applied sciences

MDPI

Article

Privacy Protection Method for Blockchain Transactions Based on the Stealth Address and the Note Mechanism

Zeming Wei, Jiawen Fang, Zhicheng Hong, Yu Zhou, Shansi Ma, Junlang Zhang, Chufeng Liang, Gansen Zhao and Hua Tang *

School of Computer Science, South China Normal University, Guangzhou 510632, China; zeming_wei@163.com (Z.W.); 2022023205@m.scnu.edu.cn (J.F.); hzhicheng@m.scnu.edu.cn (Z.H.); zhouyu_98@163.com (Y.Z.); wsmss15626216354@gmail.com (S.M.); 2021023256@m.scnu.edu.cn (J.Z.); 2019022663@m.scnu.edu.cn (C.L.); gzhao@m.scnu.edu.cn (G.Z.)
* Correspondence: tanghua@m.scnu.edu.cn

Abstract: Blockchain is a distributed ledger technology that possesses characteristics such as decentralization, tamper resistance, and programmability. However, while blockchain ensures transaction openness and transparency, transaction privacy is also at risk of being exposed. Therefore, this paper proposes the blockchain transaction privacy protection method based on the stealth address and the note mechanism to address the privacy leakage risk in blockchain public environments. Firstly, the proposed method generates a random seed known only to the parties involved based on the Diffie–Hellman key exchange protocol, ensuring the privacy of transactions. Then, it utilizes the Note Commitments table to maintain the binding relationship between the stealth address and the corresponding note, enabling efficient transfer and verification of note ownership. The uniqueness of the stealth address is utilized as an invalidation identifier for notes in the Nullifier table, ensuring efficient verification of the correctness of note invalidation identifiers. Additionally, this method employs Pedersen commitment and Bulletproofs range proof to generate proof of the legality of transaction amounts, enabling the concealment of transaction amounts and facilitating private transactions between the parties involved. Finally, this paper presents a detailed performance analysis, implementation, and testing of the method. From the results, it can be concluded that the method proposed can effectively prevent fraudulent behavior by various transaction participants and ensure the security, privacy, and integrity of the transaction. Critical processes consume only milliseconds, and the related commitments and proofs are also minimal, which is crucial for controlling transaction costs. At the same time, this method achieves a completely decentralized privacy transaction solution.

Keywords: blockchain; privacy protection; transaction

check for updates

Citation: Wei, Z.; Fang, J.; Hong, Z.; Zhou, Y.; Ma, S.; Zhang, J.; Liang, C.; Zhao, G.; Tang, H. Privacy Protection Method for Blockchain Transactions Based on the Stealth Address and the Note Mechanism. *Appl. Sci.* **2024**, *14*, 1642. https://doi.org/10.3390/app14041642

Academic Editor: Gianluca Lax

Received: 26 November 2023
Revised: 12 December 2023
Accepted: 14 December 2023
Published: 18 February 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Blockchain technology, as a distributed ledger, has distinct features and advantages, such as distributed, tamper-proof, and traceable transactions, compared to centralized systems [1]. With the continuous development of blockchain technology, it has been applied to solve the problems of low security, poor reliability, low efficiency, and high cost of the current centralized transaction model [2]. Blockchain technology is expected to reshape the shape of society and bring profound reforms in the fields of technology, finance, politics, and culture [3].

However, with the widespread adoption and development of blockchain technology, there is an increasing variety of attack methods targeting blockchain. Blockchain faces numerous technical challenges in terms of security and privacy protection. In traditional centralized transaction systems, the identities of transaction participants and transaction details are typically protected. However, on a blockchain, all transaction records are publicly stored, and attackers can access users' transaction data, network addresses, and other information [4]. Some attacks involve clustering user behavior to analyze their anonymous

identities, including a heuristic model learning of transaction behavior [5,6], IP address clustering [7], address clustering of transaction data [8], and so on, posing significant threats to the privacy protection of current blockchain transactions. The leakage of transaction privacy can expose sensitive information, such as the identities of participants, transaction amounts, and relationships, leading to various potential issues, including financial fraud, personal information disclosure, and disclosure of business secrets. Therefore, protecting the privacy of blockchain transactions has become a key challenge in current research and practice.

Regarding the issue of protecting transaction privacy in blockchain, there have been several relevant studies [9–11], mainly focusing on two approaches: direct privacy protection and indirect privacy protection. Indirect privacy protection methods employ indirect means to protect transaction privacy, mainly through techniques such as coin mixing [12], ring signature [13], and stealth address [14]. Direct privacy protection methods directly hide transaction information and are mainly based on zero-knowledge proof. However, current blockchain transaction privacy protection schemes are not yet fully mature, and they have certain limitations in terms of efficiency, security, and privacy. These issues need to be addressed, especially in scenarios such as data trading that require a balance between privacy and service performance.

Blockchain not only ensures open and transparent transactions but also brings the risk of privacy leakage. Attackers can infer the true identity of blockchain users with a high success rate through clustering analysis and other methods. At present, some mainstream privacy protection methods also have many problems. For example, in the stealth address scheme, transactions need to be accompanied by the temporary public key, resulting in the loss of certain privacy information. In addition, the recipient needs to continuously scan block data and repeatedly perform complex calculations, which is inefficient and requires high computing resources; in the ZCash privacy trading system, a note-based privacy trading scheme is adopted, which uses zk-SNARKs to prove relevant legitimacy and ownership, requiring a series of complex calculations, which is very unfriendly to controlling transaction costs. In addition, zk-SNARKs require additional trusted third parties for trusted settings, and the existence of nonprivate transactions also leads to the loss of certain privacy. How to address the shortcomings of the above solutions and effectively integrate their advantages is an important research question and challenge faced in this article.

In response to the challenges and difficulties of protecting transaction privacy in blockchain, this article aims to contribute to the privacy and security of blockchain transactions and presents the blockchain transaction privacy protection scheme BTPP (Blockchain Transaction Privacy Protection), which is based on the note mechanism in Zcash and the DKSAP protocol. In response to the temporary public key exposure problem of the former, the BTPP protocol extracts the shared knowledge of both parties as a random seed, enhancing the privacy of transactions. In response to the complex proving process and dependence on third parties in the note mechanism, the BTPP protocol uses the Note Commitments table to maintain the binding relationship between temporary addresses and corresponding notes, achieving efficient transfer and verification of note ownership. Through the Nullifier table, the uniqueness of the temporary address is used as the invalidation identifier for the note, achieving efficient verification of the correctness of note invalidation identifiers and eliminating dependence on third parties. Through these improvements, the BTPP protocol ensures transaction performance while protecting the privacy of transactions. The BTPP method consists of three stages: note construction, generation of amount Pedersen commitments and range proofs, and verification of transaction legality. It leverages these stages to generate relevant proofs of transaction legality off-chain and completes the relevant verification of transaction legality on-chain, thus confirming the transaction. This approach ensures the protection of critical information, such as the transaction relationship and amount for both parties, guaranteeing the privacy of transactions. The

BTPP method achieves fully decentralized blockchain transactions while ensuring privacy and performance.

2. Related Works

With the continuous development and application of blockchain technology, many scholars have begun to explore and develop various privacy protection mechanisms and technologies. The goal of privacy protection is to effectively safeguard the privacy of transaction participants while ensuring the security and trustworthiness of the blockchain, enabling anonymous blockchain transactions. Currently, research on blockchain transaction privacy protection primarily focuses on indirect and direct methods.

2.1. Indirect Privacy Protection

Indirect privacy protection methods employ an indirect approach to safeguarding transaction privacy information. These methods mainly rely on techniques such as obfuscation or temporary generation. Common technological solutions include coin mixing, stealth addresses, and ring signatures.

1. Coin mixing technology

Mixing technology achieves privacy by disrupting the correspondence between the addresses of the transacting parties, ensuring transaction unlinkability. In centralized approaches, an additional third party serves as a mixer, disrupting the relationship between the currency and the transactors during the mixing service. Malicious entities find it difficult to obtain critical transaction information. However, this approach relies on a third party, requires payment for the mixing service, and carries the risk of token theft. A decentralized approach, Coinjoin [15], was introduced to address these issues. It eliminates the need for an anonymous third-party intermediary. The coordinator of the mixing service combines multiple transactions, severing the relationship between the transactions and the transacting parties to blur the linkability of the transactions. However, this approach introduces new challenges. It is difficult to ensure that all participants in the mixing process are honest users, and it cannot eliminate the risk of Sybil attacks or guarantee internal unlinkability. To tackle these limitations, a new decentralized mixing scheme called CoinShuffle [16] was proposed. Users shuffle the transaction relationships using a shuffling protocol, and their address information is protected to ensure the security of the mixing transactions. However, users need to be online throughout the mixing process. Addressing this drawback, the CoinParty scheme [17] was introduced. Unlike CoinShuffle, it does not require users to be online throughout the process. Instead, participating users only need to send encrypted output addresses and corresponding funds to the mixing group before the mixing service begins.

Decentralized mixing technologies have improved privacy and convenience, but they still have some limitations, such as susceptibility to DoS attacks and relatively low mixing efficiency. Currently, they are not able to provide fully mature privacy protection.

2. Ring signature technology

Ring signature is a simplified form of group signature where the core idea is that users (members of the ring set) can utilize a "ring" signature that represents the members of the ring [18,19], but it cannot be attributed to any specific signer. The ring set is a collection without any central authority, providing features such as non-repudiation, privacy, and traceability, significantly enhancing the privacy of blockchain transactions.

CryptoNote [20] is one of the most prominent protocols based on ring signature technology. It utilizes linkable ring signatures to conceal the sender's address in transactions. Monero [21], a cryptocurrency, is built upon this key technology. Monero implements ring confidential transactions (RingCT), which leverage homomorphic commitments and ring signatures to achieve range proofs, thereby hiding transaction amounts while ensuring public verifiability.

Due to its anonymity and anti-counterfeiting properties, ring signature technology has been widely applied in fields such as intellectual property protection and anonymous voting. However, ring signature privacy protection schemes also face security threats. Research studies [4,22,23] have demonstrated privacy vulnerabilities in ring signature technology. In such schemes, the length of signatures increases as more members join, and there are also issues related to malicious repudiation of signatures by signers and vulnerability to key selection attacks.

3. Stealth address

The stealth address technology primarily aims to hide transaction addresses and provide privacy protection for the recipients [24]. The sender computes a temporary address before the transaction, and the payment tokens are sent to that address. The true recipient can confirm the transaction's validity using the corresponding keys, thereby obfuscating the association between the transacting parties.

