Journal of Southeast University (English Edition) Vol. 38, No. 1, pp. 36 - 41 Mar. 2022 ISSN 1003—7985

Cryptanalysis on an organization-friendly blockchain system

Zhang Ying Jiang Rui

(School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: To verify that an organization-friendly blockchain system may suffer from forgery and collusion attacks, forgery and collusion attacks were theoretically carried out according to the phase sequence of an organization-friendly blockchain system. Then, the organization-friendly blockchain system was improved and based on the phase sequence forgery and collusion attacks were conducted. The results show that the attacker can obtain illegal transaction data from forgery and collusion attacks on the organization-friendly blockchain system. However, for the improved organization-friendly blockchain, the attacker's forgery and collusion attacks cannot be completed. Therefore, the organization-friendly blockchain system may be subject to forgery and collusion attacks, but the improved organization-friendly blockchain system can prevent such attacks.

Key words: blockchain; identity privacy; transaction supervision; forgery attack; collusion attack

DOI: 10.3969/j.issn.1003 - 7985.2022.01.006

In 2008, SatoshiNakamoto proposed a peer-to-peer electronic cash system, which is called Bitcoin[1]. Consequently, considerable research on its underlying technology, which is called blockchain, has been conducted worldwide. However, blockchain technologies[1-2] may encounter the privacy protection problem.

To solve the privacy protection problem in blockchains, many schemes[3-8] have been proposed. The Mixcoin mechanism[3] was proposed to hide the transaction process among transaction users. However, the centralized Mixcoin scheme may result in the transactional centralization problem. A ring signature was applied to the Monero cryptocurrency[4]. However, in this anonymous technology, the ring signature operation relies on other users' public keys. Ring confidential transactions[5] improve the Monero cryptocurrency[4] by introducing a Pedersen commitment on the basis of the ring signature. In 2013, Zerocoin[6], a distributed e-cash system, was proposed to apply cryptographic techniques to unlink transactions from the payment's origin without adding trusted parties. However, Zerocoin has limited functionalities. To overcome this problem, Zerocash[7] was proposed to hide the transaction amount and the origin or destinations of the payment anonymously. However, Zerocash has weak efficiency. Bolt[8] was proposed by constructing three anonymous payment channels to ensure a secure, instantaneous, and private payment. However, all of the above schemes have inappropriate or excessive privacy protection and may result in the transaction supervision problem. Hence, no one can determine the relevant information of transaction users, and illegal crimes, such as fraud, money laundering, and drug smuggling, are prone to occur.

To solve the conflict between privacy protection and transaction supervision in blockchains, a number of schemes[9-12] have been proposed. Auditable Zerocoin[9] was proposed to allow designated auditors to extract link information from Zerocoin transactions. A decentralized anonymous payment scheme with accountability and privacy[10] was proposed to address regulatory concerns by adding the privacy-preserving policy-enforcement mechanism. The confidential and auditable payment scheme[11] was proposed to keep the transaction confidential. The organization-friendly blockchain system[12] was proposed to realize the balance between privacy protection and transaction supervision. However, the scheme may suffer from forgery and collusion attacks, from which an attacker can easily obtain the transaction amount illegally.

In this paper, the organization-friendly blockchain system[12] is briefly reviewed, the forgery and collusion attacks that the system[12] may suffer from are described, and countermeasures to remedy such attacks are presented.

1 Organization-Friendly Blockchain System

The organization-friendly blockchain system[12] has nine main phases: system setup Setup, key generation KeyGen, organization issue Issue, user registration Join, address generation AddrGen, transaction generation TransGen, transaction verification TransVer, transaction relay TransRelay, and user identity tracing UserTrace.

Received 2021-08-14, **Revised** 2021-11-10.

Biographies: Zhang Ying(1997—), female, graduate; Jiang Rui(corresponding author), male, doctor, associate professor, R. Jiang@seu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 61372103), the Natural Science Foundation of Jiangsu Province (No. BK20201265), the Key Laboratory of Computer Network Technology of Jiangsu Province.

Citation: Zhang Ying, Jiang Rui. Cryptanalysis on an organization-friendly blockchain system[J]. Journal of Southeast University (English Edition), 2022, 38(1): 36 - 41. DOI: 10.3969/j.issn.1003 - 7985.2022.01.006.

In the Setup phase, the system runs Setup to initialize the system. The system takes as input a security parameter 1^k and outputs the system public parameter P. G_1 and G_2 are bilinear groups: $e: G_1 \times G_2 \to G_T$; $|G_1| = |G_2| = p$, where p is the prime, g_1 is a generator of G_1, g_2 is a generator of G_2, $g_1 \leftarrow \psi(g_2)$, and hash function $H: \{0, 1\}^* \to Z_p^*$. The system public parameter is $P = (G_1, G_2, g_1, g_2, p, H)$.

In the KeyGen phase, the registration node RegMan, organization node OrgMan, and member user node MebUser generate their respective key pairs.

1) RegMan randomly chooses $x, y \leftarrow Z_p^*$ and computes $u = g_2^x$ and $v = g_2^y$. The public key of RegMan is $r_{pk} = (u, v)$, and the private key is $r_{sk} = (x, y)$.

