В данной работе предложена и реализована система тендеров, которая удовлетворяет критериям безопасности, открытости и конфиденциальности. Вопрос доверия решён с помощью технологии блокчейн, а сокрытие приватной информации — с помощью криптографического протокола неинтерактивного доказательства знания с нулевым разглашением zk-SNARK [2]. Система основана на платформе Ethereum. Вся ключевая информация о тендерах сохраняется в блокчейне, а проверка правил и отслеживание выполнения условий участниками реализованы в виде кода смарт-контрактов.

Для реализации алгоритма сокрытия информации о заявках в Ethereum C++ клиент добавлен отдельный модуль *tenderzkp*. Он построен на базе протокола zk-SNARK с предобработкой для NP-полного языка системы ограничений ранга 1. Протокол использует эллиптическую кривую Баррето — Наерига. Реализация криптографической схемы предоставлена библиотекой libsnark [3].

В модуле *tenderzkp* реализованы функции создания и верификации доказательства о корректности заявки. Доказательство строится на основе ограничений на приватные и открытые входные данные заявки, выраженных с помощью базовых схем библиотеки libsnark.

Для работы с добавленной криптографической схемой в Ethereum C++ клиент созданы новые предкомпилированные контракты с адресами 0x00...09 и 0x00...0a и разработана Solidity-библиотека, которая инкапсулирует низкоуровневое взаимодействие с предкомпилированными контрактами и предоставляет интерфейс для работы с ними в виде Solidity-функций. Чтобы добавить возможность вызывать методы разработанной криптографической схемы из сторонних приложений, расширен JSON-RPC API Ethereum клиента.

Предложенный метод может быть использован не только для тендеров, но и в других системах, где есть необходимость скрывать часть информации в открытой блокчейн-сети. Он расширяет область применения технологии блокчейн в промышленных программных комплексах.

ЛИТЕРАТУРА

1. *Hardwick F. S., Akram R. N., and Markantonakis K.* Fair and transparent blockchain based tendering framework — A step towards open governance // IEEE Intern. Conf. TrustCom/BigDataSE, New York, USA, 2018. P. 1342–1347.
2. *Ben-Sasson E., Chiesa A., Genkin D., et al.* SNARKs for C: Verifying program executions succinctly and in zero knowledge // CRYPTO'2013. LNCS. 2013. V. 8043. P. 90–108.
3. `https://github.com/scipr-lab/libsnark` — libsnark: a C++ library for zkSNARK proofs.

UDC 004.056 DOI 10.17223/2226308X/13/27

VALIDATION-FREE OFFCHAIN TRANSACTIONS WITH UNLINKABLE DOUBLE SPEND DETECTION

S. N. Kyazhin, K. A. Klimenko

The so-called layer-two protocols are a class of blockchain scaling solutions. They allow to minimize onchain traffic, and therefore make state transitions (payments, for example) faster and more suitable for everyday use, while still preventing double spend attacks. Unfortunately, these solutions also have some downsides and tradeoffs (channel capacity, route availability, operator availability, etc.). In this work we study the possibility of simplifying and improving existing protocols for offchain transactions and describe a scheme that, without transaction validation, allows to detect a double spender and not trace other transactions. This scheme is based on the anonymous transferable e-cash system. We use an offchain analogue of the UTXO model, therefore there are offchain transactions for issue, transfer and redeem of a so-called note, containing a number that can be used as a secret key to make the corresponding token transfer transaction onchain.

Ключевые слова: *blockchain, offchain, unlinkability, double spend detection.*

1. Introduction

Layer-two protocols are the trend of blockchain scalability solutions right now. Such protocols allow users to make offchain transactions. In [1] the authors summarize and systematize existing solutions: payment/state channels and commit-chains (or hubs). It would be interesting to create a system in which some users transfer tokens to others, while they can dynamically join the two-layer solution (free establishment property [1]). Next, any user who received tokens offchain can receive them onchain. There are commit-chains with unlinkability and anonimity properties (e.g. TumbleBit [2] and Bolt [3]), which are suitable for this problem.

