Secure Voting Protocol Using Balance Scale

Shohei Kaneko[1], Pascal Lafourcade[2], Lola-Baie Mallordy[2], Daiki Miyahara[1,3], Maxime Puys[2], and Kazuo Sakiyama[1]

[1] The University of Electro-Communications, Tokyo, Japan
[2] Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Etienne, LIMOS, 63000 Clermont-Ferrand, France
[3] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract. Voting has been one of the most widely used cryptographic protocols. The use of ballot boxes allows us to ensure voter privacy and to ensure the accuracy of the voting process. This method effectively hides individual votes, but inherently reveals vote counts during the tallying process. In this study, we present an advanced cryptographic protocol that employs the use of a physical balance scale to compare votes, while ensuring the confidentiality of individual vote counts. The proposed protocol not only addresses this challenge but also allows the voting of multiple candidates. We discuss the efficiency of our proposed protocol in comparison to other voting protocols that use physical objects and show that our protocol is efficient despite achieving the secrecy of the number of votes.

Keywords: Secure multiparty computation · Voting · Balance scale.

1 Introduction

Voting is one of the most prevalent and widely recognized examples of a cryptographic protocol utilizing physical instruments. A traditional voting entails voters inserting ballots bearing their chosen candidates into a ballot box, which is then shuffled to conceal the individual votes while only the tally is calculated. This method is a widely used and highly effective means of preserving voter privacy and ensuring fair election outcomes. Nevertheless, devising a protocol that permits vote tallying while simultaneously concealing the vote counts themselves, solely through the use of physical tools, represents a significant challenge. In traditional methods that employ a ballot box, the tallying process inherently reveals the vote counts, necessitating the implementation of additional mechanisms to maintain the confidentiality of these counts during the computation. In a small community, disclosing the vote counts can lead to the risk of individuals' voting choices being inferred, potentially compromising the fairness of the voting process. Our approach helps protect voter privacy and ensures that the integrity of the election is preserved without unfair influence or pressure on voters.

Card-based protocols using a deck of physical cards enable us to perform secure multiparty computations. A number of research efforts have tackled a secure

Table 1: Voting protocols using physical objects

Object	#Candidates	#Votes	Reference
Cards	Two	Open	[19]
Cards	Any	Open	[29]
Cards	Two	Hide	[23]
Cards	Two	Hide	[2]
Cards	Two	Hide	[3]
Cards	Any	Open	[30]
Balance	**Any**	**Hide**	**Ours**

voting protocol [3, 19, 23, 29, 30]; for example, Abe et al. [3] in 2023 proposed a card-based majority voting protocol that computes the majority over input bits without revealing anything. However, these protocols have a limitation in that the number of candidates is limited to two.[4]

1.1 Contribution

In this study, we address a secure voting protocol and present the one for multiple candidates. For this, we employ the physical property of a *balance scale* and facilitate the comparison of votes while maintaining the confidentiality of the individual vote counts. Our proposed protocol does not reveal any information other than necessary, such as the rankings of the candidates. That is, it outputs only the candidates with the highest number of votes and keeps the rankings of the other candidates secret. The proposed protocol can also be applied to new research directions, such as developing an auction protocol using everyday objects (cf., [8]).

Table 1 demonstrates a comparison between our proposed protocol and the existing ones. From this table, one can observe that our proposed protocol is the first one that is constructed for any number of candidates and for hiding the number of votes. It should be noted that the existing protocols for multiple candidates [29, 30] do not focus on the secrecy of the number of votes, and these protocols could be extended to achieve it using a general protocol that computes an arbitrary function. However, it implies that additional costs could be introduced for computing. In Sect. 4, we discuss the efficiency of our proposed protocol in comparison to card-based protocols and show that our protocol is efficient despite achieving the secrecy of the number of votes.

