Confidential and auditable payments*

Tatsuo Mitani[1,2] and Akira Otsuka[1]

[1] Institute of Information Security, Yokohama, Japan
{dgs187101,otsuka}@iisec.ac.jp
[2] Mitsubishi Chemical Systems, Inc.

Abstract. In this paper, we construct the Confidential and Auditable Payments (CAP) scheme. We keep the transaction confidential by writing ciphertexts of transactions in a ledger. We also realize the soundness of the CAP scheme by the soundness of the zero-knowledge proof. A court controls a unique secret key of the ciphertexts written in the ledger. The court can enforce confidential transactions open with the secret key according to the legal procedure. Although there are many works for protecting transaction's privacy strictly, the CAP scheme eliminates concerns about money laundering caused by excessively confidential transactions. It contributes to the sound use of blockchain.

Keywords: blockchain · homomorphic encryption · zero-knowledge proof.

1 Introduction

Bitcoin [13] has spread around the world in the past decade. This virtual technology is called a blockchain. Bitcoin makes transaction information public. For this reason, Bitcoin is transparent. The bank keeps the customer's transaction information confidential. Therefore, there is a need to keep transaction information concealed even in the blockchain. However, excessive confidentiality of transaction information may cause money laundering.

In this paper, we construct the Confidential and Auditable Payments (CAP) scheme that allows a court to audit transactions while keeping the transaction information confidential. Every participant writes their account balance as a ciphertext of homomorphic encryption in a ledger with a unique public key. They realize transactions by calculating ciphertexts. The court controls its secret key and can forcefully decrypt the ciphertexts and confirm the information.

1.1 Related work

In Bitcoin, transaction information is open. For this reason, there are some works for concealing transaction information. Zerocoin [10] and Zerocash [15] are extensions based on Bitcoin. They realized strong anonymity by designing anonymous coins that skillfully combined commitments. Zerocash uses the zero-knowledge

* Supported by Mitsubishi Chemical Corporation.

succinct non-interactive argument of knowledge (zk-SNARK) [6] to show to others that there is no fraud such as double-spending. Recently, Zether [4], a cryptocurrency that can conceal transaction information, has been proposed as an extension based on Ethereum. Zether has also used the zero-knowledge proof Bulletproofs [5] proposed by their group.

Mitani and Otsuka expressed the state transition of a permissioned blockchain using homomorphic encryption [11]. In their scheme, a zero-knowledge proof of plaintext knowledge shows that the equation of the encrypted model is established to outsiders of the blockchain, and proves the validity of the state transition.

1.2 Our approach

There are many works for protecting transaction privacy strictly. However, we intend to reduce the excessive confidentiality inducing money laundering. We propose a confidential and auditable scheme.

Let us state our approach. We describe auditability at first. A pair of a public key pk_0 and a secret key sk_0 of homomorphic encryption is issued. The court controls this secret key sk_0. They can decrypts ciphertexts according to the judicial procedure when requested. In this sense, the scheme is auditable.

Next, we state confidentiality. As shown in Fig. 1, we consider that a column vector of each participant's account balance $\vec{x}$ is updated to the next time state $\vec{x}'$ by the transition matrix A. That is, $\vec{x}' = \vec{x}A$. The transition matrix A corresponds to the remittance by each participant. Each participant uses the public key pk_0 in common. Each participant writes their balance in the ledger as ciphertext $\vec{\mathbf{x}}$. As for the transition matrix, each participant writes each element in the ledger as ciphertext $\mathbf{A}$. Then, each participant writes their balance of the next time, reflecting the remittance in the ledger as a ciphertext $\vec{\mathbf{x}}'$. All the ciphertexts in the ledger are the ciphertexts of homomorphic encryption. $\vec{\mathbf{x}}' - \vec{\mathbf{x}}\mathbf{A}$ is the ciphertext of zero if and only if $\vec{x}' = \vec{x}A$.

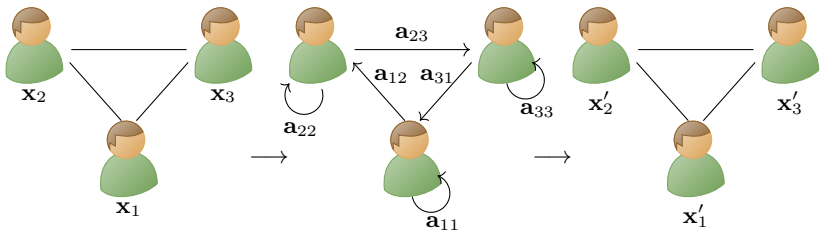

Fig. 1. Illustration of a state transition for three participants. $\vec{x} = (x_1, x_2, x_3), \vec{x}' = (x'_1, x'_2, x'_3)$ and $A = (a_{ij})$ are plaintexts. The corresponding ciphertexts are $\vec{\mathbf{x}}, \vec{\mathbf{x}}'$ and $\mathbf{A}$. Because of the homomorphic encryption, $\vec{\mathbf{x}}' - \vec{\mathbf{x}}\mathbf{A}$ is the ciphertext of zero if and only if $\vec{x}' = \vec{x}A$.

Let us confirm the remittance procedure when a sender i transfers to a recipient j. The element a_{ij} of the transition matrix A corresponds to this remittance. We use zero-knowledge proof of plaintext knowledge of ciphertext to show that the remittance is legitimate. The sender also writes the proof regarding a_{ij} in the ledger. This proof shows that the sender has not sent more than their account balance. Knowledge corresponds to the randomness in creating the ciphertext of homomorphic encryption. At the time of remittance, the sender encrypts the randomness with the recipient's public key pk_j. Then the sender sends it to the recipient. The recipient decrypts with their secret key sk_j and obtains the randomness of the remittance. From the knowledge of the randomness collected, the recipient proves that their updated account balance x'_j is correct by the operation $\vec{x}' = \vec{x}A$.

