TumbleBit++: A Comprehensive Privacy Protocol Providing Anonymity and Amount-Invisibility

Yi Liu[1,2], Zhen Liu[1(✉)], Yu Long[1(✉)], Zhiqiang Liu[1(✉)], Dawu Gu[1(✉)],
Fei Huan[1(✉)], and Yanxue Jia[1]

[1] School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai, China
{1780790324,liuzhen,longyu,ilu_zq,dwgu,huanfei,jiayanxue}@sjtu.edu.cn
[2] Shanghai Viewsource Information Science and Technology Co., Ltd.,
Shanghai, China

Abstract. Since the advent of bitcoin, the privacy of bitcoin has become a hot issue. Many coin mixing protocols guarantee the anonymity and unlinkability of the payer and payee of a transaction. However, due to the publicity of blockchain, the confidentiality of transaction amounts has not been provided. Everyone has the chance to get the amount of a transaction, which poses a challenge to the privacy of users.

To overcome the problem, we propose an improved mixing protocol based on TumbleBit, which is named TumbleBit++. TumbleBit++ combines confidential transactions with centralized untrusted anonymous payment hub, and achieves the protection of transaction amounts without undermining the anonymity of TumbleBit. TumbleBit++ allows multiple payers to trade in different transaction amounts, and Tumbler, as an untrusted third party, does not know the exact amount of each transaction and the flow of funds between the payer and payee of one transaction.

Keywords: TumbleBit · Confidential transactions · Bitcoin

1 Introduction

The most important aspect of bitcoin's privacy is the hiding of transaction information, such as transaction address and transaction amount. In order to achieve the anonymity of bitcoin, the technology of coin mixing [1,2,9] has been adopted to separate the relationship between the input and output addresses. TumbleBit [2], as a centralized mixing protocol, uses an untrusted third party, Tumbler, to offer mixing service with transaction flow invisible to the third party. However, an attacker can still get information about the flow of transactions by the increasing or decreasing amount of money [2]. Confidential transactions [3] realized the protection of transaction amounts on blockchain, but with no concern of anonymity.

© Springer Nature Switzerland AG 2019
R. Steinfeld and T. H. Yuen (Eds.): ProvSec 2019, LNCS 11821, pp. 339–346, 2019.
https://doi.org/10.1007/978-3-030-31919-9_21

Several currencies have contributed to the protection of amounts. Monero is a cryptocurrency based on the CryptoNote protocol [4], which provides unlinkability and untraceability by ring signature, stealth address and Pedersen commitment [11]. However, the ring signature requires space and verification overheads on blockchain and makes it difficult for clients to distinguish the spent transaction outputs for pruning [5]. Zerocoin [6] is a zero-knowledge-proof-based currency. Users can mint bitcoin into zerocoin with hidden addresses for trading. Zerocash [7] uses the non-interactive zero-knowledge proof technology which is zk-SNARK to achieve privacy and anonymity and to support arbitrary denomination transactions. Because of the need for complex mathematical calculation, the cost of Zerocoin and Zerocash is high. Besides, the dependence on trusted setup and the non-falsifiable cryptographic assumptions [8] makes it have low acceptance. Valueshuffle [10] based on Coinshuffle++ [9], aims at hiding the amounts of transactions by combining confidential transactions and stealth address. In Valueshuffle, DiceMix protocol is run to mix output triples, which consist of output addresses, value commitments and range proofs. However, since the range proof is quite large, Valueshuffle splits the output triple into chunks to mix and recombines the messages after mixing. This arrangement demands high computation costs and more redundance. Inheriting the features of Coinshuffle++, the scheme can not resist DoS attacks and Sybil attacks.

Our Contribution: TumbleBit++. In this paper, we present TumbleBit++, a complete privacy protection protocol that combines confidential transactions with centralized coin mixing protocol, TumbleBit. TumbleBit++ provides the invisibility of amounts on the basis of anonymity, which makes the amounts and flow of transactions invisible to not only users but also the third party.

TumbleBit++ modifies the 2-of-2 escrow smart contract of TumbleBit, and allows multiple bitcoins to be packaged in one transaction without revealing the value. Verification, blinding and zero-knowledge proof steps are applied to prevent theft and provide anonymity.

