# T-PPA: A Privacy-Preserving Decentralized Payment System with Efficient Auditability Based on TEE

1st Suozai Li
*China Electronics Corporation Hainan Joint Innovation Research Institute Co. Ltd.*
Chengmai, China
lisuozai@jiri.ac.cn

2nd Ming Huang
*China Electronics Corporation Hainan Joint Innovation Research Institute Co. Ltd.*
Chengmai, China
huangming@jiri.ac.cn

3th Qinghao Wang
*SSC Holding Company Ltd.*
Chengmai, China
*the College of Computer Science and Engineering Northeastern University*
Shenyang, China
qinghao_wang@qq.com

4th Yongxin Zhang
*SSC Holding Company Ltd.*
Chengmai, China
yongxin@oxhainan.org

5th Ning Lu
*the College of Computer Science and Engineering Northeastern University*
Shenyang, China
*the School of Computer Science and Technology Xidian University*
Xi'an, China
luning@neuq.edu.cn

6th Wenbo Shi
*the College of Computer Science and Engineering Northeastern University*
Shenyang, China
shiwb@neuq.edu.cn

7th Hong Lei*
*SSC Holding Company Ltd.*
Chengmai, China
*the School of Cyberspace Security, Hainan University*
Haikou, China
leihong@oxhainan.org

2022 IEEE 8th International Conference on Computer and Communications (ICCC) | 978-1-6654-5051-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCC56324.2022.10065866

***Abstract*—Cryptocurrencies such as Bitcoin and Ethereum achieve decentralized payment by maintaining a globally distributed and append-only ledger. Recently, several researchers have sought to achieve privacy-preserving auditing, which is a crucial function for scenarios that require regulatory compliance, for decentralized payment systems. However, those proposed schemes usually cost much time for the cooperation between the auditor and the user due to leveraging complex cryptographic tools such as zero-knowledge proof. To tackle the problem, we present T-PPA, a privacy-preserving decentralized payment system, which provides customizable and efficient auditability by leveraging trusted execution environments (TEEs). T-PPA demands the auditor construct audit programs based on request and execute them in the TEE to protect the privacy of transactions. Then, identity-based encryption (IBE) is employed to construct the separation of power between the agency nodes and the auditor and to protect the privacy of transactions out of TEE. The experimental results show that T-PPA can achieve privacy-preserving audits with acceptable overhead.**

***Index Terms*—Auditable, Blockchain, Confidential transactions, Decentralized payment system, Trusted execution environments**

This work was supported in part by the Finance Science and Technology Project of Hainan Province (No. ZDKJ2020009); in part by National Key R&D Program of China (No.2021YFB2700601); in part by the National Natural Science Foundation of China (Nos. 62163011, 62072092, 62072093 and U1708262); in part by the Fundamental Research Funds for the Central Universities (No.N2023020); in part by the Natural Science Foundation of Hebei Province (No.F2020501013); in part by the China Postdoctoral Science Foundation(No.2019M653568); and in part by the Key Research and Development Project of Hebei Province (No.20310702D).

## I. Introduction

Cryptocurrencies such as Bitcoin [1] and Ethereum [2] implement decentralized peer-to-peer payment by maintaining an append-only public ledger. However, Bitcoin-like and

Ethereum-like cryptocurrencies record all transactions in the public ledger, which could breach the privacy of the users. While those cryptocurrencies provide pseudonym (*i.e.,* it is unlinkable between account addresses and real identities) to alleviate the problem, some researchers have proved that associating account addresses with real identities is feasible [3].

As a result of the issue, a number of proposals to improve privacy have been proposed, including novel cryptocurrencies with privacy guarantees (*e.g.,* Zcash [4] and Monero [5]) and privacy-enhancing schemes for existing cryptocurrencies (*e.g.,* CoinJoin [6] and TumbleBit [7]). However, strong privacy guarantees provide malicious users the ability to engage in criminal behavior (*e.g.,* money laundering or tax evasion). A feasible solution is providing the audit functions in those privacy-preserving decentralized payment systems. For example, in some financial institutions, the designated auditors frequently check the accuracy of corporate statements using transaction information from banks.

Recently, a number of audit schemes [8]–[12] have been proposed for privacy-preserving decentralized payment systems. However, they triggered concerns on several issues. First, some works rely on the centralized auditor, which takes charge of excessive power. For example, the centralized auditor in Zcash extension [8] always knows the transaction amounts and can arbitrarily reveal the real identity of users. Second, those schemes only provide limited audit functions, which cannot support complex scenarios. In PRCash [10], the auditor only audits the single transaction (*e.g.,* the amount of a transaction), but the fact that statistical audits on multiple transactions are also significant. Third, poor efficiency is also a common problem. Some schemes [11], [12] achieve privacy-preserving audit using complex cryptographic tools (*e.g.,* zero-knowledge proof), which result in poor efficiency. Thus, designing a privacy-preserving decentralized payment system that supports customizable and efficient auditability remains a challenge.

