Blockchain-Assisted Conditional Anonymity Privacy-Preserving Public Auditing Scheme With Reward Mechanism

Jie Zhao, Hejiao Huang, Chonglin Gu, Zhongyun Hua, *Member, IEEE*, and Xiaojun Zhang

***Abstract*—In real-world scenarios, in order to encourage one to report others crimes, judicial department usually rents independent cloud storage spaces to receive the precious evidences from whistleblowers. Since the uploaded data are not controlled by cloud users, remote data integrity is very important. Public cloud auditing enables an auditor to periodically check the integrity of outsourcing data on behalf of users, without retrieving the entire data file. However, most existing data auditing schemes have potential security vulnerabilities, and thus cannot defense many security attacks (e.g., the man-in-the-middle attack). Meanwhile, it is significant to protect whistleblower's identity privacy, reward the real data uploader, and further trace the responsibility of slanders accurately. From the aforementioned requirements, we present an efficient blockchain-assisted conditional anonymity privacy-preserving public auditing (BA-CAPPPA) scheme with reward mechanism. The Ethereum blockchain is integrated into BA-CAPPPA to enhance the security level of the whole public auditing mechanism. Theoretical analysis results show that the BA-CAPPPA achieves man-in-the-middle attack resistance, storage correctness guarantee, data privacy-preservation, conditional identity anonymity, and reward mechanism. Performance evaluations and comparisons demonstrate that BA-CAPPPA could outperform some state-of-the-art data auditing schemes.**

***Index Terms*—Conditional anonymity, ethereum blockchain, public cloud auditing, remote data integrity, reward mechanism.**

I. Introduction

ALONG with the popularity of wireless mobile devices and the rise of "We Media" era [1]–[3], a growing number of people utilize portable smart devices to record other people's criminal evidence (e.g., pictures, videos, or texts), thus actively participating in social management. As the amount of data produced by smart devices increases continuously, storing these data has brought a heavy burden on users with limited resources. Simultaneously, the vast quantities of data from different whistleblowers every day severely increase the pressure of judicial department (JD) on local data storage and management. Cloud storage technology [4] can address the aforementioned hard problems; it provides massive data storage space and powerful information processing capacity for cloud users.

Although cloud storage services provide great benefits to users, there exist various security threats that may affect the trust of organizations and individuals on cloud storage [5]. In particular, once sensitive reporting materials are outsourced to the cloud server (CS), due to hardware/software failures or human errors, cloud users could lose physical control for these data and always worry about whether the outsourced data are corrupted. As a matter of fact, the CS is a semitrusted entity, it may betray the auditing protocol for grabbing the money, power, and reputation. All these corrupt behaviors could bring serious trouble to whistleblowers, and even lead to the threat of death [6]. Therefore, it is necessary for cloud users to accomplish remote data integrity auditing. Nevertheless, it is not practical for users to periodically check the integrity of the whole data file in person, because it could incur enormous communication overhead and computation costs.

Public cloud auditing could delegate a third-party auditor (TPA) to check the integrity of outsourced data on behalf of users, without downloading the whole dataset. Shockingly, such a near-perfect remote data auditing mechanism still faces several fatal flaws. A typical example is that an adversary launches a man-in-the-middle attack [7] during the challenge-verification process between TPA and CS, and it may destroy the security of public cloud auditing thoroughly. Specifically, a well-trained adversary can intercept TPA's challenge message, and replays it to the CS. Subsequently, the adversary hijacks the response proof information of the CS, and resends it to TPA, but the response proof information is destroyed or forged by the adversary. Finally, the result of auditing verification is always "False." In this case, the user could constantly make requests to the CS for data recovery and economical compensation [8]. The CS tries to recover the original data files through mobilizing erasure codes, and compensates for the users, but these data stored in CS are still intact. Consequently, users could lose their final trust in the CS, whereas the CS will waste extensive computing resources and fall into huge risks of claim settlements.

In addition to the aforementioned issues, we also note that user's identity privacy protection and anonymous rewarding are

Manuscript received 5 March 2021; revised 4 July 2021 and 21 September 2021; accepted 1 November 2021. Date of publication 30 November 2021; date of current version 26 August 2022. This work was supposed in part by the Science and Technology Program of Shenzhen under Grant JCYJ20210324132406016 and in part by the National Natural Science Foundation of China under Grant 61902327 and Grant 61732022. *(Corresponding author: Hejiao Huang.)*

Jie Zhao, Hejiao Huang, Chonglin Gu, and Zhongyun Hua are with the Department of Computer Science and Technology, Mobile Internet and Cloud Computing Research Center, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: zhaojswpu2017@163.com; hjhuang@aliyun.com; guchonglin@hit.edu.cn; huazyum@gmail.com).

Xiaojun Zhang is with the School of Computer Science, Research Center for Cyber Security, Southwest Petroleum University, Chengdu 610500, China (e-mail: zhangxjdzkd2012@163.com).

Digital Object Identifier 10.1109/JSYST.2021.3125835

two significant functionalities. To encourage whistleblowers to provide timely and effective evidences of the criminal to JD, we need to achieve the following two goals: First, protection of identity privacy is a basic security requirement in a realistic network reporting scenarios [9]. It not only ensures that honest and courageous whistleblowers are protected from retaliation by defendants and their henchman, but also is immensely conducive to tracking and revealing malicious users from submitting junk information. Second, the anonymous rewarding should be improved [10], since whistleblowers and their families need to muster a great deal of courage and even risk their lives to safeguard social justice, he/she would like to get a reporting incentive without disclosing any sensitive identity information. As far as we are concerned, such problem has not been well addressed in previous research works. Hence, it is a challenging task to realize the user's identity anonymity with incentive policy (IP) in public auditing scheme.

To well solve the aforementioned issues for secure outsourced data in clouds, we propose an efficient blockchain-assisted conditional anonymity privacy-preserving public auditing scheme with reward mechanism, called BA-CAPPPA. The main contributions of this article are summarized as follows.

1) We investigate the security of outsourcing storage data in cloud computing, and integrate blockchain technology (BT) [11] into BA-CAPPPA, it can assist CS to check the correctness of the original storage data uploaded by users, ensure the traceability of these storage data, and avoid the CS from damaging the stored data intentionally. The proposed BA-CAPPPA scheme could enhance the security level of the whole public auditing mechanism while it can still be executed efficiently.
2) We propose an IP with conditional identity anonymity (CIA) to encourage courageous whistleblowers. It can prove that he/she is the real data uploader without leaking any sensitive identity information when JD plans to pay for the whistleblower. Even if there are events of libelers or spammers in system, the PKG could trace and reveal the real identities of misbehaved users accurately.
3) We reconstruct the response auditing proof information with verifiable random masking code technique to resist the man-in-the-middle attack. When the TPA's auditing result is $False$, it can indicate that some remote storage data have indeed been destroyed or discarded, the real data uploader should get compensation from the CS, whereas others cannot obtain the compensation.
4) We prove the correctness guarantee and provide the detailed security analysis of BA-CAPPPA. The comprehensive performance analysis and evaluation demonstrate that BA-CAPPPA achieves desirable efficiency, especially in the auditing verification costs, it is more suitable in the deployment of practical cloud storage reporting systems, compared with some state-of-the-art data auditing schemes.

The rest of this article is organized as follows. A related work is illustrated in Section II. In Section III, we review preliminaries, including the elliptic curve cryptosystem, ECDLP problem, Ethereum blockchain, system model, formal definition, threat model, and design goals. In Section IV, we describe our concrete construction of BA-CAPPPA. In Section V, we prove the correctness and security of BA-CAPPPA. The comprehensive performance evaluation is conducted in Section VI. Finally, Section VII concludes this article.

II. Related Work

Cloud storage service could provide cloud users with a convenient, flexible, and efficient way of data management, under the era of big data explosion. Owing to the sensitivity of reporting data, and the high frequency of attacks in public network environment, the reporting data of users are faced with several security threats, especially the confidentiality, reliability, and integrity of data [12]. Up to now, a large number of novel cryptographic techniques [5], [13]–[15] have been proposed to protect the data confidentiality. Meanwhile, some other outstanding schemes [6], [9], [16], [17] have been flexibly deployed in cloud storage to ensure data reliability. For cloud users, how to ensure the integrity of remote data is the most important and urgent security issue.

In order to check the integrity of remote data stored in cloud, Ateniese *et al.* [18] first proposed a notion of provable data possession (PDP) paradigm in 2007. By achieving the sampling inspection with a high detection rate, it can drastically reduce the overhead of challenge verification. In the same year, Juels and Kaliski [19] proposed a brand new auditing model of Proofs of Retrievability (PoR). In the model, some "Tags" masqueraded as normal encrypted data blocks are embedded evenly throughout the whole file block, it could be used to scale the integrity of the entire outsourced data in cloud. In 2008, based on short signature technique, Shacham and Waters [20] reconstructed the compact PoR with enhanced security. In 2013, Wang *et al.* [4] proposed a privacy-preserving remote data integrity auditing scheme with the employment of a random masking code technique. Recently, a majority of existing remote data integrity auditing schemes [8], [21], [22] were proposed to extend new features. Particularly, the scheme in [8] proposed a blockchain-based private provable PDP, it skillfully introduces the nontamper of transactions in blockchain to resist the corrupt CS.

However, majority of the aforementioned schemes rely on public key infrastructure system, which requires considerable overhead from the complex certificate management. To simplify certificate management process, Shamir [23] innovatively designed an identity-based cryptosystem. In 2017, Yu *et al.* [24] leveraged the zero-knowledge proof protocol to construct an identity-based auditing scheme with perfect data privacy preserving. In 2020, to guarantee subsequent secure communications of users and servers, and enhance the computational efficiency of key establishment process, Abbasinezhad-Mood *et al.* [25] combined the identity-based cryptosystem and elliptic curve signature algorithm to propose a novel privacy-preserving signature-based key establishment protocol. After that, several identity-based public auditing protocols [26]–[28] have been proposed to further enhance security and performance. Specifically, a provable secure identity-based public auditing scheme has been proposed in [26], it supports the strong secure proxy process between original data owner and proxy. To resist the internal attacks in the identity-based public auditing scheme, scheme in [27] takes advantage of new technique-blockchain to solve its inherent weaknesses. Scheme in [28] combines lattice-based linear homomorphic signature with an identity-based data outsourcing public verification scheme in clouds to achieve postquantum security.