We organize the rest of this paper as follows. Section 2 describes two ingredients: the ring learning with errors(RLWE) and the zero-knowledge proof(ZKP). Section 3 describes the definitions for a secure CAP scheme. Section 4 describes the construction of the CAP scheme, including data structures, algorithms, and security proofs. Section 5 states the conclusion and future work.

2 Building blocks

Notation. We summarize the parameters and notation in this paper in Table 1. We identify a vector $(a_0, \ldots, a_{n_d-1})$ with a polynomial $a_0 + a_1 X + \cdots + a_{n_d-1} X^{n_d-1}$.

RLWE. Fully homomorphic encryption is capable of both addition and multiplication in ciphertexts. Gentry realized it in 2009 [7]. The RLWE scheme [9] is one of the most promising forms. We introduce the definitions used in this paper.

Definition 1 (Syntax of the RLWE scheme (Definition 6 in [2])). *We describe the RLWE scheme represented by a tuple of PPT algorithms*

$$\mathsf{RLWE} := (\mathsf{RLWE.Gen}, \mathsf{RLWE.Enc}, \mathsf{RLWE.Dec})$$

with the following syntax. We denote a message space as $\mathbf{M}$ *and a ciphertext space as* $\mathbf{C}$*.*

- $\mathsf{RLWE.Gen}(1^\lambda)$ *returns a key pair* $(\mathsf{pk}, \mathsf{sk})$ *from an input* 1^λ*.*
- $\mathsf{RLWE.Enc}(\mathsf{pk}, m, r)$ *returns a ciphertext* $\mathbf{c} \in \mathbf{C}$ *from an input of the public key* pk*, a message* $m \in \mathbf{M}$ *and a randomness* r*.*
- $\mathsf{RLWE.Dec}(\mathsf{sk}, \mathbf{c})$ *returns a message* $m \in \mathbf{M}$ *or* $\perp$ *from an input of the secret key* sk *and a ciphertext* $\mathbf{c} \in \mathbf{C}$*.*

Definition 2 (Pseudorandomness of ciphertexts (Definition 7 in [2])). *We say a RLWE scheme* $\mathsf{RLWE} := (\mathsf{RLWE.Gen}, \mathsf{RLWE.Enc}, \mathsf{RLWE.Dec})$ *satisfies pseudorandomness of ciphertexts or simply* RLWE *is secure, if for every PPT adversary* $\mathcal{A}$ *the advantage*

$$\mathrm{Adv}^{\mathsf{pr}}_{\mathsf{RLWE},\mathcal{A}}(\lambda) := \mid \Pr[\mathrm{Exp}^{\mathsf{pr}}_{\mathsf{RLWE},\mathcal{A}}(\lambda) = 1] - 1/2 \mid$$

is negligible in λ*, where* $\mathrm{Exp}^{\mathsf{pr}}_{\mathsf{RLWE},\mathcal{A}}(\lambda)$ *is as defined in Fig. 2.*

Table 1. List of Parameters and Notation

Parameters	Explanation
$\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{R}$	the sets of natural numbers, integers, rational numbers and real numbers
$\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$	the finite field $\{0, 1, \ldots, q-1\}$
n_d	the degree of the polynomial
$\mathbf{R} = \mathbb{Z}[X]/\langle X^{n_d}+1\rangle$	the ring of integers
$\mathbf{R}_q = \mathbb{Z}_q[X]/\langle X^{n_d}+1\rangle$	the ring of integers modulo q
$\lvert a\rvert = \sqrt{a_0^2 + a_1^2 + \cdots + a_{n_d-1}^2}$	l_2-norm of $a \in \mathbf{R}_q$. a_i is the coefficient of X^i.
$\mathrm{negl}(n)$	a negligible function f such that $\forall n > n_c, f(n) < 1/n^c$ under $\forall c \in \mathbb{N}, \exists n_c$
$a \xleftarrow{\$} A$	randomly sampled from the distribution A or from a uniform distribution over the set A.
$\tilde{\mathcal{O}}(\cdot)$	Bachmann-Landau notation
$\omega(f(n))$	the ω notation is a function growing asymptotically faster than $f(n)$.
$\mathcal{O}$	an oracle
$D_{v,\sigma} = \rho^{n_d}_{v,\sigma}(x)/\rho^{n_d}_{0,\sigma}(\mathbb{Z}^{n_d})$ and $D_\sigma = \rho^{n_d}_{0,\sigma}(x)/\rho^{n_d}_{0,\sigma}(\mathbb{Z}^{n_d})$	D_σ and $D_{v,\sigma}$ are discrete gaussian distributions on $\mathbf{R}$ with mean $0, v$ and standard deviation σ. $\rho^{n_d}_{v,\sigma}(x) = (\frac{1}{\sqrt{2\pi}\sigma})^{n_d} e^{\frac{-\lvert x-v\rvert^2}{2\sigma^2}}$ over $\mathbb{R}^{n_d}$ $\rho^{n_d}_{0,\sigma}(\mathbb{Z}^{n_d}) = \Sigma_{z\in\mathbb{Z}^{n_d}} \rho^{n_d}_{0,\sigma}(z)$
Adv	the advantage of an adversary in a game
Exp	the experiment in a game
$\Pr[E]$	the probability that an event E occurs.

Remark 1. We consider RLWE scheme that is secure in the sense of Definition 2. We write a ciphertext $\mathbf{c}$ in the boldface like this. We concretely consider Brakerski-Vaikuntanathan (BV) scheme [3] in Table 2.

ZKP. Goldwasser *et al.* originally formulated this method [8]. ZKP is an interactive protocol that a prover that has a secret can inform a verifier of the fact that the proposition on the secret is correct. A prover can prevent the secret from being leaked. A verifier can reject a malicious prover with no secret.