In this paper, we present T-PPA[1], a privacy-preserving decentralized payment system, which provides customizable and efficient auditability. In T-PPA, we employ a set of agency nodes to confidentially verify transactions in their trusted execution environments (TEEs). All agency nodes are in a distributed setting and cooperate to maintain user accounts using a TEE-based consensus algorithm (*e.g.*, Proof of Luck [13]). All certified transactions are encrypted by the TEEs and synchronized to a blockchain network for future audits. A TEE-enabled auditor is introduced to customize the audit programs and execute it in the TEE. All the audit programs are publicly verifiable because of the remote attestation mechanism provided by TEE. Moreover, we employ the identity-based encryption (IBE) to protect the privacy of transactions that are out of TEEs. The auditor maintains the master key in its TEE and generates the private keys for the agency nodes. The agency nodes serve their unique identity strings as the public key to encrypt transactions in their TEEs. In this case, the agency nodes only takes charge of verifying transactions but can not obtain all transactions for auditing. The auditor is in opposite power. Thus, IBE constructs the separation of power between the agency nodes and the auditor. Our key contributions in this paper are as follows:

- We present a privacy-preserving decentralized payment system T-PPA which can provide customizable and efficient auditability by relying on TEE.
- We employ IBE to construct the separation of power between the agency nodes and the auditor. The scheme protects the privacy of transactions that are out of TEE and ensures the auditor is accessible to confidential transactions.
- We give the security analysis and performance evaluation of T-PPA. The results illustrate the feasibility and effectiveness of our scheme.

The rest of this paper is organized as follows. In section II, we introduce the related work. We give a brief overview of the bilinear maps, blockchain, identity-based encryption, and TEE in section III. In section IV, we define the system model, design goals, and the basic audit functions. In section V, we present the design of our scheme. In section VI, we demonstrate that our scheme achieves privacy and security. In section VII, we carry out experiments to evaluate the computing overhead in our scheme. The results show that our scheme is efficient. We finally present the conclusion in section VIII.

## II. Related Work

In this section, we review some related works, including distributed payment systems with auditability and the TEE-based scheme in the blockchain area.

### A. Distributed Payment Systems with Auditability

Recently, some works offer auditability in the decentralized payment systems (*i.e.,* using a common public ledger) with privacy guarantees. Zcash extension [8] is designed to insert auxiliary information into tokens to achieve transaction accountability. However, the scheme employs a centralized auditor and inherits the limitations of Zcash. Monero extension [9] also uses a centralized auditor to implement accountability, which is similar to the Zcash extension scheme. PRCash [10] achieves accountability based on the transaction limit. In PRCash, a centralized regulator issues anonymous credentials to users and checks the users' transaction amounts. If a user's transaction amount exceeds the limit, the regulator will reject the transaction. Users can voluntarily remove the transaction limit by approaching the regulator for de-anonymizing. However, PRCash only provides limited audit functions. zkLedger [11] proposes a two-dimensional table-based architecture to enable various types of interactive auditing. zkLedger provides confidential auditing based on the interactive zero-knowledge proof protocol between users and auditors. However, the zero-knowledge proof protocol adds additional storage and computational costs, which leads to poor efficiency. Moreover,

[1]T-PPA is a **T**EE-based **P**rivacy-preserving decentralized **P**ayment system with **A**uditability.

the two-dimensional table architecture will incur low scalability. MiniLedger [12] also uses the two-dimensional table architecture and achieves the transaction pruning to improve scalability. However, MiniLedger employs the RSA accumulators, which involves a trusted setup and extra overhead.

### B. TEE-Based Schemes for Privacy-Preserving Payment

TEE technology is used to enhance the confidentiality of payment in the blockchain area [14] in recent years. Obscuro [15] is a centralized mixer scheme to implement the privacy-preserving payment. The scheme employs a TEE-enabled mixer to execute the secure mix operations. BITE [16] is a lightweight scheme for Bitcoin, which uses the TEE-based full node to improve the privacy of transactions. Moreover, some works [17]–[19] are dedicated to protecting the privacy of smart contact in the blockchain-based payment systems, *e.g.,* Ethereum. However, there was no existing effort to implement a confidential payment with auditability.

## III. Preliminaries

In this section, we first review the definition of the bilinear pairing and then describe the identity-based encryption, blockchain, and trusted execution environments.

### A. Bilinear Pairing

Given two cyclic groups $G_1$ and $G_T$ with large prime order $q$. Let $g_1$ be a generator of $G_1$, and $g_2$ be a generator of $G_T$. Then a cryptographic bilinear map is defined as $e$: $G_1 \times G_1 \rightarrow G_T$. The bilinear map need to satisfy three properties:

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

### B. Identity-Based Encryption

In 1984, Shamir [20] proposed the concept of identity-based encryption (IBE), which enables encrypting a message with users' identities instead of the random strings generated by the key generator. We implement our scheme based on the well-known IBE scheme proposed by Boneh *et al.* [21].

In IBE, a user's unique identity string (*e.g.,* such as phone numbers or emails) can be regarded as the public key. IBE requires a trusted third party, called the key generation center (KGC), to provide key generation services for users. Before encrypting the message, a user needs to provide his identity to the KGC for registering, and the KGC will calculate a pair of public and private keys for the user using its master private and public keys. When sending a confidential message, the user needs no certificate but the receiver's identity (*i.e.,* the receiver's public key) to encrypt messages, which eliminates the computation and storage overhead with certificates.

