IEICE TRANS. FUNDAMENTALS, VOL.E107–A, NO.8 AUGUST 2024

PAPER

Coin-Based Cryptographic Protocols without Hand Operations

Yuta MINAMIKAWA[†], *Nonmember and* **Kazumasa SHINAGAWA**[†,††a)], *Member*

SUMMARY Secure computation is a kind of cryptographic techniques that enables to compute a function while keeping input data secret. Komano and Mizuki (International Journal of Information Security 2022) proposed a model of coin-based protocols, which are secure computation protocols using physical coins. They designed AND, XOR, and COPY protocols using so-called hand operations, which move coins from one player's palm to the other palm. However, hand operations cannot be executed when all players' hands are occupied. In this paper, we propose coin-based protocols without hand operations. In particular, we design a three-coin NOT protocol, a seven-coin AND protocol, a six-coin XOR protocol, and a five-coin COPY protocol without hand operations. Our protocols use random flips only as shuffle operations and are enough to compute any function since they have the same format of input and output, i.e., committed-format protocols.

key words: *secure computation, coin-based protocols, hand operations, random flips*

1. Introduction

Secure computation [13], [14] is a kind of cryptographic techniques that enables to compute a function while keeping input data secret. While secure computation is typically assumed to be implemented on computers, there exists a distinct line of research that focuses on implementing secure computation using everyday physical objects instead of computers, known as physical cryptography [3]. In physical cryptography, various everyday objects are used so far: a deck of cards [1], [2], a dial lock [9], a 15 puzzle [10], balls and bags [7], and so on. This paper focus on secure computation protocols using physical coins, which are known as *coin-based protocols*.

A model of coin-based protocols was proposed by Komano and Mizuki [5], [6]. In this model, secure computation is performed by manipulating a set of identical and indistinguishable coins. They designed coin-based protocols for elementary functions: a five-coin COPY protocol, a six-coin AND protocol, and a six-coin XOR protocol using *hand operations*, which move a bunch of coins from one player's palm to another palm. Although hand operations are interesting operations specific to coin-based protocols, it cannot be executed when all players have their hands occupied. Consequently, when the number of players is small, it becomes difficult to compute large functions. Furthermore, their protocols start with the input coins held in the players' palms, while the output is obtained in the form of a bunch of coins on the table, i.e., the input and output formats are distinct. It is not desirable for protocol composition since it requires to transfer the output bunch of coins into the players' palms without leaking information.

In this paper, we propose coin-based protocols for elementary functions without hand operations. In particular, we design a four-coin commitment conversion protocol, a three-coin NOT protocol, a five-coin COPY protocol, a seven-coin AND protocol and a six-coin XOR protocol without hand operations (see Table 1): In Table 1, "hand-free" refers to a protocol without hand operations, "coin" refers to the number of coins in the protocol, "swap" refers to the number of swap shuffles (also known as random bisection cuts in card-based cryptography) in the protocol, and "flip" refers to the number of *random flips*, which is the most fundamental shuffle in coin-based protocols, in the protocol. Since the input and output formats of our protocols are the same, we can compute any function by composing our protocols. In addition, our protocols require random flips only, which are easy to implement physically.

In Komano-Mizuki model [5], [6], a commitment was defined as the state in which the player held the coin in his hand. In our definition, a commitment to $x \in \{0,1\}$ consists of two coins where the top one is called a dummy coin and the bottom one is face-up if $x = 1$ and face-down otherwise. Thus, there are two types of commitments depending on either the dummy coin is face-up or face-down. They are converted to each other by our commitment conversion protocol.

Table 1 Comparison of existing and proposed protocols.

	hand-free	coin	swap	rflip
◦ Commitment conversion protocol				
Ours (Section3)	✓	4	0	0
◦ NOT protocol				
Ours (Section4)	✓	3	0	0
◦ COPY protocol				
Komano-Mizuki [6]		5	0	1
Ours (Section5)	✓	5	0	1
◦ AND protocol				
Komano-Mizuki [6]		6	1	0
Ours (Section6)	✓	7	0	1
◦ XOR protocol				
Komano-Mizuki [6]		6	1	0
Ours (Section7)	✓	6	0	2

Manuscript received July 12, 2023.
Manuscript revised November 7, 2023.
Manuscript publicized December 13, 2023.

