How to Play Old Maid with Virtual Players[⋆]

Kazumasa Shinagawa[1,4], Daiki Miyahara[2,4], and Takaaki Mizuki[3,4]

[1] Ibaraki University, Ibaraki, Japan
kazumasa.shinagawa.np92[atmark]vc.ibaraki.ac.jp
[2] The University of Electro-Communications, Tokyo, Japan
[3] Tohoku University, Sendai, Japan
[4] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract. Old Maid is a popular card game. While typically played with three or more players, it is less enjoyable with only two people. To address this, we propose a protocol to create a virtual player, Carol, by making use of card-based cryptography when only two people, Alice and Bob, are available to play Old Maid. Specifically, we design a card-based protocol to remove any pair of cards having the same number in Carol's hand (namely, the virtual player's hand) without leaking any information about Carol's hand (more than necessary); our protocol uses additional cards aside from playing cards that are used in Old Maid. Using our protocol, without any third human player, Alice and Bob can have fun with Old Maid!

Keywords: Card-based cryptography · Cryptology · Old Maid · Card games.

1 Introduction

Old Maid is a card game that is popular all over the world. It is typically played with a standard deck of 53 playing cards, including one joker. The rule is roughly as follows:

- all players are dealt an (almost) equal number of cards;
- any pair of cards having the same number is removed from a hand;
- each player takes one card from the next player's hand in turn;
- the last player to have the joker loses.

Technically, Old Maid can be played by two or more players, but preferably three or more. This is because if only two people play the game, the player

[⋆] This paper appears in Proceedings of IJTCS-FAW 2024. This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/978-981-97-7752-5_4. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use: https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

holding the joker can be completely identified (by both the players), and the game tends to become monotonous and somewhat boring. On the other hand, if there are three or more players, it is not possible to immediately identify who has the joker, and the game tends to become more fluid and tactical.

1.1 What If Only Two Players Are Available?

So, what should we do if there are only two people who are eager to play Old Maid? One solution would be to create a *virtual* player just like virtual players in video games, so that we have three players to play the game. In this paper, we propose a method to play Old Maid as if there were three players, even when there are only two people.

Suppose that Alice and Bob want to create a virtual player, Carol, to play Old Maid. The difficulty in creating such a virtual player Carol is how to remove any pair of cards having the same number in Carol's hand. Of course, it would be easy to achieve this if Alice and Bob were allowed to look at Carol's hand; however, to keep the game fun, they are not allowed to know any information about Carol's hand. How would they be able to remove any pair in Carol's hand (without looking at her hand)?

1.2 Contribution

In this paper, we design a cryptographic protocol that removes any pairs in Carol's hand without revealing any information about Carol's hand (more than necessary); we call it a *removal protocol*. Our removal protocol is a so-called *card-based protocol* [1, 2, 15], which uses a deck of physical cards to perform a cryptographic task. Our protocol takes Alice's, Bob's, and Carol's hands as input and removes any pairs in Carol's hand using some "helping cards."

More specifically, our removal protocol is a generic one: it can remove any pairs in a virtual player's hand for the case where there are one or more virtual players along with any number of real human players. The protocol uses 108 "helping cards" in addition to 54 playing cards (in a standard deck) when Old Maid is played with a single standard deck of playing cards.

After providing preliminaries in Section 2, we explain the rule of Old Maid precisely and give a framework for execution of the game with virtual players in Section 3. Next, we present our removal protocol in Section 4. We conclude in Section 5 with future directions.

1.3 Related Work

Card-based cryptography is a research area for designing cryptographic protocols such as secure computation protocols using a deck of physical cards. It is shown that any function can be securely computed using physical cards [1,16]. Although the main research topic in card-based cryptography has been the design of secure computation protocols for Boolean functions $f : \{0,1\}^n \to \{0,1\}$, various other applications have been explored in recent years. A partial list of these studies is given below.

- **Zero-Knowledge Proof for Puzzles**: Gradwohl et al. [3] proposed a zero-knowledge proof protocol for a Sudoku puzzle using a deck of cards. This protocol enables us to convince someone that we know a solution of a Sudoku puzzle without revealing any information on the solution. Subsequent studies improved on its efficiency [20, 24, 26] and considered other puzzles, e.g., [5, 8, 19, 21–23].
- **No-Fixed Point Problem**: Crépeau and Kilian [1] proposed a protocol for generating a random permutation with no fixed point. This protocol can be used to exchange gifts without receiving a gift of one's own, e.g., in a Christmas party. Recent studies improved its efficiency [9, 10, 17] and gave a lower bound on the number of cards required [6].
- **Werewolf Game**: Hashimoto et al. [7] proposed a secure grouping protocol. This can be used to determine a character role without a moderator in the Werewolf game. Their study has the same motivation as our study in that it applies card-based cryptography to a game. However, even with their protocol, not all parts of the Werewolf game can be played without a moderator.
- **Covert Lottery**: Shinoda et al. [25] proposed a covert lottery protocol. For a two-player board game such as Chess or Shogi, this protocol can be used to determine who makes the first move according to the players' requests without revealing them.