However, existing solutions also have the following tradeoffs. For example:

— unlike the regular blockchain security model, when using a layer-two solution, the user may need to monitor his funds from time to time;

— there may be some constraints or prerequisites for using such a solution (channel capacity, route availability for payment/state channels, operator availability for commit-chains, etc.);

— the user may need to use additional complex software that stores sensitive information (the history of transactions, including the so-called "breach remedy transactions", or other data required to create a proof of fraud and prevent loss of funds).

Consider the following case: there is no transaction validation, however, there is an operator that checks for double spend when the current owner wants to receive his tokens in the blockchain. Obviously, in this case, the operator will not be able to prevent double spend, but can only detect it.

This case is very similar to an e-cash system. Moreover, for example, Bolt uses an offline anonymous e-cash scheme [4]. But this protocol does not have the transferability property (for a coin there can be only one transfer transaction). The paper [5] describes a modified protocol that provides transferability (the main modification is related to the ability for the receiver to spend the received coin later).

Our current research aims to implement the approaches proposed in [4, 5] to create a simpler offchain transaction scheme that allows to detect a double spender without validation and linking transactions.

Let $\mathbb{G}$ be an additive group of prime order q, $s \in \mathbb{Z}_q$ be a number that can be used as a secret key to make a transaction for transfer of some tokens onchain. All other blockchain details are beyond the scope of this paper.

Suppose we have an offchain analogue of the UTXO model. For each s there is a so-called note. Therefore, an offchain token transfer transaction means a transaction for transfer of a note with s value.

Let G, H be generators in $\mathbb{G}$, $x_i \in \mathbb{Z}_q$ ($P_i = x_i G$) be the private (public) key of the i-th participant of the offchain transaction scheme, $i = 1, \ldots, n$. We formulate the problem as follows — to create a scheme based on [5] that allows:

— to transfer of a note between users with these keys without transaction validation;

— to reveal the public key of the user who transfered the note more than once when the current owner of the note wants to make the corresponding transaction onchain and not link the transactions for the note transfer with the corresponding public keys of other users.

2. Description of the Scheme

Let $\mathcal{H}$ be a cryptographic hash function to $\mathbb{Z}_q$, $\pi(\{a_1,\ldots,a_n\},\{b_1,\ldots,b_n\} : f_1(a_1,\ldots,a_n, b_1,\ldots,b_n)=c_1,\ldots,f_n(a_1,\ldots,a_n,b_1,\ldots,b_n)=c_n)$ be a zero knowledge proof of knowledge of such private $a_1,\ldots,a_n$ and public $b_1,\ldots,b_n$ (in general from different sets) that $f_1(a_1,\ldots,a_n,b_1,\ldots,b_n)=c_1,\ldots,f_n(a_1,\ldots,a_n,b_1,\ldots,b_n)=c_n$, where $f_1,\ldots,f_n$ are the corresponding functions, $c_1,\ldots,c_n$ are constants.

2.1. Note Issue

To issue a note, the owner of the tokens in the blockchain (he knows s):

— generates some message msg (transaction description) and computes the hash function $m_0=\mathcal{H}(msg)$;

— computes $r_0=(x+m_0)^{-1}G$, where x is the owner's private key;

— computes $\tilde{r}_0=\mathcal{H}(r_0,m_0)$;

— computes $T_0=xG+\tilde{r}_0(x+s+1)^{-1}H$;

— computes a proof

$$\pi_0=\pi(\{x\},\{s,T_0,r_0,m_0\} : T_0=xG+\tilde{r}_0(x+s+1)^{-1}H, r_0=(x+m_0)^{-1}G);$$

— creates the note $(s,\, V_0)$, where $V_0=(T_0,\pi_0,r_0,m_0)$.

2.2. Note Transfer

Assume that user A owns a note $(s,\, V)$, where $V=(V_0,\ldots,V_l)$, $V_j=(T_j,\pi_j,r_j,m_j)$, $j=0,\ldots,l$, that he legitimately received from another user. If A legitimately received the note $(s,\, V)$, it is necessary that $r_l=(x_A+m_l)^{-1}G$, where x_A is the private key of A.