[†]Ibaraki University, Hitachi-shi, 316-8511 Japan.
[††]National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, 135-0064 Japan.
a) E-mail: kazumasa.shinagawa.np92@vc.ibaraki.ac.jp

DOI: 10.1587/transfun.2023EAP1082

2. Preliminaries

2.1 Coin-Based Protocols

In this paper, we denote a face-down and face-up of a coin by • and ∘, respectively. We use the encoding as follows:

$$\bullet = 0, \quad \circ = 1.$$

For two coins $a, b \in \{\bullet, \circ\}$, a stacked on top of b is denoted as ab. For example, •∘ represents a stack of a face-down coin on top of a face-up coin. For $c_1, c_2, \ldots, c_k \in \{\bullet, \circ\}$, a stack of $c_k, c_{k-1}, \ldots, c_1$ from the bottom is denoted by

$$c_1 c_2 \cdots c_k$$

and this is called a k**-coin bundle**. For a positive integer k, we denote the set of all k-coin bundles by $\mathsf{B}^k = \{\bullet, \circ\}^k$, and the set of all i-coin bundles for any $i \leq k$ by

$$\mathsf{B}^{\leq k} = \{\epsilon\} \cup \bigcup_{i=1}^{k} \mathsf{B}^k$$

where the symbol ϵ denotes the **empty coin bundle**, i.e., the 0-coin bundle. For example, $\mathsf{B}^{\leq 2} = \{\epsilon, \bullet, \circ, \bullet\bullet, \bullet\circ, \circ\bullet, \circ\circ\}$. A sequence of coin bundles is called a *coin sequence*, and the set of all coin sequences is denoted as S. We define the set of k-coin sequences by

$$\mathsf{S}^k = \left\{ (b_1, b_2, \ldots, b_\ell) \in \mathsf{S} \;\middle|\; \sum_{i=1}^{\ell} \mathsf{size}(b_i) = k \right\}$$

where $\mathsf{size}(b_i)$ is the number of coins in b_i, defined by $\mathsf{size}(b_i) = n$ if $b_i \in \mathsf{B}^n$ and $\mathsf{size}(\epsilon) = 0$. An empty coin bundle at the end of the coin sequence can be omitted.

For a coin bundle $b \in \mathsf{B}^k$, $\mathsf{top}(b)$, $\mathsf{bottom}(b)$, and $\mathsf{turn}(b)$ denote the top of the coin bundle b, the bottom of the coin bundle b, and the flipped bundle c, respectively. For a bundle of k coins $c_1 c_2 \cdots c_k \in \mathsf{B}^k$, we have

$$\begin{aligned}
&\mathsf{top}(c_1 c_2 \cdots c_k) = c_1,\\
&\mathsf{bottom}(c_1 c_2 \cdots c_k) = \overline{c_k}, \text{ and}\\
&\mathsf{turn}(c_1 c_2 \cdots c_k) = \overline{c_k} \cdots \overline{c_2}\,\overline{c_1},
\end{aligned}$$

where $\overline{c}$ denotes the flipped coin of $c \in \{\bullet, \circ\}$. When a face-down coin • (resp. a face-up coin ∘) is flipped, it becomes ∘ (resp. •). For the empty coin bundle ϵ, we define

$$\mathsf{top}(\epsilon) = \mathsf{bottom}(\epsilon) = \mathsf{turn}(\epsilon) = \epsilon.$$

For a bit $a \in \{0, 1\}$, a **commitment** to a is a 2-coin bundle •a or ∘a, i.e., a coin bundle of a dummy coin on top of the coin $a \in \{\bullet, \circ\}$. A commitment with a face-down dummy coin • is called a **black commitment** and a commitment with a face-up dummy coin ∘ is called a **white commitment**. For a coin sequence $\Gamma = (b_1, b_2, \ldots, b_n) \in \mathsf{S}^k$, we define visible sequence as $\mathsf{top}(\Gamma) = (\mathsf{top}(b_1), \mathsf{top}(b_2), \ldots, \mathsf{top}(b_n))$. For example, the visible sequence for the coin sequence $\Gamma = (\bullet\circ, \circ\bullet, \circ\bullet, \bullet\circ)$ is $\mathsf{top}(\Gamma) = (\bullet, \circ, \circ, \bullet)$. The set of all visible sequences of k-coin sequences is defined by $\mathsf{Vis}^k = \{\mathsf{top}(\Gamma) \mid \Gamma \in \mathsf{S}^k\}$.

A coin-based protocol P is defined by a four-tuple $P = (k, U, Q, A)$ as in the model of card-based protocols [11]. Here, k is the number of coins used in the protocol, $U \subseteq \mathsf{S}^k$ is the set of input coin sequences, Q is the set of states including the initial state $q_0 \in Q$ and final state $q_f \in Q$, $A : (Q - \{q_f\}) \times \mathsf{Vis}^k \to Q \times \mathsf{Action}$ is the action function. We define the set of possible actions in the following.

