scheme in the phase *CheckProof*.

6 Conclusion

In the cloud storage, it is an important security problem to ensure the remote data integrity. Based on the faults of the existing PDP schemes, we propose the blockchain-based private PDP scheme. This paper formalizes the system model and security model of our blockchain-based private PDP. Then, we propose the first concrete blockchain-based private PDP scheme. We also analyze its efficiency from the theory and the implementation prototypal. Besides of the above properties, wa also discuss its anonymity. According to the anonymity discussion, our scheme can also realize *Client*'s anonymity.

There also exist many requirements in this field. In order to further improve the efficiency, it is also necessary to eliminate the certificate verification process. Thus, it is interesting to study the identity-based blockchain-based PDP scheme. In order to further improve the private key security, it is necessary to study key-evolving blockchain-based PDP scheme. Since it is a novel model, it is necessary and important to further extend the model to more application scenarios.

Acknowledgments

The work was supported by National Natural Science Foundation of China (Nos. 61872192, 61932016, and 61972294), Natural Science Foundation of Jiangsu Province of China (No. BK20181394), Major Science Research Project of Jiangsu Provincial Education Department(No. 19KJA310010), Qing

TABLE 4
Comparison of Time Cost in the Phase SignBlock (ms)

Schemes	Our scheme ($|F_i|$ = 100 M)	Our scheme ($|F_i|$ = 200 M)	Our scheme ($|F_i|$ = 500 M)	Ateniese [1]	Shacham [6]
Time cost (ms)	$1.4244 * 10^3$	712.2592	284.9848	$2.9042 * 10^8$	$1.4885 * 10^{10}$

The size of the stored file F is $1T$ bits, i.e., $|F| = 1T$ *bits*

Lan Project of Jiangsu Province, 1311 Talent Plan Foundation of NUPT, NSFC cultivation project of Jining Medical University (JYP2018KJ14), Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201813).

References

[1] G. Ateniese et al., "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 598–609.

[2] G. Ateniese, R. DiPietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Security Privacy Commun. Netw.*, 2008, Art. no. 9.

[3] C. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. 16th ACM Conf. Comput. Commun. Security*, 2009, pp. 213–222.

[4] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1034–1038, Aug. 2008.

[5] Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1432–1437, Sep. 2011.

[6] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Security*, 2008, pp. 90–107.

[7] Y. Zhu, H. Hu, G. J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2231–2244, Dec. 2012.

[8] H. Wang, D. He, A. Fu, Q. Li, and Q. Wang, "Provable data possession with outsourced data transfer," *IEEE Trans. Services Comput.*, Accessed: Nov. 8, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8607110, doi: 10.1109/TSC.2019.2892095.

[9] A. Juels and B. S. Kaliski Jr., "PORs: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 584–597.

[10] B. Kuang, A. Fu, S. Yu, G. Yang, M. Su, and Y. Zhang, "ESDRA: An efficient and secure distributed remote attestation scheme for IoT swarms," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8372–8383, Oct. 2019.

[11] H. Wang, D. He, and J. Han, "VOD-ADAC: Anonymous distributed fine-grained access control protocol with verifiable outsourced decryption in public cloud," *IEEE Trans. Services Comput.*, Accessed: Nov. 8, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/7886332, doi: 10.1109/TSC.2017.2687459.

[12] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1931–1940, Aug. 2017.

[13] Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared big data," *IEEE Trans. Dependable Secure Comput.*, Accessed: Nov. 8, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8345742, doi: 10.1109/TDSC.2018.2829880.

[14] H. Wang, K. Li, K. Ota, and J. Shen, "Remote data integrity checking and sharing in cloud-based health internet of things," *IEICE Trans. Inf. Syst.*, vol. 99, no. 8, pp. 1966–1973, 2016.

[15] H. Wang, D. He, J. Yu, and Z. Wang, "Incentive and unconditionally anonymous identity-based public provable data possession," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 824–835, Sep./Oct. 2019.

[16] H. Wang, "Proxy provable data possession in public clouds," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 551–559, Oct.–Dec. 2013.

[17] N. Saberhagen, "CryptoNote v 2.0," Accessed: Nov. 8, 2019. [Online]. Available: https://bravenewcoin.com/assets/Whitepapers/CryptoNote-V-2.0-whitepaper-annotated.pdf

[18] A. Kumar, C. Fischer, S. Tople, and P. Saxena, "A traceability analysis of Moneros blockchain," in *Proc. Eur. Symp. Res. Comput. Security*, 2017, pp. 153–173.

[19] S. F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "RingCT 2.0: A compact accumulator-based (linkable ring signature) scheme for blockchain cryptocurrency monero," in *Proc. 22nd Eur. Symp. Res. Comput. Security*, 2017, pp. 456–474.

[20] E. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proc. IEEE Symp. Security Privacy*, 2014, pp. 459–474.

[21] I. Miers, C. Garman, M. Green, and A. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *Proc. IEEE Symp. Security Privacy*, 2013, pp. 397–411.