Many research efforts have focused on improving stealth address schemes. Fan et al. [25] proposed a scheme that reduces the storage cost for users by only requiring them to store a key pair during initialization. In the original protocol, the recipient needs to continuously scan transaction data and compute the target address, which poses limitations in resource-constrained environments like the Internet of Things (IoT). To address this, Fan et al. [26] introduced DKSAP-IoT, a double-key stealth address protocol based on blockchain, which achieves faster communication between parties and reduces the size of transactions.

Although stealth address schemes have their merits, they are not without flaws. Each anonymous transaction requires the inclusion of a transaction's temporary public key, making the privacy of such transactions susceptible to identification, resulting in the loss of certain private information. Additionally, there are efficiency issues, such as the need for the recipient to repeatedly perform elliptic curve scalar multiplication operations. These challenges need to be addressed, especially in scenarios like data transactions that prioritize both privacy and service performance.

2.2. Direct Privacy Protection

The direct privacy protection approach aims to directly hide crucial transaction information and is primarily based on zero-knowledge proof technology. The critical aspect of this approach is that when the prover proves the correctness of a message to the verifier, they do not need to disclose specific content. As a result, the verifier cannot obtain any unnecessary information, thereby achieving the goal of protecting transaction privacy.

Zcash, proposed by Ben-Sasson et al. [27], is a typical example of applying the non-interactive zero-knowledge proof. When proving the validity of transactions, Zcash does not reveal any private information to the relevant verifiers, thus ensuring strong privacy. Although Zcash achieves good anonymity, the generation of initialized parameters requires the participation of trusted parties, making it inherently weakly centralized. Furthermore, the efficiency and throughput of proof generation in Zcash are not ideal. To address these issues, Benedikt et al. [28] designed the Bulletproof protocol, which uses short proof and does not rely on the trusted setup. It effectively improves the speed of proof generation through secure multiparty computation. Li et al. [29] applied the Bulletproof protocol to data privacy protection in energy trading scenarios. They used the Bulletproof protocol to provide range proofs for transaction amounts, hiding the specific transaction amounts. Additionally, to address the issue of the authenticity of auditing receipts, they combined zero-knowledge proof and the ElGamal encryption technique, enabling regulators to reliably monitor transaction privacy data without revealing commitments. As the number of network nodes increases, the data volume corresponding to computing proofs in the Bulletproof protocol also increases, implying a higher cost of storing this data in the blockchain and posing new challenges.

2.3. Comparison of Privacy Protection Methods

In the realm of blockchain transaction privacy protection, each privacy protection mechanism exhibits unique advantages and limitations in terms of privacy preservation and processing efficiency.

Coin mixing technology, by amalgamating transactions from multiple users, complicates the tracing of transaction origins. However, this method may decrease system efficiency, particularly when processing a large volume of transactions. The stealth address scheme generates the unique address for each transaction, offering a higher degree of privacy protection, and is relatively more efficient, especially in comparison to coin mixing. However, the key to stealth address schemes is that they can only conceal the identity of the receiver while exposing other aspects of the transaction (such as the sender's identity and amount). The ring signature technology, by obscuring the true sender within a group of users, provides robust anonymity for the sender but may incur increased complexity in the verification process and a reduction in system efficiency. In contrast, direct privacy protection schemes such as zero-knowledge proof technology provide the highest level of privacy protection, which enables the validation of transactions' legitimacy without disclosing any specific details. However, this elevated level of privacy protection is often accompanied by heightened computational complexity and reduced transaction processing efficiency.

In summary, selecting an appropriate privacy protection mechanism necessitates finding a balance between privacy and efficiency. While the coin mixing scheme and the stealth address scheme provide effective privacy protection to a certain extent, each has its limitations. The ring signature scheme and the zero-knowledge proof approach offer a deeper level of anonymity but may come at the cost of reduced processing efficiency. Therefore, in practical applications, it is crucial to choose the most suitable privacy protection strategy based on the specific context and requirements.

3. Proposed Method

In order to clearly demonstrate the method proposed in this article, this section will be divided into the following subsections: Section 3.1 (Question Quotation) mainly explains what problems the method designed in this article aims to solve and what challenges exist. Section 3.2 (Design Principle) mainly explains some of the ideas this article considers when designing solutions for these problems and challenges—that is, how some key technologies of the entire solution are designed. Section 3.3 (Framework Design) mainly describes the scheme proposed in this article from a framework perspective, clarifying the design of the technical framework of the entire method and the participants. Section 3.4 (Process Design) elaborates on the specific implementation process and technical details of the proposed method.

3.1. Question Quotation

The public and transparent nature of blockchain brings transparency to transactions, but it also introduces the risk of transaction privacy information leakage. Although users conduct transactions on the blockchain under anonymous identities, once the transaction behavior is linked to a real identity, attackers can infer the real identities of blockchain users with a high success rate using methods like clustering analysis, leading to privacy breaches.

The UTXO-based note mechanism scheme in Zcash uses zk-SNARKs to generate relevant proofs, which involves circuit encoding for proofs. This approach has slow execution speed, high transaction fees, and requires a complex trusted setup. Additionally, not all transactions in Zcash are automatically private, leading to the risk of privacy information leakage. In the DKSAP protocol, temporary addresses are generated for each transaction to hide the receiver's real address, but the transaction still needs to include the temporary public key R, resulting in the loss of certain privacy information. Moreover, the receiver needs to continuously scan the blockchain's transaction data and perform elliptic curve scalar multiplications to verify if they are the authorized recipient of the transaction. Both of these schemes have issues in terms of performance and transaction privacy security, especially in high-performance transaction service scenarios. To address

these issues, this paper improves upon the drawbacks of both schemes and combines their advantages to propose the Blockchain Transaction Privacy Protection (BTPP) method for protecting transaction privacy in blockchain.

3.2. Design Principle

The BTPP method aims to protect transaction privacy in blockchain, which is based on the stealth address and the note mechanism. In the BTPP method, a stealth address is embedded in the note as the temporary address of the recipient. The uniqueness of the stealth address is used to identify the note, ensuring that even if the note content is leaked, the specific recipient information cannot be inferred. In the calculation of the stealth address, the BTPP method utilizes the randomness seed for generating the temporary private key r based on the Diffie–Hellman key exchange protocol. This approach avoids the issue of exposing the temporary public key in the transaction. Both parties involved in the transaction can independently compute their corresponding temporary addresses, allowing the recipient to determine if they are the intended real recipient.

In the maintenance of the world state of the note, the Note Commitments table maintains the mapping between the note and its corresponding stealth address. The Nullifier table utilizes the uniqueness of the stealth address to serve as the invalidation identifier for the note. By leveraging the binding relationship between the note and its stealth address, the ownership of the note is transferred to the ownership of the confirmation key t corresponding to the stealth address of the note. Only the genuine recipient can compute the confirmation key t, enabling efficient transfer, proof, and verification of note ownership. The Nullifier table uses the stealth address associated with the note as its invalidation identifier, and when verifying the correctness of the invalidation identifier for a note, it only needs to validate the binding relationship between the stealth address and the note, achieving efficient verification of the invalidation identifier for the note.

3.3. Framework Design

The framework of the BTPP method is illustrated in Figure 1 and involves the following roles:

- Transaction Sender (Bob): Bob is responsible for constructing the output notes, generating Pedersen commitments and Bulletproofs range proofs for the relevant amounts, proving ownership of input notes, and transferring ownership of the output notes to the recipient.
- Transaction Recipient (Alice): Alice is responsible for generating Pedersen commitments and Bulletproofs range proofs for the relevant amounts, verifying the correctness and ownership of the payment note received, and receiving payment notes representing the income.
- Smart Contract: In the BTPP method, the smart contract acts as the executor of the transaction and is responsible for carrying out the necessary verification and transaction logic to ensure fair transactions between the parties.
- Blockchain: The blockchain, maintained by relevant nodes, is responsible for maintaining the world state of the notes, verifying the validity of transactions, achieving consensus, and providing a secure execution environment for smart contracts.

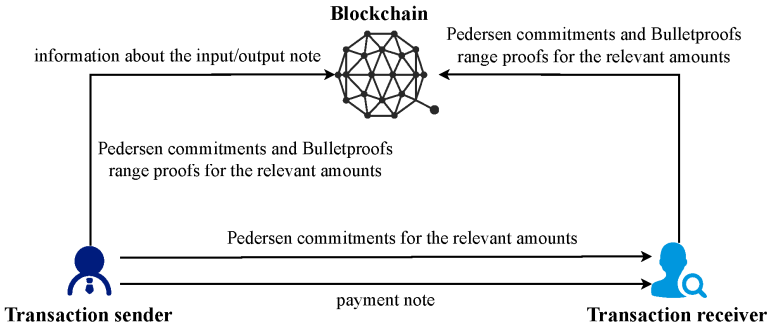


Figure 1. Framework of the BTPP method.