We can observe that it is enough to have $(k+1)$ positions on the table to deal with k coins. This implies that any k-coin sequence can be naturally identified with $(k+1)$ coin bundles by inserting empty coin bundles. Let $\Gamma = (b_1, b_2, b_3, \ldots, b_{k+1}) \in \mathsf{S}^k$ be the current coin sequence. In this paper, we use a set of actions as follows:

Move (move, n, i, j)**:** Here, $1 \leq n \leq \mathsf{size}(b_i), i, j \in \{1, 2, \ldots, k+1\}, i \neq j$. The top n coins of the coin bundle b_i is moved to the top of the coin bundle b_j. For a coin sequence $\Gamma = (\ldots, b_i^0 b_i^1, \ldots, b_j, \ldots)$ with $\mathsf{size}(b_i^0) = n$, this operation results in the coin sequence $\Gamma' = (\ldots, b_i^1, \ldots, b_i^0 b_j, \ldots)$.

Flip (flip, i)**:** Here, $i \in \{1, 2, \ldots, k+1\}$. The coin bundle b_i is flipped. For a coin sequence $\Gamma = (\ldots, b_i, \ldots)$, this operation results in the coin sequence $\Gamma' = (\ldots, \mathsf{turn}(b_i), \ldots)$.

Random flip (rflip, i)**:** Here, $i \in \{1, 2, \ldots, k+1\}$. The coin bundle b_i is flipped with probability $1/2$. For a coin sequence $\Gamma = (\ldots, b_i, \ldots)$, this operation results in Γ with probability $1/2$ and the coin sequence $\Gamma' = (\ldots, \mathsf{turn}(b_i), \ldots)$ with probability $1/2$.

2.2 Extended Diagrams

In this paper, we use extended diagrams [8] to show the correctness and security of a protocol, which is based on the idea of KWH tree [4] and used in coin-based protocols [5], [6] (see [5], [6], [8] for the detail of extended diagrams). See Fig. 1 for an example of an extended diagram. Each node represents a state of the protocol, and an arrow represents a transition by an action from a current state to the next state. In particular, the root node represents an initial state and the protocol starts with it. A branching occurs when a move operation results in two possible visible sequences. When the protocol reaches a leaf node, it terminates and outputs a coin bundle as output.

Each node has three columns: the left column represents the visible sequence of the current coin sequence, the middle column has possible coin sequences in the current state, and the right column has the probability traces, each corresponding to the coin sequence in the middle column. For the case of two-bit input, a probability trace for a coin sequence is denoted by a four-tuple $(a_{00}, a_{01}, a_{10}, a_{11})$, where a_{xy} denotes the conditional probability that the input is $(x, y) \in \{0, 1\}^2$ conditioned on the coin sequence.