1.2 Related Work

Research on implementing cryptographic functions using everyday physical objects has garnered attention due to its potential for educating the general public,

[4] It should be noted that the first voting protocol [19] requires a logarithmic number of ballots (cards) to conduct a secure voting. There are also general card-based protocols [24, 25] that compute any Boolean function over multiple inputs.

who has no specialized knowledge in security. Representative studies in this field include card-based cryptography [5, 6, 12, 20], as well as the use of objects such as balls in bags [17], coins [13, 16], and a PEZ dispenser [1, 4, 21]. These studies focus on implementing cryptographic functions such as secure computations [7, 15, 22, 31] and zero-knowledge proofs for pencil puzzles [9, 10, 14, 18, 26–28, 32]. Very recently, Kaneko et al. [11] proposed balance-based zero-knowledge proof protocols for puzzles such as Sudoku. Their protocols employ the physical property of a balance scale to achieve the security requirements. In this study, we extend the use of a balance scale to construct a voting protocol for multiple candidates.

1.3 Outline

This paper is organized as follows. In Sect. 2, we formulate a balance and coins to be used in this voting protocol. In Sect. 3, we present our voting protocols and prove their correctness and security. In Sect. 4, we discuss the efficiency of our protocols and their application to auction. Section 5 concludes the paper.

2 Preliminaries

In this section, we define the function of voting and present our computational model to use a balance scale in cryptographic protocols.

2.1 Voting

Assume there are k candidates, $2 \leq k$, denoted as C_i, $1 \leq i \leq k$, and n voters, $2 \leq n$, denoted as P_i, $1 \leq i \leq n$. Each voter P_i has a private value $x_i \in \{1, 2, \ldots, k\}$, where $x_i = j$ indicates that the voter P_i votes for a candidate C_j. To calculate the number of votes each candidate C_j receives, we use an indicator function $v(x_i, j)$, defined as follows:

$$v(x_i, j) = \begin{cases} 1 & \text{if } x_i = j, \\ 0 & \text{otherwise.} \end{cases}$$

The vote count s_j for each candidate C_j is computed as follows: $s_j = \sum_{i=1}^{n} v(x_i, j)$. The winner is determined with the maximum vote count $s_{\max}$ among all candidates, i.e., $s_{\max} = \max\{s_1, s_2, \ldots, s_k\}$. A set of winners W is defined as those candidates whose vote count equals $s_{\max}$, i.e., $W = \{C_j \mid s_j = s_{\max}, 1 \leq j \leq k\}$.

In this paper, we say that a voting protocol is *correct* if it always outputs a set of winners W. A voting protocol is *secure* if it does not reveal any information beyond W.

2.2 Model

We assume an ideal balance that compares coins placed on both sides of the plates and outputs either which side is heavier or even. That is, it tilts at a constant angle toward the heavier side regardless of the weight difference. We use coins whose weight cannot be distinguished by looking at them. Such a single coin is denoted by ◯, and a stack of coins is denoted by ◯))). We represent a non-integer by its weight, and we omit the unit for simplicity.

Our protocols use two operations on coins and a balance scale: *compare* and *shuffle*.

Compare: A comparison of two stacks of coins using a balance is represented as follows: ◯))) | ◯))). This operation outputs either left or right depending on which of the two stacks is heavier. If the weights are the same, it outputs even.

Shuffle: A shuffling is denoted by $[\cdot|\ldots|\cdot]$ and acts on several stacks of coins as follows: [◯)))|◯)))|$\cdots$|◯)))] → ◯))) ◯))) $\cdots$ ◯))). That is, the order of the stacks becomes randomized, but the order of coins within each stack remains unchanged.

In our balance-based protocol, we choose an action depending on the result of a comparison using a balance scale. Therefore, the security of our balance-based protocol means that the result of a comparison is stochastically independent to information beyond a set of winners, i.e., W.

3 Voting Protocols

In this section, we present balance-based voting protocols. We first construct the one for 2-candidate and extend it to have the one for k-candidate, $k > 2$.

3.1 Voting Protocol for 2-Candidate

We present a balance-based voting protocol for two candidates, C_1 and C_2, where n voters P_i, $1 \leq i \leq n$, cast their votes.