Benhamouda *et al.* originally proposed the zero-knowledge proof of knowledge of randomness in RLWE encryption [1]. Then, Mitani and Otsuka adopt its scheme for the zero-knowledge proof of plaintext $m = 0$ knowledge in RLWE encryption [11, 12]. We state this detail in Appendix A. In this paper, we denote the zero-knowledge proof and its verification as below syntax.

- $\mathsf{Prove}(\mathbf{x}; r)$ inputs the ciphertext $\mathbf{x}$ and its randomness r and outputs the proof π for the knowledge of $\mathsf{RLWE.Dec}(\mathsf{sk}, \mathbf{x}) = 0$. We denotes $\mathsf{Prove}(\mathbf{x}; r_1, \ldots, r_n)$ if there are some randomnesses $r_1, \ldots, r_n$.
- $\mathsf{Verify}(\pi)$ inputs the proof π and outputs 1 if it is valid, otherwise 0.

$\underline{\mathrm{Exp}^{\mathsf{pr}}_{\mathsf{RLWE},\mathcal{A}}(\lambda)}$:	$\underline{\mathcal{O}_{\mathsf{RLWE.Enc}}(m)}$:
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{RLWE.Gen}(1^\lambda)$	if $\beta = 0$
$\beta \leftarrow \{0,1\}$	$\quad \mathbf{c} \leftarrow \mathsf{RLWE.Enc}(\mathsf{pk}, m)$
$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{RLWE.Enc}}(\cdot)}(1^\lambda, \mathsf{pk})$	else
if $\beta = \beta'$ return 1	$\quad \mathbf{c} \xleftarrow{\$} \mathbf{R}_q^2$
else return 0	return $\mathbf{c}$

Fig. 2. Security challenge experiment for pseudorandomness of ciphertexts.

Table 2. Brakerski-Vaikuntanathan (BV) scheme

Parameters	Explanation
$\mathbf{M} = \mathbf{R}_p$	the plaintext space
(a, b) and s	(a, b) is a public key. s is a secret key. Then, $a, s \xleftarrow{\$} \mathbf{R}_q,\ e_s \xleftarrow{\$} D_\sigma,\ b = as + e_s$.
$\mathbf{c} = (c_1, c_2) = (bv + pe + m, av + pf)$	the ciphertext of a plaintext $m \in \mathbf{M}$ Then, a set of randomness $v, e, f \xleftarrow{\$} D_\sigma$.
$m = (c_1 - s \cdot c_2 \mod q) \mod p$	decryption of $\mathbf{c} = (c_1, c_2)$

3 Secure CAP scheme

In this section, we define the security of a CAP scheme. Regarding security, we follow Zerocash [15]. Zerocash defines and satisfies the three properties; ledger indistinguishability, transaction non-malleability, and balance. In this sense, the DAP scheme of Zerocash is secure. We adjust the three properties for a CAP scheme. We modify ledger indistinguishability. Balance corresponds to the proof regarding the transition matrix, according to Lemma 1. We combine transaction non-malleability and balance into non-malleability. In this sense, a CAP scheme Π is secure. Let us confirm the below definitions.

Definition 3 (Ledger indistinguishability). *A CAP scheme*

$$\Pi := (\mathsf{Setup}, \mathsf{CreateAccount}, \mathsf{Transition}, \mathsf{Send}, \mathsf{Receive})$$

satisfies ledger indistinguishability, if for any $\lambda \in \mathbb{N}$ and any PPT adversary $\mathcal{A}$, the advantage

$$\mathrm{Adv}^{\mathsf{L-IND}}_{\Pi,\mathcal{A}}(\lambda) := \mid \Pr[\mathrm{Exp}^{\mathsf{L-IND}}_{\Pi,\mathcal{A}}(\lambda) = 1] - 1/2 \mid$$

is negligible in λ, where $\mathrm{Exp}^{\mathsf{L-IND}}_{\Pi,\mathcal{A}}(\lambda)$ is defined in Fig. 3.

In Fig. 3, a pair of the queries (Q, Q') must be the same type; CreateAccount *or* Transition*. If the query type is* Transition*, then $Q = (\mathsf{Transition}, A)$ and $Q' = (\mathsf{Transition}, A')$. A and A' must be same size. Each element is generally different. That is, $A \neq A'$. $\vec{Q}$ and $\vec{Q}'$ are the lists of all the queries that $\mathcal{A}$ sent to the oracles.* Append *is the function to append the latest query to the lists.*

$\underline{\mathrm{Exp}^{\mathsf{L-IND}}_{\Pi,\mathcal{A}}(\lambda)}$:
$\mathsf{pp}, \mathsf{cp} \leftarrow \mathsf{Setup}(1^\lambda)$
$L_0 \leftarrow \mathcal{O}_0^\Pi(\mathsf{pp})$; $L_1 \leftarrow \mathcal{O}_1^\Pi(\mathsf{pp})$
$b \xleftarrow{\$} \{0,1\}$
while:
  $(Q, Q') \leftarrow \mathcal{A}(\mathsf{pp}, \vec{Q}, \vec{Q}', L_b, L_{1-b})$
  $L_b \leftarrow \mathcal{A}^{\mathcal{O}_b^\Pi(\cdot)}(\mathsf{pp}, Q, L_b)$; $L_{1-b} \leftarrow \mathcal{A}^{\mathcal{O}_{1-b}^\Pi(\cdot)}(\mathsf{pp}, Q', L_{1-b})$
  $\vec{Q} \leftarrow \mathsf{Append}(\vec{Q}, Q)$; $\vec{Q}' \leftarrow \mathsf{Append}(\vec{Q}', Q')$
  $c \leftarrow \mathcal{A}(\mathsf{pp}, \vec{Q}, \vec{Q}', L_b, L_{1-b})$
  if $c = 1$ break
  else continue
$b' \leftarrow \mathcal{A}(\mathsf{pp}, \vec{Q}, \vec{Q}', L_b, L_{1-b})$
if $b = b'$ return 1
else return 0