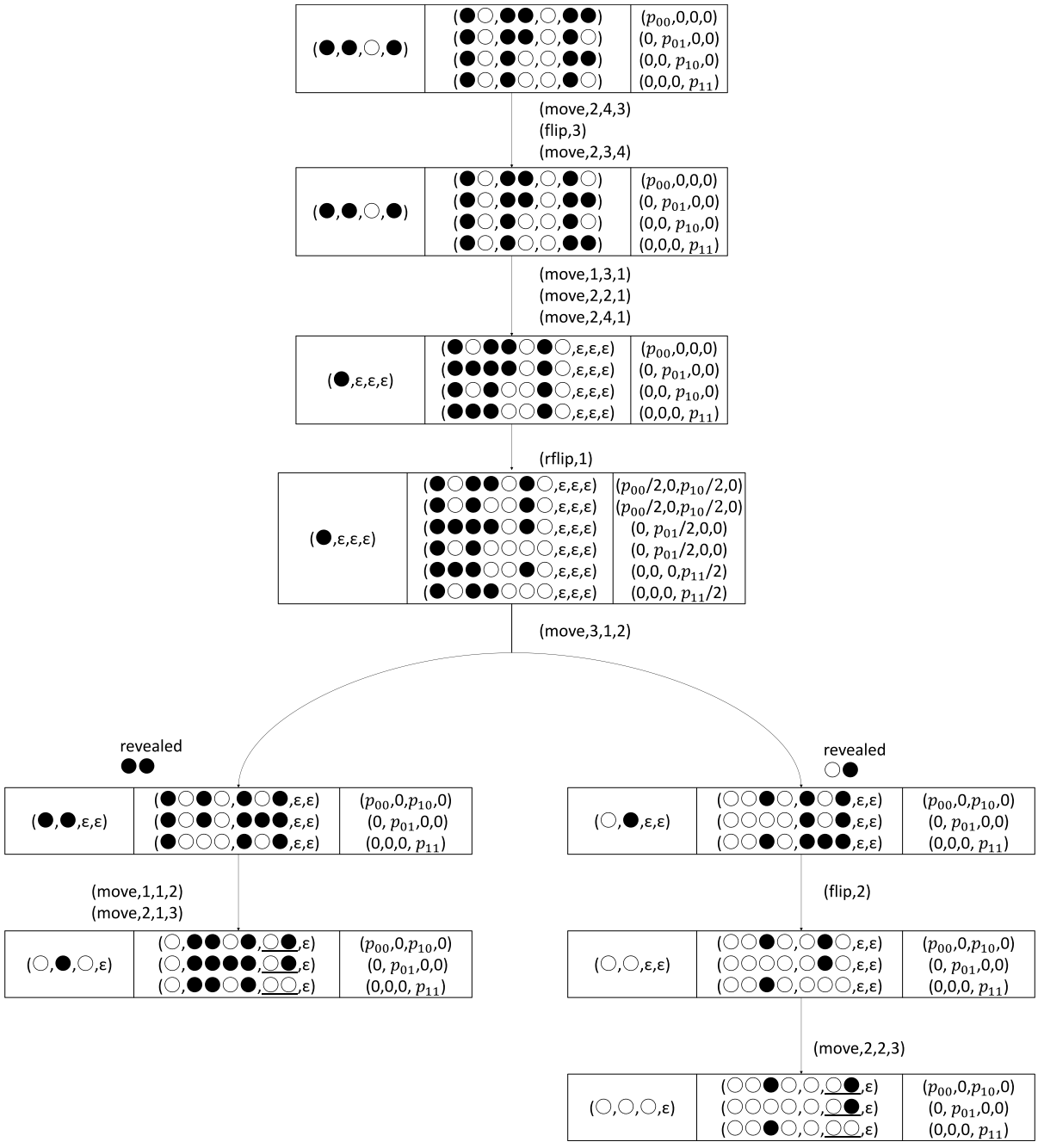

Fig. 1 The KWH tree of our AND protocol (the output position of each leaf node is underlined).

Since we use p_{00}, p_{01}, p_{10}, and p_{11} to denote the probability that the input is $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ respectively, the initial probability trace is either $(p_{00},0,0,0)$, $(0,p_{01},0,0)$, $(0,0,p_{10},0)$, or $(0,0,0,p_{11})$.

The correctness of a protocol can be checked by the fact that every leaf node has a coin bundle corresponding to the output in a fixed position. The security of a protocol can be checked by the fact that, for each node, the sum of all probability traces is $(p_{00},p_{01},p_{10},p_{11})$, which means that it leaks no input information.

3. NOT Protocol

This section gives a three-coin NOT protocol. The input for this protocol can be either a black or a white commitment. It can output any color (black or white) of commitment depending on the coin placed in the first coin bundle at the beginning of the protocol. The following is the protocol procedure when both input and output are black commitments.

1. Arrange the coins as follows:

$$(\circ,\bullet a).$$

Note that the output can be made a white commitment by setting the coin placed in the first coin bundle to •.

2. (move, 2, 2, 1): Move the second coin bundle to the first coin bundle as follows:

$$(\bullet a \circ, \epsilon).$$

3. (flip, 1): Flip the first coin bundle as follows:

$$(\bullet \overline{a} \circ, \epsilon).$$

4. (move, 2, 1, 2): Move upper two coins of the first coin bundle to the second position as follows:

$$(\circ, \bullet \overline{a}).$$

Output the second coin bundle as the resulting commitment and terminate the protocol.

Correctness follows from the description of the protocol. Security follows from the fact that the input coin (the coin of a) has never appeared on the top of the coin bundle.

We note that an n-input NOT protocol can be executed with a single flip operation by combining n commitments into a single bundle.

4. Commitment Conversion Protocol

This section gives a commitment conversion protocol, which converts a black commitment $\bullet a$ into a white commitment $\circ a$. To convert a white commitment to a black commitment, we can initially set the coin in the third coin bundle to ∘.

1. Arrange the coins as follows:

$$(\bullet, \bullet a, \bullet).$$

Note that when the input is a white commitment, the coin to be placed in the third coin bundle is ∘.

2. Apply the NOT protocol (from a black commitment to a white commitment) to the first and second coin bundles as follows:

$$(\circ, \circ \overline{a}, \bullet).$$

3. Apply the NOT protocol (from a white commitment to a white commitment) to the second and third coin bundles as follows:

$$(\circ, \circ a, \bullet).$$

The second coin bundle is the output commitment of the protocol.

Correctness follows from the description of the protocol. Security follows from the fact that the input coin (the coin of a) has never appeared on the top of the coin bundle.

We note that an n-input commitment conversion protocol can be also constructed by the n-input NOT protocol.

5. COPY Protocol

This section gives a five-coin COPY protocol, which takes a commitment as input and outputs two copies of the input commitment. This protocol is based on the Komano-Mizuki COPY protocol. Recall • = 0 and ∘ = 1. We note that our protocol is applicable to white input commitments by replacing • with ∘ and ∘ with • in the initial state.