[22] S Meiklejohn et al., "A fistful of bitcoins: characterizing payments among men with no names," in *Proc. Conf. Internet Meas.*, 2013, pp. 127–140.

[23] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," *Proc. Int. Conf. Financial Cryptography Data Security*, 2013, pp. 6–24.

[24] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for Bitcoin with accountable mixes," in *Proc. Int. Conf. Financial Cryptography Data Security*, 2014, pp. 486–504.

[25] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://www.cryptovest.co.uk/resources/Bitcoin

[26] F. Gao, L. Zhu, M. Shen, K. Sharif, Z. Wan, and K. Ren, "A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks," *IEEE Network*, vol. 32, no. 6, pp. 184–192, Nov./Dec. 2018.

[27] H. Wang, D. He, and Y. Ji, "Designated-verifier proof of assets for bitcoin exchange using elliptic curve cryptography," *Future Gener. Comput. Syst.*, Accessed: Nov. 8, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8345742, doi: 10.1016/j.future.2017.06.028.

[28] A. Dorri, M. Steger, S. Kanhere, and R. Jurdak, "BlockChain: A distributed solution to automotive security and privacy," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 119–125, Dec. 2017.

[29] L. Li et al., "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol 19, no. 7, pp. 2204–2220, Jul. 2018.

[30] Y. Tian, J. Guo, Y. Wu, and H. Lin, "Towards attack and defense views of rational delegation of computation," *IEEE Access*, vol. 7, pp. 44037–44049, 2019.

[31] L. Wang, Y. Tian, D. Zhang, and Y. Lu, "Constant-round authenticated and dynamic group key agreement protocol for D2D group communications," *Inf. Sci.*, vol. 503, pp. 61–71, 2019.

[32] H. Wang, D. He, Z. Liu, and R. Guo, "Blockchain-based anonymous reporting scheme with anonymous rewarding," *IEEE Trans. Eng. Manag.*, Accessed: Nov. 8, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8704256, doi: 10.1109/TEM.2019.2909529.

[33] H. Wang, Q. Wang, D. He, Q. Li, and Z. Liu, "BBARS: Blockchain-based anonymous rewarding scheme for V2G networks," *IEEE Internet Things J.*, vol 6, no. 2, pp. 3676–3687, Apr. 2019.

[34] W. Tong, X. Dong, Y. Shen, and Xi. Jiang, "A hierrchical sharding protocol for multi-domain IoT blockchains," in *Proc. IEEE Int. Conf. Commun.*, May, 2019, pp. 1–6.

[35] G. Liang, S. Weller, F. Luo, J. Zhao, and Z. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3162–3173, May 2019.

[36] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 5, pp. 840–852, Oct. 2018.

[37] M. Bellare and A. Palacio, "The knowledge-of-exponent assumptions and 3-round zero-knowledge schemes," in *Proc. 24th Annu. Int. Cryptology Conf.*, 2004, pp. 273–289.

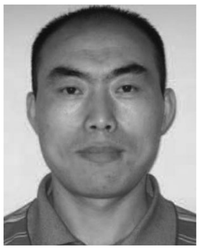
[38] C. Research, "SEC 2: Recommended elliptic curve domain parameters," Accessed: Nov. 8, 2019. [Online]. Available: http://www.secg.org/collateral/sec_final.pdf

[39] The GNU Multiple Precision Arithmetic Library (GMP), Accessed: Nov. 8, 2019. [Online]. Available: http://gmplib.org/

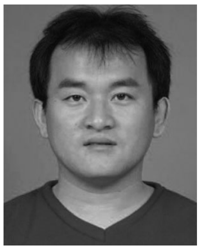
[40] Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL), Accessed: Nov. 8, 2019. [Online]. Available: http://certivox.com/

[41] The Pairing-Based Cryptography Library (PBC), Accessed: Nov. 8, 2019. [Online]. Available: http://crypto.stanford.edu/pbc/howto.html

Huaqun Wang received the BS degree in mathematics education from Shandong Normal University, in China, in 1997, the MS degree in applied mathematics from East China Normal University, in China, in 2000, and the PhD degree in cryptography from the Nanjing University of Posts and Telecommunications, in 2006. Now, he is a professor in Nanjing University of Posts and Telecommunications. His research interests include applied cryptography, blockchain, network security, and cloud computing security.



Qihua Wang received the BS degree in computer science and technology from Shandong Normal University, in 2003, and the MS degree in computer science and technology from the Harbin University of Science and Technology, in 2008, and the PhD degree in control theory and control engineering from the Dalian Maritime University, in 2015. He is currently a teacher with the Jining Medical University and postdoctoral research at the University of Electronic Science and Technology of China. His research interests include applied cryptography, Internet of Things, network security, and network location.



Debiao He received the PhD degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2009. He is currently a professor of the State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, China. His main research interests include cryptography and information security, in particular, cryptographic schemes.