The BTPP method comprises a set of polynomial-time algorithms, including *GSetup*, *UserKeyGen*, *NoteSRCH*, *Random*, *NoteCreate*, *Commit*, *Open*, *RangeProve*, *RangeVerify*, *OVerify*, *NfVerify*, and *TVerify*. These algorithms are formally defined as follows:

1. $\mathrm{pp} \leftarrow GSetup(1^{\lambda})$: Public parameter generation: Given the security parameter λ as input, the algorithm outputs the public parameters pp.
2. $(pk_{\mathrm{u}}, sk_{\mathrm{u}}) \leftarrow UserKeyGen(pp, pk_{\mathrm{ca}}, ID_{\mathrm{u}})$: User key pair generation: Given the public parameters *pp*, the main key pk_{ca} of the CA, and the user identifier ID_{u} as input, the algorithm outputs the user's asymmetric key pair (pk_{u}, sk_{u});
3. $note_{\mathrm{in}} \leftarrow NoteSRCH(v, note_{\mathrm{u}})$: Search for eligible input notes: Given the user's list of notes and the transaction amount v as input, the algorithm searches for notes that exactly match the value v. If there is a note with a value equal to v, it is outputted. Otherwise, if there are multiple notes whose combined value is greater than or equal to v, they are outputted.
4. $r \leftarrow Random(seed)$: Random number generation: The *Random* function takes a random seed as input and generates a random number r. The random number r is generated differently for different random seeds. The *Random* function is primarily used when calculating stealth addresses to generate consistent temporary public–private key pairs based on shared knowledge between the transaction parties.
5. $note_{\mathrm{out}} \leftarrow NoteCreate(v, r, S, B)$: Construction of transaction output note. The input includes the note value v, the transaction temporary private key r, the recipient's scan public key S, and the spend public key B. The output is the transaction's corresponding output note $note_{\mathrm{out}}$.
6. $C \leftarrow Commit(x, r)$: Pedersen commitment generation. The input consists of the target value x to be bound and hidden in the commitment and the random blinding factor r. The output is the commitment C that binds x while revealing no information about x.
7. $1/0 \leftarrow Open(C, x, r)$: Validity verification of Pedersen commitment. The input consists of the commitment C that binds the target value x, the target value x that is bound and hidden in the commitment, and the blinding factor r. The output indicates whether the commitment C is indeed bound to the claimed value x.
8. $\pi \leftarrow RangeProve(x, r, C)$: Generation of range proof based on Bulletproofs. The input consists of the value x, the Pedersen commitment C to x, and the blinding factor r used for verifying whether a hidden integer in a Pedersen commitment is equal to a given value. The output is a proof π that demonstrates that the value x is within a specified range, specifically proving that x is non-negative.
9. $1/0 \leftarrow RangeVerify(\pi, C)$: Verification of the correctness of a range proof based on Bulletproofs. The input consists of the range proof π and the corresponding Pedersen commitment C. If π matches C and the hidden value in commitment C is a non-negative integer, the verification passes. Otherwise, the verification fails.
10. $1/0 \leftarrow OVerify(t, T)$: Ownership verification of a note. The input consists of the stealth address T and its corresponding address confirmation key t. The *OVerify* algorithm determines whether the confirmation key t can successfully confirm the stealth address T. The output indicates whether the owner of the confirmation key t has ownership of the note bound to the stealth address T.
11. $1/0 \leftarrow NfVerify(nullifier, T)$: Note validity verification. The *NfVerify* algorithm is primarily responsible for verifying whether the input note has a "double spend" issue. It takes the Nullifier table, which contains nullifiers of invalidated notes in the current note world state, and the nullifier T corresponding to the note, as input. The *NfVerify* algorithm checks whether T exists in the Nullifier table. The output indicates whether the note bound to the stealth address T has been double-spent.
12. $1/0 \leftarrow TVerify(C_{\mathrm{p}}, C_{\mathrm{q}}, r_{\mathrm{p}}, r_{\mathrm{q}})$: Transaction input–output balance verification. The *TVerify* algorithm is primarily responsible for verifying whether the transaction input and output amounts are balanced, meaning that the sum of input amounts equals the sum of output amounts. The algorithm takes the sender's commitment C_{p} to the amounts and chosen blinding factors r_{p}, as well as the receiver's commitment C_{q} to

the amounts and chosen blinding factors r_q, as input. The output indicates whether the transaction inputs and outputs are balanced.

3.4. Process Design

BTPP consists of three stages: the construction of output notes, the generation of transaction amount commitments and range proofs, and the verification of transaction legality. The processes for each stage are shown in Figure 2. The following text provides a detailed description of each stage's specific procedures. The meanings of the symbols used are listed in Table 1.

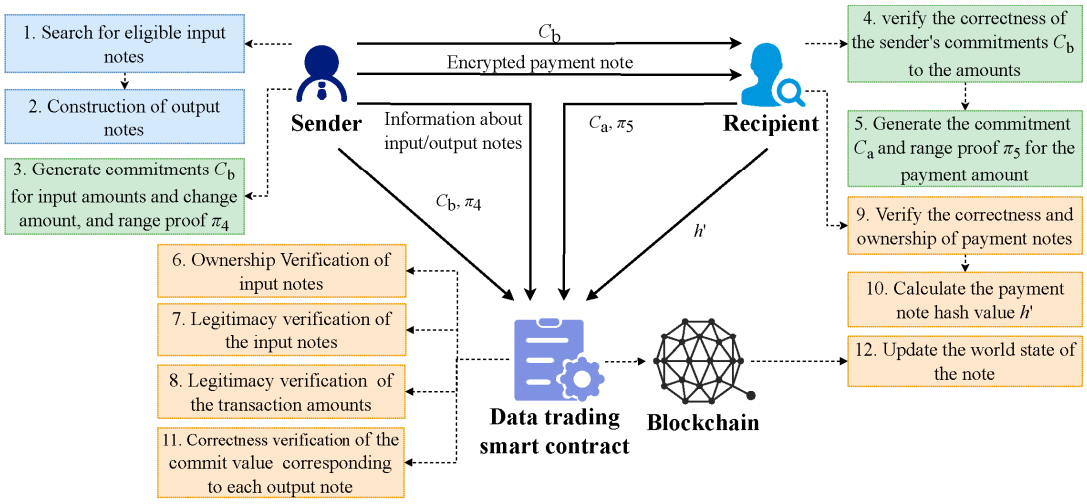


Figure 2. Process of the BTPP method.

Table 1. Meaning of each symbol.

Symbol	Meaning
(s_a, S_a)	The scan key pair of the recipient
(s_b, S_b)	The scan key pair of the sender
(b_a, B_a)	The spend key pair of the recipient
(b_b, B_b)	The spend key pair of the sender
$note_i$	The i-th note
T_i	The corresponding stealth address of the note $note_i$
t_i	The confirmation key corresponding to the stealth address T_i
(r_i, R_i)	The temporary public–private key pair for the transaction corresponding to the note $note_i$
v_i	The value corresponding to the note $note_i$
c_i	The transaction shared key corresponding to the note $note_i$
C_i	Pedersen commitment to the amount v_i
π_i	Range proof for the amount v_i
h_i	The hash value corresponding to the note $note_i$

3.4.1. The Initialization Phase

Initialization is primarily performed by the CA, which accepts the security parameter λ. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be the additive cyclic group and the multiplicative cyclic group, **G** be the elliptic curve group, G be the generating element of the group **G**, H $\in$ **G**, and the discrete logarithm of H is unknown. Choose two different generating elements P and q from $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and choose the bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, two secure hash functions $H_1 : \{0,1\}^* \times \mathbb{G}_1 \to Z_q^*$ and $H_2 : \{0,1\}^* \times \mathbb{G}_2 \to Z_q^*$, and randomly choose $s \in Z_q^*$ as the private key sk_{ca} of the CA, and the public key $pk_{ca} = sP$, $sP \to \mathbb{G}_1$. Finally, the common parameter pp of the system is obtained :

$$Gsetup\,(1^\lambda) \to pp \tag{1}$$

where $pp = (\lambda, \mathbb{G}_1, \mathbb{G}_2, e, P, q, pk_{ca}, H_1, H_2)$. The system defines two large prime numbers p and a, where p is a prime number, and a is a primitive root modulo p. The transaction user provides unique identifier *ID* to the CA, and the CA generates the corresponding user key pair (*pk*, *sk*) using the key generation algorithm *UserKenGen*:

$$UserKeyGen(pp, pk_{ca}, ID) \rightarrow (pk, sk) \tag{2}$$

Suppose the public–private key pair of the transaction sender is (pk_b, sk_b), the public–private key pair of the transaction receiver is (pk_a, sk_a), and each transaction participant user publishes its public key *pk*.

The transaction user generates the dual key pairs, i.e., the scan key pair is (s, S) and the spend key pair (b, B), where $S = s \cdot G$, $B = b \cdot G$. Assume that the dual key pairs for the transaction sender are (s_b, S_b) and (b_b, B_b), and the dual key pairs for the transaction receiver are (s_a, S_a) and (b_a, B_a).

3.4.2. The Construction Phase of the Output Note

Suppose the transaction sender Bob's note list is $note_b = [note_1, note_2, note_3]$, where $note_1 = \langle T_1, v_1 = 5, R_1 \rangle$, $note_2 = \langle T_2, v_2 = 6, R_2 \rangle$, $note_3 = \langle T_3, v_3 = 2, R_3 \rangle$, and the amount of the transaction is $v = 9$. The interaction flow of each participant is shown in Figure 3, and the detailed process is described below.

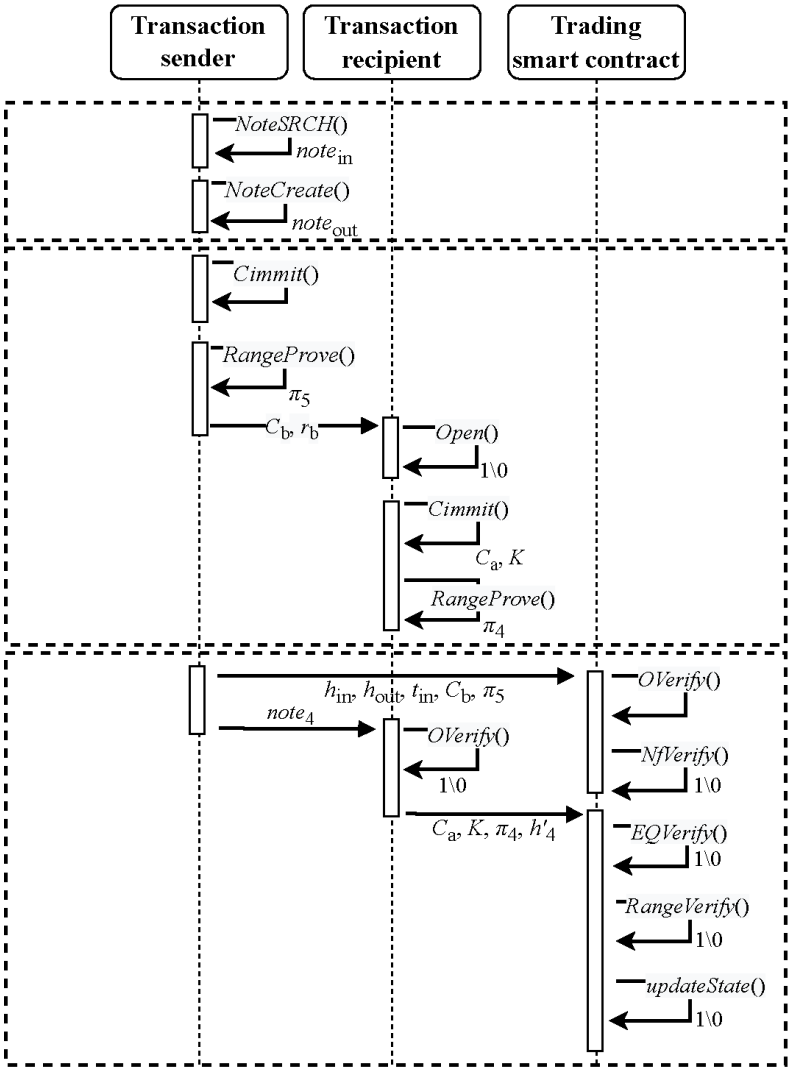


Figure 3. BTPP timing diagram.