In this paper, we design a protocol for the card game Old Maid. Although several protocols with a standard deck of playing cards were proposed [4, 11–14, 18, 20], they do not apply card-based cryptography to card games. Our work is of significant value in that it discovers a new application of card-based cryptography.

2 Preliminaries

In this section, we introduce the definitions of physical cards and a shuffling action, the "pile-scramble shuffle," which are necessary to describe our card-based protocols later; they follow the standard computation model of card-based protocols [15].

2.1 Cards

Our proposed protocol employs three types of cards: a standard deck of playing cards, numbered cards, and dummy cards.

A standard deck consists of 54 cards of a combination of 13 numbers and four suits along with two jokers, as follows:

$$\begin{gathered}
\boxed{\mathrm{A}^{\heartsuit}}\boxed{2^{\heartsuit}}\boxed{3^{\heartsuit}}\boxed{4^{\heartsuit}}\boxed{5^{\heartsuit}}\boxed{6^{\heartsuit}}\boxed{7^{\heartsuit}}\boxed{8^{\heartsuit}}\boxed{9^{\heartsuit}}\boxed{10^{\heartsuit}}\boxed{\mathrm{J}^{\heartsuit}}\boxed{\mathrm{Q}^{\heartsuit}}\boxed{\mathrm{K}^{\heartsuit}}\\
\boxed{\mathrm{A}^{\clubsuit}}\boxed{2^{\clubsuit}}\boxed{3^{\clubsuit}}\boxed{4^{\clubsuit}}\boxed{5^{\clubsuit}}\boxed{6^{\clubsuit}}\boxed{7^{\clubsuit}}\boxed{8^{\clubsuit}}\boxed{9^{\clubsuit}}\boxed{10^{\clubsuit}}\boxed{\mathrm{J}^{\clubsuit}}\boxed{\mathrm{Q}^{\clubsuit}}\boxed{\mathrm{K}^{\clubsuit}}\\
\boxed{\mathrm{A}^{\diamondsuit}}\boxed{2^{\diamondsuit}}\boxed{3^{\diamondsuit}}\boxed{4^{\diamondsuit}}\boxed{5^{\diamondsuit}}\boxed{6^{\diamondsuit}}\boxed{7^{\diamondsuit}}\boxed{8^{\diamondsuit}}\boxed{9^{\diamondsuit}}\boxed{10^{\diamondsuit}}\boxed{\mathrm{J}^{\diamondsuit}}\boxed{\mathrm{Q}^{\diamondsuit}}\boxed{\mathrm{K}^{\diamondsuit}}\\
\boxed{\mathrm{A}^{\spadesuit}}\boxed{2^{\spadesuit}}\boxed{3^{\spadesuit}}\boxed{4^{\spadesuit}}\boxed{5^{\spadesuit}}\boxed{6^{\spadesuit}}\boxed{7^{\spadesuit}}\boxed{8^{\spadesuit}}\boxed{9^{\spadesuit}}\boxed{10^{\spadesuit}}\boxed{\mathrm{J}^{\spadesuit}}\boxed{\mathrm{Q}^{\spadesuit}}\boxed{\mathrm{K}^{\spadesuit}}\boxed{\mathrm{Jo}}\boxed{\overline{\mathrm{Jo}}}\,.
\end{gathered}$$

Their backs are all identical and written as $\boxed{?}$.

The numbered cards have only numbers referred to as $\boxed{1}\boxed{2}\boxed{3}\cdots$. The cards having the same number are identical. The dummy cards have no symbol referred to as $\boxed{}$. The backs of the numbered cards and the dummy cards are all identical. We remark that the back of the standard deck and the back of the numbered cards may be different or the same. For simplicity, we assume that they are the same and written as $\boxed{?}$.

2.2 Pile-Scramble Shuffle

A shuffling action is operated on a sequence of cards, rearranging it randomly. Our proposed protocol employs the *pile-scramble shuffle* [10]. This divides a sequence into multiple piles of cards and rearranges them randomly.