2) The transaction sending organization SedOrg randomly chooses $h_1 \in G_1 \setminus \{1_{G_1}\}$, $x_1, y_1 \leftarrow Z_p^*$, and $u_1, v_1 \in G_1$, such that $u_1^{x_1} = v_1^{y_1} = h_1$. SedOrg randomly chooses $r_1 \leftarrow Z_p^*$, and computes $w_1 = g_2^{r_1}$. Then, it randomly chooses $\nu_1 \leftarrow Z_p^*$. SedOrg also randomly chooses two large primes p_1 and q_1, and computes $n_1 = p_1 q_1$, such that gcd $(n_1, (p_1 - 1)(q_1 - 1)) = 1$. SedOrg computes $\lambda_1 = \text{lcm}(p_1 - 1, q_1 - 1)$. SedOrg randomly chooses $\tilde{g}_1 \in Z_{n^2}^*$, such that gcd $(L(\tilde{g}_1^{\lambda_1} \bmod n_1^2), n_1) = 1$, where $L(x) = x - 1/n_1$. The public key of SedOrg is $o_{pks} = (h_1, u_1, v_1, w_1, \nu_1, n_1, \tilde{g}_1)$, and the private key is $o_{sks} = (x_1, y_1, r_1, \lambda_1)$. In the same way, the transaction receiving organization RecOrg can obtain its public key $o_{pkr} = (h_2, u_2, v_2, w_2, \nu_2, n_2, \tilde{g}_2)$ and private key $o_{skr} = (x_2, y_2, r_2, \lambda_2)$.

3) The transaction sender SedUser randomly chooses $h_3 \in G_1 \setminus \{1_{G_1}\}$, $x_3 \leftarrow Z_p^*$, and computes $u_3 = h_3^{x_3}$. The public key of SedUser is $u_{pks} = (h_3, u_3)$, and the private key is $u_{sks} = x_3$. In the same way, the transaction receiver RecUser can obtain its public key $u_{pkr} = (h_4, u_4)$ and private key $u_{skr} = x_4$.

In the Issue phase, OrgMan and RegMan interactively generate an organization certificate C_o. SedOrg and RecOrg submit respective organization public keys o_{pks} and o_{pkr} and other identifying information to RegMan for registration. Once the identity verification for the organization is passed, RegMan sends certificates $C_{so} = (\sigma_0 = g^{1/(x+\nu_1+yr_1)}, r$ and $C_{ro} = (\sigma_1 = g^{1/(x+\nu_2+y\tilde{r})}, \tilde{r})$ to SedOrg and RecOrg, respectively. Once SedOrg and RecOrg have verified their respective organization certificates, RegMan binds the organization certificate to the organization's public key and places it in the certificate library C_{lo}.

In the Join phase, MebUser and OrgMan interactively generate a sub-certificate C_u. SedUser and RecUser submit respective public keys u_{pks} u_{pkr} and and other identifying information to SedOrg and RecOrg for registration. Once the identity verification for the user is passed, SedOrg and RecOrg send sub-certificates $C_{su} = (A = (g_1/u_3)^{1/(r_1+a)}, a)$ and $C_{ru} = (A' = (g_1/u_4)^{1/(r_2+a_1)}, a_1)$ and organization certificates C_{so} and C_{ro} to SedUser and RecUser, respectively. Once SedUser and RecUser have verified respective sub-certificates and organization certificates, OrgMan binds the sub-certificate to the user public key and places it in the sub-certificate library C_{lu}.

In the AddrGen phase, OrgMan and MebUser generate their respective wallet addresses. SedOrg and RecOrg compute their respective wallet addresses $a_{so} = H(o_{pks})$, and $a_{ro} = H(o_{pkr})$. SedUser and RecUser compute their respective wallet addresses $a_{su} = H(u_{pks})$ and $a_{ru} = H(u_{pkr})$.

In the TransGen phase, SedUser performs an operation to generate a transaction and broadcast it to the blockchain network.

SedUser encrypts the transaction amount for each input address m_i, $i \in \{1, 2, \ldots, l\}$ and RecUser's addresses a_{ru} with RecOrg's public key o_{pkr}, where l is the number of input addresses. SedUser applies the Paillier algorithm[13] for encryption. The corresponding ciphertexts are $c_i = \tilde{g}_2^{m_i} r_i^{n_2} \bmod n_2^2$ and n_2^2.

SedUser proves that every transaction amount m_i is greater than 0. SedUser generates a commitment to his account amount m and computes $c_{l+1} = \tilde{g}_2^m (r_1 r_2 \ldots r_l)^{n_2} \bmod n_2^2$.

SedUser signs the transaction information with its private key u_{sks}. The signature is $\sigma = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, c, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$, where $T_1 = u_1^\alpha$, $T_2 = v_1^\beta$, and $T_3 = Ah_1^{\alpha+\beta}$, where u_1, v_1, h_1 are the values in SedOrg's public key $o_{pks} = (h_1, u_1, v_1, w_1, \nu_1, n_1, \tilde{g}_1)$, α, and β are random numbers, and A is a value in SedUser's sub-certificate $C_{su} = (A = (g_1/u_3)^{1/(r_1+a)}, a)$.

SedUser attaches C_{so} as the transaction certificate to generate a transaction $T = (a_{so}, a_{ro}, \sigma, h_3, o_{pks}, C_{so})$. Then, SedUser broadcasts the transaction T to the blockchain network.

In the TransVer phase, the miner node Miner verifies the validity of the transaction $T = (a_{so}, a_{ro}, \sigma, h_3, o_{pks}, C_{so})$ according to the following equations:

$$c \overset{?}{=} H(c_0 \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1 \| T_2 \| T_3 \| \tilde{R}_1 \| \tilde{R}_2 \| \tilde{R}_3 \| \tilde{R}_4 \| \tilde{R}_5) \quad (1)$$

$$c_1 c_2 \ldots c_l \overset{?}{=} c_{l+1} \quad (2)$$

$$e(\sigma_0, u g_2^{\nu_1} v^r) \overset{?}{=} e(g_1, g_2) \quad (3)$$

Once Eqs. (1), (2), and (3) hold, Miner broadcasts the transaction $T = (a_{so}, a_{ro}, \sigma, h_3, o_{pks}, C_{so})$ and generates a block B to complete the transaction based on the blockchain trading system.