Fig. 3. Security challenge experiment for ledger indistinguishability

Definition 4 (Non-malleability). *A CAP scheme*

$$\Pi := (\mathsf{Setup}, \mathsf{CreateAccount}, \mathsf{Transition}, \mathsf{Send}, \mathsf{Receive})$$

satisfies non-malleability, if for any $\lambda \in \mathbb{N}$ and any PPT adversary $\mathcal{A}$, the advantage

$$\mathrm{Adv}^{\mathsf{NM}}_{\Pi,\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\mathsf{NM}}_{\Pi,\mathcal{A}}(\lambda) = 1]$$

is negligible in λ, where $\mathrm{Exp}^{\mathsf{NM}}_{\Pi,\mathcal{A}}(\lambda)$ is defined in Fig. 4. Besides, $\mathcal{O}^\Pi$ and Append *are the same as Definition 3.*

We lead the below definition of a secure CAP scheme.

Definition 5 (Security). *A CAP scheme*

$$\Pi := (\mathsf{Setup}, \mathsf{CreateAccount}, \mathsf{Transition}, \mathsf{Send}, \mathsf{Receive})$$

is secure if it satisfies ledger indistinguishability of Definition 3 and non-malleability of Definition 4.

4 Construction

In this section, we describe data structures, algorithms, and security analysis.

4.1 Data structures

In this subsection, we describe data structures of the CAP scheme. Following the ideas of Mitani and Otsuka [11], we consider that the state of the assets held by each participant will transition to the next state.

$\underline{\mathrm{Exp}^{\mathsf{NM}}_{\Pi,\mathcal{A}}(\lambda)}$:
$\mathsf{pp}, \mathsf{cp} \leftarrow \mathsf{Setup}(1^\lambda)$
$L \leftarrow \mathcal{O}^{\Pi}(\mathsf{pp})$
while:
  $Q \leftarrow \mathcal{A}(\mathsf{pp}, \vec{Q}, L)$
  $L \leftarrow \mathcal{A}^{\mathcal{O}^{\Pi}(\cdot)}(\mathsf{pp}, Q, L)$
  $\vec{Q} \leftarrow \mathsf{Append}(\vec{Q}, Q)$
  $c \leftarrow \mathcal{A}(\mathsf{pp}, \vec{Q}, L)$
  if $c = 1$ break
  else continue
$A^* \leftarrow \mathcal{A}(\mathsf{pp}, \vec{Q}, L)$
$L' \leftarrow \mathcal{A}^{\mathcal{O}^{\Pi}(\cdot)}(\mathsf{pp}, (\mathsf{Transition}, A^*), L)$
if ($\mathsf{Verify}(\pi^a)$ or $\mathsf{Verify}(\vec{\pi}^b)$ or $\mathsf{Verify}(\Pi^c)$ or $\mathsf{Verify}(\vec{\pi}')$) and $(\sum_i(\mathbf{x}'_i - \mathbf{x}_i) \notin \mathbf{C}_0)$ in L'
  return 1
else return 0

Fig. 4. Security challenge experiment for non-malleability

- $\vec{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{Q}^n$. $\vec{x}$ is the amount of assets in the blockchain at time t. n is the number of the participants in a blockchain.
- $\vec{x}' = (x'_1, x'_2, \ldots, x'_n) \in \mathbb{Q}^n$. $\vec{x}'$ is the amount of assets in the blockchain at time $t+1$.
- $A = (a_{ij})$ is a transition matrix such that $\vec{x}' = \vec{x}A$. Its size is $n \times n$. We suppose that the total volume moving from x_i to x'_j is v. That is, $x'_i = x_i - v$ and $x'_j = x_j + v$. We define the distribution rate $a_{ij} := v/x_i$. In particular, we consider a_{ii} as the "staying" rate ($x_i \to x'_i$). In the transition, the amount x_i is distributed to $x'_1, \ldots, x'_i, \ldots, x'_n$ at the ratio of $a_{i1}, \ldots, a_{ii}, \ldots, a_{in}$. The sum of all the ratios must be equal 1 because of the preservation. That is, $\sum_{j=1}^n a_{ij} = 1$. Let us confirm the following lemma.

  Lemma 1 (Lemma 1 in [11]). *If $\vec{x}' = \vec{x}A$ and $\sum_{j=1}^n a_{ij} = 1$, then the equation $\sum_{i=1}^n x'_i = \sum_{i=1}^n x_i$ holds.*
- Ledger L is a distributed ledger each participants in the blockchain holds and updates. L contains the ciphertexts $\vec{\mathbf{x}}, \vec{\mathbf{x}}', \mathbf{A}$ and related proofs.

Remark 2. It is impractical to force all participants to join the transition matrix A every time. We assume a transition matrix A' that pertains only to participants trading at a particular time t. That is, A' is the submatrix of the transition matrix A for all participants. Since there is essentially no difference, we will discuss A and A' indiscriminately.

4.2 Algorithms

We describe the CAP scheme

$$\Pi := (\mathsf{Setup}, \mathsf{CreateAccount}, \mathsf{Transition}, \mathsf{Send}, \mathsf{Receive})$$

in Fig. 5. We can verify the completeness of the CAP scheme Π by confirming the construction in Fig. 5.

4.3 Security analysis

We describe the below lemmas for the security of the CAP scheme Π in Fig. 5.