1. Each voter P_i places one coin ◯ in front of each candidate C_1 and C_2. Specifically, the voter places ◯ of weight 2 in front of the candidate they vote for and ◯ of weight 1 in front of the other candidate. Each P_i stacks their coin on top of the previous voter's coins, and all voters sequentially place their coins in front of the candidates.
2. After all the voters have placed their coins, the coins corresponding to each candidate are compared using a balance scale. That is, the stack of coins for candidate C_1 (resp. C_2) is placed on the left (resp. right) plate of the balance scale: ◯))) | ◯))).
   - If the comparison results in left, then C_1 is declared the winner.
   - If it results in right, then C_2 is declared the winner.
   - If it results in even, then the result is a tie, and both candidates are considered winners.

Efficiency: This protocol is executed using $2n$ coins, distributed as two coins per voter for n voters, and requires only one comparison.

Correctness and Security: This voting protocol is straightforward, and the correctness and security are clear.

3.2 Voting Protocol for *k*-Candidate

We extend the previous protocol to one for k candidates, $k > 2$. Voters cast their votes for k candidates, and the candidate with the highest number of votes is declared the winner. A simple approach to extending the voting protocol is to repeatedly apply the 2-candidate protocol to determine the winner. However, this simple extension has the limitation of potentially revealing the rankings of all candidates, thereby compromising the security of the voting results. To address this issue and prevent the leakage of candidate rankings, our protocol introduces a novel mechanism for managing and concealing vote counts.

Idea: To protect against the leakage of candidate rankings, our protocol involves a shuffling procedure for a stack of coins ◯))) placed in front of each candidate. That is, we shuffle a sequence of k stacks of coins and then compare them one by one to reveal the winner. However, if there are ties among non-winning candidates, the balance scale may output `even` during comparisons, potentially leaking information about intermediate rankings. To avoid this, our protocol assigns a coin of specific weight (less than 1) to each candidate. That is, before the shuffling, such a coin of predetermined weight is placed on top of each stack of coins. Since the weights of such coins are unique to each candidate and are less than 1, the balance scale does not output `even` even if there are ties, and the winner is correctly determined. Moreover, this approach allows us to later identify the candidate associated with each stack based on the weight of such a coin. By using these weighted coins, we can determine which candidate received the most votes, without revealing the intermediate counts or rankings of the other candidates.

Procedure: The voting procedure is as follows.

1. Each candidate C_j is assigned a coin ◯ of weight w_j, where w_j is a distinct weight assigned to candidate C_j such that $0 < w_j < 1$. The specific weights are assigned to ensure that each candidate has a unique weight, and these weights are known to all participants in the protocol.
2. Each voter V_i places one coin ◯ in front of each candidate C_j, following the same procedure as in the one for two candidates (Sect. 3.1). That is, V_i places a coin of weight 2 in front of the candidate they votes for and places a coin of weight 1 in front of each of the other candidates.
3. Each candidate C_j places its assigned coin ◯ on top of its respective stack of coins ◯))). Now we have the k stacks of the $n + 1$ coins each:

$$\underbrace{\bigcirc\!)))}_{n+1}\quad\underbrace{\bigcirc\!)))}_{n+1}\quad\cdots\quad\underbrace{\bigcirc\!)))}_{n+1}.$$

4. All the k stacks are collected and shuffled as follows:

   [◯ | ◯ | $\cdots$ | ◯] $\rightarrow$ ◯ ◯ $\cdots$ ◯.

   The resulting sequence of stacks are denoted as $c_i =$ ◯, $1 \leq i \leq k$.
5. Each pair of stacks is compared and sorted in descending order as follows:

   $$c_i \mid c_j \quad \text{for all } i \text{ and } j \text{ such that } 1 \leq i < j \leq k.$$

   Note that the result of these comparisons does not leak any information because the correspondence between the k stacks and the k candidates is lost due to the shuffling action applied in the previous step.
6. The top coins are removed from the two heaviest stacks, and the resulting stacks are compared: ◯ | ◯.
   - If the comparison results in either left or right, it means that the heavier stack is the heaviest among the k stacks of coins, and the weight of the top coin from the heavier stack implies the winner, because in step 1 the top coin is assigned to a unique candidate.
   - If the comparison results in even, it means that at least the two stacks are the heaviest among the k stacks. Therefore, the protocol should determine the number of ties from the top. That is, the top coin is removed from the next heaviest stack, and their weights are compared: ◯ | ◯. This process is repeated until the balance outputs no longer even. When the balance finally tilts, the weight of the top coins from the stacks that were previously equal to the heaviest stack is revealed, and all candidates corresponding to these coins are declared winners.