In the TransRelay phase, RecOrg receives the transaction $T = (a_{so}, a_{ro}, \sigma, h_3, o_{pks}, C_{so})$ broadcasted by Miner and decrypts the wallet address $a_{ru} = \dfrac{L(c_0^{\lambda_2} \bmod n_2^2)}{L(\tilde{g}_2^{\lambda_2} \bmod n_2^2)} \bmod n_2$ and transaction amount $m_i = \dfrac{L(c_i^{\lambda_2} \bmod n_2^2)}{L(\tilde{g}_2^{\lambda_2} \bmod n_2^2)} \bmod n_2$, i

$\in \{1, 2, \ldots, l\}$. Then, RecOrg relays the transaction amount to RecUser.

In the UserTrace phase, the system tracks the identity of the malicious transaction user when an abnormal transaction occurs. The whole process is divided into external tracing and internal tracing.

In external tracking, RegMan receives the transaction sent by Miner and tracks the public key o_{pks} of SedOrg according to the organization certificate C_{so}.

In internal tracking. SedOrg receives the transaction sent by RegMan and decrypts the user's sub-certificate C_{su} with its private key $o_{\text{sks}} = (x_1, y_1, \lambda_1, r_1)$. After a given σ, SedOrg computes $A = (T_3/(T_1^{x_1}T_2^{y_1}))$ and obtains A of C_{su}. The internal malicious user's public key u_{sks} is further tracked according to the sub-certificate C_{su}.

2 Forgery Attack and Collusion Attack

2.1 Forgery attack

In this section, the forgery attack is described in detail as follows. The forgery attack has two phases: the preparation phase and the implementation phase.

At the forgery attack preparation phase, the attacker A_0 registers with the legitimate OrgMan.

In the KeyGen phase of the scheme[12], the attacker A_0, as MebUser, generates key pairs. A_0 randomly chooses $h_A \in G_1 \setminus \{1_{G_1}\}$, $x_A \leftarrow Z_p^*$, and computes $u_A = h_A^{x_A}$. The public key of A_0 is $u_{\text{pkA}} = (h_A, u_A)$, and the private key is $u_{\text{skA}} = x_A$.

In the Join phase of the scheme[12], A_0 registers with the legitimate OrgMan. As an example for registration to OrgMan, A_0 submits its public key u_{pkA} and other identifying information to OrgMan and easily passes the identity verification. OrgMan will randomly choose $a' \in Z_p^*$, compute $A_A = (g_1/u_A)^{1/(r_1+a')}$, and generate the sub-certificate $C_A = (A_A, a')$. OrgMan sends the sub-certificate C_A and organization certificate C_{o} to A_0. In the same way, A_0 can obtain the sub-certificate and organization certificate from other legitimate OrgMan.

In the AddrGen phase of the scheme[12], the attacker A_0 computes the wallet address $a_A = H(u_{\text{pkA}})$.

Having finished the forgery attack preparation phase, the attacker A_0 can start the forgery attack implementation phase.

Firstly, the attacker A_0 immediately intercepts the transaction when MebUser broadcasts a transaction $T = (a_{\text{so}}, a_{\text{ro}}, \sigma, h_3, o_{\text{pks}}, C_{\text{so}})$ at the TransGen phase of the scheme[12]. A_0 modifies the original transaction T as $T' = (o_{\text{so}}, o_{\text{ro}}, \sigma', h_A, o_{\text{pks}}, C_{\text{so}})$, and broadcasts T' to the blockchain network. A_0 modifies $\sigma = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, c, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$ as $\sigma' = (T_1', T_2', T_3', c_0', c_1, \ldots, c_l, c_{l+1}, c', s_\alpha', s_\beta', s_a', s_{x_A}', s_{\delta_1}', s_{\delta_2}')$, and changes h_3 to h_A as follows.

To modify σ as σ', A_0 randomly chooses $r_0' \in Z_{n^2}^*$, and computes $c_0' = \tilde{g}_2^{a_A} r_0'^{n_2} \bmod n_2^2$. A_0 randomly chooses $\alpha', \beta' \leftarrow Z_p^*$, and computes $T_1' = u_1^{\alpha'}$, $T_2' = v_1^{\beta'}$, $T_3' = A_A h_1^{\alpha'+\beta'}$, where A_A is the sub-certificate issued by SedOrg to A_0 during the forgery attack preparation phase, and computes $\delta_1' = a'\alpha'$, $\delta_2' = a'\beta'$. A_0 randomly chooses r_α', r_β', r_a', r_{δ_1}', r_{δ_2}', $r_{x_A}' \leftarrow Z_p^*$ and computes $R_1' = u_1^{r_\alpha'}$, $R_2' = v_1^{r_\beta'}$, $R_3' = e(T_3', g_2)^{r_a'} e(h_1, w_1)^{-r_\alpha'-r_\beta'} e(h_1, g_2)^{-r_{\delta_1}'-r_{\delta_2}'} e(h_A, g_2)^{r_{x_A}'}$, $R_4' = T_1'^{r_\alpha'} u_1^{-r_{\delta_1}'}$, $R_5' = T_2'^{r_\alpha'} v_1^{-r_{\delta_2}'}$. Then, A_0 computes $c' = H(c_0' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1' \| T_2' \| T_3' \| R_1' \| R_2' \| R_3' \| R_4' \| R_5')$, and $s_\alpha' = r_\alpha' + c'\alpha'$, $s_\beta' = r_\beta' + c'\beta'$, $s_a' = r_a' + c'a'$, $s_{x_A}' = r_{x_A}' + c'x_A$, $s_{\delta_1}' = r_{\delta_1}' + c'\delta_1'$, $s_{\delta_2}' = r_{\delta_2}' + c'\delta_2'$. The modified signature is $\sigma' = (T_1', T_2', T_3', c_0', c_1, \ldots, c_l, c_{l+1}, c', s_\alpha', s_\beta', s_a', s_{x_A}', s_{\delta_1}', s_{\delta_2}')$. A_0 changes h_3 to h_A. The modified transaction $T' = (a_{\text{so}}, a_{\text{ro}}, h_A, o_{\text{pks}}, C_{\text{so}})$.