### C. Blockchain

Blockchain was originally presented by Satoshi [1] to solve the consensus problem in a distributed network. Blockchain implements the characteristics of tamper-proof, decentralization and transparency, and has attracted a great deal of attention from researchers. Based on different access permission, existing blockchains can be classified into public blockchains, private blockchains, and consortium blockchains. Public blockchains are completely decentralized, and the data on the chains is publicly accessible. Private blockchains are not decentralized because of a clear hierarchy of permissions settings. In the paper, we focus on the consortium blockchain, which is a hybrid form of public and private blockchains. In consortium blockchains, a small number of nodes is employed as validators to verify the transaction from user nodes. When new nodes join a consortium blockchain, they need to be authorized by the validators. Consortium blockchains are useful in some situations, such as several institutions deal transactions or share information to each other. Hyperledger Fabric [22] is one of the most famous consortium blockchain, which is widely used in enterprise use cases. In this paper, we implement our scheme based on it.

Hyperledger Fabric is a permissioned blockchain platform that provides secure interactions between a group of entities and helps them to handle disputes. It supports the flexibility to fit particular use cases or trust models because of pluggable consensus protocols. Thus, we consider the certain consensus protocol in this paper. Moreover, Fabric also provides the private data and channel architecture to achieve the confidential transaction.

### D. Trusted Execution Environments

TEE technologies, such as ARM TrustZone [23] and Intel software guard extensions (SGX) [24], provide a secure processing environment, which guarantees confidentiality and integrity of code and data. We implement our scheme based on SGX since TrustZone is more popular on mobile devices.

SGX provides the secure processing environment called enclave in the SGX-enabled platform. The enclave is protected by some hardware modules and can prevent malicious roles even the privileged software (*e.g.,* kernel, hypervisor) from breaking the confidentiality of internal programs. SGX provides the remote attestation mechanism to prove that the certain code is running in an enclave of a real SGX-enabled platform. Moreover, the remote attestation mechanism can help the remote party securely communicate with the enclave. Specifically, a remote party can establish a secure channel with an enclave by the key exchange protocols (*e.g.*, authenticated Diffie-Hellman protocol [25]) in the user custom field of remote attestation protocols.

## IV. System Model and Goals

### A. System Model

As shown in Fig. 1, this paper proposes a novel privacy-preserving decentralized payment system T-PPA which in-

volves five parties: Users, T-Agency Nodes (TANs), T-Auditor (TA), Audit Requester (AR), and Blockchain Network.

**Users**. In our scheme, users can be arbitrary individuals or organizations, which want to transfer to others. All users can act as the senders or receivers of transactions. Each user should register their account with a TAN before the transaction.

**T-Agency Nodes (TANs)**. Each TAN should provide the TEE to verify the transaction requests from users and generate the encrypted transaction. All of the TANs form a peer-to-peer network and cooperate to maintain the identity and account of users.

**T-Auditor (TA)**. The TA also supports the TEE and executes the secure audit process in it. By observing the blockchain, TA can obtain the relevant transactions. TA should run a setup to generate the public parameters in the system initialization stage. Moreover, the TA authenticates the identity of each TAN and generates private keys for them.

**Audit Requesters (ARs)**. An AR is the benefit correlation person of some transactions. When someone conflicts their interest with others, AR sends the audit request to the TA.

**Blockchain Network**. The blockchain network is formed from a number of nodes, who cooperate to perform the consensus algorithm in the system and maintain the blockchain. The nodes can behave as miners to receive transactions from some TANs.

According to Fig. 1, T-PPA consists of three phases. In phase I (1-3), TA first generates public parameters and master private key, and generates private keys for TANs. Then, the user registers with one TAN for an account under the confirmation. TAN will maintain the account in the TEE and synchronize the account to the other TANs. In phase II (4-6), when a user sends a transaction request to a TAN, the TAN first authenticates the account of the user. The TAN then verifies the transaction and encrypts it with its private key. The TAN final synchronizes the encrypted transaction to the blockchain network. In phase III (8-12), when the TA receives an audit request from an AR, he/she first constructs the TEE code based on the request. Any parties can verify the TEE code using the remote attestation mechanism. Then, the TA searches the related transactions from the blockchain and executes the TEE code inside the TEE to audit those transactions. Finally, the TA will return the audit result to the AR.

### B. Design Goals

We aim to employ the TEE technology to protect the privacy of transactions in the audit process. In this section, we consider the two aspects of our scheme, which are performance and security. For performance, our scheme demands that the audit process is practiced and only induces an acceptable computation cost. For security, we assume that the TEE is trusted, and it will execute the protocol according to a predetermined program (without considering factors such as side-channel attacks). Our scheme ensures privacy in the audit process, which implements the privacy goals as follows:

- *Value privacy*: If all of the TEE-based nodes are non-compromised, the transaction value is unknown to the adversary in the audit process.
- *Sender privacy*: If all of the TEE-based nodes are non-compromised, the sender identity is unknown to the adversary in the audit process.
- *Receiver privacy*: If all of the TEE-based nodes are non-compromised, the receiver identity is unknown to the adversary in the audit process.