Lemma 2 (Ledger indistinguishability). *The CAP scheme Π in Fig. 5 satisfies ledger indistinguishability in Definition 3.*

Proof. We consider the games in Fig. 6. We denote events as follows.

- $E_0 : \mathrm{G}^0_{\mathsf{CPA},\mathcal{A}}(\lambda) = 1$
- $E_1 : \mathrm{G}^1_{\mathsf{CPA},\mathcal{A}}(\lambda) = 1$
- $B_0 : \beta = 0$
- $B_1 : \beta = 1$

Since B_0 and B_1 are disjoint events, we have

$$\Pr[E_i] = \Pr[E_i \cap B_0] + \Pr[E_i \cap B_1] = \Pr[E_i|B_0] \cdot \Pr[B_0] + \Pr[E_i|B_1] \cdot \Pr[B_1].$$

where $i = \{0, 1\}$. We construct an adversary $\mathcal{B}$ from an adversary $\mathcal{A}$ who can distinguish $\mathrm{G}^0_{\mathsf{CPA},\mathcal{A}}(\lambda)$ and $\mathrm{G}^1_{\mathsf{CPA},\mathcal{A}}(\lambda)$. That is,

$$\begin{aligned}
&| \Pr[\mathrm{G}^0_{\mathsf{CPA},\mathcal{A}}(\lambda) = 1] - \Pr[\mathrm{G}^1_{\mathsf{CPA},\mathcal{A}}(\lambda) = 1] | \\
&=| \Pr[E_0] - \Pr[E_1] | \\
&=| (\Pr[E_0|B_0] \cdot \Pr[B_0] + \Pr[E_0|B_1] \cdot \Pr[B_1]) - (\Pr[E_1|B_0] \cdot \Pr[B_0] + \Pr[E_1|B_1] \cdot \Pr[B_1]) | \\
&=| \Pr[B_0] \cdot (\Pr[E_0|B_0] - \Pr[E_1|B_0]) + \Pr[B_1] \cdot (\Pr[E_0|B_1] - \Pr[E_1|B_1]) | \\
&\leq \Pr[B_0] \cdot | \Pr[E_0|B_0] - \Pr[E_1|B_0] | + \Pr[B_1] \cdot | \Pr[E_0|B_1] - \Pr[E_1|B_1] | \\
&= (\Pr[B_0] + \Pr[B_1]) \cdot \mathrm{Adv}^{\mathsf{pr}}_{\mathsf{RLWE},\mathcal{B}}(\lambda) \\
&= \mathrm{Adv}^{\mathsf{pr}}_{\mathsf{RLWE},\mathcal{B}}(\lambda)
\end{aligned}$$

where we applied $\Pr[B_0] + \Pr[B_1] = 1$ in the last equation. Since the game $\mathrm{G}^1_{\mathsf{CPA},\mathcal{A}}(\lambda)$ is independent of β, we have thus $\Pr[\mathrm{G}^1_{\mathsf{CPA},\mathcal{A}}(\lambda) = 1] = 1/2$. Moreover, $\mathrm{Exp}^{\mathsf{L-IND}}_{\Pi,\mathcal{A}}(\lambda)$ consists of the game $\mathrm{G}^0_{\mathsf{CPA},\mathcal{A}}(\lambda)$. Because of Definition 2, we obtain

$$\mathrm{Adv}^{\mathsf{L-IND}}_{\Pi,\mathcal{A}}(\lambda) \leq | \Pr[\mathrm{G}^0_{\mathsf{CPA},\mathcal{A}}(\lambda) = 1] - 1/2 | \leq \mathrm{Adv}^{\mathsf{pr}}_{\mathsf{RLWE},\mathcal{B}}(\lambda) < \mathrm{negl}(\lambda).$$

Lemma 3 (Non-malleability). *The CAP scheme Π in Fig. 5 satisfies non-malleability in Definition 4.*

Proof. Let us consider the below violations of verification in the cases $\mathcal{A}$ wins without the knowledge of randomness.

- $\mathcal{A}$ wins but violates $\mathsf{Verify}(\pi^a)$ in $\mathsf{CreateAccount}$.
- $\mathcal{A}$ wins but violates $\mathsf{Verify}(\vec{\pi}^b)$ in $\mathsf{Transition}$.
- $\mathcal{A}$ wins but violates $\mathsf{Verify}(\mathit{\Pi}^c)$ in $\mathsf{Transition}$.

Setup

- Inputs: security parameter λ
- Outputs: public parameters pp and court parameters cp

1. Compute $(\mathsf{pk}_0, \mathsf{sk}_0) := \mathsf{RLWE.Gen}(1^\lambda)$.
2. Set $\mathsf{pp} := \mathsf{pk}_0$.
3. Set $\mathsf{cp} := \mathsf{sk}_0$.
4. Output pp and cp.

CreateAccount

- Inputs: public parameters pp and security parameter λ
- Outputs: account ciphertext $\mathbf{x}$, proof π^a in L and key pair $(\mathsf{pk}, \mathsf{sk})$

1. Choose randomly $r_{\mathbf{x}}$.
2. Compute $\mathbf{x} := \mathsf{RLWE.Enc}(\mathsf{pk}_0, 0, r_{\mathbf{x}})$.
3. Compute $\pi^a := \mathsf{Prove}(\mathbf{x}; r_{\mathbf{x}})$.
4. Compute $(\mathsf{pk}, \mathsf{sk}) := \mathsf{RLWE.Gen}(1^\lambda)$.
5. Output $\mathbf{x}$, π and $(\mathsf{pk}, \mathsf{sk})$.