The following steps describe the interactive procedure for transfer of the note $(s,\, V)$ from user A to user B.

First, the Receiver (B):

— generates some message msg (transaction description) and computes the hash function $m_{l+1}=\mathcal{H}(msg)$;

— computes $r_{l+1}=(x_B+m_{l+1})^{-1}G$, where x_B is the receiver's private key;

— sends m_{l+1}, r_{l+1} to the Sender.

Next, the Sender (A):

— computes $\tilde{r}_{l+1}=\mathcal{H}(r_{l+1},m_{l+1})$;

— computes $h_{l+1}=\mathcal{H}(s,T_0,\ldots,T_l)$;

— computes $T_{l+1}=x_AG+\tilde{r}_{l+1}(x_A+s+h_{l+1})^{-1}H$;

— computes a proof

$$\pi_{l+1}=\pi(\{x_A\},\{s,T_{l+1},r_{l+1},m_{l+1}\}:T_{l+1}=x_AG+\tilde{r}_{l+1}(x_A+s+h_{l+1})^{-1}H, r_l=(x_A+m_l)^{-1}G);$$

— creates and sends the note (s,V), where $V=(V_0,\ldots,V_{l+1})$, $V_{l+1}=(T_{l+1},\pi_{l+1},r_{l+1},m_{l+1})$, to the Receiver.

The Receiver can optionally verify the proof π_{l+1}.

2.3. Note Redeem

When the current owner of the note wants to make the corresponding transaction in the blockchain, he sends the note (s, V), $V = (V_0, \dots, V_t)$, $V_j = (T_j, \pi_j, r_j, m_j)$, to the Operator.

The Operator verifies that:

— the proof π_j is valid for all $j = 0, \dots, t$;
— the note with s has not been redeemed.

If the note with s has been redeemed, there was a double spend.

2.4. Double Spender Detection

A double spend is equivalent to the fact that the Operator received notes with the same s and different $V = (V_0, \dots, V_k, \dots, V_t)$ and $V' = (V_0, \dots, V'_k, \dots, V'_{t'})$.

The Operator:

— looks for the minimal k that $V_k = (T_k, \pi_k, r_k, m_k) \neq V'_k = (T'_k, \pi'_k, r'_k, m'_k)$;
— computes $\tilde{r}_k = \mathcal{H}(r_k, m_k)$ and $\tilde{r}'_k = \mathcal{H}(r'_k, m'_k)$;
— computes the public key of the double spender:

$$P = (\tilde{r}'_k - \tilde{r}_k)^{-1}(\tilde{r}'_k T_k - \tilde{r}_k T'_k).$$

3. Conclusion

This paper is dedicated to the research of the possibility of constructing a protocol for offchain transactions that, without transaction validation, allows to detect a double spender and not trace other transactions. We describe a possible scheme based on the transferable anonymous e-cash system proposed in [5]. In future papers, we would like to reformulate the security properties from [5] and provide the proofs.

REFERENCES

1. *Gudgeon L., Moreno-Sanchez P., Roos S., et al.* SoK: Off The Chain Transactions. Cryptology ePrint Archive: Report 2019/360.
2. *Heilman E., Alshenibr L., Baldimtsi F., et al.* TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub. Cryptology ePrint Archive: Report 2016/575.
3. *Green M. and Miers I.* Bolt: Anonymous Payment Channels for Decentralized Currencies. Cryptology ePrint Archive: Report 2016/701.
4. *Camenisch J., Hohenberger S., and Lysyanskaya A.* Compact E-Cash // EUROCRYPT 2005. LNCS. 2005. V. 3494. P. 302–321.
5. *Canard S., Gouget A., and Traore J.* Improvement of efficiency in (unconditional) anonymous transferable E-Cash // Financial Cryptography and Data Security. LNCS. 2008. V. 5143. P. 202–214.