Efficiency: This protocol uses $k(n+1)$ coins, one shuffle, and $\binom{k}{2} + |W|$ comparisons, where $|W|$ denotes the number of winners. The number of comparisons in step 5 is $\binom{k}{2}$ because it performs comparisons for all possible pairs of stacks. The number of comparisons in step 6 is $|W|$ to determine the number of winners. By applying a fast sorting algorithm such as the quick sort during the sorting process in step 5, the number of comparisons can be reduced to $\mathcal{O}(k \log k)$.

3.3 Correctness and Security

We prove the correctness and security of our voting protocol for k-candidate presented in Sect. 3.2.

Lemma 1 (Correctness). *The protocol always outputs the candidates who receive the most votes.*

Proof. In the protocol, the result of the voting process is determined by a comparison of the total weight of the coins placed in front of each candidate. This process ensures that the total weight of the coins is an accurate reflection of the number of votes each candidate received. After the stacks of coins are shuffled in step 4, a comparison process is initiated, whereby the candidate whose stack is the heaviest is identified as the winner. Because the coins placed in step 2 have a weight of either 1 or 2, when comparing the weights of the stacks without placing the coin assigned to the candidate on top, if the balance tilts, it means that the difference in weight between the stacks is at least 1. The coin assigned to the candidate has a weight smaller than 1, and hence, placing it on top of the stack will not change the result. Because there is a possibility of a tie for first place, step 6 involves checking if there are any stacks with the same weight as the first-place stack. □

Lemma 2 (Security). *No information other than a set of winners W is leaked.*

Proof. Note that our balance-based protocol is executed in public. Since the coins placed are identical in appearance, it is impossible to identify which candidate received a particular vote, thus preserving the secrecy of the vote. Before the stacks are compared, coins weighing less than weight 1 are assigned to each candidate, which are used to identify the candidates after the final comparison. This information is only revealed after the final comparison, ensuring that the rankings of the candidates and the number of votes they received remain confidential.

In step 4, the stacks of coins are shuffled, and hence, results of comparisons performed in step 5, i.e., the numbers of left and right the balance outputs, are completely independent to the votes, meaning that no information is revealed. Note that in step 5, the balance does not output even due to the coin assigned to each candidate in step 1. In step 6, the number of even the balance outputs is equal to $|W| - 1$ because it repeats to compare two stacks from the first place. Therefore, the results of comparisons performed in the protocol is either independent to the votes or equal to $|W|$, meaning that no information beyond W is leaked. □

4 Discussion

We discuss our balance-based voting protocols. We first discuss the efficiency and the security of our protocols. We then compare our protocols with other voting protocols using physical objects and show the application to auction.

4.1 Efficiency

Number of Types of Coins: When there are two candidates, the protocol can be executed using two types of coins: one weighing 1 and one weighing 2. For three or more candidates, in addition to the two types of coins, additional coin types are assigned to each candidate, increasing the number of coin types according to the number of candidates to ensure that the balance scale does not become balanced. However, as the number of candidates increases and the weight differences between assigned coins become smaller, the balance scale might become balanced due to its precision. In such cases, this problem can be resolved by increasing the weight of the coins used for voting, making the coins assigned to candidates heavier.

Table 2: The complexity of our balance-based protocol and the existing card-based ones using t-sided polygon cards [30], where n is the number of voters and k is the number of candidates.