2 Preliminaries

TumbleBit. TumbleBit [2] is a centralized coin mixing scheme with an untrusted anonymous payment hub, which is compatible with bitcoin. TumbleBit uses RSA encryption algorithm [13] and ECDSA [14] to ensure the anonymity and unforgeability of transactions. TumbleBit uses off-chain puzzle payments to replace on-chain payments, which also improves the efficiency of coin mixing.

Puzzle-promise protocol and RSA-puzzle-solver protocol are two important sub-protocols of TumbleBit, which turn bitcoin payments into off-chain puzzle payments. Puzzle-promise protocol generates puzzle pairs for off-chain payments between Tumbler and the payee. RSA-puzzle-solver protocol provides the solution to the specific puzzle through interactions between the payer and Tumbler.

The anonymity of TumbleBit is achieved by blinding. The payee B uses the blind factor r, which is only visible to B, to blind the puzzle z. So that the third party T cannot link the blinded puzzle $\bar{z}$ from the payer A to the original puzzle z, which splits the relationship between A and B. In TumbleBit, the blinding of puzzle z is based on RSA encryption process. For a blind factor r, the blinding of puzzle z is $\bar{z} = r^e z \bmod N$.

Pedersen Commitment. Pedersen commitment [11] is a scheme which allows the user to commit to a secret value without revealing it. Besides, the value can be revealed later and the user can prove the revealed value to be correct [15]. Pedersen commitment is applied in confidential transactions to hide the amounts of transactions.

Pedersen commitment is a homomorphic commitment, which means that the commitment of sum equals the sum of commitments. For example, the commitment of value x_1 is $com_1 = com(x_1, r_1)$, while the commitment of value x_2 is $com_2 = com(x_2, r_2)$. r_1 and r_2 are random values for encryption. The homomorphic property makes it that the commitment of value $(x_1 + x_2)$ is $com(x_1 + x_2, r_1 + r_2) = com_1 \oplus com_2 = com(x_1, r_1) \oplus com(x_2, r_2)$. Therefore, homomorphic commitment makes it convenient and effective to verify the balance of transaction amounts.

3 TumbleBit++

3.1 System Entities and Overview

The system entities of TumbleBit++ are similar to that of TumbleBit. The payer is Alice A, and the payee is Bob B. Tumbler T is an untrusted third party.

In TumbleBit++, the amounts of all on-chain transactions, which are the four transactions in Fig. 1, are hidden in commitments. For example, the transaction

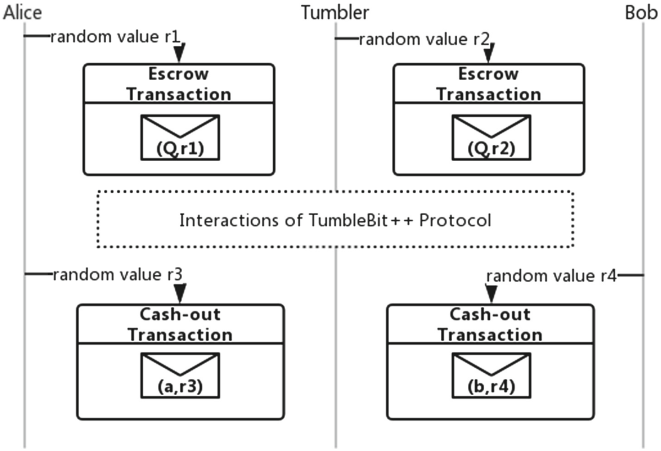

Fig. 1. System entities of TumbleBit++, in which the values in four transactions are hidden. In theory, values a and b should be equal.

amount in the escrow transaction between A and T, which is the upper limit Q of one round, is committed with the random value $r1$ of A; The transaction amount in the escrow transaction between T and B, which is also the upper limit Q, is committed with the random value $r2$ of T. In cash-out transactions which are used to activate the escrow transactions, the amounts a and b are committed with random values of A and B separately.

Before the interactions, A escrows Q BTCs on chain and T also escrows Q BTCs on chain. Through the interactions of TumbleBit++ protocol, a BTCs in escrow transaction flow from A to T, and b BTCs flow from T to B. In one payment, a and b should be equal. Meanwhile, T does not know the relationship between A and B and the transaction amounts which are a and b. After one round of payments, which includes multiple transactions, unspent bitcoins in escrow transactions will be withdrawn.