1. The transaction sender Bob finds the notes that meet the transfer criteria

If Bob's note list happens to have note with v = 9, then the number of input and output notes is 1 each; otherwise, Bob needs to make a combined payment. This process corresponds to step 1 in Figure 2. The process is as follows:

(1) Bob finds one or more of notes that have not yet been spent, i.e., $note_1$, $note_2$ and $note_3$;
(2) Bob decrypts the notes by the private key sk_b to obtain the values and other information;
(3) Bob finds the notes that matche the transfer amount v as the input notes for this transaction, i.e., $note_\mathrm{in} = [note_1, note_2]$, satisfying $v_\mathrm{in} = v_1 + v_2 \geq v$, and the whole process is shown as Equation (3):

$$NoteSRCH(v, note_\mathrm{b}) \rightarrow note_\mathrm{in} \tag{3}$$

2. The transaction sender Bob constructs the output note

This process corresponds to step 2 in Figure 2. When the total value v_in of the input notes $note_\mathrm{in}$ exceeds the amount v, Bob needs to generate payment note $note_4$ and change note $note_5$ with values $v_4 = 9$ and $v_5 = v_\mathrm{in} - v_4 = 2$, respectively, with the following construction process:

(1) Bob obtains the scan public key S_a and the spend public key B_a of Alice;
(2) Bob chooses a random integer k_1, i.e., $0 < k_1 < p - 1$, and computes and sends the public key Pub_b:

$$pub_\mathrm{b} = a^{k_1} \bmod p \tag{4}$$

(3) Alice chooses a random integer k_2, i.e., $0 < k_2 < p - 1$, and computes and sends the public key pub_a:

$$pub_\mathrm{a} = a^{k_2} \bmod p \tag{5}$$

(4) Bob receives the public key pub_a calculated by Alice and uses the private key k_1 to calculate the shared key K_1:

$$K_1 = {pub_a}^{k_1} \bmod p \tag{6}$$

(5) Alice receives the public key pub_b computed by Bob and uses the private key k_2 to compute the shared key K_2:

$$K_2 = {pub_b}^{k_2} \bmod p \tag{7}$$

(6) Both sides of the transaction have the same shared key, i.e., $K_1 = K_2$, only both sides of the transaction know the shared key, and both sides use the shared key to compute the randomness seed:

$$seed = Hash(k_1) = Hash(k_2) \tag{8}$$

where Hash(·) is the secure hash function.
(7) Bob calculates the temporary public–private key pair (r_4, R_4) for the payment note $note_4$ transaction based on $seed$:

$$Random(seed) \rightarrow r_4 \tag{9}$$

$$R_4 = r_4 \cdot \mathrm{G} \tag{10}$$

(8) Bob randomly selects the random number r_5 of the change note $note_5$ transaction to generate the public–private key pair (r_5, R_5);
(9) Bob computes the shared keys $c_4 = H(r_4 \cdot S_\mathrm{a})$, $c_5 = H(r_5 \cdot S_\mathrm{b})$ for notes $note_4$ and $note_5$, respectively;
(10) Bob calculates the stealth addresses $T_4 = c_4 \cdot \mathrm{G} + B_\mathrm{a}$ and $T_5 = c_5 \cdot \mathrm{G} + B_\mathrm{b}$ of $note_4$ and $note_5$, respectively, and the corresponding receiver Alice is also able to calculate the same shared key c_4 of the payment note $note_4$ based on the randomness seed $seed$,

thus calculating the corresponding stealth address T_4. The whole process is shown in Equation (11):

$$\begin{array}{c} NoteCreate(v_4, r_4, S_a, B_a) \rightarrow note_4 \\ NoteCreate(v_5, r_5, S_b, B_b) \rightarrow note_5 \end{array} \tag{11}$$

3.4.3. Generation Phase of Pedersen Commitment and Bulletproofs Range Proof for Relevant Amounts

In order to achieve concealment of the transaction amount, both parties to the transaction calculate the Pedersen commitments for the relevant amounts separately and generate the range proofs for the corresponding amounts based on Bulletproofs. This process corresponds to steps 3, 4, and 5 in Figure 2, the specific process of which is described below.

1. The transaction sender generates the commitments and range proofs for the relevant amounts

(1) The transaction sender Bob selects random numbers r_{b_1}, r_{b_2}, and r_{b_5} as blind factors and generates commitments C_1, C_2, and C_5 for input amounts v_1, v_2, and change amount v_5, respectively:

$$\begin{array}{c} Commit(v_1, r_{b_1}) = r_{b_1} \cdot G + v_1 \cdot H \rightarrow C_1 \\ Commit(v_2, r_{b_2}) = r_{b_2} \cdot G + v_2 \cdot H \rightarrow C_2 \\ Commit(v_5, r_{b_5}) = r_{b_5} \cdot G + v_5 \cdot H \rightarrow C_5 \end{array} \tag{12}$$

(2) Bob generates the range proof π_5 for the change amount v_5 based on Bulletproofs:

$$RangeProve(v_5, r_{b_5}, C_5) = \pi_5 \tag{13}$$

(3) Bob sends the commitments $C_b = [C_1, C_2, C_5]$ and the blind factor r_b to the transaction receiver Alice, where $r_b = r_{b_1} + r_{b_2} - r_{b_5}$.

2. The transaction receiver generates the commitments and range proofs for the relevant amounts

(1) After receiving the commitments from Bob, Alice first verifies the correctness of the commitments:

$$C_1 + C_2 = C_5 + r_b \cdot G + v_4 \cdot H \tag{14}$$

where $C_1 + C_2$ is Bob's commitment to the input of the transaction; C_5 is Bob's commitment to the change amount; and $r_b \cdot G + v_4 \cdot H$ is Bob's commitment to transfer the amount to Alice, i.e., to verify that the input and output of the transaction are balanced: $v_1 + v_2 = v_4 + v_5$;

(2) Alice chooses the random number r_a, generates the Pedersen commitment C_4 for the payment amount v_4, and computes K:

$$\begin{array}{c} Commit(v_4, r_a) = r_a \cdot G + v_4 \cdot H \rightarrow C_4 \\ K = (r_b - r_a) \cdot G \end{array} \tag{15}$$

(3) Alice generates the range proof π_4 of payment amount v_4 based on Bulletproofs:

$$RangeProve(v_4, r_a, C_4) = \pi_4 \tag{16}$$

3.4.4. Verification Stage of Transaction Legitimacy

In order to describe the change of the note world state before and after the transaction, assume that the current note world state is shown in Table 2. The Note Commitments table in the BTPP method maintains the mapping relationship between the hash of note and its stealth address, while the Nullifier table maintains the stealth address of the invalid note. The verification of transaction legitimacy includes the ownership and legitimacy verification of the input notes, the legitimacy verification of the transaction amount, and the correctness verification of the payment note. The entire process corresponds to steps 6 to 11 in Figure 2, the specific process of which is described below.

Table 2. The world state of the note before the transaction.

Note Commitments	Nullifier Set
$(h_1 = Hash(note_1) \rightarrow T_1)$ $(h_2 = Hash(note_2) \rightarrow T_2)$ $(h_3 = Hash(note_3) \rightarrow T_3)$	

1. Verify the sender's ownership of the input notes

(1) The sender Bob sends to the verification network the hash $h_{\text{in}} = [h_1, h_2]$ of the input note $note_{\text{in}}$, the corresponding confirmation key $t_{\text{in}} = [t_1, t_2]$, and the hash $h_{\text{out}} = [h_4, h_5]$ of the output notes $note_4$, $note_5$;

(2) The verification nodes calculate the corresponding stealth addresses $T_1{}' = t_1 \cdot \mathrm{G}$ and $T_2{}' = t_2 \cdot \mathrm{G}$ by the confirmation keys t_1 and t_2 and find the stealth addresses T_1 and T_2 corresponding to the note hash values h_1 and h_2 through the Note Commitments table, then determines Bob's ownership of $note_1$ by whether $T_1{}'$ and T_1 are equal. Bob's ownership of $note_2$ is determined by whether $T_2{}'$ and T_2 are equal. The whole verification process is shown in Equation (17):

$$\begin{array}{l} OVerify(t_1, T_1) \rightarrow 1/0 \\ OVerify(t_2, T_2) \rightarrow 1/0 \end{array} \tag{17}$$

2. Verify the legality of the input notes

The legitimacy verification of input notes is to detect whether there is a "double spend" problem with the input notes. Since the Note Commitments table maintains the binding relationship between the notes and their stealth addresses, while the Nullifier table maintains the stealth addresses of the invalid notes, the legitimacy of the input notes can be verified by determining whether the stealth addresses T_1 and T_2 bound to the input notes $note_1$ and $note_2$ exist in the Nullifier table, as shown in Equation (18):

$$\begin{array}{l} NfVerify(nullifier, T_1) \rightarrow 1/0 \\ NfVerify(nullifier, T_2) \rightarrow 1/0 \end{array} \tag{18}$$

3. Verify the legality of the transaction amounts

(1) Bob publishes the commitments $C_{\text{b}} = [C_1, C_2, C_5]$ and the range proof π_5 to the verification network;

(2) Alice publishes the commitment $C_{\text{a}} = [C_4]$, K and range proof π_4 to the verification network;

(3) The verification nodes verify the balance of the transaction input–output amounts, as shown in Equation (19):

$$C_1 + C_2 = C_5 + C_{\text{a}} + K \tag{19}$$

(4) The verification nodes verify that the transaction output amounts are non-negative, as shown in Equation (20):

$$\begin{array}{l} RangeVerify(\pi_4, C_4) \rightarrow 1/0 \\ RangeVerify(\pi_5, C_5) \rightarrow 1/0 \end{array} \tag{20}$$

4. The recipient verifies the correctness of the payment note

(1) Bob encrypts $note_4$ using Alice's public key pk_{a} to generate $note_4^{\text{enc}}$ and sends it to Alice;

(2) Alice decrypts $note_4^{\text{enc}}$ using the private key sk_{a} to obtain the payment note $note_4$ and verifies ownership of $note_4$ and the specific information of the note, completing the verification of the correctness of $note_4$:

$$OVerify(t_4, T_4) \rightarrow 1/0 \tag{21}$$

(3) Alice calculates the hash value $h_4{}' = Hash(note_4)$ of the payment note $note_4$ and publishes $h_4{}'$ to the verification network;
(4) The verification nodes verify the correctness of the commit value of the note, i.e., verifies that h_4 is equal to $h_4{}'$.