Secondly, in the TransVer phase of the scheme[12], Miner verifies the validity of the transaction $T' = (a_{\text{so}}, a_{\text{ro}}, \sigma', h_A, o_{\text{pks}}, C_{\text{so}})$. If Eqs. (1), (2), and (3) will hold, then the modified transaction T' can be verified.

1) Miner verifies Eq. (1), which is changed as $c' \stackrel{?}{=} H(c_0' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1' \| T_2' \| T_3' \| \tilde{R}_1 \| \tilde{R}_2 \| \tilde{R}_3 \| \tilde{R}_4 \| \tilde{R}_5)$. Miner calculates $\tilde{R}_1 = u_1^{s_\alpha'} T_1'^{-c'}$, $\tilde{R}_2 = v_1^{s_\beta'} T_2'^{-c'}$, $\tilde{R}_3 = e(T_3', g_2)^{s_a'} e(h_1, w_1)^{-s_\alpha'-s_\beta'} e(h_1, g_2)^{-s_{\delta_1}'-s_{\delta_2}'} e(h_A, g_2)^{s_{x_A}'} (e(T_3', w_1)/e(g_1, g_2))^{c'}$, $\tilde{R}_4 = T_1'^{s_\alpha'} u_1^{-s_{\delta_1}'}$, $\tilde{R}_5 = T_2'^{s_\alpha'} v_1^{-s_{\delta_2}'}$. Then, it verifies whether the equation $c' \stackrel{?}{=} H(c_0' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1' \| T_2' \| T_3' \| \tilde{R}_1 \| \tilde{R}_2 \| \tilde{R}_3 \| \tilde{R}_4 \| \tilde{R}_5)$ holds or not. Apparently, Eq. (1) will hold. The left side of Eq. (1) is c', which is included in σ' at the modified transaction T' and is equal to $H(c_0' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1' \| T_2' \| T_3' \| R_1' \| R_2' \| R_3' \| R_4' \| R_5')$. The right side of Eq. (1) is $H(c_0' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1' \| T_2' \| T_3' \| \tilde{R}_1 \| \tilde{R}_2 \| \tilde{R}_3 \| \tilde{R}_4 \| \tilde{R}_5)$. Here,

$$\tilde{R}_1 = u_1^{s_\alpha'} T_1'^{-c'} = u_1^{r_\alpha'+c'\alpha'} u_1^{-c'\alpha'} = u_1^{r_\alpha'} = R_1'$$

$$\tilde{R}_2 = v_1^{s_\beta'} T_2'^{-c'} = v_1^{r_\beta'+c'\beta'} v_1^{-c'\beta'} = v_1^{r_\beta'} = R_2'$$

$$\tilde{R}_4 = T_1'^{s_\alpha'} u_1^{-s_{\delta_1}'} = T_1'^{r_\alpha'+c'\alpha'} u_1^{-r_{\delta_1}'-c'a'\alpha'} = T_1'^{r_\alpha'} u_1^{-r_{\delta_1}'} = R_4'$$

$$\tilde{R}_5 = T_2'^{s_\alpha'} v_1^{-s_{\delta_2}'} = T_2'^{r_\alpha'+c'\alpha'} v_1^{-r_{\delta_2}'-c'a'\beta'} = T_2'^{r_\alpha'} v_1^{-r_{\delta_2}'} = R_5'$$

$$\begin{aligned}\tilde{R}_3 &= e(T_3', g_2)^{s_a'} e(h_1, w_1)^{-s_\alpha'-s_\beta'} e(h_1, g_2)^{-s_{\delta_1}'-s_{\delta_2}'} e(h_A, g_2)^{s_{x_A}'} \cdot (e(T_3', w_1)/e(g_1, g_2))^{c'} = \\ &e(h_1, w_1)^{-r_\alpha'-r_\beta'-c'(\alpha'+\beta')} e(h_1, g_2)^{-r_{\delta_1}'-r_{\delta_2}'-c'a'(\alpha'+\beta')} \cdot \\ &e(T_3', g_2)^{r_a'+c'a'} e(h_A, g_2)^{r_{x_A}'+c'x_A} (e(T_3', w_1)/e(g_1, g_2))^{c'} = \\ &R_3' e(T_3', g_2)^{c'a'} e(h_1, g_2^{r_1})^{-c'(\alpha'+\beta')} e(h_1, g_2)^{-c'a'(\alpha'+\beta')} \cdot \\ &e(h_A, g_2)^{c'x_A} e(T_3', g_2^{r_1})^{c'} e(g_1, g_2)^{-c'} = \\ &R_3' e((g_1/u_A)^{1(r_1+a')} h_1^{\beta'+\alpha'}, g_2)^{c'(r_1+a')} e(h_1, g_2)^{-c'(r_1+a')(\alpha'+\beta')} \cdot \\ &e(h_A, g_2)^{c'x_A} e(g_1, g_2)^{-c'} = \\ &R_3' e((g_1/h_A^{x_A})^{1/(r_1+a')}, g_2)^{c'(r_1+a')} e(h^{\beta'+a'}, g_2)^{c'(r_1+a')} \cdot \\ &e(h_1, g_2)^{-c'(r_1+a')(\alpha'+\beta')} e(h_A, g_2)^{c'x_A} e(g_1, g_2)^{-c'} = \\ &R_3' e(g_1/h_A^{x_A}, g_2)^{c'} e(h_A, g_2)^{c'x_A} e(g_1, g_2)^{-c'} = R_3'\end{aligned}$$

2) The two sides of Eq. (1) are equal. Therefore, Eq. (1) can hold.