Let ℓ be the number of cards and k be the number of piles (i.e., divisions) such that k is a divisor of ℓ. An ℓ-card k-pile pile-scramble shuffle is to divide a sequence of ℓ cards into k piles (each consisting of ℓ/k cards) and rearrange them completely randomly. There are $k!$ possibilities in total, but this shuffling is executed such that nobody can trace it. Note that when $k = 2$, this shuffle is called a *random bisection cut.*

For example, a six-card three-pile pile-scramble shuffle rearranges a sequence of cards as follows:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}} \rightarrow \begin{cases} \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}, \\ \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}, \\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\ \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}, \\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\ \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}, \\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\ \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}, \\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\ \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}. \end{cases}$$

The resulting sequence is chosen among the six sequences shown on the right side with a probability of 1/6. This pile-scramble shuffle is denoted as follows by surrounding each pile with $|\cdot|$

$$\left|\,\boxed{?}\boxed{?}\,\middle|\,\boxed{?}\boxed{?}\,\middle|\,\boxed{?}\boxed{?}\,\right|.$$

This notation is also used when a sequence of cards is represented as a matrix. For example, the following notation shows a 20-card five-pile pile-scramble shuffle:

$$\left|\begin{matrix}\boxed{?}\boxed{?}\\ \boxed{?}\boxed{?}\end{matrix}\middle|\begin{matrix}\boxed{?}\boxed{?}\\ \boxed{?}\boxed{?}\end{matrix}\middle|\begin{matrix}\boxed{?}\boxed{?}\\ \boxed{?}\boxed{?}\end{matrix}\middle|\begin{matrix}\boxed{?}\boxed{?}\\ \boxed{?}\boxed{?}\end{matrix}\middle|\begin{matrix}\boxed{?}\boxed{?}\\ \boxed{?}\boxed{?}\end{matrix}\right|.$$

3 Old Maid

In this section, we explain the rule of Old Maid, and present a framework for playing Old Maid with virtual players.

3.1 Rule of Old Maid

First, we explain the rule of Old Maid. It is played with a standard deck of 54 playing cards. In the game, the second joker $\boxed{\text{Jo}}$ is not used[5] and the number of required cards is 53. Suppose that there are m players $P_1, P_2, \dots, P_m$. Hereinafter, a *pair* in a hand means a pair of cards having the same number in the hand. The game proceeds as follows:

1. **Dealing phase:** All 53 cards are randomly dealt to the players; each player receives $\lfloor 53/m \rfloor$ or $\lceil 53/m \rceil$ cards.
2. **Initial phase:** Each player removes all pairs from his/her hands. (If some player has no cards at this point, he/she wins and is out of the game.)
3. **Playing phase:** From the first player P_1, each player P_i draws one card from the next player's hand in turn. If the drawn card makes a pair in the P_i's hand, P_i removes the pair. If P_i has no cards at this point, P_i wins and is out of the game.
4. The player who has the joker at the end loses.

3.2 Playing Old Maid with Virtual Players

In this subsection, we explain how to play Old Maid with virtual players, while the main protocol called a *removal protocol* will be introduced in the succeeding section.

Suppose that there are m_{r} real players and m_{v} virtual players. A typical case is $(m_{\mathsf{r}}, m_{\mathsf{v}}) = (2, 1)$: Two real players Alice and Bob want to play Old Maid with a virtual player Carol because two-player Old Maid is somewhat boring. Another typical case is $m_{\mathsf{r}} = 1$ and $m_{\mathsf{v}} \geq 1$: A real player Alice wants to play Old Maid in solitary. An extreme case is $m_{\mathsf{r}} = 0$ and $m_{\mathsf{v}} \geq 2$: This is a simulation of Old Maid with no real players.

Old Maid with virtual players proceeds as follows:

1. **Dealing phase:** All 53 cards are randomly dealt to the players; each player receives $\lfloor 53/(m_{\mathsf{r}} + m_{\mathsf{v}}) \rfloor$ or $\lceil 53/(m_{\mathsf{r}} + m_{\mathsf{v}}) \rceil$ cards.
2. **Initial phase:** Each real player removes all pairs from his/her hand. For each virtual player, execute a removal protocol, which removes all pairs from the virtual player's hand. (If some player has no cards at this point, he/she wins and is out of the game.)

[5] However, $\boxed{\text{Jo}}$ will be used in our removal protocol presented in Section 4.