1. Arrange the coins as follows:

$$(0 \circ, \bullet 0 a, \epsilon).$$

2. (move, 3, 2, 1): Move the second coin bundle to the first coin bundle as follows:

$$(\bullet 0 a 0 \circ, \epsilon, \epsilon).$$

3. (rflip, 1): Apply a random flip to the first coin bundle as follows:

$$(\bullet 0 a 0 \circ, \epsilon, \epsilon) \longrightarrow (\bullet r a' r \circ, \epsilon, \epsilon),$$

where $r \in \{0, 1\}$ is a uniformly random bit chosen by the random flip and $a' = a \oplus r$.

4. (move, 2, 1, 2): Move the top two coins of the first coin bundle to the second position as follows:

$$(a' r \circ, \bullet r, \epsilon).$$

If $a' = \bullet$, go to Step 5. Otherwise, go to Step 6.

5. (move, 2, 1, 3): Move upper two coins of the first coin bundle to the third position as follows:

$$(\circ, \bullet r, \bullet r).$$

Output the second and third coin bundles as the copy result and terminate the protocol. (Note that $a = r$ in this case.)

6. (move, 2, 2, 1): Move the second coin bundle to the first coin bundle as follows:

$$(\bullet r \circ r \circ, \epsilon, \epsilon).$$

7. (flip, 1): Flip the first coin bundle as follows:

$$(\bullet \overline{r} \bullet \overline{r} \circ, \epsilon, \epsilon).$$

8. (move, 2, 1, 2): Move upper two coins of the first coin bundle to the second position as follows:

$$(\bullet \overline{r} \circ, \bullet \overline{r}, \epsilon).$$

9. (move, 2, 1, 3): Move upper two coins of the first coin bundle to the third position as follows:

$$(\circ, \bullet \overline{r}, \bullet \overline{r}).$$

Output the second and third coin bundles as the copy result and terminate the protocol. (Note that $a = \overline{r}$ in this case.)

Correctness follows from the description of the protocol. Security follows from the fact that $a' = a \oplus r$ distributes uniformly at random since r is a uniformly random bit.

6. AND Protocol

This section gives a seven-coin AND protocol, which takes $\bullet a$ and $\bullet b$ as input, and outputs $\circ(a \wedge b)$. We note that our protocol is applicable to white input commitments by replacing $\bullet$ with $\circ$ and $\circ$ with $\bullet$ in the initial state.

1. Arrange the coins as follows:
$$(0\circ, \bullet a, \circ, \bullet b).$$
2. Apply the NOT protocol to the third and fourth coin bundles as follows:
$$(0\circ, \bullet a, \circ, \bullet \overline{b}).$$
3. $(\mathsf{move}, 1, 3, 1)$: Move the third coin bundle to the first coin bundle as follows:
$$(\circ 0\circ, \bullet a, \epsilon, \bullet \overline{b}).$$
4. $(\mathsf{move}, 2, 2, 1)$: Move the second coin bundle to the first coin bundle as follows:
$$(\bullet a \circ 0\circ, \epsilon, \epsilon, \bullet \overline{b}).$$
5. $(\mathsf{move}, 2, 4, 1)$: Move the fourth coin bundle to the first coin bundle as follows:
$$(\bullet \overline{b} \bullet a \circ 0\circ, \epsilon, \epsilon, \epsilon).$$
6. $(\mathsf{rflip}, 1)$: Apply a random flip to the first coin bundle as follows:
$$(\bullet \overline{b} \bullet a \circ 0\circ, \epsilon, \epsilon, \epsilon) \longrightarrow (\bullet c \bullet a' \circ d\circ, \epsilon, \epsilon, \epsilon),$$
where $r \in \{0, 1\}$ is a uniformly random bit chosen by the random flip, $a' = a \oplus r$, and $(c, d) = (\overline{b}, 0)$ if $r = 0$ and $(c, d) = (1, b)$ if $r = 1$.
7. $(\mathsf{move}, 3, 1, 2)$: Move upper three coins of the first coin bundle to the second position as follows:
$$(a' \circ d\circ, \bullet c\bullet, \epsilon, \epsilon).$$
If $a' = \bullet$, go to Step 8. Otherwise, go to Step 10.
8. $(\mathsf{move}, 1, 1, 2)$: Move the upper coin of the first coin bundle to the second coin bundle as follows:
$$(\circ d\circ, \bullet \bullet c\bullet, \epsilon, \epsilon).$$
9. $(\mathsf{move}, 2, 1, 3)$: Move upper two coins of the first coin bundle to the third position as follows:
$$(\circ, \bullet \bullet c\bullet, \circ d, \epsilon).$$
Output the third coin bundle as the resulting white commitment and terminate the protocol.
10. $(\mathsf{flip}, 2)$: Flip the second coin bundle as follows:
$$(\circ \circ d\circ, \circ \overline{c}\circ, \epsilon, \epsilon).$$
11. $(\mathsf{move}, 2, 2, 3)$: Move upper two coins of the second coin bundle to the third position as follows:
$$(\circ \circ d\circ, \circ, \circ \overline{c}, \epsilon).$$
Output the third coin bundle as the resulting white commitment and terminate the protocol.