Balance-Based			Card-Based	
#Coins	#Shuffles	#Comparisons	#Cards	#Shuffles
$k(n+1)$	1	$2k-2$	$k(\lceil \frac{n+1}{t} \rceil + n + 1)$	$n+1$

Number of Comparisons: Our proposed protocol for k-candidate sorts the stacks of coins in step 5 to determine the winner. However, there is no need to sort the stacks; instead, the winner can be determined simply by finding the stack with the maximum weight. That is, after shuffling, we start by comparing two stacks chosen randomly and then repeatedly compare the heavier stack with the remaining untested stacks to identify the heaviest. With this method, the number of comparisons required in step 5 is reduced to $k-1$. Moreover, the second heaviest stack can be detected by comparing the heaviest stack with the other stacks to know a tie; the number of comparisons is $k-1$, totaling to $2k-2$.

4.2 Multiple Voting Avoided

In the protocol, a voter could place a coin of weight 2 in front of several candidates to vote multiple times. This cheating cannot be detected during the execution of the protocol because the appearance of the coins used is assumed to be identical. However, after the protocol ends, it can be detected as follows: we collect all the coins used for voting, shuffle them, and then check the weight of each coin individually. If even one of the coins is not equal to the coins of weights 1 and 2, or if the number of coins with weight 2 is not k, this indicates that cheating has occurred. In addition, this additional verification can also detect that a voter is using only coins of weight 1 and 2. This verification requires $\frac{3kn}{2}$ comparisons.

4.3 Comparison with Card-based Protocols

Table 2 shows the number of coins, shuffles, and comparisons used in our balance-based protocol. In this table we also show a very recent protocol presented by Takahashi and Shinagawa [30], which uses a deck of specialized cards to deal with integers for multiple candidates. Compared to this existing protocol, we observe that the number of coins used in our protocol is less than the number of cards used in the card-based protocol. Moreover, our balance-based protocol considers the *tie* problem, meaning that multiple winners with the same number of votes can be detected.

4.4 Application to Auction

Our voting protocol can be adapted for use in *sealed bid auction* because it can hide the amount of a bid. In this adaptation, each bidder place a single coin

◯ that represents their bid amount. That is, the lightest coin represents the lowest bid, and as the coins get heavier, the bid amount increases. The highest bid is represented by the heaviest coin. The bid amounts are scaled accordingly. In addition, each bidder places an assigned coin on top of it, and the winner is determined in the same way as in the proposed protocol. This adaptation also allows us to achieve second-price auctions; however, information about the number of ties for the first place is necessary to determine the second place in the use of a balance because we determine it from the top by comparison.

4.5 Limitations of Implementation

In our protocol, it is necessary to place the stacks of coins representing vote counts onto the balance plates in a single attempt without dividing them. However, due to the physical limitations of a real balance, there is a limit to the number of coins that can be placed simultaneously, which restricts the number of voters in practical implementations. More voters means more comparisons, as in step 5 described in Sect. 3.2. Moreover, as the number of voters increases, the weight difference between the top coins becomes smaller, given that the top coin must be prepared with a weight of $0 < w_j < 1$, as discussed in Sect. 4.1. This reduction in weight difference has the potential to impact the accuracy of comparisons, particularly if the sensitivity of the balance is not sufficient. To address this problem, we propose increasing the weight of each coin per vote. For example, whereas the current setup employs coins of 1 and 2 for voting, the coin weights could be doubled to 2 and 4, respectively.

5 Conclusion

In this paper, we introduced a secure voting protocol that uses a balance scale to compare votes while preserving the security of individual vote counts. This protocol ensure that only the candidates with the highest vote count is revealed. The protocol's security and correctness is proved. This work opens new avenues for the application of physical objects in cryptographic protocols, demonstrating the potential for further innovation in the use of a balance scale in cryptography.

Acknowledgments. We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Number JP23H00479, the ANR project MobiS5 (ANR-18-CE39-0019), the ANR project SEVERITAS (ANR-20-CE39-0009) and the ANR Project PRIVA-SIQ (ANR-23-CE39-0008).