3) The original signature is $\sigma = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, c, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$, and the modified signa-

ture is $\sigma' = (T_1', T_2', T_3', c_0', c_1, \ldots, c_l, c_{l+1}, c', s_\alpha', s_\beta', s_a', s_{x_A}', s_{\delta_1}', s_{\delta_2}')$. The transaction amount c_i, $i \in \{1, 2, \ldots, l, l+1\}$ is not modified, so the equation $c_1 c_2 \ldots c_l \overset{?}{=} c_{l+1}$ can hold. The verification process is $c_1 c_2 \ldots c_l = \tilde{g}_2^{m_1 m_2 \ldots m_l} (r_1 r_2 \ldots r_l)^{n_2} \bmod n_2^2 = \tilde{g}_2^m (r_1 r_2 \ldots r_l)^{n_2} \bmod n_2^2 = c_{l+1}$

4) The original transaction is $T = (a_{so}, a_{ro}, \sigma, h_3, o_{pks}, C_{so})$, and the modified transaction is $T' = (a_{so}, a_{ro}, \sigma', h_A, o_{pks}, C_{so})$. SedOrg's certificate C_{so} has not been modified, so the equation $e(\sigma_0, u g_2^{\nu_1} v^r) \overset{?}{=} e(g_1, g_2)$ holds. The verification process is $e(\sigma_0, u g_2^{\nu_1} v^r) = e(g_1^{1/(x+\nu_1+yr)}, g_2^{x+\nu_1+yr}) = e(g_1, g_2)$.

5) Having checked the three equations, Miner broadcasts the transaction T' and generates a new block B' to complete the transaction based on the blockchain trading system.

Finally, in the TransRelay phase of the scheme[12], RecOrg receives the transaction $T' = (a_{so}, a_{ro}, \sigma', h_A, o_{pks}, C_{so})$ broadcasted by Miner and decrypts the wallet address $a_A = \dfrac{L(c_0'^{\lambda_2} \bmod n_2^2)}{L(\tilde{g}_2^{\lambda_2} \bmod n_2^2)} \bmod n_2$ and transaction amount $m_i = \dfrac{L(c_i^{\lambda_2} \bmod n_2^2)}{L(\tilde{g}_2^{\lambda_2} \bmod n_2^2)} \bmod n_2$, $i \in \{1, 2, \ldots, l\}$. Then, RecOrg relays the transaction amount to the wallet address a_A of the attacker node A_0 instead of the real legal RecUser.

2.2 Collusion attack

In this study, the collusion attack is regarded as an attack where some nodes in the blockchain conspire to exchange effective information and modify transaction content to illegally obtain other legal nodes' transaction amounts.

Specifically, the collusion attack is launched as the malicious node A_2 sends its own address to another malicious node A_1, where A_2 is a MebUser belonging to the same organization as the original RecUser and A_1 is a MebUser belonging to the same organization as the original SedUser. Then, A_1 modifies the original transaction information and changes the receiving address of the original transaction to A_2's address. Finally, A_2 can illegally obtain the transaction amount of the original SedUser.

The collusion attack has two phases: the preparation phase and the implementation phase.

At the collusion attack preparation phase, attackers A_1 and A_2 register with the legitimate OrgMan, and A_2 may send its wallet address to A_1.

In the KeyGen phase of the scheme[12], attackers A_1 and A_2, as MebUser, generate their respective key pairs. A_1 randomly chooses $h_{A_1} \in G_1 \backslash \{1_{G_1}\}$, $x_{A_1} \leftarrow Z_p^*$, and computes $u_{A_1} = h_{A_1}^{x_{A_1}}$. The public key of A_1 is $u_{pA_1} = (h_{A_1}, u_{A_1})$, and the private key is $u_{sA_1} = x_{A_1}$. In the same way, the attacker A_2 can generate its public key $u_{pA_2} = (h_{A_2}, u_{A_2})$ and private key $u_{sA_2} = x_{A_2}$.

In the Join phase of the scheme[12], A_1 and A_2 register with the legitimate OrgMan, respectively. As an example for registration to OrgMan, A_1 can get sub-certificates $C_{A_1} = (A_{A_1} = (g_1 / u_{A_1})^{1/(r_1 + a^*)}, a^*)$.

In the AddrGen phase of the scheme[12], A_1 and A_2 compute their respective wallet addresses $a_{A_1} = H(u_{pA_1})$ and $a_{A_2} = H(u_{pA_2})$. Then, A_2 sends its wallet address a_{A_2} to A_1.

After the collusion attack preparation phase, attacker A_1 can start the collusion attack implementation phase.

Firstly, attacker A_1 immediately intercepts the transaction when the member user node broadcasts a transaction $T = (a_{so}, a_{ro}, \sigma, h_3, o_{pks}, C_{so})$ at the TransGen phase of the scheme[12]. Then, A_1 modifies the original transaction T as $T'' = (a_{so}, a_{ro}, \sigma'', h_{A_1}, o_{pks}, C_{so})$, and broadcasts T'' to the blockchain network. A_1 modifies $\sigma = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, c, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$ as $\sigma'' = (T_1'', T_2'', T_3'', c_0'', c_1, \ldots, c_l, c_{l+1}, c'', s_\alpha'', s_\beta'', s_a'', s_{x_{A_1}}'', s_{\delta_1}'', s_{\delta_2}'')$, where $c_0'' = \tilde{g}_2^{a_{A_1}} r_{20}''^n \bmod n_2^2$. The rest of the modification process is the same as that at the forgery attack.