See Fig. 1 for the extended diagram of our AND protocol. The correctness follows from the fact that all leaf nodes have the output commitment $\circ(a \wedge b)$ at the third position. The security follows from the fact that, for each node, the sum of the probability traces is $(p_{00}, p_{01}, p_{10}, p_{11})$, which means that it leaks no input information.

Some readers familiar with card-based cryptography will notice the similarities between our protocol and the Mizuki-Sone's AND protocol [12]. We note that our protocol has more steps compared to Mizuki-Sone's AND protocol due to the move and flip operations. If successive move/flip operations can be considered as one step, our protocol can be executed with almost the same number of steps as Mizuki-Sone's AND protocol.

7. XOR Protocol

This section gives a six-coin XOR protocol, which takes $\bullet a$ and $\bullet b$ as input, and outputs $\bullet(a \oplus b)$. We note that our protocol is applicable to white input commitments by replacing $\bullet$ with $\circ$ and $\circ$ with $\bullet$ in the initial state.

1. Arrange the coins as follows:
$$(\circ, \circ, \bullet a, \bullet b).$$
2. $(\mathsf{move}, 2, 3, 1)$: Move the third coin bundle to the first coin bundle as follows:
$$(\bullet a\circ, \circ, \epsilon, \bullet b).$$
3. $(\mathsf{move}, 2, 4, 2)$: Move the fourth coin bundle to the second coin bundle as follows:
$$(\bullet a\circ, \bullet b\circ, \epsilon, \epsilon).$$
4. $(\mathsf{move}, 3, 2, 1)$: Move the second coin bundle to the first coin bundle as follows:
$$(\bullet b \circ \bullet a\circ, \epsilon, \epsilon, \epsilon).$$
5. $(\mathsf{rflip}, 1)$: Apply a random flip to the first coin bundle as follows:
$$(\bullet b \circ \bullet a\circ, \epsilon, \epsilon, \epsilon) \longrightarrow (\bullet c \circ \bullet d\circ, \epsilon, \epsilon, \epsilon),$$
where $(c, d) \in \{(b, a), (\overline{a}, \overline{b})\}$.
6. $(\mathsf{move}, 3, 1, 2)$: Move upper three coins of the first coin bundle to the second position as follows:
$$(\bullet d\circ, \bullet c\circ, \epsilon, \epsilon).$$
7. $(\mathsf{flip}, 2)$: Flip the second coin bundle as follows:
$$(\bullet d\circ, \bullet \overline{c}\circ, \epsilon, \epsilon).$$