References

1. Abe, Y., Iwamoto, M., Ohta, K.: Efficient private PEZ protocols for symmetric functions. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography. LNCS, vol. 11891, pp. 372–392. Springer, Cham (2019), https://doi.org/10.1007/978-3-030-36030-6_15

2. Abe, Y., Nakai, T., Kuroki, Y., Suzuki, S., Koga, Y., Watanabe, Y., Iwamoto, M., Ohta, K.: Efficient card-based majority voting protocols. New Gener. Comput. **40**, 173–198 (2022), https://doi.org/10.1007/s00354-022-00161-7
3. Abe, Y., Nakai, T., Watanabe, Y., Iwamoto, M., Ohta, K.: A computationally efficient card-based majority voting protocol with fewer cards in the private model. IEICE Trans. Fundam. **106**(3), 315–324 (2023), https://doi.org/10.1587/transfun.2022cip0021
4. Balogh, J., Csirik, J.A., Ishai, Y., Kushilevitz, E.: Private computation using a PEZ dispenser. Theor. Comput. Sci. **306**(1), 69–84 (2003), https://doi.org/10.1016/S0304-3975(03)00210-X
5. Boer, B.D.: More efficient match-making and satisfiability the five card trick. In: Quisquater, J.J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 208–217. Springer, Heidelberg (1990), https://doi.org/10.1007/3-540-46885-4_23
6. Crépeau, C., Kilian, J.: Discreet solitary games. In: Stinson, D.R. (ed.) Advances in Cryptology—CRYPTO' 93. LNCS, vol. 773, pp. 319–330. Springer, Berlin, Heidelberg (1994), https://doi.org/10.1007/3-540-48329-2_27
7. Doi, A., Ono, T., Abe, Y., Nakai, T., Shinagawa, K., Watanabe, Y., Nuida, K., Iwamoto, M.: Card-based protocols for private set intersection and union. New Gener. Comput. pp. 1–22 (2024), https://doi.org/10.1007/s00354-024-00268-z
8. Dreier, J., Jonker, H., Lafourcade, P.: Secure auctions without cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) Fun with Algorithms. LNCS, vol. 8496, pp. 158–170. Springer, Cham (2014), https://doi.org/10.1007/978-3-319-07890-8_14
9. Hand, S., Koch, A., Lafourcade, P., Miyahara, D., Robert, L.: Efficient card-based zkp for single loop condition and its application to Moon-or-Sun. New Gener. Comput. pp. 1–29 (2024), https://doi.org/10.1007/s00354-024-00275-0
10. Hatsugai, K., Ruangwises, S., Asano, K., Abe, Y.: NP-completeness and physical zero-knowledge proofs for Sumplete, a puzzle generated by ChatGPT. New Gener. Comput. pp. 1–20 (2024), https://doi.org/10.1007/s00354-024-00267-0
11. Kaneko, S., Lafourcade, P., Mallordy, L.B., Miyahara, D., Puys, M., Sakiyama, K.: Balance-based ZKP protocols for pencil-and-paper puzzles. In: Information Security Conference, Oct 2024, Washington DC, United States (2024)
12. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology—ASIACRYPT 2015. LNCS, vol. 9452, pp. 783–807. Springer, Berlin, Heidelberg (2015), https://doi.org/10.1007/978-3-662-48797-6_32
13. Komano, Y., Mizuki, T.: Coin-based secure computations. Int. J. Inf. Secur. **21**, 833–846 (2022), https://doi.org/10.1007/s10207-022-00585-8
14. Komano, Y., Mizuki, T.: Physical zero-knowledge proof protocols for Topswops and Botdrops. New Gener. Comput. pp. 1–30 (2024), https://doi.org/10.1007/s00354-024-00272-3
15. Manabe, Y., Ono, H.: Card-based cryptographic protocols with a standard deck of cards using private operations. New Gener. Comput. pp. 1–25 (2024), https://doi.org/10.1007/s00354-024-00257-2
16. Minamikawa, Y., Shinagawa, K.: Coin-based cryptographic protocols without hand operations. IEICE Trans. Fundamentals **E107.A**(8), 1178–1185 (2024), https://doi.org/10.1587/transfun.2023EAP1082