Send

- Inputs:
  - public parameters pp
  - account ciphertext $\mathbf{x}$ in Ledger L
  - randomness $r_{\mathbf{x}}$
  - plaintext x
  - sending volume v
  - public key pk_j
- Outputs:
  - copy proof π^c
  - copy account $\mathbf{x}^c$
  - distribution ratio $\mathbf{a}_{ij}$
  - volume $\mathbf{v}$
  - randomness $\mathbf{r_v}$

1. Choose randomly $r_{\mathbf{x}^c}$, where $r_{\mathbf{x}^c} \neq r_{\mathbf{x}}$.
2. Compute $\mathbf{x}^c := \mathsf{RLWE.Enc}(\mathsf{pk}_0, x, r_{\mathbf{x}^c})$.
3. Compute $\pi^c := \mathsf{Prove}(\mathbf{x}^c - \mathbf{x}; r_{\mathbf{x}^c} - r_{\mathbf{x}})$.
4. Compute $a := v/x$.
5. Choose randomly $r_{\mathbf{a}_{ij}}$.
6. Compute $\mathbf{a}_{ij} := \mathsf{RLWE.Enc}(\mathsf{pk}_0, a, r_{\mathbf{a}_{ij}})$.
7. Compute $r_{\mathbf{v}}$ from $r_{\mathbf{x}^c}$ and $r_{\mathbf{a}_{ij}}$.
8. Compute $\mathbf{v} := \mathsf{RLWE.Enc}(\mathsf{pk}_j, v, r_{\mathbf{v}})$.
9. Compute $\mathbf{r_v} := \mathsf{RLWE.Enc}(\mathsf{pk}_j, r_{\mathbf{v}}, r_{\mathbf{r_v}})$.
10. Output $\pi^c, \mathbf{x}^c, \mathbf{a}_{ij}, \mathbf{v}, \mathbf{r_v}$.

Transition

- Inputs:
  - public parameters pp
  - old accounts state $\vec{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ in L
  - accounts randomness $r_{\mathbf{x}_1}, \ldots, r_{\mathbf{x}_n}$
- Outputs in L:
  - new accounts state $\vec{\mathbf{x}}' = (\mathbf{x}'_1, \ldots, \mathbf{x}'_n)$
  - new transition matrix $\mathbf{A} = (\mathbf{a}_{ij})$
  - balance proofs $\vec{\pi}^b = (\pi^b_1, \ldots, \pi^b_n)$
  - accounts copy proofs $\mathit{\Pi}^c = (\pi^c_{ij})$
  - transition proofs $\vec{\pi}' = (\pi'_1, \ldots, \pi'_n)$

1. For $i \in \{1, \ldots, n\}$ (i sends v to j.)
   (a) For $j \in \{1, \ldots, n\}$
      i. $\mathsf{Send}(\mathsf{pp}, \mathbf{x}_i, r_{\mathbf{x}_i}, x_i, v, \mathsf{pk}_j)$.
      ii. Write $\pi^c_{ij}, \mathbf{x}^c_{ij}, \mathbf{a}_{ij}$ in L.
   (b) Compute
   $$\pi^b_i := \mathsf{Prove}(\sum_j \mathbf{a}_{ij} - \mathbf{1}; r_{\mathbf{a}_{i1}}, \ldots, r_{\mathbf{a}_{in}}).$$
   (c) Write π^b_i in L.
2. $\mathsf{Verify}(\mathit{\Pi}^c)$
3. $\mathsf{Verify}(\vec{\pi}^b)$
4. For $j \in \{1, \ldots, n\}$ (j receives v from i.)
   (a) For $i \in \{1, \ldots, n\}$
      i. $\mathsf{Receive}(\mathsf{pp}, \mathsf{sk}_j, \mathbf{x}^c_i, \mathbf{a}_{ij}, \mathbf{v}, \mathbf{r_v}, \pi^c_{ij})$.
   (b) Compute $x'_j := \sum_i v_{ij}$.
   (c) Choose $r_{\mathbf{x}'_j}$ randomly.
   (d) Compute $\mathbf{x}'_j := \mathsf{RLWE.Enc}(\mathsf{pk}_0, x'_j, r_{\mathbf{x}'_j})$.
   (e) Compute
   $$\pi'_j := \mathsf{Prove}(\mathbf{x}'_j - \sum_i \mathbf{a}_{ij} \cdot \mathbf{x}_i; r_{\mathbf{x}'_j}, r_{\mathbf{v}_{1j}}, \ldots, r_{\mathbf{v}_{nj}}).$$
   (f) Write $\mathbf{x}'_j, \pi'_j$ in L.
5. $\mathsf{Verify}(\vec{\pi}')$

Receive

- Inputs:
  - public parameters pp
  - secret key sk_j
  - ciphertexts $\mathbf{x}^c, \mathbf{a}_{ij}, \mathbf{v}, \mathbf{r_v}$
  - proof π^c in L
- Outputs:
  - volume v
  - randomness $r_{\mathbf{v}}$

1. Compute $v := \mathsf{RLWE.Dec}(\mathsf{sk}_j, \mathbf{v})$.
2. Compute $r_{\mathbf{v}} := \mathsf{RLWE.Dec}(\mathsf{sk}_j, \mathbf{r_v})$.
3. If $\mathbf{a}_{ij} \cdot \mathbf{x}^c = \mathsf{RLWE.Enc}(\mathsf{pk}_0, v, r_{\mathbf{v}})$ and $|r_{\mathbf{v}}|$ is small: Output $v, r_{\mathbf{v}}$.
4. Else: Output $\perp$.