5. Update the world state of the note

Each verification node updates the Note Commitments table to record the binding relationship between the hash values $h_{\text{out}} = [h_4, h_5]$ of output notes ($note_4$, $note_5$) and the corresponding stealth addresses T_4 and T_5, indicating the output of this transaction, and updates the Nullifier table to record the invalid identifiers T_1 and T_2 of input notes ($note_1$, $note_2$), indicating the input of this transaction.

The entire process of this transaction is now complete. The world state of the note after the transaction is completed is shown in Table 3.

Table 3. The world state of the note after the transaction.

Note Commitments	Nullifier Set
$(h_1 = Hash(note_1) \rightarrow T_1)$	$nf_1 = T_1$
$(h_2 = Hash(note_2) \rightarrow T_2)$	$nf_2 = T_2$
$(h_3 = Hash(note_3) \rightarrow T_3)$	
$(h_4 = Hash(note_4) \rightarrow T_4)$	
$(h_5 = Hash(note_5) \rightarrow T_5)$	

4. Performance Analysis

The security of the BTPP scheme is based on cryptographic techniques, including the discrete logarithm assumption, secure hash functions, and public-key cryptography. This section will analyze the BTPP scheme from the perspectives of security and privacy.

4.1. Transaction Information Privacy Analysis

The BTPP scheme ensures the privacy of transaction information, including the transaction relationship and transaction amount of the parties involved. Apart from the transaction participants, no one, including the transaction verification party, can know the specific details of the transaction.

In terms of transaction relationship, the sender uses the recipient's dual public key information to generate the stealth address T for the transaction. This address is bound to the corresponding note as the recipient's temporary address, effectively hiding the recipient's actual address. Only the corresponding recipient possesses the confirmation key for the stealth address, which proves their ownership of the note. This process severs the transaction relationship between the trading parties, achieving unlinkability of the transaction relationship.

In terms of transaction amounts, the BTPP scheme employs commitments for amounts. For example, the commitment form of an amount v, where r is a random blinding factor chosen by the proving party, relies on the elliptic curve discrete logarithm problem, making it difficult for an attacker to compute the amount v, even if they obtain r. This ensures the protection of the transaction amount. The Pedersen commitment for the amount hides the transaction amount while completing the verification of transaction legality.

4.2. Transaction Security Analysis

4.2.1. Security of Ownership Verification of Notes

In the BTPP scheme, the transaction parties utilize the Diffie–Hellman key exchange protocol to compute the shared secret keys k_1 and k_2 based on their respective knowledge. Using this shared secret key, they calculate the stealth address T for the payment note. However, only the authorized recipient possesses the confirmation key $t = c + b$, where b represents the spend private key known only to the recipient, and c represents the transaction shared secret key known only to the transaction parties. As a result, only the

legitimate recipient can prove ownership of the note by computing $T = t \cdot \mathrm{G}$. Therefore, the BTPP scheme ensures that only the true recipient of the transaction can provide proof of ownership for the note.

4.2.2. Security of Legitimacy Verification of Notes

The BTPP scheme has updated the maintenance of the Note Commitments table and the Nullfier table. It utilizes the uniqueness of the stealth address as the invalid identifier for note and binds the stealth address to the corresponding note. When verifying the legitimacy of an input note, it is only necessary to check the correctness of its invalid identifier, which is the stealth address, by verifying whether there is a binding relationship between the invalid identifier and the input note. No one can tamper with the data on the blockchain. Therefore, the BTPP scheme ensures the security of validating the legitimacy of input notes and prevents "double spend" issues.

4.2.3. Security of the Stealth Address

For the calculation of the stealth address $T = c \cdot \mathrm{G} + B = Hash(r \cdot S) \cdot \mathrm{G} + B$, where S, B, and G are public information, but for the temporary private key $r = Random(seed)$ of the transaction, the randomness seed of its calculation comes from the common knowledge known only to the two parties of the transaction, and only the two parties of the transaction know the temporary private key r to calculate the same stealth address T_{r}, and only the corresponding recipient has the confirmation key t for the stealth address and can prove ownership of it.

4.2.4. Security of Legitimacy Verification of the Relevant Transaction Amounts

The BTPP scheme achieves concealment of the transaction amounts via the Pedersen commitments and range proofs based on Bulletproofs. Taking the previously described transaction as an example, for the sender's commitments $C_{\mathrm{b}} = [C_1, C_2, C_5]$ for the amounts and the blind factor $r_{\mathrm{b}} = r_{\mathrm{b}_1} + r_{\mathrm{b}_2} - r_{\mathrm{b}_5}$, the receiver verifies the correctness of the payment amount based on Equation (14), i.e., $v_1 + v_2 = v_4 + v_5$, and only the sender's correctly generated commitments to the input amounts and the change amount can pass the receiver's verification of the payment amount. Meanwhile, for the sender's C_{b} and the receiver's C_{a} and K, the verification nodes verify the balance of transaction inputs and outputs as follows:

$$\begin{aligned} C_1 + C_2 &= C_5 + C_{\mathrm{a}} + K \\ &= r_{\mathrm{b}_1} \cdot \mathrm{G} + v_1 \cdot \mathrm{H} + r_{\mathrm{b}_2} \cdot \mathrm{G} + v_2 \cdot \mathrm{H} = r_{\mathrm{b}_5} \cdot \mathrm{G} + v_5 \cdot \mathrm{H} + r_{\mathrm{a}} \cdot \mathrm{G} + v_4 \cdot \mathrm{H} + (r_{\mathrm{b}} - r_{\mathrm{a}}) \cdot \mathrm{G} \\ &= v_1 \cdot \mathrm{H} + v_2 \cdot \mathrm{H} = v_5 \cdot \mathrm{H} + v_4 \cdot \mathrm{H} \\ &= v_1 + v_2 = v_4 + v_5 \end{aligned}$$

Only when both transaction parties correctly generate commitments to the relevant amounts can the verification notes verify the balance of input and output amounts of the transaction. Additionally, the range proofs provided by both parties for the output amounts ensure that the output amounts are not negative. Therefore, the BTPP scheme can securely verify the legality of transaction amounts.

4.2.5. Integrity of Blockchain Transaction

The stealth address protects the recipient's identity by generating the temporary address for each transaction, meaning outside observers cannot trace the recipient's identity or transaction history through the address. Although the BTPP protocol uses the temporary address, the integrity of the transaction is still verified through the public ledger of the blockchain, ensuring the non-tamperability and integrity of the transaction. In addition, the BTPP protocol utilizes the Pedersen commitment combined with the Bulletproofs range proof to ensure the legitimacy of the transaction amount without revealing any specific information. Although the transaction amount is hidden, through the generated commitments and range proofs, network participants (such as miners or validators) can still

verify the validity of the transaction. This means that the transaction still follows the basic rules of the blockchain, such as not exceeding the account balance and maintaining the integrity of the transaction. At the same time, through the note mechanism, the blockchain network can perform necessary compliance and integrity checks, such as preventing double-spending, without revealing the privacy details of the transaction.

5. Experiment and Result Analysis

5.1. Experiment Environment

The experiments in this section are all implemented by VMware, a virtual machine with 4 GB of memory, based on the Javascript language, using the secp256k1 curve $y^2 = x^3 + 7$ provided by the noble-secp256k1 library; utlizing the libsecp256k1 library and the elliptic library to implement the related scalar operations, modulo, power and other operations on the elliptic curve secp256k1; and providing the source of randomness based on the bigint-crypto-utils library. The experimental hardware environment configuration is shown in Table 4.

Table 4. Hardware environment of the experiment.

Hardware	Configuration
Operating System	Ubuntu 20.04.5 LTS (64 bit)
Processor	Intel (R) Core (TM) i7-7700 CPU @ 3.60 GHz (4 cores)
Memory	4 GB
Graphics board	AMD Radeon (TM) R7 350

5.2. Experiment Purpose

The experimental design is based on the key processes involved in the proposed BTPP scheme, including specific implementation details such as the construction of the transaction output note, the generation of the Pedersen commitments and range proofs, and the legitimacy verification of the transaction so as to verify and evaluate the efficiency and feasibility of the BTPP scheme, and the experimental purposes can be divided into the following points:

- To verify the correctness of the BTPP scheme. One of the purposes is to verify the feasibility and correctness of the key steps of the BTPP scheme and thus verify that the BTPP scheme is correct.
- To test the efficiency improvements brought by the relevant designs in the BTPP scheme and verify its efficiency and feasibility. The BTPP scheme is able to achieve efficient note ownership transfer and verification operations by improving the generation of stealth addresses and the note mechanism. One of the purposes is to verify the efficiency of the relevant operations, thus verifying the efficiency and feasibility of the BTPP scheme.

5.3. Experimental Process and Result Analysis

5.3.1. Construction of the Output Notes

1. Initialization of the receiver's dual key pair

The receiver Alice initializes the dual key pair, i.e., the scan key pair (s, S) and the spend key pair (b, B), where the information of the generation element G and the associated key pair parameters are shown in Figure 4.