We also note that anonymity is an extremely important security requirement, it can effectively protect the identity privacy of user, especially in the medical information systems and the network reporting systems. This is because any user's identity

information is leaked, it may cause serious consequences [29]. In 2015, Yang *et al.* [30] used the ring signature to construct a secure anonymous authentication protocol for ad hoc group. In 2019, Tang *et al.* [31] designed a personalized and trusted healthcare service approach to enable user's identity privacy in social media health networks. By utilizing the efficient Chebyshev chaotic map-based public key cryptosystem, an anonymous key agreement scheme was proposed by Abbasinezhad-Mood *et al.* [32], it could provide the expected security requirements, and have a certain performance advantages compared with other related schemes. Jia *et al.* [33] indicated that identity anonymity is essential in mobile edge computing, and thus presenting a provably secure and efficient identity-based anonymous authentication scheme. However, most of these schemes are based on bilinear pairings, which may cause considerable time-consuming cryptographic operations.

In addition, it is necessary to integrate IP and compensation functionality (CF) into public auditing schemes. In 2016, Gong *et al.* [34] designed a privacy-preserving scheme for incentive-based demand response programs in smart grids. To encourage the public to disclose adverse events, Wang *et al.* [35] proposed the concept of identity-based public PDP with incentive and unconditional anonymity, it can reward the whistleblower. In 2020, Huang *et al.* [10] developed a novel framework IPANM, which integrated the incentive mechanism based on blockchain, and thus realizing an incentive privacy-preserving public auditing scheme. Recently, some existing results have been proposed in other research field of reward mechanism [11], [36], [37].

III. Preliminaries

A. Elliptic Curve Cryptography (ECC) and Hardness Problem

ECC plays an extremely significant role in the construction of existing cryptographic protocols [25]. The ECC means that the coefficients of the curve equation as defined by $Weierstrass$ are all elements over a finite field F_p, where p is a large prime number. The elliptic curve equation is defined as $y^2 \equiv x^3 + ax + b \bmod p$, where $a, b \in F_p$, $0 \leq x < q$, and $4a^3 + 27b^2 \neq 0 \bmod p$. In particular, the ECC provides a better performance than Rivest–Shamir–Adleman (RSA), because RSA's key size is 1024 b under the identical security strength. According to the ANSI X9 and National Bureau of Standards [8], the base ECC's key length requirement is 160 b. Due to the MOV specification of bilinear map, it needs to increase the number of bits of the hypersingular elliptic curve to ensure the same security intensity as the ECC. Therefore, the ECC has more advantages in calculation efficiency. Now, we provide the definition of elliptic curve discrete logarithm problem (ECDLP) as follows.

Definition 1: Given two large prime numbers p and q, set $\mathbb{E}$ to be an elliptic curve in $\mathbb{Z}_p$. Suppose P is a point of the elliptic curve $\mathbb{E}$, and P is the generator of the additive cyclic group $\mathbb{G} =< P >$ with order q. For any $R, P \in \mathbb{G}$, the goal of ECDLP is to find an integer $a \in \mathbb{Z}_q$, such that $R = aP$.

B. Ethereum Blockchain: Blockchain is composed of a variety of sophisticated technologies, such as cryptography, distributed data storage, point-to-point networks, and consensus protocols. At present, industry and commerce are mainly based on public blockchain [9]. It can be executed by any participant in the public network, and the electronic currency system represented by it has been very mature, such as Bitcoin, Ethereum, and Libra [38]. In this article, we take full advantage of the Ethereum blockchain technique to construct the BA-CAPPPA, considering the following two extremely important properties.

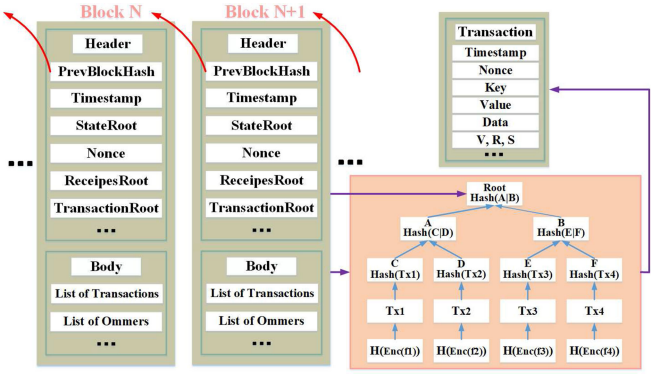

Fig. 1. Example of Ethereum blockchain.

Imtamperability: Ethereum is a growing linear collection of data elements, in which each data element is called a block. The chaining is completed by adding the hash value of the previous block to the current block, and each current block contains a hash pointer as a link to a previous block. Continuous nested blocks ensure that transaction data are generated in a chronological order, so a transaction cannot be backdated without changing its block and all subsequent blocks. Meanwhile, based on the Merkel-tree-structure and secure cryptographic hash functions [39], the transaction data at the bottom of the tree are stored in the chunks, which are divided into buckets; then the hash value of each bucket is obtained and repeated until only one root hash value remains. When the transaction of node in the tree is modified, it will cause an avalanche effect. If an adversary tries to tamper with a transaction on a block (except for the initial block), it has to modify all previous blocks. Moreover, the consensus mechanism in blockchain has ensured that each block database would not be tampered with, unless the adversary owns 51 stakes in whole blockchain. Here, we show an example of Ethereum in Fig. 1. Each block contains a hash pointer as a link to a previous block, a $Timestamp$ represents the actual time a transaction was chained, the $Nonce$ enables each transaction to be executed sequentially, and a $TransactionRoot$, etc., these fields work together to ensure that transaction data are tamper-proof and traceable.

Secure storage and query: Ethereum has secure storage-query functions, since the transaction data are protected by secure cryptographic algorithms, and its storage modality is distributed. Data on the chain of Ethereum system include block body and transaction record field. The block body data are mainly generated and maintained by the system, which protects the security of system parameters in the chain. User's data are stored in the transaction record filed, it can be queried and shared in the future. To be specific, a user submits transaction data to the blockchain in a fixed format. Once these data are received, Ethereum converts it into a string value $Value$ by using Recursive Length Prefix Encoding, then it computes the digital signature Key of $Value$. Finally, the $< Key, Value >$ is stored in LevelDB [40]. When retrieving a certain data, the user first hunts for the block header accurately according to the height of block, and anatomizes the $< Key, Value >$ from the query statement $SearchByKey(Key) = \{Value_i | <$

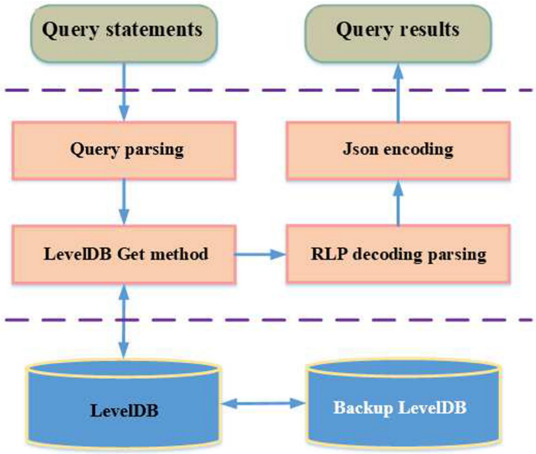

Fig. 2. Simple flow diagram for data query in blockchain.

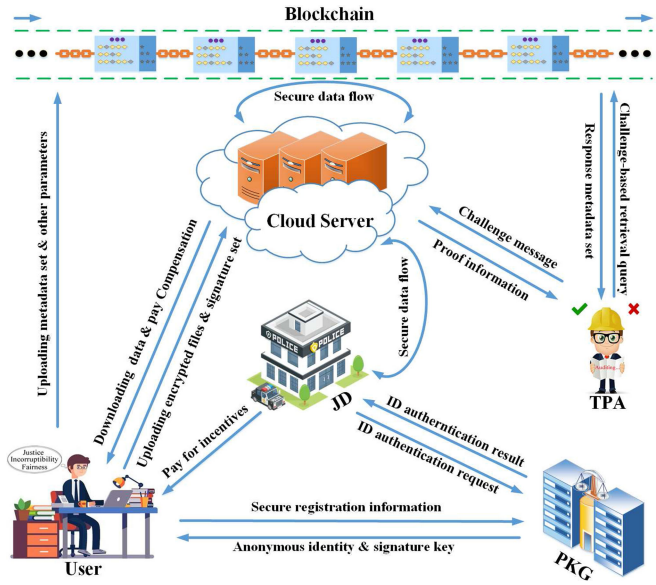

Fig. 3. Basic system model of BA-CAPPPA.

$Key_i, Value_i >, Key = Key_i, i = 1, 2, \ldots, n\}$ to recover the Key_i, where n is the number of transaction data. Then, it calls LevelDB's Get method to obtain the corresponding $Value_i$, as well as decodes it. Ultimately, the search data are sent to the user via a secure channel. The flow diagram for data query in the blockchain is shown in Fig. 2 .