Secondly, in the TransVer phase of the scheme[12], Miner verifies the validity of the transaction $T'' = (a_{so}, a_{ro}, \sigma'', h_{A_1}, o_{pks}, C_{so})$. If Eqs. (1), (2), and (3) will hold, then the verification process is the same as that at the forgery attack. Therefore, the modified transaction T'' can be verified.

Finally, in the TransRelay phase of the scheme[12], RecOrg receives the transaction T'' broadcasted by Miner and decrypts the ciphertexts c_0'', c_i, $i \in \{1, 2, \ldots, l\}$ with its private key $o_{skr} = (x_2, y_2, r_2, \lambda_2)$ to obtain the transaction receiver's wallet address a_{A_2} and transaction amount m_i. Then, RecOrg relays the transaction amount to attacker A_2.

3 Counter measures

3.1 Improvement

In this section, the improvement of the scheme[12] is proposed. The TransGen and TransVer phases of the scheme[12] are modified, and the details are presented as follows:

At the TransGen phase of the scheme[12], the original transaction $T = (a_{so}, a_{ro}, \sigma, h_3, o_{pks}, C_{so})$ is modified as $\bar{T} = (a_{in}, a_{out}, \bar{\sigma}, s_1, o_{pks}, o_{pkr}, C_{so})$. The original transaction input and output addresses a_{so} and a_{ro}, SedUser's public key h_3, and the signature $\sigma = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, c, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$ are modified, and RecOrg's public key o_{pkr} is added.

1) The transaction input address $a_{so} = H(o_{pks})$ is modified as $a_{in} = H(o_{pks} \| T_3)$. Randomly choose $\alpha, \beta \leftarrow Z_p^*$, and compute $T_3 = A h_1^{\alpha+\beta}$, where A is the value in SedUser's sub-certificate $C_{su} = (A = (g_1 / u_3)^{1/(r_1+a)}, a)$ and h_1 is the value in SedOrg's public key $o_{pks} = (h_1, u_1, v_1, w_1, v_1, n_1, \tilde{g}_1)$.

2) The transaction output address $a_{ro} = H(o_{pkr})$ is modified as $a_{out} = H(o_{pkr} \| T_{3r})$. Randomly choose $\alpha_r, \beta_r \leftarrow Z_p^*$, and compute $T_{3r} = A' h_2^{\alpha_r + \beta_r}$, where A' is the value in RecUser's sub-certificate $C_{ru} = (A' = (g_1/u_4)^{1/(r_2+a_1)}, a_1)$ and h_2 is the value in RecOrg's public key $o_{pkr} = (h_2, u_2, v_2, w_2, \nu_2, n_2, \tilde{g}_2)$.

3) h_3 is modified as $s_1 = e(h_3, g_2)$, and to modify σ as $\bar{\sigma} = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, \bar{c}, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$, the $R_3 = e(T_3, g_2)^{r_a} e(h_1, w_1)^{(-r_\alpha - r_\beta)} e(h_1, g_2)^{(-r_{\delta_1} - r_{\delta_2})} e(h_3, g_2)^{r_{x_3}}$ is modified as $\bar{R}_3 = e(T_3, g_2)^{r_a a_{in}} e(h_1, w_1)^{(-r_\alpha - r_\beta) a_{in}} e(h_1, g_2)^{(-r_{\delta_1} - r_{\delta_2}) a_{in}} s_1^{r_{x_3} a_{in}}$. Then, $c = H(c_0 \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1 \| T_2 \| T_3 \| R_1 \| R_2 \| R_3 \| R_4 \| R_5)$ is modified as $\bar{c} = H(c_0 \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1 \| T_2 \| T_3 \| R_1 \| R_2 \| \bar{R}_3 \| R_4 \| R_5)$.

4) RecOrg's public key o_{pkr} is added to the transaction T. Finally, the modified transaction is $\bar{T} = (a_{in}, a_{out}, \bar{\sigma}, s_1, o_{pks}, o_{pkr}, C_{so})$.

At the TransVer phase of the scheme[12], the verification equation $c \overset{?}{=} H(c_0 \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1 \| T_2 \| T_3 \| \tilde{R}_1 \| \tilde{R}_2 \| \tilde{R}_3 \| \tilde{R}_4 \| \tilde{R}_5)$ is modified as $\bar{c} \overset{?}{=} H(c_0 \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1 \| T_2 \| T_3 \| \tilde{R}_1 \| \tilde{R}_2 \| \tilde{\tilde{R}}_3 \| \tilde{R}_4 \| \tilde{R}_5)$, where $\tilde{\tilde{R}}_3 = e(T_3, g_2)^{s_a a_{in}} e(h_1, w_1)^{(-s_\alpha - s_\beta) a_{in}} e(h_1, g_2)^{(-s_{\delta_1} - s_{\delta_2}) a_{in}} s_1^{s_{x_3} a_{in}} (e(T_3, w_1)/e(g_1, g_2))^{c a_{in}}$, and $a_m = H(o_{pks} \| T_3)$.

3.2 Forgery attack resistance

The improvement of the system[12] can resist forgery attacks. An attacker cannot successfully conduct a forgery attack. The detailed description is as follows.

After the forgery attack preparation phase, the attacker A_0 may start the forgery attack implementation phase.

