J Comb Optim (2010) 19: 501–530
DOI 10.1007/s10878-009-9252-7

Unconditional secure communication: a Russian Cards protocol

Zhenhua Duan · Chen Yang

Published online: 30 June 2009

Abstract This paper investigates Russian Cards problem for the purpose of unconditional secure communication. First, a picking rule and deleting rule as well as safe communication condition are given to deal with the problem with 3 players and 7 cards. Further, the problem is generalized to tackle n players and $n(n-1)+1$ cards. A new picking rule for constructing the announcement is presented, and a new deleting rule for players to determine each other's cards is formalized. Moreover, the safe communication condition is proved. In addition, to illustrate the approach, an example for 5 players and 21 cards is presented in detail.

Keywords Russian Cards problem · Picking rule · Deleting rule

1 Introduction

The security of cryptographic protocols generally depends upon several assumptions such as the agents are computationally limited and certain computational problems are intractable with these computational limits. In protocols based on public key encryption schemes such as RSA (Rivest et al. 1978), for example, decryption of messages is tractable for the intended recipient but assumed to be impossible for an intruder, because it requires factoring a large product of primes, a problem assumed to be intractable (Vasilenko 2006). There do exist, however, unconditionally secure protocols, whose security does not rely upon such assumptions. Some of such protocols

This research is supported by the NSFC Grant Nos. 60433010, and 60873018.

Z. Duan (✉) · C. Yang
Institute of Computing Theory and Technology, Xidian University, Xi'an 710071, China
e-mail: zhhduan@mail.xidian.edu.cn

C. Yang
e-mail: chyang@mail.xidian.edu.cn

have been studied recently in the cryptography and information theory community (Fische and Wright 1996; Makarychev 2001). These protocols can be shown to be secure even against the adversaries with unlimited computational powers, because they ensure that the adversary cannot learn secrets for information theory rather than computational reasons.

'Russian Cards' problem was originally presented at the Moscow Mathematics Olympiad in 2000 with 3 players and 7 cards. The problem can be described as follows (van Ditmarsch 2003): *From a pack of seven known cards two players each draw three cards and a third player gets the remaining card. How can the players with three cards openly* (*publicly*) *inform each other about their cards, without the third player learning from any of their cards who holds it?*

This type of problem has been studied in Fische and Wright (1996), Koichi et al. (2004), Ramanujam and Suresh (2001), Roehling (2005), Stiglic (2001) as the Key-set protocol for communication among a team of parties and a passive eavesdropper whose computational power is unlimited. Significantly, it is implied that a solution to this type of problem can communicate information among parties in a distributed setting securely without using any encryption. Although Russian Cards protocol is similar to Key-set protocol, parties in the latter only wish to share a common one-bit secret key while parties in the former want to know the actual card deal. Since such problems are often rather subtle, dynamic epistemic logic is used to deduce possible solutions to Russian Cards problem in van Ditmarsch (2003, 2005). This has been used as a basis for a model checking approach in van Ditmarsch et al. (2006). It has been proved that at least two announcements can solve this problem (Albert et al. 2005; Atkinson et al. 2007). Nevertheless, how to figure out these solutions are not given. Moreover, a generic safe communication protocol based on Russia Cards problem has not intensively been studied. Cyriac and Krishnan studied the Lower Bound for the Communication Complexity of the Russian Cards Problem (Cyriac and Krishnan 2008). They also pointed out that it is interesting to study the possible generalization of the problem to an n player m card game, and to derive a possible lower bound for announcement in this scenario. However, to the best of our knowledge, there is no published work on this generalization. Therefore, in this paper, we are motivated to generalize the Russian Cards problem so that more players can communicate using the unconditional protocol. The main contribution of the paper is as follows: (1) formalizing picking and deleting rules as well as safe communication condition with 3 players and 7 cards; (2) generalizing the problem to n players and $n(n-1)+1$ cards and further formalizing picking and deleting rules and safe communication condition.

To deal with the initial Russian Cards problem, we first formalize two algorithms called picking rule and deleting rule to construct announcements (a set of cards an announcer holds) and to manage communication among players. Moreover, a safe communication condition is proposed and proved. Further, we generalize the problem to n players and $n(n-1)+1$ cards so that a safe public communication protocol can be established. To do so, an assumed cards dealer randomly dispatches n cards from $n(n-1)+1$ cards to each party as his hand, and leave the remaining one card for intruder. In the communication with n $(n \geq 4)$ players, each party has to announce his hand by means of the announcement one by one. Each announcement is actually a matrix containing announcer's hand and other fake hands. After all of the announcements, all parties but intruder know each other's hand.

The rest of the paper is organized as follows. Section 2 models and analyzes the original Russia Cards problem. The picking rule and deleting rule as well as the safe communication condition are formalized. In Sect. 3, we generalize the problem to a generic case with n players and $n(n-1)+1$ cards. By the mathematical analysis, new picking and deleting rules are given. To prove the communication based on our approach is safe, some lemmas and two theorems are proved. In Sect. 4, to illustrate our approach, an example with 5 player and 21 cards is given in detail. Finally, the conclusion is drawn in Sect. 5.