Furthermore, this paper assumes that all actors in the system communicate securely with each other by hiding their network information, such as using the Tor anonymity network [26]. The anonymity network is a significant scheme to prevent the malicious controllers of TANs from eroding the sender privacy. Specifically, a TEE relies on the operating system, which is controlled by the untrusted controller, to run communication operations. A malicious controller can observe the source address of a transaction request and confirm the sender of the transaction. Tor anonymity network can hide the source addresses of the IP packages, therefore, the sender privacy can be protected.

The controller of a TEE controls the hardware and software out of the TEE (*e.g.,* the disk and the operating system), particularly the TEE relies on the operating system to run communication operations. In this case, all the messages between the TEE and other parties are forwarded by the controller. Thus, when users send transactions to a TAN, the malicious controller of the TAN can observe the source addresses and erode the sender privacy. Tor anonymity network can prevent adversaries from learning the source addresses of the IP packages in the network layer. In our scheme, when a user sends a transaction to a TAN, he/she must employ Tor to hide the source. Then, the controller of the TAN cannot learn who sends the transaction, therefore, the sender privacy is protected. We have added the details to explain the effect of Tor in paragraph 2 of section IV-B.

### C. Basic Audit Functions

To understand the audit process of our scheme, we list the basic audit functions in Table I, which can be classified into two types, *i.e.,* the value and the identity. Specifically, F1 can reveal the exact value of a transaction, while F2 only confirms whether the value of a transaction is kept within reasonable bounds. F3 and F4 reveal the identities of the sender and receiver in a transaction, respectively. Moreover, F5 can verify whether the user is a participant in a transaction. We only implement those basic audit functions in a TEE instance (*i.e.,* SGX), but note that based on these basic audit functions, we can construct any more complex audit functions. All of the audit functions will be executed in plaintext speedily in the TEE and the confidentiality will be maintained by the TEE.

## V. T-PPA Design

In this section, we present the design of T-PPA. We first provide a high-level overview of T-PPA and then describe the protocol in detail.

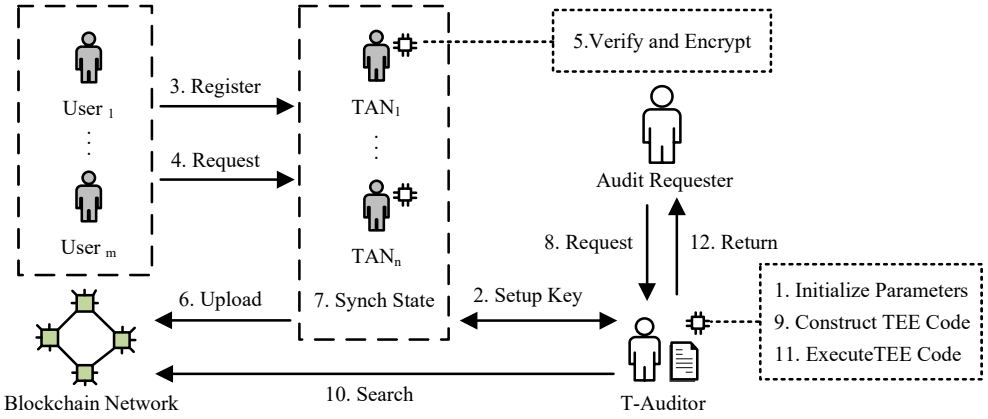


Fig. 1. System model

TABLE I
BASIC ADUIT FUNCTIONS

| Number | Audit Type | Function | Description |
|---|---|---|---|
| F1 | Value | Transaction Value | Value of the certain transaction |
| F2 | Value | Transaction Value Limit | Value range of the certain transaction |
| F3 | Identity | Sender Identity | Sender identity of the certain transaction |
| F4 | Identity | Receiver Identity | Receiver identity of the certain transaction |
| F5 | Identity | Participation/Non-participation | Whether the certain user participates in the certain transaction |

### A. Overview

T-PPA comprises three stages, (1) initialization (phase I): the TA initializes the system parameters and users register accounts; (2) authenticated transaction upload (phase II): users generate the transactions and TANs synchronize to the blockchain network, and (3) privacy-preserving auditing (phase III): the TA performs the privacy-preserving audit process. Especially, TA and TANs execute the trusted code in their enclaves, and other parties will verify the credibility and establish the secure channel with them using the remote attestation mechanism, as mentioned in section III-D. Moreover, we give the definitions of symbols used in our protocol in Table II.

TABLE II
DESCRIPTION OF SYMBOLS

| Symbol | Description |
|---|---|
| $msk$ | The master private key holded by TA |
| $mpk$ | The master public key |
| $sk_i$ | The private key of $\text{TAN}_i$ |
| $SKs$ | The set of all TANs' private keys |
| $id_i$ | The ID of $\text{TAN}_i$ |
| $IDs$ | The set of all TANs' IDs |
| $User_i$ | The identity information of $\text{user}_i$ |
| $k_{<user_i,TAN_i>}$ | The symmetric key between $\text{user}_i$ and $\text{TAN}_i$'s TEE |
| $k_{<AR,TA>}$ | The symmetric key between AR and TA's TEE |
| $bal_i$ | The balance of $\text{user}_i$ |
| $v$ | The value of a transaction |
| $sen$ | The sender of a transaction |
| $rec$ | The receiver of a transaction |
| $Tx$ | The plaintext of a transaction |
| $Tx_{enc}$ | The ciphertext of a transaction |
| $x, r$ | The random numbers |
| $P$ | The random generator |
| $\oplus$ | The bitwise-and operator |
| $Tag$ | The basic information of the transaction |
| $T$ | A time period |
| $v_{limit}$ | The limit value of a transaction |
| $n_{limit}$ | The limit number of transactions one account can transfer in a time period |
| $v_{thres}$ | The threshold of transactions one account can transfer in a time period |