The interactions of TumbleBit++ protocol involve interactions between T and B through puzzle-promise protocol, interactions between A and T through RSA-puzzle-solver protocol, and interactions between A and B for parameter values. The details are described in Sect. 3.2.

In the enhanced 2-of-2 escrow smart contract of TumbleBit++, we stipulate that the transaction value in $T_{fulfill}$ [2] is a commitment of the actual amount of transaction between A and B, and the amount that $T_{fulfill}$ can take from T_{escrow} is exactly the amount committed in $T_{fulfill}$, rather than the fixed 1 BTC in TumbleBit.

3.2 Concrete Protocol

As Fig. 2 shows, TumbleBit++ has three phases.

Escrow Phase. In this phase, there are three steps.

- On-chain escrow transactions.
  A and T escrow Q BTCs in commitments in escrow transactions separately on chain. The detailed output addresses are described in TumbleBit.
- Puzzle-promise protocol.
  If B aims to get b BTCs from T, B and T generate a puzzle pair (c, z) through puzzle-promise protocol. Different from TumbleBit, the signature σ of T to the cash-out transaction is committed in $comm(\sigma, r_c)$ and r_c is a random value of T. The commitment is encrypted by puzzle solution ε.
- Blinding.
  Besides the blinding of puzzle z, the commitment of value b, which is c_B is also blinded by blinding factor r_B of B. However, the blinding of puzzle z is based on RSA encryption algorithm which can be found in Sect. 2, while the blinding of c_B is based as follows.

For a commitment $c = comm(v, r) = v \cdot H + r \cdot G$, the random blind factor r_1 is selected and the commitment is blinded to $\bar{c} = c + r_1 \cdot G$.

After blinding, Δr_1 is calculated by B at the same time. $\bar{z}$, $\bar{c_B}$ and Δr_1 are prepared to send to A for verification and puzzle solution.

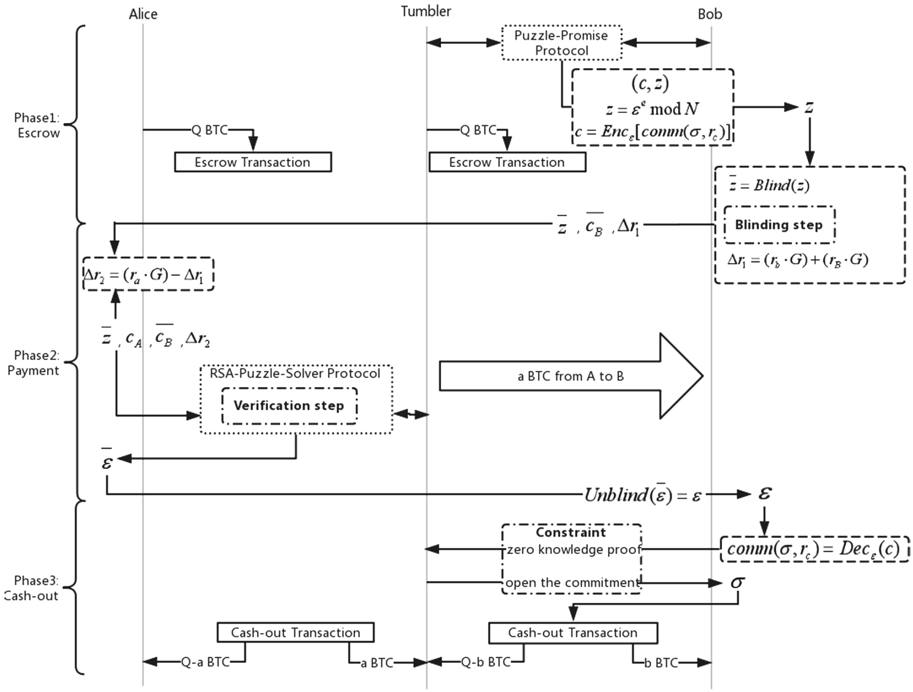

Fig. 2. TumbleBit++ protocol.