2. Construction of the output notes

The sender Bob needs to construct the payment note by first calculating the stealth address T for this transaction, which is shown in Figure 5, including the calculation of the randomness seed, the temporary key pair (r, R), and the shared key c. Taking the above transaction as an example, the information of the final payment note is shown in Figure 5.

```
G: [55066263022277343669578718895168534326250603453777594175500187360389116729240n,
    32670510020758816978083085130507043184471273380659243275938904335757337482424n]

------Alice's dual key pairs-----
The scan key pair (s, S):
   s: efc937c102e9fba4a637a31ad7e42025ff744331dda55a70e49ffe65011e0516
   S: 0268d096e8a597246b6cbfbaf67d71d3737d9bedc3317d1254206ab4298b4ba90c
The spend key pair (b, B):
   b: fa6026bf304b8c0ab2226d335a8418866a3ba2de84614ce43e957840eacd4c8
   B: 023ec681d17578f14537eeebb4ca86827a3c5fb2e9f9b405cf3733c99d9603d613

Initialization time consumption ofthe receiver's dual key pairs: 133.096 ms
```

Figure 4. Initialization of the receiver's dual key pair.

```
1. Bob calculates the randomness seed
   seed: d9a0012624759efcfe41c711712ac2b58770f968632addb5f19d1a7d05d2d8
2. Bob generates r based on seed and calculates R
   r: 1327a5d68b40db550307edffcaa5f384d00eec0ed183b45d8e0973793bb90f06
   R: 02d7781d74fe017b929cff94dcd42e6d3b5dda718c150738a548437b97f3543eb
3. Bob calculates the transaction shared key c=H(S·r) and the stealth address T=c·G+B
   c: 698b69020f3247b1546750a2b2aaeed830f52cbb19ef457b6b6014c8130c00d9
   T: 034d52d51df16657a638b7bd4ce76b438068fb832888371ba3066f695adb47965
Time consumed by Bob to calculate the stealth address T: 9.779 ms

The payment note: [
   034d52d51df16657a6387bd4ce76b438068fb83288837f1ba3066f695adb47965,
   9,
   1327a5d68b40db550307edffcaa5f384d00eecOed183b45d8e0973793bb90f06 ]
Time consumed by Bob to construct the payment note: 12.162 ms
```

Figure 5. Construction of the payment note.

After testing, the time consumed for initialization of dual public key pairs, calculation of stealth address and construction of output note for transaction users are 133.096 ms, 9.779 ms and 12.162 ms respectively, all in milliseconds.

5.3.2. Generation of Pedersen Commitments and Range Proofs

Taking the previously described transaction as an example, in order to achieve the hiding of the transaction amount, both parties of the transaction generate the Pedersen commitments for the relevant amounts and the range proofs for the output amounts based on Bulletproofs, respectively. The specific process is as follows.

1. Parameter initialization

The initialization information for generating elements G, H, and the reasonable range of amounts $[2^{lower}, 2^{upper}]$ is shown in Figure 6.

```
--- Parammeters Initialization ---
G: [  2181798a056c528d95936cb7383029bfc3ac2eeb15627ef9dc1e6f99,
      310d4b81f423fe14199c41229a15fd1764384422a24fbfc0311957627726a3120eb6 ]
H: [ 2803ac03fba6b316d547b3ca3b22078a5a17a5e97b4b60312552de374c1a01424a6,
     3a38904308c4fc2968c3022bc5d9236dac26d7d021e637cb249b8122863973c74f1 ]
Setting of the legal amount range [2 ^ lower, 2 ^ upper]:  { lower: 0n, upper: 32n }

Time consumed for initialization of parameters: 2.963 ms
```

Figure 6. Initialization of parameters related to the generation of Pedersen commitments and range proofs.

2. The sender generates the Pedersen commitments for the relevant amounts and range proofs based on Bulletproofs

The sender Bob first generates the Pedersen commitments C_1, C_2, and C_5 for input amounts v_1, v_2, and change amount v_5. The generation process is shown in Figure 7, while the range proof π_5 for output amount v_5 is generated through the global reasonable range $[2^{lower}, 2^{upper}]$ of output amounts. The whole process is shown in Figure 8.

```
--- The sender Bob generates the Pedersen commitments for the relevant amounts ---
1. Bob randomly generates blind factors rbl, rb2, rb5
    rb1: 55338067807769969670935055776253102871166629655703779270349862445458034231204n
    rb2:95740227587188792006625217288033594890524001114669414572577276541763926812095n
    rb5:38024494950089430999362125273251269808193280196788394576231867382992312856747n

 2. Bob generates Pedersen commitments C1, C2, C5 for v1, v2 and v5 respectively
    C1: 04e562c940a7034d6b5d5c8e1637bffd37b18807b285a45268eea74befece352ca13be5dbcb35dc3a8
        1b0b609d62cc93904ccd1b48ea09cad940459713b151e
    C2: 0482435a322d643cab6c0a3faef97c5982208d59def054b300806bd2751b464c13e558a0f59c68c507b
        e082acc6103af034a7802ad930c35a66c77ccb206284ee5
    C5: 0417d00ef4b9c4b359a6bdea5e29e496733fd137267dcf1a6b23be0ec19e1750da30494254e5fcd5effb
        7026618e7ea712fd9b97c6118418abae2cbf8b8b059046

  3. Bob calculates rb = rbl + rb2 − rb5
     rb: 113053800444869330678198147791035427953497350573584799266695271604229648186552n

  Time consumed by the sender to generate Pedersen commitments C1, C2, C5 respectively:
  15.901 ms, 16.393 ms, 16.177 ms
```

Figure 7. The sender generates the Pedersen commitments for the relevant amounts.

```
--- Sender Bob generates the range proof for the output amount v5 ---
1. Reasonable range setting for output amount [2 ^ ower, 2 ^ upper]: { lower: 0n, upper: 32n }
2. Generate the range proof for the output amount v5: π5 = RangeProof(v5, rb5, C5) :
π5:{
   'V': '0417d00ef4b9c4b359a6bdea5e29e496733fd137267dcfla6b23be0ec19e1750da30494254e5fcd5e
        fb7026618e7ea712fd9b97c61f84f8abae2cbf8b8b059046',
   ,,,
   'ind': [
     { 'L': '04633f91dce03696b22367619961857ec9ab9ce4e047e43f768930efe7f770e84139497a4
            e12at31d4lebtlea5896736e88517610701635303',
       'R': '04fe8ada0944bb635337d64bab7e66792d4642276f0cdf9312bfc65aaea3de45ccdab3fe01c92
            1274cf78074e98008d8b37362280eb7688503c5dd84352bda97c5'
     }, ...
   ], ...
}

Time consumed by the sender to generate the range proof for the output amount v5: RangeProve(v5,rb5,C5): 640.071 ms
Size of the range proof π5: 2.78027 kb
```

Figure 8. The sender generates the Bulletproofs-based range proof for the output amount v_5.

3. The recipient generates the Pedersen commitments for the relevant amounts and range proofs based on Bulletproofs

The receiver Alice generates the Pedersen commitment C_4 for the payment amount v_4 (the information related to its calculation results is shown in Figure 9) and generates the range proof π_4 for the output amount v_4. The calculation results are shown in Figure 10.

The time consumed by both the sender and the receiver to generate Pedersen commitments for the relevant amounts based on the algorithm *Commit* is about 16 ms, and the time consumed by the algorithm *RangeProof* to generate range proofs for the output amounts is about 640 ms. The time consumed for the relevant operations is in the millisecond range, and the size of the range proof is only about 2.8 kb, which is efficient in proving the legitimacy of the transaction amounts.

```
------Receiver Alice calculates the Pedersen commitment C4 for v4------
1. Alice randomly selects the blind factor ra
   ra: 427212888643826525296006108345138685949180949965759111402079200009506100616 1n
2. Alice calculates the commitment C4 as well as K
   C4: 0447fe61d9cf96e65c7f6a2280b13b3fa5c94ecb96cea4d631d23b10a27eeeda74423813d6396b
       44cfe1e981dcf1e9f80c7b315d5feed3e151bb5c492057d15f20
   K: 046220abafc7a82f9430cf7079698659649fceaf9c323b582947b079e7bfc3699c8aea090cf9e40
      f6973c57b890007e0ad765982a66f7037962b55ed130f20c640

Time consumed by receiver Alice to generate the Pedersen commitment C4: Commit(v4,ra): 16.014 ms
```

Figure 9. The recipient generates the Pedersen commitments for the relevant amounts.

```
------Receiver Alice generates the range proof for the output amount v4------
1. Reasonable range setting for output amount [2^lower, 2^upper]:  { lower: 0n, upper: 32n }
2. Generate the range proof for the output amount v4: π4=RangeProof(v4,ra,C4):
π4: {
  'V': '0447fe61d9cf96e65c7f6a2280b13b3fa5c94ecb96cea4d631d23b10a27eeeda74423813d6396b44c
        fe1e981dcf1e9f80c7b315d5feed3e151bb5c492057d15f20',
  ...
  'ind': [
    { 'L': '04175acb74e780c414072df32c575a03ceed23dca58896b1b5562f6c9dda9ec51ece1e049e91a
          d25127875ef3bc3744fdd98b4ada71ce707c35148f5a90022e6fc',
      'R': '045625a1f2ecb9548caa6d3d11dd9202bd7051a871b88141fadf34dd2d255c2aedc22f9b9ec63
          9030f8586c42a3c7dfe5de44a289a70511b36fc48c4c703976b0f'
    },  ...
  ],  ...
}

Time consumed by the receiver to generate the range proof of the output amount v4: RangeProof(v4,ra,C4): 639.685 ms
Size of the range proof π4: 2.84766 kb
```

Figure 10. The receiver generates the range proof of the output amount v_4 based on Bulletproofs.

5.3.3. Legality Verification of Transactions

The verification of transaction legitimacy includes the verification of ownership of the input notes, the legitimacy of the transaction amounts, and the verification of the correctness of the payment note by the receiver, etc. The specific verification process and performance are described below.

1. Verification of the ownership of the input notes

The sender Bob proves ownership of the input notes $note_1$ and $note_2$ by means of the ownership keys t_1 and t_2 corresponding to the stealth addresses T_1 and T_2, and the related information and verification process are shown in Figure 11.

```
------The sender Bob's two public-private key pairs------
The scan key pair (s, S):
  s: d119f97c52439d99bf5f4cb4fe73096c75589864e6c1bf919b6cc0bc23af2951
  S: 027cb3ece8023584d57b907d7613e5cfcaa5191c637dae42476339c26c7670036b
The spend key pair (b, B):
  b: 135dfd580bb09c7b187d47f7456cae8eef2868189c16d4dc3f3c10eded4b5835
  B: 03f7f66f4e662743e88c761dea6977f0ed9dfe544f2193b9081e1a698204f22034

The stealth addresses T1 and T2 corresponding to the input notes note1 and note2:
  T1: 03570ea7e44bd138923a43ce01fb38da6d2db10c124942b7841bdf972370f8c143
  T2: 030753e10f4c7a491be012c8232fbc69ef5c60c4d15fd99a10ba77e6d7d6e9ab40
The sender Bob has ownership keys t1, t2 corresponding to the stealth addresses T1 and T2:
  t1: 76af5c8f25120e370f53691cc9bde9af081eca0c887fb9adfe767f50e974a6b4
  t2: 1f482ede93b57890e82beece702ffc820070893518bba6e0c10d0d8c0f222d27

------Verify the sender Bob's ownership of the input notes------
The stealth address T1' obtained by t1·G: 03570ea7e44bd138923a43ce01fb38da6d2db10c124942b7841bdf972370f8c143
OVerify(t1,T1): true
The stealth address T2' obtained by t2·G: 030753e10f4c7a491be012c8232fbc69ef5c60c4d15fd99a10ba77e6d7d6e9ab40
OVerify(t2,T2): true

Verify Bob's ownership of input note note1 OVerify(t1,T1): 0.402 ms
Verify Bob's ownership of input note note2 OVerify(t2,T2): 0.338 ms
```

Figure 11. Ownership verification of input notes.