17. Miyahara, D., Komano, Y., Mizuki, T., Sone, H.: Cooking cryptographers: Secure multiparty computation based on balls and bags. In: Computer Security Foundations Symposium. pp. 389–404. IEEE, NY (2021), https://doi.org/10.1109/CSF51468.2021.00034
18. Miyahara, D., Robert, L., Lafourcade, P., Mizuki, T.: ZKP protocols for Usowan, Herugolf, and Five Cells. Tsinghua Science and Technology **29**(6), 1651–1666 (2024), https://doi.org/10.26599/TST.2023.9010153
19. Mizuki, T., Asiedu, I.K., Sone, H.: Voting with a logarithmic number of cards. In: Mauri, G., Dennunzio, A., Manzoni, L., Porreca, A.E. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 7956, pp. 162–173. Springer, Berlin, Heidelberg (2013), https://doi.org/10.1007/978-3-642-39074-6_16
20. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) Frontiers in Algorithmics. LNCS, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009), https://doi.org/10.1007/978-3-642-02270-8_36
21. Murata, S., Miyahara, D., Mizuki, T., Sone, H.: Public-PEZ cryptography. In: Susilo, W., Deng, R.H., Guo, F., Li, Y., Intan, R. (eds.) Information Security. LNCS, vol. 12472, pp. 59–74. Springer, Cham (2020), https://doi.org/10.1007/978-3-030-62974-8_4
22. Nakai, T., Iwanari, K., Ono, T., Abe, Y., Watanabe, Y., Iwamoto, M.: Card-based cryptography with a standard deck of cards, revisited: Efficient protocols in the private model. New Gener. Comput. pp. 1–14 (2024), https://doi.org/10.1007/s00354-024-00269-y
23. Nakai, T., Misawa, Y., Tokushige, Y., Iwamoto, M., Ohta, K.: Secure computation for threshold functions with physical cards: Power of private permutations. New Gener. Comput. **40**, 95–113 (2022), https://doi.org/10.1007/s00354-022-00153-7
24. Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Card-based protocols for any Boolean function. In: Jain, R., Jain, S., Stephan, F. (eds.) Theory and Applications of Models of Computation. LNCS, vol. 9076, pp. 110–121. Springer, Cham (2015), https://doi.org/10.1007/978-3-319-17142-5_11
25. Ono, H., Manabe, Y.: Card-based cryptographic logical computations using private operations. New Gener. Comput. **39**(1), 19–40 (2021), https://doi.org/10.1007/s00354-020-00113-z
26. Ruangwises, S.: Verifying the first nonzero term: Physical ZKPs for ABC End View, Goishi Hiroi, and Toichika. Journal of Combinatorial Optimization **47**(4), 69 (2024), https://doi.org/10.1007/s10878-024-01170-6
27. Ruangwises, S., Iwamoto, M.: Printing protocol: Physical ZKPs for decomposition puzzles. New Gener. Comput. pp. 1–13 (2024), https://doi.org/10.1007/s00354-024-00266-1
28. Sasaki, S., Shinagawa, K.: Physical zero-knowledge proof for Sukoro. New Gener. Comput. pp. 1–18 (2024), https://doi.org/10.1007/s00354-024-00271-4
29. Shinagawa, K., Mizuki, T., Schuldt, J., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Card-based protocols using regular polygon cards. IEICE Trans. Fundam. **E100.A**(9), 1900–1909 (2017), https://doi.org/10.1587/transfun.E100.A.1900
30. Takahashi, Y., Shinagawa, K.: Extended addition protocol and efficient voting protocols using regular polygon cards. New Gener. Comput. pp. 1–18 (2024), https://doi.org/10.1007/s00354-024-00275-0

31. Takahashi, Y., Shinagawa, K., Shikata, H., Mizuki, T.: Efficient card-based protocols for symmetric functions using four-colored decks. In: ACM ASIA Public-Key Cryptography Workshop. pp. 1–10. ACM, New York (2024), https://doi.org/10.1145/3659467.3659902

32. Tamura, Y., Suzuki, A., Mizuki, T.: Card-based zero-knowledge proof protocols for the 15-puzzle and the token swapping problem. In: ACM ASIA Public-Key Cryptography Workshop. pp. 11–22. ACM, New York (2024), https://doi.org/10.1145/3659467.3659905