In our article, the auxiliary data $\{Tag, filename, H(f_i'), (T_i, W_i)\}$ uploaded by user to blockchain are equivalent to the transaction data. Once receiving the auxiliary data from users, blockchain first verifies the validity of user's identity and auxiliary data, if both the two conditional hold, these data will be absorbed into the block with a $Timestamp$, then the blockchain computes the encrypted data block digest $H(f_i')$ to obtain the root hash value $H(\Lambda)_B$ according to the MTH and hash function H of Ethereum. After that, it stores metadata (T_i, W_i) and file tag $filename$ in the LevelDB with time order, and generates the corresponding signature query value Key_i, where $i \in [1, n]$. At last, the block synchronizes transaction data to the entire blockchain. By taking advantage of the imtamperability and secure storage and query of Ethereum, it not only check whether the storage data uploaded by users to the CS are correct, but also trace whether these data are uploaded by users originally when they have disputes. Moreover, it can prevent the CS from launching intention attack to damage the stored data, and thus enhancing the security level of the auditing mechanism.

C. System Model and Formal Definition: In this section, we first define a basic system model of BA-CAPPPA with reward mechanism, which is depicted in Fig. 3. It consists of six different entities: User, private key generator (PKG), CS, TPA, blockchain, and JD.

1) User: The user is a general term for whistleblower, data uploader, or original data owner. To uncover crimes committed by criminals, corrupt officials, and organized gangs, he/she has the ability to gather a great many real evidences or secrets by smart devices, and uploads it to the CS for enjoying more convenient services.
2) PKG: The PKG is trusted, it is in charge of issuing system public parameters, and generating the anonymous identity of user. Concurrently, under the premise of fully ensuring the privacy of user's identities, PKG assists JD to reward the real user without leaking any identity information.
3) CS: The CS is managed by cloud service provider. It provides massive data storage services and powerful computing capability for cloud user. If existing a fact that remote data have been changed, the CS needs to pay compensations to the corresponding user.
4) TPA: The TPA is a honest entity. Once an outsourcing auditing agency agreement is reached, it can periodically check the integrity of the remote data stored in public CS on behalf of the user.
5) Blockchain: The blockchain can store some important auxiliary data for each user, and generate secure verification proof for CS and TPA, respectively. Based on the data imtamperability and traceability of blockchain, it could assist the CS to verify whether the original storage data received from the user is correct and traceable. When receiving TPA's challenge message, it can regenerate the corresponding metadata set and return it to the TPA in time, thus helping the TPA to complete the auditing task.
6) JD: The JD is an authority, it can not only accept the reports of illegal crime in time, but also reward the honest and courageous user.

In our system model, a user needs to send a secure registration information to PKG for obtaining the corresponding anonymous identity, as well as its signature key. In order to ensure that the original storage data uploaded by users to the CS is well-documented, and further enhance the security level of existing public auditing schemes. The system model requires that the original outsourced data of users be divided into two parts: auxiliary data and storage data, which are uploaded to blockchain and CS, respectively. In particular, the auxiliary data mainly denote the block tags besides of file properties and block properties. Meanwhile, to overcome the inherent shortcomings and technical bottlenecks of Ethereum, including the limited storage capacity, as well as slow response and query speed, the auxiliary data should be as light as possible. Once receiving the auxiliary data from users, the blockchain first verifies the legitimacy of user's identities and checks the validity of these data. Then, it computes the root hash value for all encrypted

data block digest, and generates the corresponding storage-query value for each metadata. Finally, it transmits the secure data flows to CS. Within a certain time range, the CS checks whether the root hash uploaded and the root hash value sent by the user are equal, and further verifies the validity of the filename tag. If both the two conditional hold, the CS accepts the storage data uploaded by user; otherwise, it refuses to store these data.

To reduce the remote data auditing burden of user, TPA is employed to execute the integrity checking tasks frequently. To be specific, the TPA sends the challenge message and challenge-based retrieval query to CS and blockchain, severally. After that, the CS and blockchain return auditing proof information and response metadata set to TPA, respectively. That is, the auditing proof information received by TPA is no longer provided by the CS alone, and the important proof information is also supplied by blockchain. When the entire response proof information is received successfully, TPA checks its integrity by auditing equation. If the auditing result is "False," it means that the reporting data stored by the user on the CS has indeed been damaged. Thus, the corrupt CS must pay compensations to the user. Moreover, when JD intends to pay for the user, PKG assists JD in identifying user's real identity, and sends the identity authentication result to show that it is the real data uploader.

The formal definition of BA-CAPPPA scheme consists of the following seven polynomial-time algorithms (PPTs).

1) *Setup:* The system initialization is a probabilistic PPT, which takes a secure parameter 1^k as the input, and outputs the system master secret key msk and system public parameters $Para$.
2) *AnonyIDGen and KeyExtract:* This is anonymity and extraction algorithm run by the PKG. It takes the master secret key msk, the system public parameters $Para$, the user's real identity id, and a valid period $Time$ of an anonymous identity as inputs. The PKG outputs an anonymous identity Aid as well as its signature private key SK_{Aid}.
3) *SignGen and DataOutsourcing:* This is a signature and metadata generation algorithm run by the user. It takes a report data file F with unique index $filename$, the pseudorandom function Prf with a private key sk_{Prf}, the anonymous identity Aid of user, and the signature private key SK_{Aid} as the inputs. It outputs a blind data file F', and its corresponding signature aggregation ψ, the metadata set (T, W), the root hash value $H(\Lambda)$ of MHT in blockchain, and a file tag Tag.
4) *ChallengeGen:* This is a challenge message generation algorithm run by TPA, which takes the system public parameters $Para$ as inputs, and outputs the challenge message $Chal$.
5) *ProofGen:* This is a proof information generation algorithm run by the CS and blockchain, severally. It takes the challenge message $Chal$, a storage-query set Key, a blind set L of data blocks, and the corresponding signature aggregation ψ as input. It outputs the response proof information $Proof$, as well as the metadata set (T, W).
6) *ProofVerification:* This is a proof verification algorithm run by the TPA. It takes the system public parameters $Para$, the auditing challenge message $Chal$, the response proof information $Proof$, and metadata set (T, W) with the file tag Tag as the input. It outputs the auditing results $0/1$.
7) *RewardCertification:* This algorithm is divided into the following two parts: First, this is an incentive verification algorithm run by the JD and PKG. It takes the system public parameters $Para$, and a valid period $Time$ of an anonymous identity Aid as inputs. It outputs the real identity comparison results $0/1$. Second, this is a compensation algorithm run by the JD and CS, it takes the pseudorandom function Prf with a private key sk_{Prf}, and the identity-filename tag σ as inputs. It outputs the authentication results $0/1$.

D. Security Threats and Design Goals: In the security model, the proposed BA-CAPPPA scheme is mainly confronted with five types of active attacks. First, a trained adversary initiates a man-in-the-middle attack during the integrity auditing of remote data without being detected. Second, to maintain the reputation or avoid compensation, the CS may hide the fact that the uploaded reporting data are destroyed or lost, due to hardware failures or software bugs, even worse, the malicious CS directly delete or replace the reporting data block for economic interests. Third, an adversary (including the CS) may be awfully curious about the sensitive reporting data stored by user, who derives the primitive reporting data through powerful computing devices. Fourth, the defendants or mafia try their utmost to deduce the real identity of user, thus putting the real data uploader and their families under the risk of injury or even death. Finally, a speculator tries to impersonate the real data uploader to receive incentives, and a dishonest user may upload incomplete storage data to CS, and thus making a false charge to the innocent CS. To further enhance the reliability of the final auditing verification results, and ensure that the designed CF can operate with a high security, we require that the semitrusted CS will not collude with TPA or other entities in the system. For the transmission of data between any two logical entities in public network environment, we focus on the man-in-the-middle attack. Actually, considering the specific characteristics of Denial of Service (DoS) attack and other more complex reasons [41], the cloud-based network reporting system cannot against such attacks at present.

Therefore, the proposed BA-CAPPPA scheme with reward mechanism is required to achieve the following design goals.

1) *Man-in-the-middle attack resistance:* In public networks, although there is strong adversary intercepting and changing the response auditing proof information between the CS and TPA, such attack should be detectable and resistable by public verifiers.
2) *Storage correctness guarantee:* On behalf of the original data owner, TPA could periodically check the integrity of remote data without retrieving whole storage data set. Moreover, A high-security auditing mechanism should consider the intentional attack of the malicious CS. When the CS replaces these data blocks challenged by TPA with other data blocks, the proposed auditing scheme should still be work.
3) *Data privacy protection:* Since the reporting data are highly sensitive, the proposed BA-CAPPPA scheme should prevent any adversary (including curious CS) from obtaining enough plain-ciphertext pairs to address a system of linear equations, and thus deriving the primitive data information.
4) *Conditional identity anonymity:* No adversary in cloud-based reporting system can obtain an anonymous identity

of user to deduce the real identity, other than the PKG. The PKG not only enables whistleblowers to get the reward from JD without leaking any identity information, but also could track and reveal the malicious user.

5) *Perfect reward mechanism:* Once it is determined, the user plays a key role in the whole complaint or tip-off case, under the premise of not revealing any sensitive identity information, the JD can pay corresponding incentives to the real data uploader. Meanwhile, when some data are lost or modified, users can prove that he/she is the real data owner, and then gets compensations from CS, whereas others cannot obtain the compensations. Moreover, a dishonest user in system cannot frame the innocent CS to get the additional compensations.
6) *High efficiency:* Due to the volume of reporting users is large, TPA may be entrusted by multiple users at the same time to execute the integrity verification of outsourcing data, thus the auditing task will become more and more onerous. On the other hand, users expect to receive the TPA's auditing results in real time. Therefore, TPA is required to conduct the whole auditing verification process (including the auditing communication overhead and computation costs) with higher efficiency.