### B. Initialization

In the beginning, the TA generates public parameters for building an IBE system in the TEE. The public parameters are generated as follows. Two cyclic groups of prime order $p$ are denoted as $G_1$ and $G_T$, respectively, and $g$ is a generator of $G_1$. Then a bilinear map $e$ is defined as $G_1 \times G_1 \rightarrow G_T$. Let $H_1 : \{0,1\}^* \rightarrow G_1$ be the hash function that maps a string data to an element in $G_1$, and let $H_2 : G_1 \rightarrow Z_q^*$ be another hash function that maps an element in $G_1$ to an element in $Z_q^*$. The TA chooses a value $x \in Z_q^*$ and a generator $P \in G_1$ randomly. Then, TA regards $x$ as the master private key $msk$ and calculates the master public key $mpk = xP$. It is worth noticing that the $mpk$ is publicly visible to all parties. Then, TANs publish their $IDs = \{id_1, id_2, \cdots, id_n\}$, respectively, which $n$ represents the number of TANs. TA constructs the private key in the TEE for each TAN, specifically, the private key of $\text{TAN}_i$ is $sk_i = xH_1(id_i)$. Finally, TA store all the private keys in the TEE, which are noted as $SKs = \{sk_1, sk_2, \cdots, sk_n\}$. When $\text{user}_i$ wants to register an account with $\text{TAN}_i$, $\text{user}_i$ must verify

the TEE of $\text{TAN}_i$ to guarantee the correctness and exchange a symmetric key $k_{<user_i,TAN_i>}$ with the TEE using remote attestation as mentioned in section III-D. Then, $\text{user}_i$ registers with the TEE of $\text{TAN}_i$ by its identity information $User_i$, and the TEE then creates an account for the user and initializes the balance $bal_i$ . The TAN then synchronizes users' states with other TANs in their TEEs by the secure channels. Note that the TEE-based consensus algorithm (*e.g.*, Proof of Luck [13]) can be used to synchronize status between TEEs.

### C. Authenticated Transaction Upload

To make a payment, $\text{user}_i$ first encrypts the $Tx = \{sen, rec, v\}$ with symmetric key $k_{<user_i,TAN_i>}$, and then sends the encrypted transaction $Tx_{enc}$ to the $\text{TAN}_i$, where $sen$ and $rec$ are the sender and receiver respectively, and $v$ is the amount of the payment to be transferred. Note that any TAN can be used because of the synchronized information. The TAN then verifies the account of the user (*e.g.,* user's identity and account balance) and whether the transaction is legal (*e.g.,* transaction limits) in its TEE. Specifically, we consider that transactions should be verified to ensure compliance with counter-terrorist financing and anti-money laws. For example, many countries or institutions limit each user's transaction amount of the cryptocurrency at a time or over a period of time (*e.g.,* Financial Crimes Enforcement Network inspects the transactions over $1000 in the US, and some exchanges such as Huobi and Binance also propose a restriction on the amount of transactions one account per day). To cater the requirements, we list the validation policies in our scheme as follows:

- *Legal identity*: All users need to register themselves with a TAN by their identity information in the initialization phase. Thus, the TAN will verify the user's identity $User_i$ is legal.
- *Legal balance*: The transaction is only performed when the user has a sufficient balance. Thus, the TAN will verify the transaction value $v$ is less than or equal to the user's balance $bal$.
- *Legal transaction value*: The value of a transaction an account can transfer is limited to $v_{limit}$. Thus, the TAN will verify the transaction value $v$ is less than $v_{limit}$.
- *Legal transaction number*: The number of a transaction an account can transfer in the time period $T$ is limited to $n_{limit}$. Thus, the TAN will verify the account transfer the number is less than $n_{limit}$ in the time period $T$.
- *Legal transaction threshold*: The total amount of transaction an account can transfer in the time period $T$ is limited to $v_{thres}$. Thus, the TAN will verify the account transfer the amount is less than $v_{thres}$ in the time period $T$.

If the transaction has been validated, the TAN will encrypt the transaction using its private key in the TEE, *i.e.,* $Tx_{enc} =< rP, Tx \oplus H_2(g_{id}^r) >$, where $r$ is a random value from $Z_q^*$ and $g_{id}^r = e(H_1(id_i), mpk)$. Meanwhile, the TAN needs to generate the tag $Tag$ for the transaction, which represents basic information such as the timestamp, index, and ID of the TAN. Finally, the TAN synchronizes users' states with other TANs in their TEEs by the secure channels, and uploads the tuple ($Tag$, $Tx_{enc}$) to the blockchain network.