According to Figure 11, it can be observed that the stealth addresses T_1' and T_2', calculated by the sender Bob using the confirmation keys t_1 and t_2 of the input notes,

successfully complete the ownership proving and verification of the input notes. By comparing T_1 with T_1', the ownership of $note_1$ by the sender Bob can be verified. Similarly, by comparing T_2 with T_2', the ownership of $note_2$ by the sender Bob can be verified. The ownership verification operation *OVerify* consumes only approximately 0.37 ms. The BTPP scheme demonstrates correctness and efficiency in verifying the ownership of input notes.

2 Verification of the legitimacy of the transaction amounts

The legitimacy of the transaction amounts consists of two aspects: (1) the balance between the input and output amounts; (2) the range of the output amounts is reasonable. The balance of the input and output amounts is verified by the transaction receiver and the verification nodes through Equation (14) and Equation (19), respectively. Whether the range of the output amounts is reasonable is verified by the range proofs π_4 and π_5 generated by both parties, and then the verification of the proofs is completed by the verification nodes, as shown in Figure 12. As can be seen in Figure 12, when the sender correctly generates the commitments to the input amounts and the change amounts, the receiver is able to correctly complete the verification of the correctness of the payment amount by calculating the commitment to the payment amount. Equations (14) and (19) can only hold when both parties correctly generate Pedersen commitments for the relevant amounts, thus completing the balanced verification of the input and output of the transaction. The time consumed for verification is approximately 16.732 ms for Equation (14) and 0.346 ms for the other case, where Equation (14) includes one *Commit* operation in the verification process, and it is known from the previous section that the *Commit* operation consumes about 16 ms. When the output amounts v_4 and v_5 both satisfy the specified reasonable range $[2^{lower}, 2^{upper}]$, the corresponding range proofs can pass the legitimacy verification by the network. The time consumption of the verification algorithm *RangeVerify* for range proofs is only 12.025 ms on average.

```
------The receiver Alice verifies the correctness of the transaction amounts------
1. Alice calculates the commitment for the amount paid by sender Bob: C'=rb·G+v4·H
 C': 049a370c68788433f33c653869070c2b97a4f60117db2cae603d903cf75f557d1a22506c1717dd9ad1d54
     e8fbf94868495177c156fd5d43161d3cbb652d16c9439
2. Alice verifies whether the equation C1+C2=C5+rb·G+v4·H holds
 C1+C2: 044d6a315cce2f81ffb60028c913f455e09dd29f5b7384283423c3a13bf7212b5fe9c842c0a073cf31fed
        7442a2a75b146a94e2c1865d8ac7cd770c4ac11bd54b6
 C5+C': 044d6a315cce2f81ffb60028c913f455e09dd29f5b7384283423c3a13bf7212b5fe9c842c0a073cf31fed
        7442a2a75b146a94e2c1865d8ac7cd770c4ac11bd54b6
C1+C2=C5+rb·G+v4·H: true
Time consumption of the receiver to verify the correctness of the transaction amounts: 16.732 ms

------Verification nodes verify the legitimacy of the transaction amounts------
1. Verify that the inputs and outputs of the transaction are balanced, i.e. whether the equation C1+C2=C5+C4+K holds
 C1+C2: 044d6a315cce2f81ffb60028c913f455e09dd29f5b7384283423c3a13bf7212b5fe9c842c0a073cf31fed
        7442a2a75b146a94e2c1865d8ac7cd770c4ac11bd54b6
 C5+C4+K: 044d6a315cce2f81ffb60028c913f455e09dd29f5b7384283423c3a13bf7212b5fe9c842c0a073cf31fed
          7442a2a75b146a94e2c1865d8ac7cd770c4ac11bd54b6
C1+C2=C5+C4+K: true
Time consumption of the verification nodes to verify that the inputs and outputs of the transaction are balanced: 0.346 ms
2. verify the correctness of the range proofs for the transaction output amounts (v4, v5)
RangeVerify(π4,C4): true
RangeVerify(π5,C5): true
Time consumed to verify the range proof π4 of the output amount v4: RangeVerify(π4,C4): 12.11 ms
Time consumed to verify the range proof π5 of the output amount v5: RangeVerify(π5,C5): 10.94 ms
```

Figure 12. Verification of the legality of transaction amounts.

3. Verification of the correctness of payment note by the recipient

The verification process of the correctness of the payment note by the receiver is shown in Figure 13. First, the receiver verifies whether the relevant note information is correct by decrypting the payment note $note_4$ and then calculates the temporary key pair (r_4', R_4') of the note $note_4$ using the shared knowledge of both parties. They further calculate the shared key c_4' and the confirmation key t_4, which can calculate the same stealth address using the confirmation key t_4. Finally, the receiver verifies the ownership of the payment note and completes the verification of the correctness of the payment note.

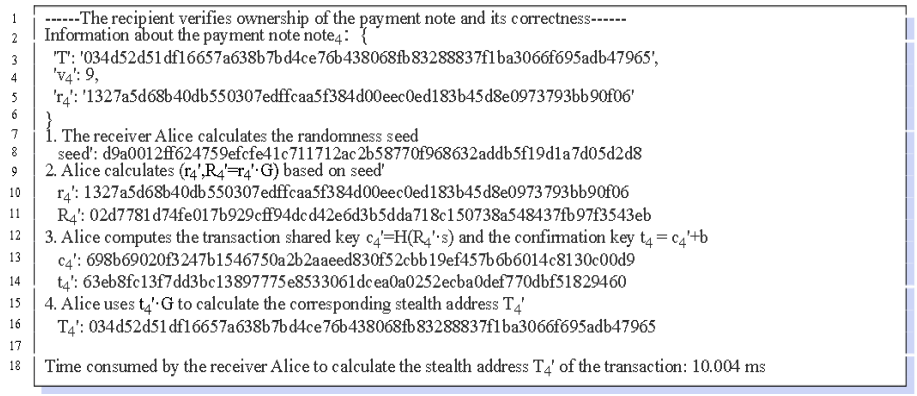


Figure 13. The receiver verifies ownership of the payment note.

The verification operations of transaction legitimacy in the BTPP scheme, including ownership verification of input notes, balance verification of transaction input and output, verification of the range proofs, the output amounts, and verification of correctness of payment notes. These operations are all performed in milliseconds and are highly efficient. At the same time, a series of verifications of transaction legitimacy can be passed only when the relevant proofs are correct. In summary, the BTPP scheme has correctness, feasibility, and efficiency.

6. Method Comparison

The previous sections provide a detailed analysis of the performance of the BTPP method and verify that the BTPP scheme has correctness, feasibility, and efficiency through relevant implementations and tests. This subsection will compare the BTPP scheme with its composition-related privacy protection methods to highlight the advantages of the BTPP scheme and its research significance.

6.1. Compare with the DKSAP Protocol

The DKSAP protocol is the privacy protection technology based on address hiding, which has been landed in many blockchain projects since its announcement. The BTPP scheme is proposed based on the DKSAP protocol, and the performance comparison results with the DKSAP protocol are shown in Table 5.

Table 5. Comparison of DKSAP protocol and BTPP method.

	The Protection of Transaction Amount	The Protection of Trading Relationship	Risk Of Leakage of Temporary Key	The Receiver Needs to Continuously Scan the Block Data	Receiver's Computation
DKSAP	no	yes	yes	yes	larger
BTPP	yes	yes	no	no	smaller

The DKSAP protocol generates the temporary address for the receiver in each transaction and uses it as the receiver's address for this transaction, obscuring the transaction relationship between the two parties. The receiver verifies itself as the authorized receiver of this transaction by calculating the confirmation key of the transaction and thus cannot link the transaction relationship between the two parties. The BTPP method combines the stealth address and the note mechanism to conceal both transaction relationships and transaction amounts. The transaction sender in the DKSAP protocol needs to generate the temporary key pair (r, R) when calculating the stealth address and send the public key R together with the transaction. The temporary public key R identifies the transaction as a stealth transaction, resulting in the loss of some key privacy information. Additionally, the receiver needs to continuously retrieve transaction data from the blockchain and perform

elliptic curve scalar multiplication operations based on R to determine whether he is the authorized receiver, which is computationally intensive and inefficient. The BTPP method uses the Diffie–Hellman key exchange protocol to generate a random seed known only to both parties when generating the temporary key pair. From this, both parties to the transaction can calculate the temporary address of this transaction by themselves, avoiding the risk of exposure of the temporary public key. At the same time, the receiver no longer needs to continuously scan the block data and repeatedly perform relevant calculations to verify whether it is the recipient of the transaction. The receiver only needs to compare the temporary address of the transaction with the calculated temporary address, thereby reducing the demand for node computing resources and improving computing efficiency. The BTPP method ensures transaction privacy while maintaining trading efficiency.

6.2. Compare with the Zerocash

Zcash is a cryptocurrency designed to provide enhanced privacy, while the BTPP scheme incorporates the UTXO-based note mechanism in Zerocash. The comparison of the BTPP scheme with the Zerocash system is shown in Table 6.

Table 6. Comparison of Zerocash and BTPP method.

	The Protection of Transaction Amount	The Protection of Trading Relationship	Trusted Setup	Relies on Trusted Third Party
Zerocash	yes	yes	yes	yes
BTPP	yes	yes	no	no

The Zerocash system improves transaction privacy using the zk-SNARKs, achieving protection for both transaction relationships and amounts. However, it relies on the zk-SNARKs for proof of note ownership and requires trusted setup involving the third party, resulting in lower transaction efficiency. Essentially, the Zerocash system is a weakly centralized system. In contrast, the BTPP method embeds the transaction's stealth address into the payment note. The mapping between the note and its corresponding stealth address is maintained on the blockchain through the Note Commitments table. This binding between the note and the stealth address enables efficient transfer of note ownership. The receiver can independently compute the confirmation key for the stealth address based on common knowledge. By leveraging the binding relationship between the stealth address and the note, efficient proof of note ownership is achieved. Additionally, the integration of the stealth address and note mechanism enhances transaction privacy. As mentioned earlier, the time consumption for relevant operations is milliseconds, demonstrating high efficiency. Compared to the Zerocash system, the advantage of the BTPP method is that it is completely decentralized, and the whole process does not require the participation of third parties.