IV. Proposed BA-CAPPPA

A. Overview of BA-CAPPPA

As an integration of identity-based signature algorithm, ECC, blinding technique, and verifiable random masking codes, the proposed BA-CAPPPA can provide an efficient identity anonymity and reward mechanism for whistleblowers in clouds. At a high level, it seems that full anonymity could preserve the identity privacy of whistleblowers completely, but it is impossible to trace and reveal a malicious user who has submitted a large amount of junk information to slander others. Meanwhile, it is difficult to realize a secure anonymous reward mechanism without leaking the sensitive identity information of users. To resist the malicious behavior of CS, the proposed BA-CAPPPA takes advantage of the BT to enhance the security of the whole public auditing mechanism while it is still able to be executed efficiently. Since the sensitivity of reporting materials, data privacy protection (DPP) is as important as anonymity of whistleblower. When the remote data stored on CS is destroyed or lost, the real data uploader is supposed to get the compensations, because any data loss or tampering may make the user unable to provide the complete and important witness information to the JD, and thus it cannot obtain the corresponding rewards and crack down on offenders. On the contrary, the honest and courageous user may be blacklisted by the JD, and even mired in lawsuits, which will make it lose their enthusiasm for participating in social management. In conclusion, our BA-CAPPPA scheme consists of the seven functionality properties: CIA, DPP, BT, public batch auditing (PBA), IP, traceability of original storage data (TOSD), and CF. Particularly, we also remark that no other related schemes in the literature, at the time of this research, achieves the aforementioned functionalities simultaneously (see Table I).

TABLE I
COMPARISON OF FUNCTIONAL PROPERTY WITH EXISTING RELEVANT SCHEMES

Schemes	CIA	DPP	BT	PBA	IP	TOSD	CF
PPIR [3]	Yes	Yes	No	Yes	Yes	No	No
IPANM [10]	No	No	Yes	Yes	Yes	No	No
RDIC [24]	No	Yes	No	Yes	No	No	No
CPVPA [27]	No	No	Yes	Yes	No	No	No
ETPPH [31]	Yes	Yes	No	No	Yes	No	No
IAID-PDP [35]	Yes	No	No	Yes	Yes	No	No
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes

B. Construction of BA-CAPPPA

Now, we construct the blockchain-assisted conditional anonymity privacy-preserving public auditing scheme with reward mechanism.

1) Setup: This algorithm is to generate the system parameters in the following steps.

1) Two large prime numbers p and q are selected to define an elliptic curve $\mathbb{E}$ over a finite field F_p. Suppose P is a point of the elliptic curve $\mathbb{E}$, and P is the generator of the additive cyclic group $\mathbb{G} =< P >$ with order q. Choose an integer $s \in \mathbb{Z}_q^*$, and compute $P_{\text{pub}} = sP$ as the master public key.
2) Set a pseudorandom function $Prf : SK_{Prf} \times \{0,1\}^* \times I \rightarrow \mathbb{Z}_q^*$, where Sk_{Prf} denotes the set of secret key for Prf and I is a set of serial numbers. System randomly chooses sk_{Prf}, where $sk_{Prf} \in Sk_{Prf}$, and it is shared secretly by user and TPA.
3) Define five secure hash functions: $h_1 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \longrightarrow \{0,1\}^*$, $h_2 : \{0,1\}^* \times \mathbb{G} \longrightarrow \mathbb{Z}_q^*$, $h_3 : \{0,1\}^* \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, and $h_4 : \{0,1\}^* \times \{0,1\}^* \longrightarrow \mathbb{G}$ and $H : \{0,1\}^* \longrightarrow \{0,1\}^\ell$, respectively, where $*$ and ℓ denote the length of bits, and $\ell << *$. Particularly, H is the hash function specified on the Ethereum blockchain.
4) Choose a secure signature-verification algorithm pair $(SSIG, VER)$, let secure public key encryption-decryption algorithm pair (ENC, DEC), and set PKG's private-public key pair be (α, β), where $\beta = \alpha P$.

The system public parameters are $Para = (p, q, \mathbb{E}, P, P_{\text{pub}}, \beta, h_1, h_2, h_3, h_4, H)$, and the system secret keys are (s, α).

2) AnonyIDGen and KeyExtract: This algorithm is designed to achieve the online registration of user in the public channel, as well as generate user's anonymous identity Aid. Details of this phase are described as follows.

1) Each user U_{id} in system has a unique and real identity $id \in \{0,1\}^\ell$, which chooses a random number $r \in \mathbb{Z}_q^*$ to compute $R = rP$. Then, the user sends the triples $(R, ENC_\beta(id, R), C_{id})$ to PKG for online registration safely. Here, the $C_{id} = SSIG_{ask}(ENC_\beta(id, R))$, the ask is a private key randomly selected by U_{id}, and its corresponding public key is apk. Moreover, U_{id} needs to make apk public and saves the private key ask secretly.
2) When receiving the triples $(R, ENC_\beta(id, R), C_{id})$ from the user, PKG first checks the validity of signature-ciphertext C_{id} with the secure verification algorithm VER_{apk}. If the verification fails, the PKG repulses it by emitting $Error$; otherwise, it decrypts $ENC_\beta(id, R)$ to obtain user's real identity id based on the corresponding decryption algorithm DEC_α. After that, it calculates $Aid = h_1(P_{\text{pub}}, sR, Time) \oplus id$, where the $Time$ shows the validity period of anonymous identity Aid.
3) With the anonymous identity Aid, PKG randomly selects an integer μ from $\mathbb{Z}_q^*$, and computes the

signature private key $SK_{Aid} = \mu + sh_2(Aid, V)$ of user, where $V = \mu P$. Finally, it transmits the quads $(V, ENC_{apk}(Aid, SK_{Aid}), Aid, C_{Aid})$ to the user, where the $C_{Aid} = SSIG_{\alpha}(ENC_{apk}(Aid, SK_{Aid}))$.

4) Once receiving the quads $(V, ENC_{apk}(Aid, SK_{Aid}), Aid, C_{Aid})$ from PKG, without loss of generality, the U_{id} receives the anonymous identity Aid, and recovers its corresponding signature private key SK_{Aid}. Then, it verifies whether the verification equation (1) is correct.

$$SK_{Aid}P \triangleq V + P_{\text{pub}}h_2(Aid, V) \tag{1}$$

If it fails, the user rejects it, and demands PKG to repeat the aforementioned steps to regenerate the private key SK_{Aid}. Otherwise, the U_{id} accepts it.

3) SignGen and DataOutsourcing: This phase is mainly executed by user U_{id} who has ample evidence material to accuse corrupt officials, lawbreakers, and immoralists, etc.

1) Given a reporting file F with the name $filename \in \{0,1\}^*$, user U_{id} compresses it into n blocks, and $F = \{f_1, \ldots, f_i, \ldots, f_n\} \in \mathbb{Z}_q$, and computes $Tag = filename \| n \| SSIG_{SK_{Aid}}(filename \| n)$ as the signature label for F, where $i \in [1, n]$. Simultaneously, user U_{id} randomly chooses an integer $\tau_i \longleftarrow \mathbb{Z}_q^*$, and computes $T_i = \tau_i P = (x_i, y_i)$, $W_i = x_i \mod q$, and $\delta_i = (\tau_i W_i + SK_{Aid} f_i) \mod q$. Then, the U_{id} outputs the metadata set $\{(T_i, W_i)\}_{1 \le i \le n}$, and the signature collection $\psi = \{\delta_i\}_{1 \le i \le n}$.
2) To guarantee the confidentiality of reporting file F, user U_{id} utilizes the pseudorandom function Prf to generate $n+1$ blind factors $(\omega_1, \ldots, \omega_n, \omega) \leftarrow Prf_{sk_{Prf}}(filename, i)$, which is used to blind each file block f_i as $f_i' = f_i + {\omega_i}^{-1} h_3(Tag || i)$ under the secret key sk_{Prf}. Thus, the reporting file $F = \{f_1, \ldots, f_i, \ldots, f_n\}$ is blinded to be $F' = \{f_1', \ldots, f_i', \cdots f_n'\}$. In addition, the U_{id} computes $\sigma = \omega h_4(Aid, Tag)$.
3) According to the structure of MHT and every leaf node $H(f_i')_{i \in [1,n]}$, the user computes root hash value $H(\Lambda)_U$ in chronological order [39]. After that, it sends the auxiliary data $(Tag, filename, \{H(f_i'), (T_i, W_i)\}_{i \in [1,n]})$ to the blockchain, and uploads the storage data $(F', Tag, filename, \psi, H(\Lambda)_U, \sigma)$ to CS.
4) When receiving the auxiliary data that need to be chained, the blockchain checks the validity of $Tag = filename \| n \| SSIG_{SK_{Aid}}(filename \| n)$ with the secure verification algorithm VER. If the verification fails, the blockchain believes it invalid; otherwise, it computes a signature value $key_i^{filename}$ for each metadata $\{(T_i, W_i)\}_{i \in [1,n]}$ based on the description of Ethereum blockchain in Section II-B, and stores the data pairs $\langle key_i^{filename}, (T_i, W_i) \rangle$ in LevelDB. Finally, it transmits storage-query message $SQ = (H(\Lambda)_B, Key)$ and Key to CS and Blockchain, severally. Here, the $H(\Lambda)_B$ is the root hash of MHT, and $Key = \{key_i^{filename}\}_{i \in [1,n]}$.
5) Once receiving the storage-query message SQ from blockchain, CS verifies the validity of Tag with the VER, and checks whether the formula $H(\Lambda)_B \triangleq H(\Lambda)_U$ holds. If both the two conditions hold, the CS accepts the storage data $(F', Tag, \psi, H(\Lambda)_U, \sigma, Key)$.

4) ChallengeGen: This phase is mainly run by TPA, and the detailed steps are as follows.

1) A random subset $L = \{\ell_1, \ldots, \ell_\theta\}$ of the universal set $[1, n]$ is picked, where the subset L locates reported file blocks, which need to be validated.
2) A random integer $\nu_j \in \mathbb{Z}_q^*$ for each $j \in L$ is selected. Then, the TPA sends challenge massage $Chal = \{j, \nu_j\}_{j \in L}$ to CS.

5) ProofGen: Upon receiving the challenge message Chal $= (j, \nu_j)$, the CS locates the corresponding outsourced subfiles L based on the file name $filename$, and generates the respond proof information as follows.