3. **Playing phase:** Each player P_i draws a card from the next player's hand in turn. If the drawn card makes a pair in the P_i's hand, P_i removes the pair. When P_i is a virtual player, choose a random card from the next player's hand using a shuffle and then execute a removal protocol. If P_i has no cards at this point, P_i wins and is out of the game.
4. The player who has the joker at the end loses.

4 Removal Protocol

In this section, we construct a removal protocol, which is the main protocol for playing Old Maid with virtual players. In Section 4.1, we present the description of our protocol. In Sections 4.2 and 4.3, we give the efficiency evaluation and security proof of our protocol, respectively.

4.1 Protocol Description

Suppose that there are m players: P_1 is a virtual player whose hand will be examined to remove pairs by the protocol and $P_2, P_3, \ldots, P_m$ are real or virtual players. We call a numbered card $\boxed{i}$ an *i-card* hereinafter. Our removal protocol proceeds as follows.

1. Place all players' hands in a horizontal line as follows:

$$\underbrace{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}}_{P_1}\;\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{P_2}\;\underbrace{\boxed{?}\boxed{?}}_{P_3}\;\cdots\;\underbrace{\boxed{?}\boxed{?}\boxed{?}\boxed{?}}_{P_m}.$$

2. Place i-cards on the bottom of the P_i's cards $(1 \leq i \leq m)$ as follows:

$$\begin{array}{cccccc}
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?}\boxed{?}\boxed{?} \\
\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1} & \boxed{2}\boxed{2}\boxed{2} & \boxed{3}\boxed{3} & \cdots & \boxed{m}\boxed{m}\boxed{m}\boxed{m}.
\end{array}$$

3. Place 1-cards on the bottom of the P_1's cards and dummy cards ($\boxed{}$) on the bottom of the P_i's cards $(2 \leq i \leq m)$ as follows:

$$\begin{array}{cccccc}
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?}\boxed{?}\boxed{?} \\
\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1} & \boxed{2}\boxed{2}\boxed{2} & \boxed{3}\boxed{3} & \cdots & \boxed{m}\boxed{m}\boxed{m}\boxed{m} \\
\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1} & \boxed{}\boxed{}\boxed{} & \boxed{}\boxed{} & \cdots & \boxed{}\boxed{}\boxed{}\boxed{}.
\end{array}$$

4. Place three cards $\boxed{\overline{\text{Jo}}}\boxed{}\boxed{}$ next to the rightmost column as follows:

$$\begin{array}{cccccc}
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?}\boxed{?}\boxed{?} & \boxed{\overline{\text{Jo}}} \\
\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1} & \boxed{2}\boxed{2}\boxed{2} & \boxed{3}\boxed{3} & \cdots & \boxed{m}\boxed{m}\boxed{m}\boxed{m} & \boxed{} \\
\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1} & \boxed{}\boxed{}\boxed{} & \boxed{}\boxed{} & \cdots & \boxed{}\boxed{}\boxed{}\boxed{} & \boxed{}.
\end{array}$$

5. Turn over all face-up cards as follows:

$$\begin{array}{l}
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\\
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\\
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}.
\end{array}$$

6. Repeat the following procedure for $(\alpha, \beta, \gamma, \delta) = (\spadesuit, \heartsuit, \diamondsuit, \clubsuit)$, $(\spadesuit, \diamondsuit, \heartsuit, \clubsuit)$, and $(\spadesuit, \clubsuit, \heartsuit, \diamondsuit)$.
   (a) Apply a pile-scramble shuffle with each column as a pile as follows:

$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c}
\boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?}\\
\boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?}\\
\boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?}
\end{array}\right].$$

   (b) Turn over the cards in the first row as follows:

$$\begin{array}{l}
\boxed{8^\diamondsuit}\boxed{6^\heartsuit}\boxed{\mathrm{A}^\spadesuit}\boxed{\mathrm{Jo}}\boxed{2^\diamondsuit}\boxed{\mathrm{A}^\heartsuit}\boxed{\overline{\mathrm{Jo}}}\boxed{4^\clubsuit}\boxed{5^\clubsuit}\boxed{10^\spadesuit}\boxed{\mathrm{J}^\clubsuit}\boxed{3^\diamondsuit}\boxed{\mathrm{Q}^\clubsuit}\boxed{\mathrm{Q}^\heartsuit}\;\cdots\;\boxed{3^\spadesuit}\\
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\\
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\cdots\;\boxed{?}.
\end{array}$$