Payment Phase. Compared to TumbleBit, the most important step in this phase is verification. Since multiple bitcoins are packaged into one transaction, it is necessary for third-party Tumbler to ensure the balance of revenue and expenditure of one transaction. In commitments, the value of a is committed in $c_A = comm(a, r_a)$, and the value of b is committed in $c_B = comm(b, r_b)$. The additive homomorphism of Pedersen commitment [11] ensures verifiability.

The phase has two steps.

- Off-chain puzzle payment.
  A makes a payment to B for a blinded puzzle $\bar{z}$, $\bar{c_B}$ and Δr_1. For the later verification, A calculates Δr_2 for prepare.

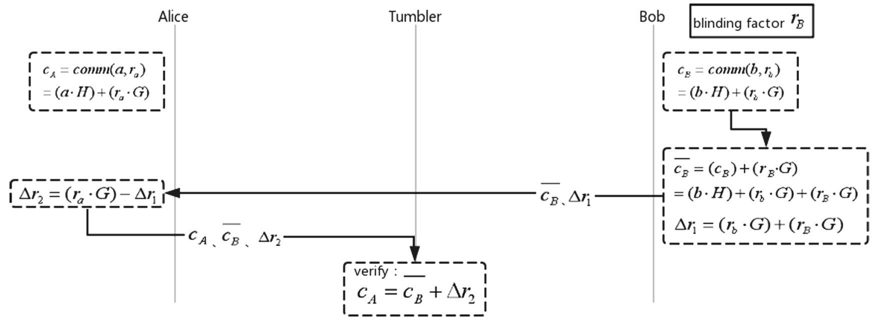
Alice
Tumbler
Bob
blinding factor r_B
$c_A = comm(a, r_a) = (a \cdot H) + (r_a \cdot G)$
$c_B = comm(b, r_b) = (b \cdot H) + (r_b \cdot G)$
$\overline{c_B} = (c_B) + (r_B \cdot G) = (b \cdot H) + (r_b \cdot G) + (r_B \cdot G)$
$\Delta r_1 = (r_b \cdot G) + (r_B \cdot G)$
$\overline{c_B}$、Δr_1
$\Delta r_2 = (r_a \cdot G) - \Delta r_1$
c_A、$\overline{c_B}$、Δr_2
verify : $c_A = \overline{c_B} + \Delta r_2$

Fig. 3. Verification of TumbleBit++.

- RSA-puzzle-solver protocol and verification.
  In the RSA-puzzle-solver protocol, it is necessary for T to verify $c_A = \bar{c_B} + \varDelta r_2$. The correctness of the verification can be verified in Fig. 3.

  If the verification is proved, A obtains the solution $\bar{\varepsilon}$ of $\bar{z}$ through RSA-puzzle-solver protocol with T and sends it to B.

Cash-Out Phase. In order to prevent A and B from cheating T with wrong verification information, constraint using zero-knowledge proof is added in this phase.

Three steps are involved in the cash-out phase.

- Unblinding.
  After receiving $\bar{\varepsilon}$ from A, B unblinds it by the blind factor r_B and gets the solution ε. As mentioned in the escrow phase, in puzzle-promise protocol of TumbleBit++, the signature σ of $T_{cash(T,B)}$ from T, are protected by commitment. The information B gets by decrypting after getting ε is $comm(\sigma, r_c)$ rather than the signature σ.
- Constraint.
  In order to open the commitment to get σ, B is supposed to provide T with a zero-knowledge proof, which can proof that the blinded value $\bar{c_B}$ of c_B is included in a set of $\bar{c_B}$ maintained by T and T doesn't know exactly which $\bar{c_B}$ is the value. Obviously, T knows a set of $\bar{c_B}$ from all previous interactions with A.
  In this scheme, if A generates $\bar{c_B}$ and $\varDelta r_2$ at will, the set of $\bar{c_B}$ will not include the corresponding commitment of c_B, so that B cannot get σ, which has no benefit to A.

  After the verification of zero-knowledge proof, T opens the commitment to B and B obtains the signature σ to complete the cash-out transaction.
- On-chain cash-out transactions.
  Finally, B claims bitcoins from T's escrow transaction by on-chain cash-out transaction. Unspent bitcoins in escrow transactions will be withdrawn.