1) Pick a random integer $\epsilon \in \mathbb{Z}_q^*$ to compute $\varepsilon = \epsilon^{-1} P$.
2) Compute the combined message $\xi = \sum_{j=1}^{j=\theta} \nu_j f_j'$, and the aggregate signature information $\delta = \sum_{j=1}^{j=\theta} \nu_j \delta_j + \epsilon^{-1} h_3(\xi, Tag)$.
3) Send the proof information $Proof = (\xi, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j \in L})$ to the TPA.

6) ProofVerification: Based on the storage-query function of Ethereum, the TPA checks the integrity of auditing proof information in the following steps.

1) Once receiving the proof information $Proof$ from the CS, the TPA first check the correctness of the inquiry index $\{key_j^{filename}\}_{j \in L}$, and then transmits it to blockchain, so as to retrieve the validating information $H(\Lambda)_B$ and $(T_j, W_j)_{\{1 \le j \le \theta\}}$. Compared whether these two root hash values $H(\Lambda)_U \triangleq H(\Lambda)_B$ are equal. If the outputs is $False$, TPA quits via emitting $Error$; otherwise, TPA performs the following steps.
2) According to the secret key sk_{Prf}, the TPA computes $\omega_j^{-1} \leftarrow Prf_{sk_{Prf}}(filename, j)$, and calculates verification coefficients $\partial = \sum_{j=1}^{j=\theta} \nu_j T_j W_j$, and $\lambda = \sum_{j=1}^{j=\theta} \nu_j \omega_j^{-1} h_3(Tag || j)$, where $j \in [1, \theta]$. After that, it checks the following verification equation (2) whether or it holds:

$$P \triangleq \partial + (\xi - \lambda)(V + P_{\text{pub}}h_2(Aid, V)) + \varepsilon h_3(\xi, Tag). \tag{2}$$

If the verification equation holds, TPA takes the auditing result as $True$; otherwise, it takes the auditing result as $False$.

7) RewardCertification: The reward mechanism is executed by the JD, which includes IP and CF. The details of the two process are described as follows.

1) When JD plans to pay for the real data or file uploader, user U_{id} will show he/she is the real data uploader. That is, U_{id} sends the triples $(R, ENC_{\beta}(id, R), C_{id})$ to PKG, which obtains the real identity (id, R). With the master private key s and user's anonymous identity Aid, PKG recovers the user's real identity id by calculating $id = h_1(P_{\text{pub}}, sR, time) \oplus Aid$. If the identity of U_{id} is consistent, PKG sends the result $True$ to JD via a secure channel. Finally, the real data uploader is rewarded without revealing any sensitive identity information.
2) Once the auditing result output is $False$, the CS should provide the corresponding compensations to user U_{id}. Specifically, the U_{id} generates the random number $\omega \leftarrow Prf_{sk_{Prf}}(filename, i+1)$ with the secret key sk_{Prf}, and computes $\sigma = \omega h_4(Aid, Tag)$. Meanwhile, the U_{id} calculates the signature value $SSIG_{ask}(\sigma)$ with the signature algorithm $SSIG(\cdot)$ and its private key ask. Then, it sends $SSIG_{ask}(\sigma)$ to CS, as well as submits the whole

claim event to JD for filing. By taking full advantage of the preimage $SSIG_{ask}(\sigma)$, CS verifies its validity. If it is true, CS pays a compensation to U_{id}; otherwise, the CS refuses to pay it compensations and the JD will penalize it.

V. Evaluation of the Proposed Mechanism

A. Correctness

For the verification equation (1), the correctness is proved as follows:

$$\begin{aligned} SK_{Aid}P &= (\mu + sh_2(Aid, V))P \\ &= V + P_{pub}h_2(Aid, V). \end{aligned}$$

The correctness of the verification equation (2) is elaborated as follows:

$$\begin{aligned} \delta P &= \left(\sum_{j=1}^{j=\theta} \nu_j \delta_j + \epsilon^{-1} h_3(\xi, Tag)\right) P \\ &= \sum_{j=1}^{j=\theta} \nu_j (\tau_j W_j + f_j SK_{Aid})P + \epsilon^{-1} h_3(\xi, Tag)P \\ &= \sum_{j=1}^{j=\theta} \nu_j T_j W_j + \varepsilon h_3(\xi, Tag) \\ &\quad + \left(\sum_{j=1}^{j=\theta} \nu_j f'_j - \sum_{j=1}^{j=\theta} \nu_j \omega_j^{-1} h_3(Tag||j)\right) SK_{Aid}P \\ &= \partial + \varepsilon h_3(\xi, Tag) + \left(\xi - \sum_{j=1}^{j=\theta} \nu_j \omega_j^{-1} h_3(Tag||j)\right) \\ &\quad \times (V + P_{\text{pub}} h_2(Aid, V)) \\ &= \partial + (\xi - \lambda)(V + P_{\text{pub}} h_2(Aid, V)) + \varepsilon h_3(\xi, Tag). \end{aligned}$$

Thus, the verification equation (2) that $\delta P = \partial + (\xi - \lambda)(V + P_{\text{pub}} h_2(Aid, V)) + \varepsilon h_4(\varepsilon)$ holds.

B. Security Proof of BA-CAPPPA

Inspired by the literature [20], which first proposed a formal security proof of cloud storage correctness, we modify it to satisfy our auditing scheme. The security proof of BA-CAPPPA includes the five aspects: man-in-the-middle attack resistance, storage correctness guarantee, DPP, CIA, and reward mechanism.

Theorem 1: In our BA-CAPPPA scheme, it is computationally infeasible for an outside adversary A to launch a man-in-the-middle attack to pass the verification process, supposing that the hardness assumption of ECDLP problem holds.

Proof: An outside adversary A is well trained and can launch active attacks online. In *Setup*, the adversary A and challenger C query constantly to obtain public key of the system, without awareness of its secret parameter (s, α). In particular, the challenger C possesses a large number of validated parameter lists for responding to queries. In *ProofGen*, Once receiving an auditing challenge message $Chal = (j, \nu_j)_{\{j \in L\}}$ from TPA, the CS generates a correct response auditing proof information $Proof = (\xi, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j \in L})$, and sends it to the TPA, where $\xi = \sum_{j=1}^{j=\theta} \nu_j f'_j$. In *ProofVerification*, TPA verifies the integrity of the data by checking the following equation:

$$\delta P \triangleq \partial + (\xi - \lambda)(V + P_{\text{pub}} h_2(Aid, V)) + \varepsilon h_3(\xi, Tag)$$

where $\partial = \sum_{j=1}^{j=\theta} \nu_j T_j W_j$ and $\lambda = \sum_{j=1}^{j=\theta} \nu_j \omega_j^{-1} h_3(Tag||i)$.

Now, we will show how an online active adversary A can conduct the man-in-the-middle attack. An outside adversary A initially intrudes into the CS, and forges a data block as $f'_j = f'_j + m'_j$, then records m'_j. In *Challenge*, the cunning adversary A eavesdrops on the challenge message $Chal = (j, \nu_j)_{\{j \in L\}}$. In *ProofGen*, since the data block f'_j is tampered with as $f'_j + m'_j$, the CS computes the proof information to $Proof^* = (\xi^*, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j \in L})$, where $\xi^* = \sum_{j=1}^{j=\theta} \nu_j (f'_j + m'_j)$, and sends it to TPA. At the same time, the adversary A intercepts $Proof^* = (\xi^*, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j \in L})$, and computes $\chi = \sum_{j=1}^{j=\theta} \nu_j m'_j$ (the value χ can also be preprocessed), $\xi = \xi^* - \chi$. Hence, the adversary A can compute the valid auditing proof information, and forwards it to TPA, which cannot discover whether some data have been forged.

Such an adversary A can attack successfully in many public auditing schemes [17], [24], [31]. If a secure channel is established between TPA and CS, it can prevent communications between two entities from being eavesdropped, forged, and replayed. In public network environment, BA-CAPPPA also effectively resists the man-in-the-middle attack by improving a verifiable random masking code technique [37]. Specifically, in *PoorfGen*, the CS chooses a random integer $\epsilon \in \mathbb{Z}_q^*$, and computes the response aggregate signature information $\delta = \sum_{j=1}^{j=\theta} \nu_j \delta_j + \epsilon^{-1} h_3(\xi, Tag)$, where $\varepsilon = \epsilon^{-1} P$. Since ϵ is selected by the CS randomly, $\epsilon^{-1} \in \mathbb{Z}_q^*$ is the inverse of ϵ, and it would reselect for each data integrity verification. That is, each random number ϵ^{-1} is equal to 0 with a probability $1/q$. Consequently, any outside adversary A cannot pass the auditing verification equation by forging the interaction data between the CS and TPA, unless it can address the ECDLP problem.

Theorem 2: In our auditing scheme, based on the ECDLP, it is computational infeasible for an adversary A (including the malicious CS) to generate a valid response auditing proof information that can pass the verification equation. In other words, the storage correctness guarantee means that no adversary A could obtain a valid game to destroy the integrity of remote storage data with a nonnegligible probability Υ.

Proof: If existing an adversary A (including the malicious CS) breaks the storage correctness of our auditing scheme with a nonnegligible probability Υ in *Games 1* and *2*, we can construct a challenger C (including the TPA) and the adversary A to make a continuous query, and thus solving the hardness assumption of ECDLP problem with a nonnegligible probability Υ'. Now, we prove the storage correctness guarantee in terms of *Game 1* and *Game 2* with corresponding detailed security analysis as follows.