**Algorithm 1:** Privacy-preserving auditing

**Input:**
The audit request $Request = \{sen, rec, v, T, statement\}$;
The key of TA's TEE $k_{tee}$;
The set of transactions $TX = \{Tx_0, \cdots, Tx_i\}$;
**Output:**
The result of audit $result$;

**TA's TEE:**
**for** *each element* $e \in Request$ **do**
  **if** $e \neq statement$ **then**
    $e_{enc}$ := `Encrypt(`$e$, $k_{tee}$`)`;
  **else**
    **return**;
  **end**
**end**
send $Request_{enc} = \{sen_{enc}, rec_{enc}, v_{enc}, T_{enc}, statement\}$ to **TA**;
**TA:**
construct the audit program `Audit(`$TX$`)`;
perform the remote attestation with the auditor;
load `Audit(`$TX$`)` in the **TEE**;
**TA's TEE:**
**for** $i$ = 0, *each transaction* $Tx_{enc}$ *over a period of time* $T$ **do**
  **if** $Tag$ *is related to* $sen$ *and* $rec$ **then**
    $Tx_i$ := `Decrypt(`$Tx_{enc}$, $sk$`)`;
    $i$++;
  **else**
    **return**;
  **end**
**end**
$result$ := `Audit(`$TX$`)`;
**return** $result$;

### D. Privacy-Preserving Auditing

For any transaction in the blockchain, T-PPA can perform efficient, privacy-preserving auditing and will not reveal the confidential information of transactions. Algorithm 1 describes the privacy-preserving auditing process. When an AR wants to audit one or more transactions in the blockchain, he/she first encrypts a specific audit request $Request = \{sen, rec, v, T, statement\}$ with the symmetric key $k_{<AR,TA>}$, where $sen$ and $rec$ are the sender and receiver respectively, and $v$ is the amount related to the audit goal, $T$ represents a period of time used to confirm relevant transactions in the blockchain, and $statement$ determines logic audit. For example, a $statement$ can be set as "whether the sender transferred value to the receiver over a period of time?".

When the TEE of the TA receives the encrypted audit request $Request_{enc}$ from an AR, it first authenticates the identity of the AR. After authentication, the TEE returns the audit request to the TA in a privacy-preserving form. A privacy-preserving audit request manifests the audit logic without the information about identity, value, and time. For example, a privacy-preserving audit request is "whether $sen_{enc}$ transferred $v_{enc}$ to $rec_{enc}$ over a period of time $T_{enc}$?", where $sen_{enc}$, $v_{enc}$, $rec_{enc}$ and $T_{enc}$ are encrypted by the TEE. Then, the TA needs to construct a TEE code to implement the audit. We implemented all the basic audit functions (F1-F5) mentioned in section IV-C in the instance of the TA, thus the TA can construct the audit program based on the privacy-preserving audit request and the basic audit functions. Generally, most complex audit functions can be completed by the combination of the basic audit functions. Exceptionally, some complicated audit functions will require TA to encode the new trusted code in the TEE. In this case, other parties need to verify the credibility of the new trusted code using the remote attestation mechanism.

After constructing the audit program in the TEE, the AR needs to perform a remote attestation on the TEE to ensure the trustworthiness of the audit program. Then, the TEE will request the relevant transactions from the blockchain. All of the audit processes need to decrypt the ciphertext transactions in the TEE. Specifically, when TA decrypts a ciphertext transaction, it first need to search the source of the transaction, *i.e.,* the ID of the TAN, and gets the corresponding private key $sk_i$. Let the ciphertext transaction $Tx_{enc} =< rP, Tx \oplus H_2(g_{id}^r) >$ as $Tx_{enc} = (U, V)$, *i.e.,* $V = rP$ and $U = Tx \oplus H_2(g_{id}^r)$. Then, TA computes the plaintext transaction $Tx = V \oplus H_2(e(id_i, U))$. Based on the plaintext transaction, TA can execute the audit operations and response the specific audit request.

## VI. Security Analysis

In this section, we first show how our scheme achieves all the privacy goals proposed in section IV-B. Then, we analyze the influence of TEE termination.

*Value Privacy.* T-PPA can protect value privacy in two situations: 1) when verifying the transactions, the transaction value is hidden from the TANs; 2) when auditing the transactions, the transaction value is hidden from the TA. During the authenticated transaction upload stage, the user transactions are verified by the TAN in the TEE. The user trusts the TEE but does not trust the controller of the TEE. Therefore the user first performs the remote authentication to ensure the trustworthiness of the TEE and then sends the encrypted transaction request to the TAN. The TEE will decrypt the request and verify it. The controller cannot decrypt the request, so the privacy of the value can be protected. Similarly, the audit programs loaded in the TA's TEE also will be verified by others, and the encrypted transactions also can be decrypted by the TA's TEE. Therefore, value privacy can be protected in the privacy-preserving auditing stage.

*Sender Privacy and Receiver Privacy.* We preserve the sender and receiver privacy, when the transactions are stored in both blockchains or are loaded in TEEs. On the one hand, the transactions in the blockchain are encrypted to hinder the identity of the sender and receiver from the others and the encrypted. On the other hand, the encrypted transactions only can be decrypted by the TEEs, which ensures the controllers of TEEs can not reveal confidential information in the transactions. Therefore, T-PPA can protect the identity privacy of the sender and receiver at the authenticated transaction upload stage and privacy-preserving auditing stage.