   (c) Rearrange the columns so that columns with the same numbers are consecutive in the order $\alpha \to \beta \to \gamma \to \delta$. An example case for $(\alpha, \beta, \gamma, \delta) = (\spadesuit, \heartsuit, \diamondsuit, \clubsuit)$ is given as follows:

$$\begin{array}{l}
\boxed{\mathrm{A}^\spadesuit}\boxed{\mathrm{A}^\heartsuit}\boxed{\mathrm{A}^\diamondsuit}\boxed{\mathrm{A}^\clubsuit}\;\boxed{2^\diamondsuit}\boxed{2^\clubsuit}\;\boxed{3^\spadesuit}\boxed{3^\heartsuit}\boxed{3^\diamondsuit}\boxed{3^\clubsuit}\;\cdots\;\boxed{\mathrm{Q}^\spadesuit}\boxed{\mathrm{Q}^\heartsuit}\boxed{\mathrm{Q}^\diamondsuit}\boxed{\mathrm{Q}^\clubsuit}\;\boxed{\mathrm{K}^\heartsuit}\boxed{\mathrm{K}^\clubsuit}\;\boxed{\mathrm{Jo}}\boxed{\overline{\mathrm{Jo}}}\\
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\\
\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}.
\end{array}$$

   In this example, each in $\{A, 3, \dots, Q\}$ has four cards and each in $\{2, \dots, K\}$ has two cards. We call the former a *two-pair number* and the latter a *one-pair number*. For each two-pair number, make two piles with (α, β) and (γ, δ) as follows:

$$\begin{array}{l}
\boxed{\mathrm{A}^\spadesuit}\boxed{\mathrm{A}^\heartsuit}\;\boxed{\mathrm{A}^\diamondsuit}\boxed{\mathrm{A}^\clubsuit}\;\boxed{2^\diamondsuit}\boxed{2^\clubsuit}\;\boxed{3^\spadesuit}\boxed{3^\heartsuit}\;\boxed{3^\diamondsuit}\boxed{3^\clubsuit}\;\cdots\;\boxed{\mathrm{Q}^\spadesuit}\boxed{\mathrm{Q}^\heartsuit}\;\boxed{\mathrm{Q}^\diamondsuit}\boxed{\mathrm{Q}^\clubsuit}\;\boxed{\mathrm{K}^\heartsuit}\boxed{\mathrm{K}^\clubsuit}\;\boxed{\mathrm{Jo}}\boxed{\overline{\mathrm{Jo}}}\\
\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\\
\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}.
\end{array}$$

   (d) Turn over all face-up cards as follows:

$$\begin{array}{l}
\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\\
\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\\
\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\cdots\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?}.
\end{array}$$

(e) Apply a pile-scramble shuffle with each pair of columns as a pile as follows:

$$\left[\begin{array}{c|c|c|c|c|c|c|c}
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} \\
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} \\
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?}
\end{array}\right].$$

(f) Apply a random bisection cut for each pair of columns as follows:

$$\left[\begin{array}{c|c}\boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?}\end{array}\right]
\left[\begin{array}{c|c}\boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?}\end{array}\right]
\left[\begin{array}{c|c}\boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?}\end{array}\right]
\left[\begin{array}{c|c}\boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?}\end{array}\right]
\cdots
\left[\begin{array}{c|c}\boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?}\end{array}\right].$$

(g) Turn over the cards in the bottom row. Suppose that there are piles having $\boxed{1}\boxed{1}$ (i.e., there are pairs in P_1's hand) as follows:

$$\begin{array}{cccccccccc}
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} \\
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} \\
\boxed{}\boxed{} & \boxed{1}\boxed{} & \boxed{1}\boxed{1} & \boxed{}\boxed{} & \boxed{}\boxed{1} & \cdots & \boxed{}\boxed{} & \boxed{}\boxed{} & \boxed{}\boxed{} & \boxed{1}\boxed{}
\end{array}.$$

Then, open the cards in these piles as follows:

$$\begin{array}{cccccccccc}
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{A^{\diamondsuit}}\boxed{A^{\clubsuit}} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} \\
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{1}\boxed{1} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} \\
\boxed{}\boxed{} & \boxed{1}\boxed{} & \boxed{1}\boxed{1} & \boxed{}\boxed{} & \boxed{}\boxed{1} & \cdots & \boxed{}\boxed{} & \boxed{}\boxed{} & \boxed{}\boxed{} & \boxed{1}\boxed{}
\end{array}.$$

Then, the top line of these piles are removed and the other cards (i.e., the 1-cards) on these piles are returned to free cards. If there are no piles having $\boxed{1}\boxed{1}$, do nothing.