Game 1: In this game, the adversary A (including the malicious CS) could be trained to forge, delete, or replace some reporting data blocks, and further generate an imitated response proof information to pass the integrity verification process. Specifically, when receiving a challenge

message $Chal = (j, \nu_j)_{j\in L}$ from the challenger C, the adversary A does not follow the auditing procedures to generate correct response auditing proof information honestly, but successfully forges the response auditing proof information $Proof^* = (\xi^*, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j\in L})$ with a nonnegligible probability Υ, where $\xi = \sum_{j=1}^{j=\theta} \nu_j f'_j \neq \xi^*$. There at least exist an encrypted data block $f_j^{'*} \neq f'_j$ that has been tampered, replaced, or deleted by the malicious CS, and thus $\triangle f'_j = f_j^{'*} - f'_j \neq 0$, $\triangle\xi = \xi^* - \xi \neq 0$, and $h_3(\xi^*, Tag) - h_3(\xi, Tag) \neq 0$, where $j \in \{1, 2, \ldots, n\}$. Therefore, the forged response auditing proof information $Proof^* = (\xi^*, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j\in L})$ can pass the following verifications equation:

$$\delta P = \partial + (\xi^* - \lambda)(V + P_{\text{pub}} h_2(Aid, V)) + \varepsilon h_3(\xi^*, Tag)$$

where $\partial = \sum_{j=1}^{j=\theta} \nu_j T_j W_j$ and $\lambda = \sum_{j=1}^{j=\theta} \nu_j \omega_j^{-1} h_3(Tag||i)$.

As a matter of fact, when a honest CS receives a challenge message $Chal = (j, \nu_j)_{j\in L}$ from TPA, it can generate a correct response auditing proof information $Proof = (\xi, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j\in L})$ as required, where $\xi = \sum_{j=1}^{j=\theta} \nu_j f'_j$ and $\delta = \sum_{j=1}^{j=\theta} \nu_j \delta_j + \epsilon^{-1} h_3(\xi, Tag)$. Thus, it could satisfy the following verification equation:

$$\delta P = \partial + (\xi - \lambda)(V + P_{\text{pub}} h_2(Aid, V)) + \varepsilon h_3(\xi, Tag)$$

where $\partial = \sum_{j=1}^{j=\theta} \nu_j T_j W_j$ and $\lambda = \sum_{j=1}^{j=\theta} \nu_j \omega_j^{-1} h_3(Tag||i)$.

According to the aforementioned two verification equations, we first get $\xi^* M + \varepsilon h_3(\xi^*, Tag) = \xi M + \varepsilon h_3(\xi, Tag)$, so we can further obtain that

$$\sum_{j=1}^{j=\theta} \nu_j f_j^{'*} M + \varepsilon h_3(\xi^*, Tag) = \sum_{j=1}^{j=\theta} \nu_j f'_j M + \varepsilon h_3(\xi, Tag).$$

Here, we define $V + P_{\text{pub}} h_2(Aid, V) = M \in \mathbb{G}$, thus we can get the equation: $\sum_{j=1}^{j=\theta} \nu_j (f_j^{'*} - f'_j) M = \varepsilon(h_3(\xi^*, Tag) - h_3(\xi, Tag))$. Owing to $h_3(\xi^*, Tag) - h_3(\xi, Tag) \neq 0$, the malicious CS can obtain: $M = -((h_3(\xi^*, Tag) - h_3(\xi, Tag))/\sum_{j=1}^{j=\theta} \triangle f'_j \nu_j)\varepsilon$. There is at least a data block $f_j^{'*} \neq f'_j$, and $\triangle f'_j = f_j^{'*} - f'_j \neq 0$. It is clear that the random integer $\nu_j \in \mathbb{Z}_q^*$ is 0 only with a probability of $1/q$, so as to the denominator is 0 at most with the probability $1/q$. As the malicious CS wins *Game 1* by forging a valid response auditing proof information $Proof^* = (\xi^*, \delta, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j\in L})$ with a high probability of Υ. Thus, the challenger C will have a nonnegligible probability $\Upsilon' = (1 - 1/q)\Upsilon$ to address the hardness assumption of ECDLP by running adversary A (malicious CS), which leads to a contradiction.

Game 2: On the basis of *Game 1*, the adversary A (including the malicious CS) further forged aggregate signature to generate a response auditing proof information, so as to pass the integrity verification process. Particularly, the proposed BA-CAPPPA scheme introduces the booming BT [11] to ensure the storage correctness guarantee, even if an adversary A can be trained to forge the aggregate signature successfully, it cannot modify the set of metadata stored on the blockchain, because the auxiliary data cannot be tampered once it is stored in the blockchain. The details are described as follows.

Upon receiving the challenge message $Chal = (j, \nu_j)_{j\in L}$ from the challenger C, the adversary A can successfully generate the forged response auditing proof information as $Proof = (\xi, \delta^*, \varepsilon, H(\Lambda)_U, \{key_j^{filename}\}_{j\in L})$ with a nonnegligible probability Υ, where $\delta^* = \sum_{j=1}^{j=\theta} \nu_j \delta_j^* + \epsilon^{-1} h_3(\xi, Tag)$, and then transmit it to the TPA.

In the auditing process, based on the description of Ethereum blockchain in Section II-B, the TPA retrieves the corresponding metadata set $\{(T_j, W_j)_{j\in L}\}$ with the query value $\{key_j^{filename}\}_{j\in L}$. Hence, the TPA checks whether the following verification equation holds:

$$\delta^* P = \sum_{j=1}^{j=\theta} \nu_j T_j W_j + (\xi - \lambda) M + \varepsilon h_3(\xi, Tag)$$

where $\lambda = \sum_{j=1}^{j=\theta} \nu_j \omega_j^{-1} h_3(Tag||j), \xi = \sum_{j=1}^{j=\theta} \nu_j f'_j$, and $M = V + P_{\text{pub}} h_2(Aid, V) \in \mathbb{G}$.

Namely, since (T_j, W_j) is stored on the blockchain, it is imtamperability and full-transparent. No adversary A could forge it and attempt to pass the auditing verification equation. Thus, our proposed BA-CAPPPA scheme successfully take advantage of the BT to resist the malicious or intentional attack of CS.

Theorem 3: The proposed BA-CAPPPA scheme with reward mechanism achieves CIA.

Proof: The theorem is proved from the two cases: i) No adversary A (including the malicious user) could deduce the real identity id according to the anonymous identity Aid of user U_{id}, except that it can obtain the system master private key s. ii) The PKG can reveal and track the real identity of malicious user id with the system master private key s.

Case i). Before a user U_{id} sends the registration information (R, id) to the PKG, U_{id} first computes the triples $(R, ENC_\beta(id, R), C_{id})$ by its own private key ask. Then, checks the validity of verification ciphertext C_{id} with by the secure verification algorithm VER, and utilizes the PKG's private key α to decrypt the identity ciphertext $ENC_\beta(id, R)$. It not only ensures that any source sender is uniquely trusted, but also achieves the high security of (R, id) in the public network. Particularly, the $R = rP$ is randomized by a random integer $r \in \mathbb{Z}_q^*$, and $Aid = h_1(P_{pub}, sR, Time) \oplus id$ is generated under the master secret key s of the PKG, without mastering r and s, it is computationally infeasible for any adversary A to compute $sR = s(rP)$ due to the hardness assumption of ECDLP. Hence, the probability that any adversary A could successfully deduce the real identity id from anonymous identity Aid of users is $1/q^2$, which is negligible in the PPT $\Re$.

Case ii). Supposed that existing a malicious user with an anonymous identity Aid abusing the reporting system, PKG could trace and revoke the real identity of the malicious user. Specifically, with the master secret key s, PKG could compute $id = h_1(P_{pub}, sR, Time) \oplus Aid$. Consequently, our proposed BA-CAPPPA achieves CIA.

Theorem 4: The proposed BA-CAPPPA scheme with reward mechanism satisfies data privacy preservation.

Proof: Now, we prove that any adversary A (including a curious CS) could not obtain a user's original data block f_i from the upload encrypted file F'. Specifically, $f'_i = f_i + \omega_i{}^{-1} h_3(Tag||i)$. Here, $\omega_i{}^{-1}$ is generated by the pseudorandom function Prf, which has a secret key sk_{Prf} confidentially. The

adversary A could not derive the secret key sk_{Prf} for Prf, and thus cannot get the original data block f_i. In addition, since ${\omega_i}^{-1}$ is considered as the random masking in [9], so as to resist the curious CS from getting enough plain-ciphertext pairs (f_i, f_i') to solve a linear system of equations. Without loss of generality, the probability that adversary A could correctly deduce the random masking value ${\omega_i}^{-1}$ is only $1/q$, which is undoubtedly equivalent to solving the hardness assumption of ECDLP. Hence, our proposed BA-CAPPPA scheme satisfies data privacy preservation. ■

Theorem 5: BA-CAPPPA achieves the high security of reward mechanism, including the IP and CF.

Proof: From *Theorem 3*, we can get BA-CAPPPA scheme satisfies the property of IP. In particular, when JD intends to pay for the real data uploader, the user could not directly conduct the information interaction process of identity authentication with JD. Meanwhile, the source of anonymous identity Aid can be traced via PKG, which is a fully trusted in the reporting system. Then, the PKG sends user's authentication results to JD through a secure channel. As a result, the real data uploader can normally receive incentives without exposing any sensitive identifying information.

In addition, when U_{id} submits the preimage $SSIG_{ask}(\sigma)$ to CS, the CS verifies its validity by the signature public key apk. If it is true, the U_{id} can obtain compensations accordingly, and the others cannot get the compensation by impersonating $user$. Since $SSIG_{ask}(\sigma)$ is signature value, where $\sigma = \omega h_4(Aid, Tag)$, and $\omega \leftarrow Prf_{sk_{Prf}}(filename, i+1)$. h_4 is cryptographic hash function, the random integer ω and the user's private key ask is randomly picked and kept in secret, no adversary A can impersonate a real user U_{id} to deceive CS and JD. Thus, based on the security of the picked signature-verification algorithm pair $(SSIG, VER)$, our BA-CAPPPA scheme realizes the character of compensations. ■

VI. Comprehensive Performance Evaluation

In this section, we first analyze the overhead of our scheme and existing schemes PPIR [3], IPANM [10], RDIC [24], CPVPA [27], and IAID-PDP [35], including the signature computation costs, auditing communication overhead and computation costs. Then, we evaluate the real performance of our scheme in experiments. Particularly, all the experiments are done on a windows 10 system with Inter(R) Core(TM) i5-9500 CPU 3.00 GHz and 8.00 GB of RAM laptop. We employ Java language with JDK 1.7 update 1, and the jECC has been imported as the curve library. In the implementation, we set the parameters that the curve we employ is Secp256k1.[1] All the results of experiments are represent 30 trials on average.