*TEE Termination.* Unexpected TEE failures or even malicious shutdown by the controller of TEE may result in negative effects. However, most TEE technologies can not protect against such termination. There are two affected roles: TANs and TA. In TANs, TEEs are responsible for verifying users' transactions. If one of the TANs breaks down, it will only invalidate the payment and will not incur the loss of users' money. Users can select another TAN in the future, because of the distributed deployment of TAN. If the TA breaks down, it will only invalidate the audit functions because TA is only responsible for performing the transaction audit. To relieve the loss of functionality, an available scheme is deploying the distributed design, which can be the future research.

## VII. Performance Evaluation

We implement T-PPA based on Intel SGX to evaluate its efficiency. Especially, the parties (*i.e.,* the Users, the TANs, the TA) in our scheme run on the PC with Ubuntu 16.04 LTS operating system, a 3.60 GHz Intel(R) Core(TM) CPU i3-9100F, and 16GB RAM. We perform the audit program in an enclave provided by the SGX. Moreover, we run the other programs in the normal memory, especially install Hyperledger Fabric client and instantiate both prover and certifier chaincodes on all the peer nodes.

We implemented the audit functions with different basic audit functions (F1-F5) mentioned in section IV-C. Note that users can construct any more complex audit function by combining basic audit functions. Fig. 2 illustrates the computation time of basic audit functions with different numbers of the related transactions. We fixed the size of a transaction to 250 bytes (maintaining the similar size as Bitcoin) and set the TAN's ID (*i.e.,* TAN's public key) to "abcdefg". The figure shows that the computation time of all audit functions grows linearly as the number of audited transactions. Especially, the Participation/Non-Participation (F5) audit induces a heavier computation time overhead than other audits due to it contains more complex judgments involving expensive computation.

We also compare the performance with the zkLedger [11], which implements the privacy-preserving audit schemes based on the zero-knowledge proof. Specifically, we test the computation time of auditing the transaction amount from one user to another in a period of time using both schemes. We set the different numbers of transactions (the size of a transaction is fixed to 250 bytes) to simulate different time periods. In our scheme, we perform the basic audit functions F2 to

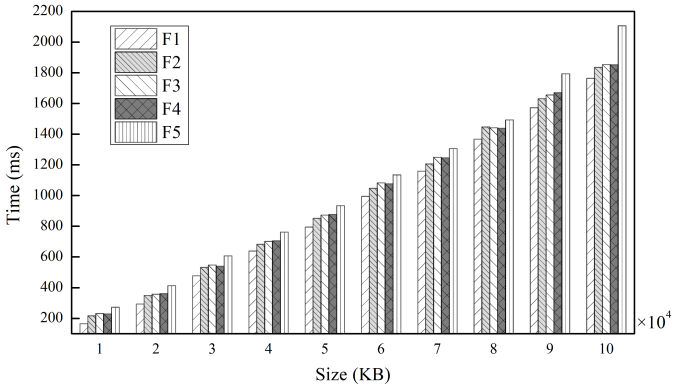


Fig. 2. The computation overhead of different audit functions.

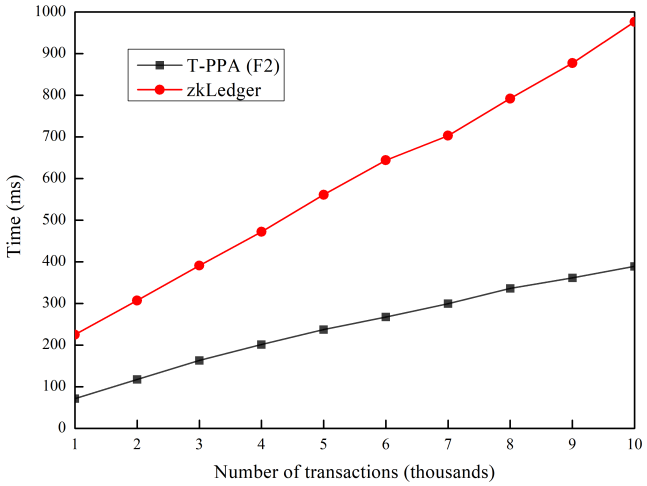


Fig. 3. The performance comparison between T-PPA and zkLedger.

implement the audits. In zkLedger, the performance overhead is linearly for increases in the number of participants. In order to facilitate comparison, we set the number of participants as five in zkLedger. The result in Fig. 3 shows that the computation time of both schemes increases linearly with the numbers of transactions, but T-PPA achieves a relatively low overhead. This is because zkLedger needs to execute complex cryptographic algorithms (*i.e.,* zero-knowledge proof).

To evaluate how the use of SGX affects performance, we implemented basic audit functions in the enclave and out of the enclave and fixed the size of audited transactions to 10 MB. We performed the different audit functions in the enclave and the normal memory, and compare the computation overhead in Fig. 4. The result indicates that performing audit functions in the enclave needs slightly more time than performing them in the normal memory. This is caused by the additional scheduling operations in the enclave, which commit to protecting confidentiality and integrity. Fortunately, all audit functions in the enclave are limited to the microsecond level, thus the scheme is a practical solution to protect the privacy of the audit process.