Fig. 5. Construction of the CAP scheme

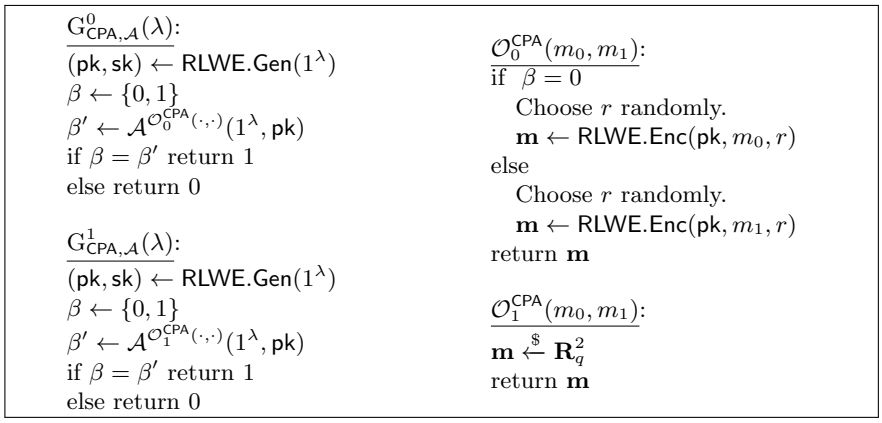

Fig. 6. Security challenge experiment for plaintexts

– $\mathcal{A}$ wins but violates $\mathsf{Verify}(\vec{\pi}')$ in $\mathsf{Transition}$.

Any violations result from the violation of zero-knowledge proof, that is, the knowledge error of zero-knowledge proof. From Lemma 6, the knowledge error of zero-knowledge proof is negligible.

Finally, we can lead the below theorem from Lemma 2 and Lemma 3.

Theorem 1 (The secure CAP scheme). *The CAP scheme* Π *in Fig. 5 is secure in the sense of Definition 5.*

5 Conclusion

In this paper, we constructed the confidential and auditable payments(CAP) scheme that allows a court to audit transactions while keeping the transaction information confidential. We proposed that every participant writes the ciphertexts of transaction information in a ledger and confirmed the ledger indistinguishability, that is, the concealment of the transaction information. Besides, we confirmed the soundness of the CAP scheme by the soundness of the zero-knowledge proof. Therefore the CAP scheme is secure. A court can forcibly clarify transaction information with a unique secret key. In this sense, we realized auditability in the CAP scheme.

Future work. In the CAP scheme, the secret key of a court can decrypt all transaction information. Therefore, we expect the court to disclose minimum requisite information. Building a decryptable scheme for each account is an open problem.

References

1. Benhamouda, F., Camenisch, J., Krenn, S., Lyubashevsky, V., Neven, G.: Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology – ASIACRYPT 2014. pp. 551–572. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
2. Boyle, E., Kohl, L., Scholl, P.: Homomorphic secret sharing from lattices without fhe. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019. pp. 3–33. Springer International Publishing, Cham (2019)
3. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-lwe and security for key dependent messages. In: Rogaway, P. (ed.) Advances in Cryptology – CRYPTO 2011. pp. 505–524. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
4. Bünz, B., Agrawal, S., Zamani, M., Boneh, D.: Zether: Towards privacy in a smart contract world. IACR Cryptology ePrint Archive **2019**, 191 (2019)
5. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 315–334. IEEE (2018)
6. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct nizks without pcps. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. pp. 626–645. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
7. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. pp. 169–178. STOC 09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1536414.1536440
8. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on computing **18**(1), 186–208 (1989)
9. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. Journal of the ACM (JACM) **60**(6), 43 (2013)
10. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy. pp. 397–411. IEEE (2013)
11. Mitani, T., Otsuka, A.: Traceability in permissioned blockchain. In: 2019 IEEE International Conference on Blockchain (Blockchain). pp. 286–293 (July 2019). https://doi.org/10.1109/Blockchain.2019.00045
12. Mitani, T., Otsuka, A.: Traceability in permissioned blockchain. IEEE Access **8**, 21573–21588 (2020). https://doi.org/10.1109/ACCESS.2020.2969454
13. Nakamoto, S., et al.: Bitcoin: A peer-to-peer electronic cash system (2008)
14. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) Advances in Cryptology — CRYPTO '91. pp. 129–140. Springer Berlin Heidelberg, Berlin, Heidelberg (1992)
15. Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy. pp. 459–474. IEEE (2014)

A Non-interactive zero-knowledge proof

Following Benhamouda *et al.* [1], and Mitani and Otsuka [11, 12], we describe the non-interactive zero-knowledge proof of plaintext $m = 0$ knowledge. First, we

describe the formal definition of the Σ'-protocol, the protocol and its theorem proving this relationship.

Definition 6 (Definition 2.5. in [1]). *Let (P, V) be a two-party protocol, where V is a PPT, and let $L, L' \subseteq \{0,1\}^*$ be languages with witness relations R, R' such that $R \subseteq R'$. Then (P, V) is called a Σ'-protocol for L, L' with completeness error α, a challenge set $\mathbb{C}$, a public input x and a private input w, if and only if it satisfies the following conditions:*

- *Three-move form: The prover P, on input (x, w), computes a commitment t and sends it to V. The verifier V, on input x, then draws a challenge $c \leftarrow \mathbb{C}$ and sends it to P. The prover sends a response s to the verifier. Depending on the protocol transcript (t, c, s), the verifier finally accepts or rejects the proof. The protocol transcript (t, c, s) is called accepting, if the verifier accepts the protocol run.*
- *Completeness: Whenever $(x, w) \in R$, the verifier V accepts with probability at least $1 - \alpha$.*
- *Special soundness: There exists a PPT algorithm E (the knowledge extractor) which takes two accepting transcripts $(t, c', s'), (t, c'', s'')$ satisfying $c' \neq c''$ as inputs, and outputs w' such that $(x, w') \in R'$.*
- *Special honest verifier zero knowledge (HVZK): There exists a PPT algorithm S (the simulator) taking $x \in L$ and $c \in \mathbb{C}$ as inputs, that outputs (t, s) so that the triple (t, c, s) is indistinguishable from an accepting protocol transcript generated by a real protocol run.*
- *High-entropy commitments: For all $(y, w) \in R$ and for all t, the probability that an honestly generated commitment by P takes on the value t is negligible.*

Let us introduce Pedersen commitments [14] for the zero-knowledge proof. we make use of the commitments as an auxiliary commitment scheme. We denote it as $(\mathsf{aCSetup}, \mathsf{aCCommit}, \mathsf{aCOpen})$.