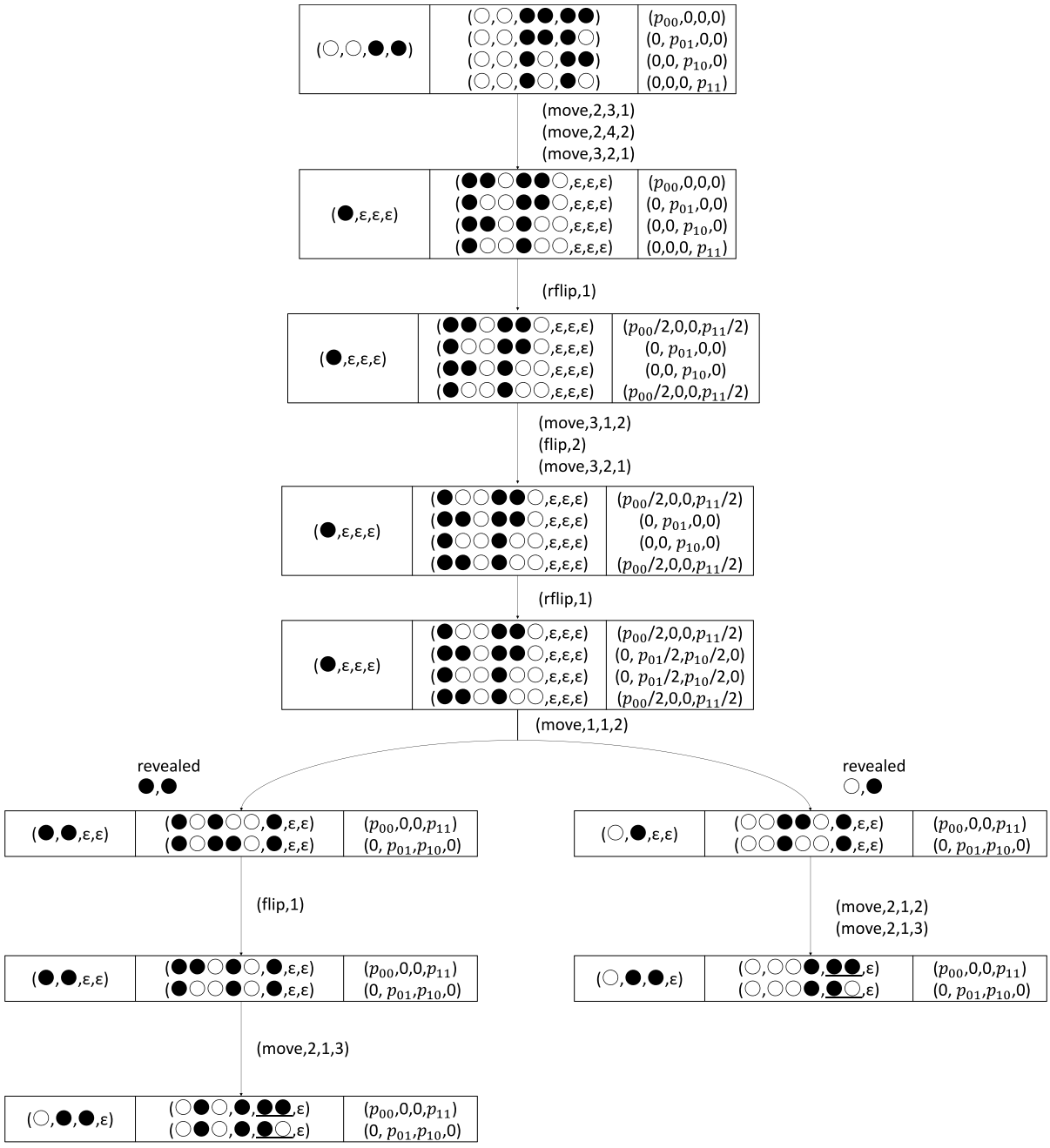

Fig. 2 The KWH tree of our XOR protocol (the output position of each leaf node is underlined).

8. (move, 3, 2, 1): Move the second coin bundle to the first coin bundle as follows:

$$(\bullet\,\overline{c}\circ\bullet\,d\circ,\epsilon,\epsilon,\epsilon).$$

9. (rflip, 1): Apply a random flip to the first coin bundle as follows:

$$(\bullet\,\overline{c}\circ\bullet\,d\circ,\epsilon,\epsilon,\epsilon)\longrightarrow(\bullet\,e\circ\bullet\,f\circ,\epsilon,\epsilon,\epsilon),$$

where $(e,f)\in\{(\overline{c},d),(\overline{d},c)\}$.

10. (move, 1, 1, 2): Move the upper coin of the first coin bundle to the second position as follows:

$$(e\circ\bullet\,f\circ,\bullet,\epsilon,\epsilon).$$

If $e=\bullet$, go to Step 11. Otherwise, go to Step 13.

11. (flip, 1): Flip the first coin bundle as follows:

$$(\bullet\overline{f}\circ\bullet\circ,\bullet,\epsilon,\epsilon).$$

12. (move, 2, 1, 3): Move upper two coins of the first coin bundle to the third position as follows:

$$(\circ\bullet\circ,\bullet,\bullet\overline{f},\epsilon).$$

Output the third coin bundle as the resulting black commitment and terminate the protocol.

13. $(\mathsf{move}, 2, 1, 2)$: Move upper two coins of the first coin bundle to the second coin bundle as follows:

$$(\bullet\ f\ \circ, \circ\ \circ\ \bullet, \epsilon, \epsilon).$$

14. $(\mathsf{move}, 2, 1, 3)$: Move upper two coins of the first coin bundle to the third position as follows:

$$(\circ, \circ\ \circ\ \bullet, \bullet f, \epsilon).$$

Output the third coin bundle as the resulting black commitment and terminate the protocol.

See Fig. 2 for the extended diagram of our XOR protocol. The correctness follows from the fact that all leaf nodes have the output commitment $\bullet(a \oplus b)$ at the third position. The security follows from the fact that, for each node, the sum of the probability traces is $(p_{00}, p_{01}, p_{10}, p_{11})$, which means that it leaks no input information.

We note that compared to the card-based XOR protocol [12], our coin-based XOR protocol seems to be more complex. In fact, our protocol requires two random flips, while the card-based XOR protocol requires a random bisection cut. We left as an open problem to find out whether a coin-based XOR protocol with one random flip can be constructed or not.