First, attacker A_0 immediately intercepts the transaction when the member user node broadcasts a transaction $\bar{T} = (a_{in}, a_{out}, \bar{\sigma}, s_1, o_{pks}, o_{pkr}, C_{so})$ at the TransGen phase of the improved scheme. A_0 modifies the original transaction $\bar{T}$ as $\bar{T}' = (a_{in}, a_{out}, \bar{\sigma}', s_1', o_{pks}, o_{pkr}, C_{so})$, and broadcasts $\bar{T}'$ to the blockchain network. A_0 modifies $s_1 = e(h_3, g_2)$ as $s_1' = e(h_A, g_2)$, where h_A is the value in attacker A_0's public key $u_{pkA} = (u_A, h_A)$, and modifies $\bar{\sigma} = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, \bar{c}, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$ to $\bar{\sigma}' = (T_1', T_2', T_3', c_0', c_1, \ldots, c_l, c_{l+1}, \bar{c}', s_\alpha', s_\beta', s_a', s_{x_3}', s_{\delta_1}', s_{\delta_2}')$, where $\bar{c}' = H(c_0' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1' \| T_2' \| T_3' \| R_1' \| R_2' \| \bar{R}_3' \| R_4' \| R_5')$, $\bar{R}_3' = e(T_3', g_2)^{r_a' a_{in}} e(h_1, w_1)^{(-r_\alpha' - r_\beta') a_{in}} e(h_2, g_2)^{(-r_{\delta_1}' - r_{\delta_2}') a_{in}} s_1'^{r_{xA}' a_{in}}$.

Then, in the TransVer phase of the improved scheme, Miner may verify the validity of the transaction $\bar{T}'$ according to Eqs. (1), (2), and (3). Here,

$$\bar{c}' \overset{?}{=} H(c_0' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1' \| T_2' \| T_3' \| \tilde{R}_1 \| \tilde{R}_2 \| \tilde{\tilde{R}}_3 \| \tilde{R}_4 \| \tilde{R}_5)$$

where

$$\bar{c}' = H(c_0' \| c_1 \| \ldots \| c_l \| T_1' \| T_2' \| T_3' \| R_1' \| R_2' \| \bar{R}_3' \| R_4' \| R_5')$$

$$\begin{aligned}
\tilde{\bar{R}}_3 &= e(T_3', g_2)^{s_a' a_{in}} e(h_1, w_1)^{(-s_\alpha' - s_\beta') a_{in}} e(h_1, g_2)^{(-s_{\delta_1}' - s_{\delta_2}') a_{in}} \cdot \\
&\quad s_1'^{s_{x_3}' a_{in}} (e(T_3', w_1)/e(g_1, g_2))^{c' a_{in}} = \\
&e(T_3', g_2)^{(r_a' + c' a') a_{in}} e(h_1, w_1)^{((-r_\alpha' - r_\beta') - c'(\alpha' + \beta')) a_{in}} s_1'^{(r_{x_3}' + c' x_A') a_{in}} \cdot \\
&e(h_1, g_2)^{((-r_{\delta_1}' - r_{\delta_2}') - c' a'(\alpha' + \beta')) a_{in}} (e(T_3', w_1)/e(g_1, g_2))^{c' a_{in}} = \\
&\bar{R}_3' e(T_3', g_2)^{c' a' a_{in}} e(h_1, g_2^{r_1})^{-c'(\alpha' + \beta') a_{in}} e(h_1, g_2)^{-c' a'(\alpha' + \beta') a_{in}} \cdot \\
&e(h_A, g_2)^{c' x_A a_{in}} e(T_3', g_2^{r_1})^{c' a_{in}} e(g_1, g_2)^{-c' a_{in}} = \\
&\bar{R}_3' e((g_1/u_A)^{1/(r_1 + a')} h_1^{\beta' + \alpha'}, g_2)^{c'(r_1 a_{in} + a' a_{in})} e(h_A, g_2)^{c' x_A a_{in}} \cdot \\
&e(h_1, g_2)^{-c'(r_1 + a')(\alpha' + \beta') a_{in}} e(g_1, g_2)^{-c' a_{in}} = \\
&\bar{R}_3' e((g_1/h_A^{x_A})^{1/(r_1 + a')}, g_2)^{c'(r_1 a_{in} + a' a_{in})} e(h_1^{\beta' + \alpha'}, g_2)^{c'(r_1 a_{in} + a' a_{in})} \cdot \\
&e(h_1, g_2)^{-c'(r_1 + a')(\alpha' + \beta') a_{in}} e(h_A, g_2)^{c' x_A a_{in}} \cdot \\
&e(g_1, g_2)^{-c' a_{in}} = \bar{R}_3' e(g_1, g_2)^{c'(r_1\ addr + a' a_{in})/(r_1 + a') - c' a_{in}} \cdot \\
&e(h_A, g_2)^{-c' x_A (r_1 a_{in} + a' a_{in})/(r_1 + a') + c' x_A a_{in}} \cdot \\
&e(h_1, g_2)^{c'(\alpha' + \beta')((r_1 a_{in} + a' a_{in}) - (r_1 + a') a_{in})} \neq \bar{R}_3'
\end{aligned}$$

Eq. (1) does not hold, and the Miner may send the transaction $\bar{T}'$ to RegMan for user identity tracing.

3.3 Collusion attack resistance

The improvement of the system[12] can resist collusion attacks. Attackers A_1 and A_2 cannot successfully launch collusion attacks. After the collusion attack preparation phase, attacker A_1 may start the collusion attack implementation phase.