(h) If $(\alpha, \beta, \gamma, \delta) = (\spadesuit, \heartsuit, \diamondsuit, \clubsuit)$ or $(\spadesuit, \diamondsuit, \heartsuit, \clubsuit)$, turn over all face-up cards as follows:

$$\begin{array}{ccccccccc}
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} \\
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} \\
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?}
\end{array}.$$

Otherwise, return the bottom row of cards to free cards as follows:

$$\begin{array}{ccccccccc}
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} \\
\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \boxed{?}\boxed{?}
\end{array}.$$

7. Apply a pile-scramble shuffle with each column as a pile as follows:

$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?} \\ \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?} \end{array}\right].$$

8. Turn over the cards in the bottom row as follows:

$$\begin{array}{l} \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ \boxed{4}\boxed{6}\boxed{3}\boxed{2}\boxed{m}\boxed{2}\boxed{1}\boxed{5}\boxed{1}\boxed{}\boxed{4}\boxed{1}\boxed{3}\boxed{m}\cdots\boxed{2}. \end{array}$$

Then, turn over the card above the dummy card as follows:

$$\begin{array}{l} \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{\text{Jo}}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ \boxed{4}\boxed{6}\boxed{3}\boxed{2}\boxed{m}\boxed{2}\boxed{1}\boxed{5}\boxed{1}\boxed{}\boxed{4}\boxed{1}\boxed{3}\boxed{m}\cdots\boxed{2}. \end{array}$$

Then, the joker and the dummy card are returned to free cards.

9. Rearrange the columns so that columns with the same numbered cards are next to each other as follows:

$$\begin{array}{ccccc} \boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?}\boxed{?} & \boxed{?}\boxed{?} & \cdots & \boxed{?}\boxed{?}\boxed{?}\boxed{?} \\ \boxed{1}\boxed{1}\boxed{1} & \boxed{2}\boxed{2}\boxed{2} & \boxed{3}\boxed{3} & \cdots & \boxed{m}\boxed{m}\boxed{m}\boxed{m}. \end{array}$$

Then, the cards above the i-cards are returned to the P_i's hand, and the cards on the bottom line are returned to free cards.

This is our removal protocol.

If the virtual player P_1 has pairs with suits $(\spadesuit, \heartsuit)$ or $(\diamondsuit, \clubsuit)$ at the beginning of the protocol, these pairs will be removed at Step 6g for the loop $(\alpha, \beta, \gamma, \delta) = (\spadesuit, \heartsuit, \diamondsuit, \clubsuit)$ because 1-cards were added to the bottom of the P_1's cards at Step 3. Similarly, if P_1 has pairs with suits $(\spadesuit, \diamondsuit)$ or $(\heartsuit, \clubsuit)$ (resp. $(\spadesuit, \clubsuit)$ or $(\heartsuit, \diamondsuit)$) at the beginning of the protocol, these pairs will be removed at Step 6g for the loop $(\alpha, \beta, \gamma, \delta) = (\spadesuit, \diamondsuit, \heartsuit, \clubsuit)$ (resp. $(\spadesuit, \clubsuit, \heartsuit, \diamondsuit)$). Therefore, all pairs in P_1's hand will be removed at the end of the protocol.

4.2 Efficiency

Let n_i $(1 \leq i \leq m)$ be the number of cards in P_i's hand. Let $n := \sum_{i=1}^{m} n_i$ be the total number of playing cards. In addition to the standard deck of 54 playing cards, we require $2n + 2$ helping cards, which consist of the following cards:

$$\underbrace{\boxed{1}\cdots\boxed{1}}_{2n_1}\ \underbrace{\boxed{2}\cdots\boxed{2}}_{n_2}\ \cdots\ \underbrace{\boxed{m}\cdots\boxed{m}}_{n_m}\ \underbrace{\boxed{}\cdots\boxed{}}_{n-n_1+2}.$$

Since $n \leq 53$, our removal protocol requires at most 108 helping cards.

For the number of shuffles, we require $3(\frac{n+1}{2} + 2) + 1$ shuffles, which consist of $\frac{3(n+1)}{2}$ random bisection cuts and 7 pile-scramble shuffles.

4.3 Security

Our protocol follows the computational model of card-based cryptography [15], in which the security of a card-based protocol is proven based on information theory. Thus, to prove the security of our protocol, it suffices to show that revealed cards during execution of the protocol do not leak any information beyond public information. In our proposed protocol, Steps 6b, 6g, and 8 reveal cards.