8. Conclusion

In this paper, we have proposed several coin-based cryptographic protocols without hand operations: a three-coin NOT protocol, a five-coin COPY protocol, a seven-coin AND protocol, a six-coin XOR protocol, and a commitment conversion protocol. Since our protocols are committed-format protocols, we can compute any function by combining these protocols. An important open problem is to reduce the number of coins and the number of steps in these protocols. We also left as an open problem to find a non-trivial relationship between coin-based protocols and other physical cryptographic protocols such as card-based protocols.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP21K17702 and JP23H00479, and JST CREST Grant Number JPMJCR22M1.

References

[1] C. Crépeau and J. Kilian, "Discreet solitary games," Advances in Cryptology - CRYPTO'93, 13th Annual International Cryptology Conference, Proceedings, Santa Barbara, California, USA, pp.319–330, 1993.

[2] B. den Boer, "More efficient match-making and satisfiability: The five card trick," Advances in Cryptology - EUROCRYPT'89, Workshop on the Theory and Application of of Cryptographic Techniques, Proceedings, Houthalen, Belgium, pp.208–217, 1989.

[3] G. Hanaoka, M. Iwamoto, Y. Watanabe, T. Mizuki, Y. Abe, K. Shinagawa, M. Arai, and N. Yanai, "Physical and visual cryptography to accelerate social implementation of advanced cryptographic technologies," IEICE Trans. Fundamentals, (Japanese Edition), vol.J106-A, no.8, pp.214–228, Aug. 2023.

[4] A. Koch, S. Walzer, and K. Härtel, "Card-based cryptographic protocols using a minimal number of cards," Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part I, T. Iwata and J.H. Cheon, eds., Auckland, New Zealand, vol.9452 of Lecture Notes in Computer Science, pp.783–807, Springer, 2015.

[5] Y. Komano and T. Mizuki, "Multi-party computation based on physical coins," Theory and Practice of Natural Computing - 7th International Conference, TPNC 2018, Proceedings, D. Fagan, C. Martín-Vide, M. O'Neill, and M.A. Vega-Rodríguez, eds., Dublin, Ireland, vol.11324 of Lecture Notes in Computer Science, pp.87–98, Springer, 2018.

[6] Y. Komano and T. Mizuki, "Coin-based secure computations," Int. J. Inf. Sec., vol.21, no.4, pp.833–846, 2022.

[7] D. Miyahara, Y. Komano, T. Mizuki, and H. Sone, "Cooking cryptographers: Secure multiparty computation based on balls and bags," 34th IEEE Computer Security Foundations Symposium, CSF 2021, Dubrovnik, Croatia, pp.1–16, IEEE, 2021.

[8] T. Mizuki and Y. Komano, "Analysis of information leakage due to operative errors in card-based protocols," Combinatorial Algorithms - 29th International Workshop, IWOCA 2018, Proceedings, C.S. Iliopoulos, H.W. Leong, and W.-K. Sung, eds., Singapore, vol.10979 of Lecture Notes in Computer Science, pp.250–262, Springer, 2018.

[9] T. Mizuki, Y. Kugimoto, and H. Sone, "Secure multiparty computations using a dial lock," Theory and Applications of Models of Computation, 4th International Conference, TAMC 2007, Proceedings, J. Cai, S.B. Cooper, and H. Zhu, eds., Shanghai, China, vol.4484 of Lecture Notes in Computer Science, pp.499–510, Springer, 2007.

[10] T. Mizuki, Y. Kugimoto, and H. Sone, "Secure multiparty computations using the 15 puzzle," Combinatorial Optimization and Applications, First International Conference, COCOA 2007, Proceedings, A.W.M. Dress, Y. Xu, and B. Zhu, eds., Xi'an, China, vol.4616 of Lecture Notes in Computer Science, pp.255–266, Springer, 2007.

[11] T. Mizuki and H. Shizuya, "A formalization of card-based cryptographic protocols via abstract machine," Int. J. Inf. Secur., vol.13, no.1, pp.15–23, 2014.

[12] T. Mizuki and H. Sone, "Six-card secure AND and four-card secure XOR," Frontiers in Algorithmics, X. Deng, J.E. Hopcroft, and J. Xue, eds., vol.5598 of LNCS, pp.358–369, Springer, Berlin, Heidelberg, 2009.

[13] A.C.-C. Yao, "Protocols for secure computations (extended abstract)," 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, pp.160–164, IEEE Computer Society, 1982.

[14] A.C.-C. Yao, "How to generate and exchange secrets (extended abstract)," 27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, pp.162–167, IEEE Computer Society, 1986.

Yuta Minamikawa received the B.E. degree from Ibaraki University in 2023. He is currently a master student in Ibaraki University.

Kazumasa Shinagawa received the B.E. and M.S. degrees from University of Tsukuba in 2015 and 2017, respectively, and the Ph.D. degree from Tokyo Institute of Technology in 2020. He is currently an assistant professor at Ibaraki University from 2021, and also serves as Collaborative Researcher at AIST. He is a member of IEICE, IPSJ, and IACR.