Attacker A_1 immediately intercepts the transaction when the member user node broadcasts a transaction $\bar{T} = (a_{in}, a_{out}, \bar{\sigma}, s_1, o_{pks}, o_{pkr}, C_{so})$ at the TransGen phase of the improved scheme. Then, A_1 modifies the original transaction $\bar{T}$ as $\bar{T}'' = (a_{in}, a_{out}, \bar{\sigma}'', s_1'', o_{pks}, o_{pkr}, C_{so})$, and broadcasts $\bar{T}''$ to the blockchain network. A_1 modifies $s_1 = e(h_3, g_2)$ as $s_1'' = e(h_{A_1}, g_2)$, where h_{A_1} is the value in attacker A_1's public key $u_{pA_1} = (u_{A_1}, h_{A_1})$, and modifies $\bar{\sigma} = (T_1, T_2, T_3, c_0, c_1, \ldots, c_l, c_{l+1}, \bar{c}, s_\alpha, s_\beta, s_a, s_{x_3}, s_{\delta_1}, s_{\delta_2})$ as $\bar{\sigma}'' = (T_1'', T_2'', T_3'', c_0'', c_1, \ldots, c_l, c_{l+1}, \bar{c}'', s_\alpha'', s_\beta'', s_a'', s_{x_3}'', s_{\delta_1}'', s_{\delta_2}'')$, where $\bar{c}'' = H(c_0'' \| c_1 \| \ldots \| c_l \| c_{l+1} \| T_1'' \| T_2'' \| T_3'' \| R_1'' \| R_2'' \| \bar{R}_3'' \| R_4'' \| R_5'')$, $\bar{R}_3'' = e(T_3'', g_2)^{r_a'' a_{in}} e(h_1, w_1)^{(-r_\alpha'' - r_\beta'') a_{in}} e(h_1, g_2)^{(-r_{\delta_1}'' - r'^{*}_{\delta_2}) a_{in}} s_1^{*\, r_{xA}^{*} a_{in}}$.

Then, in the TransVer phase of the improved scheme, Miner may verify the validity of the transaction $\bar{T}''$ according to Eqs. (1), (2), and (3). The verification process is the same as that at the forgery attack. Hence, Eq. (1) does not hold. Miner may send the transaction $\bar{T}''$ to RegMan for user identity tracing.

4 Conclusions

1) In the organization-friendly blockchain system, attacker A_0 can obtain the transaction amount without being detected, which means the forgery attack succeeds.

2) In the organization-friendly blockchain system, attacker A_2 can obtain the transaction amount without being detected, which means the collusion attack succeeds.

3) In the improved organization-friendly blockchain

system, forgery and collusion attacks can be prevented.

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system [EB/OL]. (2008) [2021-01-20]. http://www.bitcoin.org/bitcoin.pdf.

[2] Buterin V. Ethereum: A next generation smart contract and decentralized application platform [EB/OL]. (2013) [2021-01-10]. http://ethereum.org/en/whitepaper.

[3] Bonneau J, Narayanan A, Miller A, et al. Mixcoin: Anonymity for bitcoin with accountable mixes [C]// *Financial Cryptography and Data Security*. Christ Church, Barbados, 2014: 486 - 504. DOI: 10.1007/978-3-662-45472-5_31.

[4] Van Saberhagen N. Cryptonote v2.0 [EB/OL]. (2012) [2021-01-10]. http://cryptonote.org/whitepaper.pdf.

[5] Noether S, Mackenzie A. Ring confidential transactions [J]. *Ledger*, 2016, **1**: 1 - 18. DOI: 10.5195/LEDGER.2016.34.

[6] Miers I, Garman C, Green M, et al. Zerocoin: Anonymous distributed e-cash from bitcoin [C]// *IEEE Symposium on Security and Privacy*. Berkeley, CA, USA, 2013: 397 - 411. DOI: 10.1109/SP.2013.34.

[7] BenSasson E, Chiesa A, Garman C, et al. Zerocash: Decentralized anonymous payments from bitcoin [C]// *IEEE Symposium on Security and Privacy*. San Jose, CA, USA, 2014: 459 - 474. DOI: 10.1109/SP.2014.36.

[8] Green M, Miers I. Bolt: Anonymous payment channels for decentralized currencies [C]// 2017 *ACM SIGSAC Conference on Computer and Communications Security*. Dallas, TX, USA, 2017: 473 - 489. DOI: 10.1145/3133956.31 34093.

[9] Naganuma K, Yoshino M, Sato H, et al. Auditable Zerocoin [C]// *IEEE European Symposium on Security and Privacy Workshops*. Paris, France, 2017: 59 - 63. DOI: 10.1109/EuroSP.2017.51.

[10] Garman C, Green M, Miers I. Accountable privacy for decentralized anonymous payment [C]// *Financial Cryptography and Data Security*. Christ Church, Barbados, 2016: 81 - 98. DOI: 10.1007/978-3-662-54970-4_5.

[11] Mitani T, Otsuka. Confidential and auditable payments [C]// *Financial Cryptography and Data Security*. Kota Kinabalu, Malaysia, 2020: 466 - 480. DOI: 10.1007/978-3-030-544553-3 _33.

[12] Zheng H, Wu Q, Xie J, et al. An organization-friendly blockchain system [J]. *Computer & Security*, 2020, **88**: 101598. DOI: 10.1016/j.cose.2019.101598.

[13] Paillier P. Public-key cryptosystems based on composite degree residuosity classed [J]. *Lecture Notes in Computer Science*, 1999, **1592**: 223 - 238. DOI: 10.1007/3-540-48910-X16.

密码分析组织友好的区块链系统

张 颖 蒋 睿

(东南大学网络空间安全学院, 南京 210096)

摘要: 为验证组织友好的区块链系统可能遭受伪冒攻击和共谋攻击,按照组织友好的区块链系统的阶段顺序,理论上进行伪冒攻击和共谋攻击. 然后,改善组织友好的区块链系统,并根据其阶段顺序,进行伪冒攻击和共谋攻击. 结果表明:攻击者可以通过对组织友好的区块链系统进行伪冒攻击和共谋攻击,获得非法交易量;攻击者则无法对改善后的组织友好区块链系统进行伪冒攻击和共谋攻击. 由此说明,组织友好的区块链系统可能遭受伪冒攻击和共谋攻击,改善后的区块链系统则可以防止伪冒攻击和共谋攻击.

关键词: 区块链;身份隐私;交易监督;伪冒攻击;共谋攻击

中图分类号: TP309.2