In Step 6b, the cards in the first row, i.e., all players' hands are revealed. Note that in Step 6a, a pile-scramble shuffle is applied to the first row, randomizing the order of the cards. Therefore, Step 6b reveals no additional information because all the removed cards are public information in a game of Old Maid.

In Step 6g, the cards in the bottom row are revealed. Remember that a revealed 1-card $\boxed{1}$ means that the top card in the same column comes from P_1; a dummy card $\boxed{}$ means that the top card in the same column comes from a player other than P_1. If a pair $\boxed{1}\boxed{1}$ appears, it is removed, and hence, the number of $\boxed{1}\boxed{1}$ is public information. Since the size of P_1's hand is public, the number of $\boxed{1}\boxed{1}$ immediately tells us the number of $\boxed{}\boxed{}$ as well as the number of either $\boxed{1}\boxed{}$ or $\boxed{}\boxed{1}$. Because Step 6e applies a pile-scramble shuffle with each pair of columns, the position of each pair is independent of the players' hands. Moreover, either $\boxed{1}\boxed{}$ or $\boxed{}\boxed{1}$ occurs with a probability of exactly 1/2 because of the application of a random bisection cut in Step 6f. Therefore, revealed cards do not leak any information beyond public information.

Finally, in Step 8, all the cards in the bottom row are revealed. Remember that an i-card means that the card above it comes from P_i. Because Step 7 applies a pile-scramble shuffle, the positions of each i-card and the dummy are independent of the players' hands.

5 Conclusion

In this paper, we showed that Alice and Bob can enjoy playing Old Maid with a virtual player Carol. Technically, we designed a removal protocol, which is a card-based protocol for removing any pairs in Carol's hand without revealing any information about Carol's hand.

Although card-based cryptography has sometimes been inspired by the card game techniques, the reverse direction (i.e., applying card-based cryptography to card games) had rarely been studied. In this paper, we posed a novel problem of simulating virtual players in a card game using card-based cryptographic techniques. We name this problem a *player simulation problem* for card games. We believe that the player simulation problem for card games is a new research area of fun with cryptography and it will attract the attention of many researchers as well as many non-specialists, like physical zero-knowledge proofs.

Acknowledgements

We would like to thank Tatsuya Sasaki for his cooperation in preparing a Japanese draft version at an earlier stage of this work. This work was supported in part by JSPS KAKENHI Grant Numbers JP21K17702, JP23H00479, and JP24K02938, and JST CREST Grant Number JPMJCR22M1.

References

1. 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
2. Den Boer, B.: 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
3. Gradwohl, R., Naor, M., Pinkas, B., Rothblum, G.N.: Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. In: Crescenzi, P., Prencipe, G., Pucci, G. (eds.) Fun with Algorithms. LNCS, vol. 4475, pp. 166–182. Springer, Berlin, Heidelberg (2007), https://doi.org/10.1007/978-3-540-72914-3_16
4. Haga, R., Hayashi, Y., Miyahara, D., Mizuki, T.: Card-minimal protocols for three-input functions with standard playing cards. In: Batina, L., Daemen, J. (eds.) AFRICACRYPT 2022. LNCS, vol. 13503, pp. 448–468. Springer, Cham (2022), https://doi.org/10.1007/978-3-031-17433-9_19
5. Hand, S., Koch, A., Lafourcade, P., Miyahara, D., Robert, L.: Check alternating patterns: A physical zero-knowledge proof for Moon-or-Sun. In: Shikata, J., Kuzuno, H. (eds.) Advances in Information and Computer Security. LNCS, vol. 14128, pp. 255–272. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-41326-1_14
6. Hashimoto, Y., Nuida, K., Shinagawa, K., Inamura, M., Hanaoka, G.: Toward finite-runtime card-based protocol for generating a hidden random permutation without fixed points. IEICE Trans. Fundam. **E101.A**(9), 1503–1511 (2018), https://doi.org/10.1587/transfun.E101.A.1503
7. Hashimoto, Y., Shinagawa, K., Nuida, K., Inamura, M., Hanaoka, G.: Secure grouping protocol using a deck of cards. IEICE Trans. Fundam. **E101.A**(9), 1512–1524 (2018), https://doi.org/10.1587/transfun.E101.A.1512
8. Hatsugai, K., Asano, K., Abe, Y.: A physical zero-knowledge proof for Sumplete, a puzzle generated by ChatGPT. In: Wu, W., Tong, G. (eds.) Computing and Combinatorics. LNCS, vol. 14422, pp. 398–410. Springer, Cham (2024), https://doi.org/10.1007/978-3-031-49190-0_29
9. Ibaraki, T., Manabe, Y.: A more efficient card-based protocol for generating a random permutation without fixed points. In: Mathematics and Computers in Sciences and in Industry (MCSI). pp. 252–257 (2016), https://doi.org/10.1109/MCSI.2016.054
10. Ishikawa, R., Chida, E., Mizuki, T.: Efficient card-based protocols for generating a hidden random permutation without fixed points. In: Calude, C.S., Dinneen, M.J. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 9252, pp. 215–226. Springer, Cham (2015), https://doi.org/10.1007/978-3-319-21819-9_16