Pedersen commitments. Given a family of prime order groups $\{\mathbb{G}(\lambda)\}_{\lambda \in \mathbb{N}}$ such that the discrete logarithm problem is hard in $\mathbb{G}(\lambda)$ with security parameter λ, let $\tilde{q} = \tilde{q}(\lambda)$ be the order of $\mathbb{G} = \mathbb{G}(\lambda)$. To avoid confusion, we denote all elements with order $\tilde{q}$ with a tilde in the following. We will write the group $\mathbb{G}(\lambda)$ additively.

- $\mathsf{aCSetup}$: This algorithm chooses $\tilde{g}, \tilde{h} \xleftarrow{\$} \mathbb{G}$ and outputs $cpars = (\tilde{g}, \tilde{h})$.
- $\mathsf{aCommit}$: To commit to a message $m \in \mathbb{Z}_{\tilde{q}}$, it first chooses $r \xleftarrow{\$} \mathbb{Z}_{\tilde{q}}$. It then outputs a pair $(\widetilde{cmt}, o) = (m\tilde{g} + r\tilde{h}, r)$.
- aCOpen: Given a commitment $\widetilde{cmt}$, an opening o, a public key $cpars$ and a message m, it outputs accept if and only if $(\widetilde{cmt}, o) \stackrel{?}{=} (m\tilde{g} + r\tilde{h}, r)$.

Lemma 4 (Theorem 2.1. in [1]). *Under the discrete logarithm assumption for $\mathbb{G}$, the Pedersen commitment scheme is perfectly hiding and is computationally binding.*

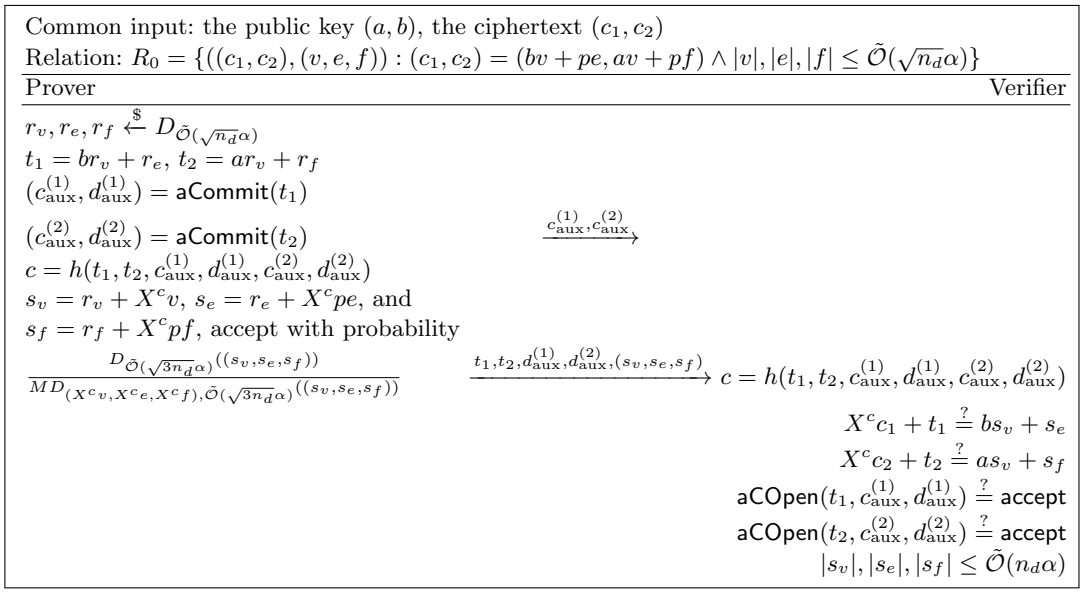

Fig. 7. Non-interactive zero-knowledge proof of a ciphertext of zero regarding Ring-LWE encryption (Figure 3 in [12])

We show a non-interactive zero-knowledge proof of ciphertext of zero in Fig. 7. h is a cryptographic hash function. This protocol satisfies Lemma 5. The parallel protocol satisfies Lemma 6.

Lemma 5 (Lemma 5 in [12]). *The protocol in Fig. 7 is an HVZK Σ'-protocol for the following relations:*

$$R_0 = \{((c_1, c_2), (v, e, f)) : (c_1, c_2) = (bv + pe, av + pf) \wedge |v|, |e|, |f| \leq \tilde{\mathcal{O}}(\sqrt{n_d}\alpha)\}$$
$$R'_0 = \{((c_1, c_2), (v, e, f)) : (2c_1, 2c_2) = (2bv + 2pe, 2av + 2pf) \wedge |2v|, |2e|, |2f| \leq \tilde{\mathcal{O}}(n_d^2\alpha)\}$$

where $2v, 2e$ and $2f$ are reduced modulo q. The protocol has a knowledge error of $1/(2n_d)$, a completeness error of $1 - 1/M$, and high-entropy commitments.

Lemma 6 (Theorem 6 in [12]). *Let us apply the protocol in Fig. 7 for λ times in parallel (the parallel protocol). Let the parallel protocol be accepting if and only if at least $\lambda/2M$ out of λ proofs were valid under the condition that an honest verifier rejects no proofs. Then, the parallel protocol has both a completeness error and knowledge error of $\mathrm{negl}(\lambda)$ under the condition $n_d \geq 2M$.*