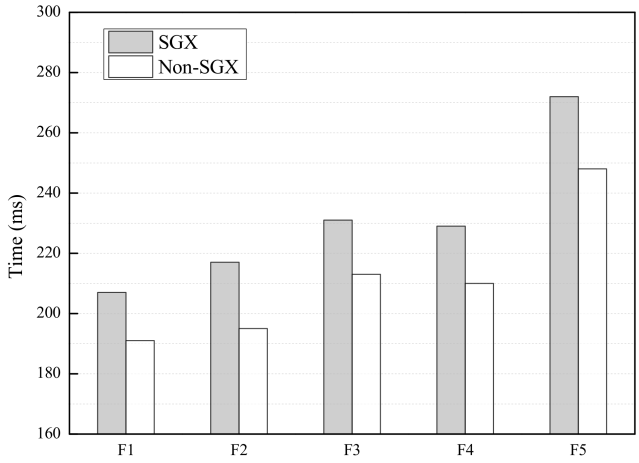


Fig. 4. The computation overhead of different audit functions with different execution environments.

## VIII. Conclusion

In this paper, we have presented T-PPA, a privacy-preserving decentralized payment system, which provides more efficient auditability than the state-of-the-art by leveraging trusted execution environments. The identity-based encryption scheme makes T-PPA highly efficient in processing the audit process. We have analyzed the performance and security of T-PPA, and implemented T-PPA based on Hyperledger Fabric source code. Experiment results showed that T-PPA is highly efficient in the audit process, and it provides high privacy-preserving for the transaction. Though T-PPA adopts the permissioned blockchain, its design can be easily extended to permissionless blockchains.

## References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[3] T. Mitani and A. Otsuka, "Confidential and auditable payments," in *International Conference on Financial Cryptography and Data Security*. Springer, 2020, pp. 466–480.

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

[5] S. Noether and A. Mackenzie, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, 2016.

[6] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world," in *Post on Bitcoin Forum*, 2013.

[7] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "TumbleBit: An untrusted bitcoin-compatible anonymous payment hub," in *Network and Distributed System Security Symposium*, 2017.

[8] C. Garman, M. Green, and I. Miers, "Accountable privacy for decentralized anonymous payments," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 81–98.

[9] Y. Li, G. Yang, W. Susilo, Y. Yu, M. H. Au, and D. Liu, "Traceable Monero: Anonymous cryptocurrency with enhanced accountability," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 679–691, 2019.

[10] K. Wüst, K. Kostiainen, V. Čapkun, and S. Čapkun, "PRcash: Fast, private and regulated transactions for digital currencies," in *International Conference on Financial Cryptography and Data Security*. Springer, 2019, pp. 158–178.

[11] N. Narula, W. Vasquez, and M. Virza, "zkLedger: Privacy-preserving auditing for distributed ledgers," in *15th USENIX Symposium on Networked Systems Design and Implementation*, 2018, pp. 65–80.
[12] P. Chatzigiannis and F. Baldimtsi, "MiniLedger: Compact-sized anonymous and auditable distributed payments," in *European Symposium on Research in Computer Security*. Springer, 2021, pp. 407–429.
[13] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of Luck: An efficient blockchain consensus protocol," in *proceedings of the 1st Workshop on System Software for Trusted Execution*, 2016, pp. 1–6.
[14] Z. Bao, Q. Wang, W. Shi, L. Wang, H. Lei, and B. Chen, "When blockchain meets SGX: An overview, challenges, and open issues," *IEEE Access*, vol. 8, pp. 170 404–170 420, 2020.
[15] M. Tran, L. Luu, M. S. Kang, I. Bentov, and P. Saxena, "OBSCURO: A bitcoin mixer using trusted execution environments," in *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 692–701.
[16] S. Matetic, K. Wüst, M. Schneider, K. Kostiainen, G. Karame, and S. Capkun, "BITE: Bitcoin lightweight client privacy using trusted execution," in *28th USENIX Security Symposium*, 2019, pp. 783–800.
[17] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *IEEE Symposium on Security and Privacy*. IEEE, 2016, pp. 839–858.
[18] P. Das, L. Eckey, T. Frassetto, D. Gens, K. Hostáková, P. Jauernig, S. Faust, and A.-R. Sadeghi, "FASTKITTEN: Practical smart contracts on bitcoin," in *28th USENIX Security Symposium*, 2019, pp. 801–818.
[19] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, and D. Song, "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts," in *IEEE European Symposium on Security and Privacy*. IEEE, 2019, pp. 185–200.
[20] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1984, pp. 47–53.
[21] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual international cryptology conference*. Springer, 2001, pp. 213–229.
[22] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, 2018, pp. 1–15.
[23] T. Alves, "TrustZone: Integrated hardware and software security," *White paper*, 2004.
[24] V. Costan and S. Devadas, "Intel SGX explained." *IACR Cryptol. ePrint Arch.*, vol. 2016, no. 86, pp. 1–118, 2016.
[25] H. Krawczyk, "SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE-protocols," in *23rd Annual International Cryptology Conference*, vol. 2729, 2003, pp. 400–425.
[26] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, 2004, pp. 303–320.