2 Russian Cards problem

Initially, the cards were named $0, \ldots, 6$. Besides being public, all announcements are assumed to be truthful. According to game rules, when all cards are randomly allocated to each player, we call the set of cards held by a player a *hand*, the set of hands appearing in an announcement a *hand set*, and the allocation of cards a *card deal*. Further, for a hand of cards such as $\{0, 1, 2\}$, we write 0-1-2 instead; and for a card deal such as 0-1-2, 3-4-5, 6, we write 0-1-2.3-4-5.6 to mean that the first player holds cards $\{0, 1, 2\}$, the second holds $\{3, 4, 5\}$, and the third holds $\{6\}$. Suppose three players are Anne, Bill and Crow. We assume that 0-1-2.3-4-5.6 is the actual card deal. A solution to the Russian Cards problem is a sequence of secure announcements such that Anne and Bill know each other's hand but Crow knows nothing about the card deal. The following is an instance of solutions (van Ditmarsch 2003): *Anne says*: "*I have one of 0-1-2, 0-3-4, 0-5-6, 1-3-5, 2-4-6.*" *After which Bill announces*: "*Crow has card 6.*" Once Bill receives Anne's announcement, he learns Anne's hand from the hand of his own. Accordingly, after having received both announcements, Crow knows just Anne has one of 0-1-2, 0-3-4, 1-3-5, but neither which one nor the hand of Bill. This sequence is safe. However, if we replace Anne's announcement by "I have one of 0-1-2, 0-3-4, 0-5-6, 1-3-4, 2-5-6" and keep other conditions, the result is different since although both Anne and Bill can learn each other's hand, Crow is also able to figure out some information (for instance, he can work out Bill holds card 5) about their cards because he holds card 6. So this sequence is unsafe.

According to the rule of the game, three players draw cards in a random manner. We can equivalently assume that there is a cards dealer who randomly dispatches cards to these players. When all players obtain their hands, only the dealer knows the card deal. The dispatching process is straightforward. The cards dealer is also responsible for constructing announcement (hand set) for Anne according the card deal. Actually, the key information communicated among players is hand set generated by the cards dealer, since it not only notifies Bill of Anne's hand but also keeps Crow from knowing any information about hands of Anne and Bill. Observe that the hand set is generated in the way in which the first three hands cover all the cards with a sharing card (0 in the instance), and each of the remaining hands consists of three cards, coming from each of the first three hands excluding the sharing card. In this way, the cards dealer can figure out hand set. We call this procedure *picking rule*. Note that only the cards dealer knows picking rule.

Before we give the algorithm of picking rule, we first introduce a matrix called *hand matrix* $B = (b_{i,j})_{3\times 3}$,

$$B = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} \qquad B_1 = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 3 & 4 \\ 0 & 5 & 6 \end{pmatrix} \qquad B_2 = \begin{pmatrix} 3 & 0 & 4 \\ 3 & 1 & 5 \\ 3 & 2 & 6 \end{pmatrix}$$

for it can more concisely denote a hand set, where each row represents a hand, and each of the last two columns also represents a hand. Since each announcement is faithful, Anne's hand should be contained in her announcement. For example, hand matrix B_1 agrees with the hand set from the instance of solution. The matrix has Anne's hand placed in the first row. However, Anne's hand is not limited to appear in rows, it can also be contained in columns to make a qualified announcement. For instance, in hand matrix B_2, Anne's hand appears in the second column. For this reason, we consider picking rule in *row* and *column* cases respectively.

For *row case*, Anne's hand can appear in any row. Without loss of generality, we assume Anne's hand is placed in the first row of B. Thus, the picking rule can be given in Algorithm 2.1. Let $S = \{0, 1, 2, 3, 4, 5, 6\}$ denote the set of all cards, and h_a, h_b and h_c represent the hand of Anne, Bill and Crow respectively. Hand matrix $B = \text{Arpick}\ (S, h_a)$.

Algorithm 2.1: Arpick (picking rule (row case))

```
1  input: set M, h;
2  output: matrix B;
3  temp variable: integer m;
4  M := M − h
   /* extract a sharing card from h_a and fill the first
      column of B with it */
5  Let m ∈ h;
6  h := h − {m};
7  for i := 1 to 3 do
8      b_{i,1} := m;
9  end
10 Let m ∈ h; b_{1,2} := m; h := h − {m};
11 Let m ∈ h; b_{1,3} := m; h := h − {m};
12 for i := 2 to 3 do /* place cards in other columns of the
                         last two rows of B */
13     for j := 2 to 3 do
14         Let m ∈ M;
15         b_{i,j} := m;   M := M − {m};
16     end
17 end
```