7. Conclusions

This paper focuses on the problem of transaction privacy leakage risk in the public environment of blockchain. This paper first elaborates on the contradictory problem of the open and transparent nature of blockchain and the risk of transaction privacy leakage and analyzes and summarizes the current research status and progress, as well as the challenges and problems in the field of blockchain transaction privacy protection. To this end, this paper proposes the blockchain transaction privacy protection method that integrates the stealth address and the note mechanism, whose implementation does not depend on any third party and guarantees the privacy of transaction relationships and transaction amounts on the blockchain.

In order to verify and evaluate the feasibility, security, and efficiency of the proposed scheme in this paper, this paper concludes with a detailed analysis of the scheme in terms of both privacy and security. From the analysis results, it can be seen that the method proposed

in this article can prevent fraudulent behavior by transaction senders, transaction receivers, and verifiers, ensure security, and also ensure the privacy and integrity of transaction information. Then, this paper evaluates and analyzes the BTPP scheme by designing relevant experiments based on different experimental purposes. Through the experimental results, it can be concluded that the proposed method consumes milliseconds in key processes, and the related commitments and proofs are also very small, which is crucial for controlling transaction costs. At the same time, it also verifies that only honest transaction participants can complete transactions. In addition, the BTTP method is compared with recent related works. The BTPP method can address the shortcomings of the original stealth address scheme, such as the inability to protect the privacy of transaction amounts, privacy loss caused by temporary key leakage, and the need for the receiver to calculate repeatedly. The BTPP method can also address the issues of requiring zk-SNARKs for relevant proving and third-party trusted settings in the Zerocash privacy trading system, achieving a completely decentralized privacy trading solution.

The findings of this article bear significant implications not only for enhancing the privacy capabilities of existing blockchain systems but also for the broader development of the digital security domain. For instance, these privacy protection methods can be applied in areas such as digital identity verification, secure data exchange, and enhancing the anonymity of financial transactions. Blockchain developers should consider integrating these privacy protection mechanisms to enhance user privacy and overall system security. Additionally, this article encourages subsequent research to build upon this foundation to make these privacy protection methods more applicable to a wide range of blockchain scenarios. Further studies are also needed to assess the feasibility and effectiveness of these methods in different blockchain architectures and their resilience in the face of advanced security threats. Through these practical applications and continuous research, the research results of this article will have a lasting and positive impact on the progress of blockchain technology and the development of digital security.

The BTPP method uses the Diffie–Hellman key exchange protocol to generate the randomness seed known only to both parties when generating the temporary key pair, which requires multiple interactions between the two parties. Although the interaction information does not affect the overall security even if it is listened to by an attacker, reducing the number of interactions between the two parties and simplifying the process of the whole scheme, or even achieving zero interaction, are still directions for future work to improve the scheme in this paper further. With the development of blockchain, its network scale will also expand, resulting in the current privacy protection solution having scalability problems in larger blockchain networks. This is also one of the studies that this article will focus on in the future.

Author Contributions: Conceptualization, Z.W.; Methodology, Z.W. and Y.Z.; Software, Z.W.; Validation, Z.W.; Formal analysis, Z.W. and C.L.; Writing—original draft, Z.W.; Writing—review & editing, Z.W., J.F., Z.H., J.Z., C.L., G.Z. and H.T.; Visualization, S.M.; Supervision, J.F., Y.Z. and S.M.; Project administration, G.Z. and H.T.; Funding acquisition, G.Z. and H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Key Technologies R&D Program of Guangdong Province (No. 2020B1111370001) and the National Natural Science Foundation of China (No. 82271267).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available in the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, B.; Qi, G.; Lu, W. Recent Advances in Privacy Protection Technologies in Blockchain. In Proceedings of the 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 19–21 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 77–82. [CrossRef]
2. Shen, X.; Pei, Q.; Liu, X. Survey of block chain. *Chin. J. Netw. Inf. Secur.* **2016**, *2*, 11–20. [CrossRef]
3. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
4. Wijaya, D.A.; Liu, J.; Steinfeld, R.; Liu, D.; Yuen, T.H. Anonymity reduction attacks to monero. In Proceedings of the Information Security and Cryptology: 14th International Conference, Inscrypt 2018, Fuzhou, China, 14–17 December 2019; pp. 86–100. [CrossRef]
5. Zheng, B.; Zhu, L.; Shen, M.; Du, X.; Guizani, M. Identifying the vulnerabilities of bitcoin anonymous mechanism based on address clustering. *Sci. China Inf. Sci.* **2020**, *63*, 132101. [CrossRef]
6. He, X.; He, K.; Lin, S.; Yang, J.; Mao, H. Bitcoin address clustering method based on multiple heuristic conditions. *IET Blockchain* **2022**, *2*, 44–56. [CrossRef]
7. Long, T.; Xu, J.; Fu, L.; Wang, X. Analyzing and de-anonymizing Bitcoin networks: An IP matching method with clustering and heuristics. *China Commun.* **2022**, *19*, 263–278. [CrossRef]
8. Lin, Y.J.; Wu, P.W.; Hsu, C.H.; Tu, I.P.; Liao, S.W. An evaluation of bitcoin address classification based on transaction history summarization. In Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Republic of Korea, 14–17 May 2019; pp. 302–310. [CrossRef]
9. Zhang, H.; Wang, M.; Li, G. The development status of frontier technology of blockchain security and privacy protection. *Inf. Technol. Netw. Secur.* **2021**, *40*, 7–12. [CrossRef]
10. Li, T.; Wang, H.; He, D.; Yu, J. Blockchain-based privacy-preserving and rewarding private data sharing for IoT. *IEEE Internet Things J.* **2022**, *9*, 15138–15149. [CrossRef]
11. Xue, Z.; Wang, M.; Zhang, Q.; Zhang, Y.; Liu, P. A regulatable blockchain transaction model with privacy protection. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 1642–1652. [CrossRef]
12. Lu, N.; Chang, Y.; Shi, W.; Choo, K.-K.R. CoinLayering: An efficient coin mixing scheme for large scale bitcoin transactions. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1974–1987. [CrossRef]
13. Noether, S.; Mackenzie, A. Ring confidential transactions. *Ledger* **2016**, *1*, 1–18. [CrossRef]
14. Banerjee, A.; Clear, M.; Tewari, H. Demystifying the Role of zk-SNARKs in Zcash. In Proceedings of the IEEE Conference on Application, Information and Network Security (AINS), Kota Kinabalu, Malaysia, 17–19 November 2020; pp. 12–19. [CrossRef]
15. Deuber, D.; Schröder, D. CoinJoin in the Wild: An Empirical Analysis in Dash. In Proceedings of the Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, 4–8 October 2021; pp. 461–480. [CrossRef]
16. Ruffing, T.; Moreno-Sanchez, P.; Kate, A. Coinshuffle: Practical decentralized coin mixing for bitcoin. In Proceedings of the Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, 7–11 September 2014; pp. 345–364. [CrossRef]
17. Ziegeldorf, J.H.; Grossmann, F.; Henze, M.; Inden, N.; Wehrle, K. Coinparty: Secure multi-party mixing of bitcoins. In *Proceedings of the Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, San Antonio, YX, USA, 2–4 March 2015*; pp. 75–86. [CrossRef]
18. Lu, X.; Au, M.H.; Zhang, Z. Raptor: A practical lattice-based (linkable) ring signature. In Proceedings of the Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, 5–7 June 2019; pp. 110–130. [CrossRef]
19. Mundhe, P.; Yadav, V.K.; Singh, A.; Verma, S.; Venkatesan, S. Ring signature-based conditional privacy-preserving authentication in VANETs. *Wirel. Pers. Commun.* **2020**, *114*, 853–881. [CrossRef]
20. Yu, J.; Au, M.H.A.; Esteves-Verissimo, P. Re-thinking untraceability in the cryptonote-style blockchain. In Proceedings of the IEEE 32nd computer security foundations symposium (CSF), Hoboken, NJ, USA, 25–28 June 2019; pp. 94–9413. [CrossRef]
21. Li, Y.; Yang, G.; Susilo, W.; Yu, Y.; Au, M.H.; Liu, D. Traceable monero: Anonymous cryptocurrency with enhanced accountability. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 679–691. [CrossRef]
22. Wijaya, D.A.; Liu, J.; Steinfeld, R.; Liu, D. Monero ring attack: Recreating zero mixin transaction effect. In Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 1196–1201. [CrossRef]
23. Wang, L.; Lin, X.; Qu, L.; Ma, C. Ring selection for ring signature-based privacy protection in VANETs. In Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [CrossRef]
24. Courtois, N.T.; Mercer, R. Stealth address and key management techniques in blockchain systems. In Proceedings of the ICISSP 2017—Proceedings of the 3rd International Conference on Information Systems Security and Privacy, Porto, Portugal, 19–21 February 2017; pp. 559–566. [CrossRef]
25. Fan, J.; Wang, Z.; Luo, Y.; Bai, J.; Li, Y.; Hao, Y. A new stealth address scheme for blockchain. In Proceedings of the ACM Turing Celebration Conference-China, Chengdu, China, 17–19 May 2019; pp. 1–7. [CrossRef]

26. Fan, X. Faster dual-key stealth address for blockchain-based internet of things systems. In Proceedings of the Blockchain–ICBC 2018: 1st International Conference, Held as Part of the Services Conference Federation, SCF 2018, Seattle, WA, USA, 25–30 June 2018; pp. 127–138. [CrossRef]
27. Sasson, E.B.; Chiesa, A.; Garman, C.; Green, M.; Miers, I.; Tromer, E.; Virza, M. Zerocash: Decentralized anonymous payments from bitcoin. In Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 18–21 May 2014; pp. 459–474. [CrossRef]
28. Bünz, B.; Bootle, J.; Boneh, D.; Poelstra, A.; Wuille, P.; Maxwell, G. Bulletproofs: Short proofs for confidential transactions and more. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 315–334. [CrossRef]
29. Li, Y.; Chen, Y.; Li, T.; Ren, X. A regulatable data privacy protection scheme for energy transactions based on consortium blockchain. *Secur. Commun. Netw.* **2021**, *2021*, 4840253. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.