Since TumbleBit++ is an enhancement of TumbleBit protocol, some details such as cut and choose scheme and smart contract can be found in TumbleBit [2].

4 Security Analysis

TumbleBit++ has several security properties. Due to the limitation of space, we conclude the properties briefly.

- Anonymity.
  Inherited from TumbleBit, anonymity is provided by blinding scheme, which includes blinding of puzzle and blinding of commitments.
- Amounts invisibility.
  Invisibility of amounts is realized by Pedersen commitments of confidential transactions. In addition, TumbleBit++ can mix transactions with different amounts, which means that it is more efficient and flexible than the traditional fixed-value transactions in TumbleBit.
- Tumbler untrustworthiness.
  Inherited from TumbleBit, Tumbler is unable to know the amount and flow information in the transaction, which is realized by commitments and blinding.
- Theft prevention.
  The verification step and constraint based on zero-knowledge proof prevent theft from payers and payees effectively. Besides, the authority of generating commitments avoids Tumbler modifying values of commitments.
- DoS resistance.
  Inherited from TumbleBit, since the independence between users of coin mixing transactions, even if some network resources are occupied, it does not affect the process of mixing.
- Sybils resistance.
  Sybil attack [12] is a form of attack in peer-to-peer networks. The property is also inherited from TumbleBit.

5 Conclusion

In the privacy protection of bitcoin, many mixing schemes can provide anonymity reasonably, but there is no systematic mechanism for the amounts concealment. Confidential transaction is a scheme that hides the amounts of transactions with Pedersen commitments.

In this paper, based on TumbleBit protocol and CT scheme, TumbleBit++ protects the amounts, and inherits the anonymity and the third party's untrustworthiness of TumbleBit, which is realized by blinding on commitments, verification, and constraint based on zero-knowledge proof. In summary, we get a comprehensive privacy protocol, TumbleBit++, which provides anonymity and amount-invisibility.

Acknowledgement. The authors are supported by the National Natural Science Foundation of China (Grant No. 61672347, 61572318, 61672339, 61872142), the National Cryptography Development Fund (No. MMJJ20170111) and Minhang Technology Innovation Program for SMEs, a finance business platform based on blockchain technology (2018MH110).

References

1. Ruffling, T., Moreno-Sanchez, P., Kate, A.: Coinshuffle: practical decentralized coin mixing for bitcoin. In: Kutylowski, M., Vaudya, J. (eds.) ESORICS 2014. LNCS, vol. 8713, pp. 345–364. Springer, Cham (2014)
2. Heilman, E., AlShenibr, L., Baldimtsi, F., Scafuro, A., Goldberg, S.: TumbleBit: an untrusted Bitcoin-compatible anonymous payment hub. In: NDSS 2017 (2017)
3. Maxwell, G.: Confidential transactions (2015). https://people.xiph.org/~greg/confidential_values.txt
4. Noether, S.: Review of CryptoNote white paper. https://downloads.getmonero.org/whitepaper_review.pdf
5. OmegaStarScream: Bitcoin Core & pruning mode. Bitcoin Forum. https://bitcointalk.org/index.php?topic=1599458.0
6. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: anonymous distributed e-cash from Bitcoin. In: S&P 2013 (2013)
7. Ben-Sasson, E., et al.: Zerocash: decentralized anonymous payments from Bitcoin. In: S&P 2014 (2014)
8. Gentry, C., Wiches, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: STOC 2011 (2011)
9. Ruffling, T., Moreno-Sanchez, P., Kate, A.: P2P mixing and unlinkable Bitcoin transactions. In: NDSS 2017 (2017)
10. Ruffing, T., Moreno-Sanchez, P.: ValueShuffle: mixing confidential transactions for comprehensive transaction privacy in bitcoin. In: Brenner, M., et al. (eds.) FC 2017. LNCS, vol. 10323, pp. 133–154. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70278-0_8
11. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
12. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45748-8_24
13. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
14. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Secur. **1**, 36–63 (2001)
15. Damgård, I.: Commitment schemes and zero-knowledge protocols. In: Damgård, I.B. (ed.) EEF School 1998. LNCS, vol. 1561, pp. 63–86. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48969-X_3