In order to facilitate the performance analysis among our scheme and several related schemes, we unify some notations to represent the corresponding cryptographic operation. Specifically, $T_{Pair}, T_{Mul}, T_{Add}, T_{Exp}, T_{mul}, T_{Ha}$ are denoted by the running time of a bilinear pairing operation, the running time of a scalar multiplication operation, the running time of a point addition operation, the running time of a modular exponentiation operation, the running time of a general multiplication operation, and the running time of a hash function operation, respectively. $|G|$ denotes the bit length of a point in the additive cyclic group $\mathbb{G}$. $|G_1|$ and $|G_2|$ severally denote the bit length of elements of groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with a bilinear map, as described in the literature [4]. We also denote $|pf|$ and $|\ell|$ by the size of the zero-knowledge proof and the size of an identity. Moreover, the $|\nabla|$ denotes the number of users participating in ring signature.

[1]The q is FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F, a is 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000, b is 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007, P is 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 as the generator, and the order of P is FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141, all the parameters are given in hex.

TABLE II
Comparison of Signature Computation Costs

Schemes	The signature costs of single data block	Execution time (μs)
PPIR [3]	$(\nabla+2)T_{Mul}+\nabla T_{Add}+T_{Ha}$	$\approx$ 26122.2
IPANM [10]	$4T_{Pair}+(2\nabla+4)T_{Exp}+$ $(\nabla+3)T_{Ha}+(\nabla+3)T_{mul}$	$\approx$ 45901.1
RDIC [24]	$4T_{Exp}+T_{mul}+2T_{Ha}$	$\approx$ 7088.7
CPVPA [27]	$5T_{Exp}+4T_{mul}+3T_{Ha}$	$\approx$ 8877.1
IAID-PDP [35]	$(3\nabla+2)T_{Exp}+$ $(\nabla+2)T_{mul}+T_{Ha}$	$\approx$ 36458.6
Our scheme	$4T_{Mul}+3T_{mul}+T_{Add}$	$\approx$ 10676.7

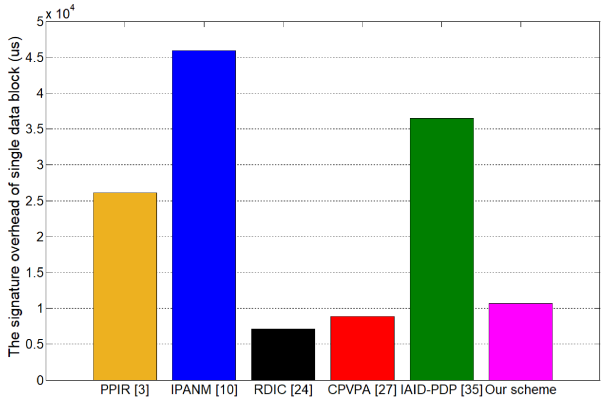

Fig. 4. Signature computation costs comparison of single data block.

The signature computation costs of single data block, in terms of the generation of signature and metadata, is listed in Table II. The comparison of the implementation results of the single data block signature computation costs between our scheme and several related schemes PPIR [3], IPANM [10], RDIC [24], CPVPA [27], and IAID-PDP [35], which are shown in Fig. 4. It demonstrates that BA-CAPPPA is much less than PPIR [3], IPANM [10], and IAID-PDP [35], and only BA-CAPPPA could achieve the confidentiality of sensitive outsourced data. Our scheme is indeed higher than RDIC [24], and slightly higher than CPVPA [27], but BA-CAPPPA could realize CIA and guarantee DPP simultaneously. Therefore, BA-CAPPPA achieves reasonable signature costs.

Based on the interaction of TPA and CS in public auditing, we give the communication overhead of the proposed scheme and several related schemes from two parts: challenge message and response proof information. Through detailed analysis and demonstration, we obtain that the total auditing communication overhead in PPIR [3] is $|G_1|\theta+(\nabla+1)|G_1| \approx 1024\theta+11164$ (bits), IPANM [10] is $2|q|\theta+|q|+|G_1|+|G_2| \approx 512\theta+3328$ (bits), RDIC [24] is $|q|\theta+2|q|+|G_1|+$

TABLE III
COMPARISON OF AUDITING COMPUTATION TASKS

Schemes	The CS's auditing computation costs (μs)	The TPA's auditing computation costs (μs)	The total auditing verification costs (μs)
PPIR [3]	$(2T_{Mul} + \nabla T_{Add} + T_{Ha})\theta$ $\approx 4470.2\theta$	$2T_{Exp}\theta + 2T_{Pair} + (T_{Mul} + T_{Add})\nabla$ $\approx 2340\theta + 32638$	$(2T_{Mul} + 2T_{Exp} + \nabla T_{Add} + T_{Ha})\theta$ $+ 4T_{Exp} + (T_{Mul} + T_{Add})\nabla$ $\approx 6810.2\theta + 32638$
IPANM [10]	$(T_{Exp} + T_{mul})\theta + T_{Pair}$ $+ T_{Exp} + T_{mul} + T_{Ha}$ $\approx 1170.9\theta + 6605.7$	$(T_{Exp} + T_{mul} + T_{Ha})\theta + 2T_{Pair}$ $+ 3T_{Exp} + T_{mul}$ $\approx 1178.7\theta + 14364.9$	$(2T_{Exp} + 2T_{mul} + T_{Ha})\theta + 3T_{Pair}$ $+ 4T_{Exp} + 2T_{mul} + T_{Ha}$ $\approx 2948.6\theta + 20970.6$
RDIC [24]	$T_{Pair} + 2T_{Exp} + T_{Ha}$ ≈ 7774.8	$(T_{Pair} + 3T_{Exp} + 2T_{mul} + T_{Ha})\theta$ $+ T_{Exp} - 2T_{mul}$ $\approx 8946.6\theta + 1168.2$	$(T_{Pair} + 3T_{Exp} + 2T_{mul} + T_{Ha})\theta$ $+ T_{Pair} + 3T_{Exp} + T_{Ha} - 2T_{mul}$ $\approx 8946.6\theta + 8943$
CPVPA [27]	$(T_{Exp} + 2T_{mul})\theta - 2T_{mul}$ $\approx 1171.8\theta$	$(3T_{Exp} + 3T_{mul} + 2T_{Ha})\theta + 4T_{Pair}$ $+ 2T_{Ha} \approx 3528.3\theta + 24048$	$(4T_{Exp} + 5T_{mul} + 2T_{Ha})\theta + 4T_{Pair}$ $+ 2T_{Exp} - 2T_{mul} \approx 4700.1\theta + 24048$
IAID-PDP [35]	$(T_{Exp} + 2T_{mul})\theta - T_{mul}$ $\approx 1171.8\theta$	$(T_{Exp} + T_{mul} + T_{Ha})\theta + T_{Ha}$ $+ (\nabla + 1)T_{Pair} + (\nabla + 1)T_{Exp}$ $+ (\nabla - 1)T_{mul} \approx 1178.7\theta + 72582$	$(2T_{Exp} + 3T_{mul} + T_{Ha})\theta + T_{Ha}$ $+ (\nabla + 1)T_{Pair} + (\nabla + 1)T_{Exp}$ $+ (\nabla - 2)T_{mul} \approx 2350.5\theta + 72582$
Our scheme	$2T_{mul}\theta + T_{Mul}$ $\approx 1.8\theta + 2165.2$	$(T_{Mul} + 2T_{mul})\theta + 4T_{Mul} + T_{mul}$ $+3T_{Add} \approx 2167\theta + 8701.3$	$(T_{Mul} + 4T_{mul})\theta + 5T_{Mul} + T_{mul}$ $+3T_{Add} \approx 2168.8\theta + 10866.5$

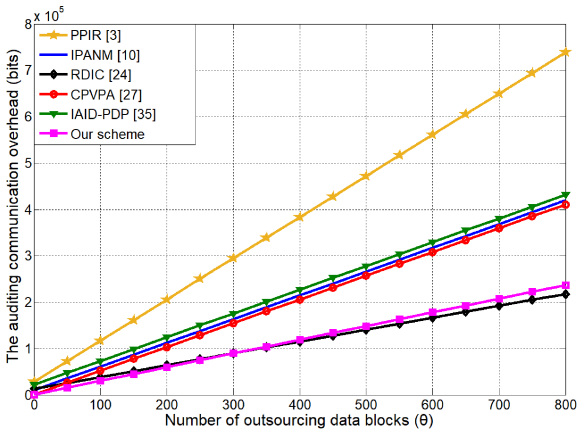

Fig. 5. Auditing communication overhead comparison.

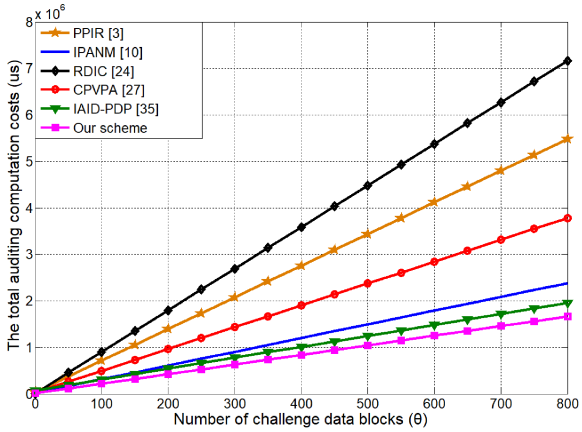

Fig. 6. Total auditing verification costs comparison.