11. Koch, A., Schrempp, M., Kirsten, M.: Card-based cryptography meets formal verification. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology–ASIACRYPT 2019. LNCS, vol. 11921, pp. 488–517. Springer, Cham (2019), https://doi.org/10.1007/978-3-030-34578-5_18
12. Koyama, H., Miyahara, D., Mizuki, T., Sone, H.: A secure three-input AND protocol with a standard deck of minimal cards. In: Santhanam, R., Musatov, D. (eds.) Computer Science – Theory and Applications. LNCS, vol. 12730, pp. 242–256. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-79416-3_14
13. Manabe, Y., Ono, H.: Card-based cryptographic protocols with a standard deck of cards using private operations. In: Cerone, A., Ölveczky, P.C. (eds.) Theoretical Aspects of Computing – ICTAC 2021. LNCS, vol. 12819, pp. 256–274. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-85315-0_15
14. Mizuki, T.: Efficient and secure multiparty computations using a standard deck of playing cards. In: Foresti, S., Persiano, G. (eds.) Cryptology and Network Security. LNCS, vol. 10052, pp. 484–499. Springer, Cham (2016), https://doi.org/10.1007/978-3-319-48965-0_29
15. Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. Int. J. Inf. Secur. **13**(1), 15–23 (2014), https://doi.org/10.1007/s10207-013-0219-4
16. 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
17. Murata, S., Miyahara, D., Mizuki, T., Sone, H.: Efficient generation of a card-based uniformly distributed random derangement. In: Uehara, R., Hong, S.H., Nandy, S.C. (eds.) WALCOM: Algorithms and Computation. LNCS, vol. 12635, pp. 78–89. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-68211-8_7
18. Niemi, V., Renvall, A.: Solitaire zero-knowledge. Fundam. Inf. **38**(1,2), 181–188 (1999), https://doi.org/10.3233/FI-1999-381214
19. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Physical ZKP protocols for Nurimisaki and Kurodoko. Theor. Comput. Sci. **972**, 114071 (2023), https://doi.org/10.1016/j.tcs.2023.114071
20. Ruangwises, S.: Two standard decks of playing cards are sufficient for a ZKP for Sudoku. New Gener. Comput. **40**, 49–65 (2022), https://doi.org/10.1007/s00354-021-00146-y
21. Ruangwises, S.: Physical zero-knowledge proof for ball sort puzzle. In: Della Vedova, G., Dundua, B., Lempp, S., Manea, F. (eds.) Unity of Logic and Computation. LNCS, vol. 13967, pp. 246–257. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-36978-0_20
22. Ruangwises, S.: Physical zero-knowledge proofs for Five Cells. In: Aly, A., Tibouchi, M. (eds.) Progress in Cryptology – LATINCRYPT 2023. LNCS, vol. 14168, pp. 315–330. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-44469-2_16
23. Ruangwises, S.: Physically verifying the first nonzero term in a sequence: Physical ZKPs for ABC end view and Goishi Hiroi. In: Li, M., Sun, X., Wu, X. (eds.) Frontiers of Algorithmics. LNCS, vol. 13933, pp. 171–183. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-39344-0_13
24. Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: Efficient card-based zero-knowledge proof for Sudoku. Theor. Comput. Sci. **839**, 135–142 (2020), https://doi.org/10.1016/j.tcs.2020.05.036

25. Shinoda, Y., Miyahara, D., Shinagawa, K., Mizuki, T., Sone, H.: Card-based covert lottery. In: Maimut, D., Oprina, A.G., Sauveron, D. (eds.) Innovative Security Solutions for Information Technology and Communications. LNCS, vol. 12596, pp. 257–270. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-69255-1_17
26. Tanaka, K., Mizuki, T.: Two UNO decks efficiently perform zero-knowledge proof for Sudoku. In: Fernau, H., Jansen, K. (eds.) Fundamentals of Computation Theory. LNCS, vol. 14292, pp. 406–420. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-43587-4_29