$2|G_2| \approx 256\theta + 5452$ (bits), CPVPA [27] is $2|q|\theta + 2|q| + 2|G_1| \approx 512\theta + 2560$ (bits), and IAID-PDP [35] is $2|q|\theta + |q| + \nabla|G_1| \approx 512\theta + 10496$ (bits), respectively. Our scheme is $(|\ell| + |q|)\theta + 3|q| + |G| \approx 320\theta + 1024$ (bits) in terms of auditing communication overhead. The implementation results of the communication overhead in Fig. 5. BA-CAPPPA realizes a sound communication overhead in the integrity verification phase. Specifically, with the growth of the number of the challenge data blocks, the communication overhead in BA-CAPPPA is much lower than PPIR [3], IPANM [10], RDIC [24] CPVPA [27], and IAID-PDP [35]. When the number of the challenge data blocks is more than 320, we can see that BA-CAPPPA has a slightly higher than RDIC [24] in communication overhead. It is noted that the detection probability is greater than 95% when $\theta = 300$ [5]. At the same time, only BA-CAPPPA could resist the malicious CS by introducing BT, as well as achieve the TOSD and compensation functionalities.

In addition, the total auditing verification costs is listed in Table III, which consists of the CS's auditing computation costs and the auditing computation costs of TPA. In implementation results of Fig. 6, it proves that the integrity computation costs are much efficient than PPIR [3], IPANM [10], RDIC [24], CPVPA [27], and IAID-PDP [35]. This is because BA-CAPPPA is based on ECC, which does not incur time-consuming bilinear pair operations, modular exponentiations, and a hash maps to the multiplicative group. Therefore, compared with the aforementioned related schemes, BA-CAPPPA is more suitable for deployment in complex cloud-based network reporting system.

VII. CONCLUSIONS AND FUTURE WORK

In this article, we have proposed BA-CAPPPA scheme with reward mechanism, and formalize its system and threat model. By using the elliptic curve cryptosystem and identity-based public key cryptography, we have designed a novel algorithm of conditional anonymity, in which only PKG could revoke, trace, and prevent the real identities of the malicious users. Meanwhile, BA-CAPPPA achieves anonymity rewording and CF. We have integrated the Ethereum blockchain technique into our scheme to ensure the integrity of remote data, it could significantly enhance the security of the whole auditing verification process while it retains data integrity checking, IP, and compensation functionalities. We provide detailed security analysis and conduct comprehensive performance evaluation to demonstrate that BA-CAPPPA is provably secure and efficient. Regarding future work, we intend to further investigate how to combine Ethereum with other public key cryptography to resist collusion, DoS, and

other attacks, without sacrificing the security, performance, and versatility of public auditing mechanism.

REFERENCES

[1] W. Fang, G. Wang, G. B. Giannakis, Q. Liu, X. Wang, and H. Deng, "Channel-dependent scheduling in wireless energy transfer for mobile devices," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3330–3340, Mar. 2020.

[2] J. Ma, R. Huang, T. Qiu, B. Chen, and A. K. Sangaiah, "A survey of mobile social networks: Applications, social characteristics, and challenges," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3932–3947, Dec. 2018.

[3] H. Wang, D. He, and J. Yu, "Privacy-preserving incentive and rewarding scheme for crowd computing in social media," *Inf. Sci.*, vol. 470, pp. 15–27, 2019.

[4] C. Wang, S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.

[5] D. He, H. Wang, J. Zhang, and L. Wang, "Insecurity of an identity-based public auditing protocol for the outsourced data in cloud storage," *Inf. Sci.*, vol. 375, pp. 48–53, 2017.

[6] H. Yan, J. Li, and Y. Zhang, "Remote data checking with a designated verifier in cloud storage," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1788–1797, Jun. 2020.

[7] M. Conti and N. Dragoni, "A survey of san in the middle attacks," *IEEE Commun. Surv. Tut.*, vol. 18, no. 3, pp. 2027–2051, Jul.–Sep. 2016.

[8] H. Wang, Q. Wang, and D. He, "Blockchain-based private provable data possession," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 5, pp. 2379–2389, Sep./Oct. 2021.

[9] X. Zhang, J. Zhao, C. Xu, H. Li, and Y. Zhang, "CIPPPA: Conditional identity privacy-preserving public auditing for cloud-based WBANs against malicious auditors," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2019.2927219.

[10] L. Huang *et al.*, "IPANM: Incentive public auditing scheme for non-manager groups in clouds," *IEEE Trans. Dependable Secur. Comput.*, to be published, doi: 10.1109/TDSC.2020.3004827.

[11] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts," in *Proc. Int. Conf. Principles Secur. Trust*, 2017, pp. 164–186.

[12] X. Liu, R. Deng, K. K. R. Choo, Y. Yang, and H. H. Pang, "Privacy-preserving outsourced calculation toolkit in the cloud," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 5, pp. 898–911, Sep./Oct. 2020.

[13] Z. Hua, Y. Zhang, and Y. Zhou, "Two-dimensional modular chaotification system for improving chaos complexity," *IEEE Trans. Signal Process.*, vol. 68, pp. 1937–1949, 2020.

[14] C. Jiang, C. Xu, and Y. Zhang, "PFLM: Privacy-preserving federated learning with membership proof," *Inf. Sci.*, vol. 567, pp. 288–311, 2021.

[15] H. Li, Y. Yang, Y. Dai, S. Yu, and Y. Xiang, "Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 484–494, Apr.–Jun. 2020.

[16] X. Zhang, C. Huang, C. Xu, Y. Zhang, and H. Wang, "Key-leakage resilient encrypted data aggregation with lightweight verification in fog-assisted smart grids," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8234–8245, May 2021.

[17] Y. Zhang, T. Zhang, R. Guo, S. Xu, and D. Zheng, "Traceable dynamic public auditing with identity privacy preserving for cloud storage," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 11, pp. 5653–5672, 2019.

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

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

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

[21] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," *IEEE Trans. Serv. Comput.*, vol. 8, no. 1, pp. 92–106, Jan./Feb. 2015.

[22] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Trans. Inf. Forensic Secur.*, vol. 11, no. 6, pp. 1362–1375, Jun. 2016.

[23] A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*. Berlin, Germany: Springer, 1984, pp. 47–53.

[24] Y. Yu *et al.*, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans. Inf. Forensic Secur.*, vol. 12, no. 4, pp. 767–778, Apr. 2017.

[25] D. Abbasinezhad-Mood, A. Ostad-Sharif, and M. Nikooghadam, "Efficient provably-secure privacy-preserving signature-based key establishment protocol," *Ad Hoc Netw.*, vol. 100, pp. 1–12, 2020.

[26] X. Zhang, J. Zhao, L. Mu, Y. Tang, and C. Xu, "Identity-based proxy-oriented outsourcing with public auditing in cloud-based medical cyber-physical systems," *Pervasive Mobile Comput.*, vol. 56, pp. 18–28, 2019.

[27] Y. Zhang, C. Xu, X. Lin, and X. S. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 923–937, Jul.–Sep. 2021.

[28] X. Zhang, J. Zhao, C. Xu, H. Wang, and Y. Zhang, "DOPIV: Post-quantum secure identity-based data outsourcing with public integrity verification in cloud storage," *IEEE Trans. Serv. Comput.*, to be published, doi: 10.1109/TSC.2019.2942297.

[29] C. M. Yang, H. C. Lin, P. Chang, and W. S. Jian, "Taiwan's perspective on electronic medical records' security and privacy protection: Lessons learned from HIPAA," *Comput. Methods Programs Biomed.*, vol. 82, no. 3, pp. 277–282, 2006.

[30] X. Yang, W. Wu, J. K. Liu, and X. Chen, "Lightweight anonymous authentication for ad hoc group: A ring signature approach," in *Proc. Int. Conf. Provable Secur.*, 2015, pp. 215–226.

[31] W, Tang, J. Ren, and Y. Zhang, "Enabling trusted and privacy-preserving healthcare services in social media health networks," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 579–590, Mar. 2019.

[32] D. Abbasinezhad-Mood, A. Ostad-Sharif, S. M. Mazinani, and M. Nikooghadam, "Provably secure escrow-less Chebyshev chaotic map-based key agreement protocol for vehicle to grid connections with privacy protection," *IEEE Trans. Ind. Inform.*, vol. 16, no. 12, pp. 7287–7294, Dec. 2020.

[33] X. Jia, D. He, N. Kumar, and K. K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Syst. J.*, vol. 14, no. 1, pp. 560–571, Mar. 2020.

[34] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, "A privacy-preserving scheme for incentive-based demand response in the smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1304–1313, May 2016.

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

[36] T. Seregina, O. Brun, R. El-Azouzi, and B. J. Prabhu, "On the design of a reward-based incentive mechanism for delay tolerant networks," *IEEE. Trans. Mobile Comput.*, vol. 16, no. 2, pp. 453–465, Feb. 2017.

[37] X. Yan, F. Ye, Y. Yang, X. Deng, and D. Wang, "Incentive facilitation for peer data exchange in crowdsensing," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2019.2926973.

[38] W. Li and M. He, "Comparative analysis of Bitcoin, Ethereum, and Libra," in *Proc. IEEE 11th Int. Conf. Softw. Eng. Serv. Sci.*, 2020, pp. 545–550.

[39] L. Cheng, J. Liu, C. Su, K. Liang, G. Xu, and W. Wang, "Polynomial-based modifiable blockchain structure for removing fraud transactions," *Future Gener. Comput. Syst.*, vol. 99, pp. 154–163, 2019.

[40] G. Zyskind, D. M. S. Zekrifa, P. Alex, and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, 2015, pp. 180–184.

[41] K. Verma, H. Hasbullah, and A. Kumar, "An efficient defense method against UDP spoofed flooding traffic of denial of service (DoS) attacks in VANET," in *Proc. 3rd IEEE Int. Adv. Comput. Conf.*, 2013, pp. 550–555.