Algorithm 2.2: Acpick (picking rule (column case))

```
1  input: set M, h;
2  output: matrix B;
3  temp variable: integer m;
4  M := M − h;
     /* apart from h_a, M contains the remaining cards */
5  Let m ∈ M; M := M − {m};
   /* extract a sharing card from M to fill the first
      column of B with it */
6  for i := 1 to 3 do
7      b_{i,1} := m;
8  end
9  for i := 1 to 3 do  /* place h_a in the second column of B */
10     Let m ∈ h; b_{i,2} := m; h := h − {m};
11 end
12 for i := 1 to 3 do
   /* place remaining three cards in last column of B */
13     Let m ∈ M; b_{i,3} := m; M := M − {m};
14 end
```

For *column case*, Anne's hand can appear in any but the first column of hand matrix. For simplicity, we assume Anne's hand is placed in the second column of B. So, the picking rule can be given in Algorithm 2.2.

The sequence of announcements, the first from Anne and the second from Bill, completes the communication procedure for the Russian Cards problem. So we call this sequence a *communication*. Note that, in a communication, once receiving the announcement from Anne, Bill and Crow can remove the hands containing cards in their own hands, from the hand set. We call this procedure *deleting rule*. Let R_i (resp. $[R_i]$) represent the ith row (resp. set of cards from R_i), and C_j (resp. $[C_j]$) the jth column (resp. set of cards from C_j) of matrix B. We use $H = \{[R_1], [R_2], [R_3], [C_2], [C_3]\}$ to denote the set of all possible hands for Anne. The rule can be formalized in Algorithm 2.3. Thus, Bill can use *deleting rule* to determine Anne's hand. $H_b = \text{Adel}\,(H, h_b)$. Note that if

$$|H_b| = 1 \tag{2.1}$$

then Bill learns Anne's hand. In deed, the above deleting rule can be used for Crow to determine Anne's hand as well. However, it is not enough to guarantee a safe communication since Crow might learn what cards Anne has not held from his announcement. For instance, suppose Anne's announcement is {0-1-2, 0-3-4, 0-5-6, 1-3-4, 2-5-6} and Bill still hold 3-4-5, by Algorithm 2.3, Bill learns Anne's hand is 0-1-2. Since both hands 0-5-6 and 2-5-6 contain card 6, by removing them from the hand set, Crow also learns Anne's hand is one of {0-1-2, 0-3-4, 1-3-4}. Although Crow cannot determine Anne's hand however he learns Anne does not hold card 5. This is not a safe communication. Therefore, to guarantee a safe communication, the

Algorithm 2.3: Adel (deleting rule)

```
input: matrix H, set h;
output: matrix H_b;
tempt variable: integer i, j;
Let H_b := H;
for i := 1 to 3 do
    if h ∩ [R_i] ≠ φ then
        H_b := H_b − {[R_i]};
    end
end
for j := 2 to 3 do
    if h ∩ [C_j] ≠ φ then
        H_b := H_b − {[C_j]};
    end
end
```

set of hands generated by using deleting rule must cover all cards except for Crow's card. As a result, $H_c = \text{Adel}(B, h_c)$.

Let H_s represent the set of cards extracted from members of H_c (set of all possible hands for Anne). It is easy to see that, if

$$|H_c| > 1 \tag{2.2}$$

and

$$h_c = S - H_s \tag{2.3}$$

then Crow learns nothing about hands of Anne and Bill. Thus, after Anne's announcement, Bill and Crow complete the communication by the deleting rule, Bill learns Anne's hand, while Crow knows neither Anne's nor Bill's hand. Also, Crow does not know what cards Anne and Bill do not hold. Based on the above analysis, the definition of safety for Russian Cards problem can be given as follow.

Definition 1 (Safe communication) *A communication with the first announcement generated according to picking rule is safe if formulas* (2.1), (2.2) *and* (2.3) *hold.*

Further, we have the following conclusion.

Theorem 1 *For the Russian Cards problem, a communication based on the hand matrix constructed by the picking rule is safe.*

Proof We consider the following two cases:

row case Since $h_a = [R_1]$, h_b contains three cards from four cards $b_{2,2}, b_{2,3}, b_{3,2}$, $b_{3,3}$, thus $h_b \cap [R_1] = \phi$, $h_b \cap [R_2] \neq \phi$ and $h_b \cap [R_3] \neq \phi$. According to picking rule, there are $C_4^3 = 4$ possible placements for h_b. With any cases,

either $|h_b \cap [R_2]| = 2$ or $|h_b \cap [R_3]| = 2$, both $h_b \cap [C_2] \neq \phi$ and $h_b \cap [C_3] \neq \phi$. Thus according to deleting rule, $H_b = \{[R_1], [R_2], [R_3], [C_2], [C_3]\} - \{[R_2], [R_3], [C_2], [C_3]\} = \{[R_1]\}$, so formula (2.1) holds. Further, Crow holds just one card h_c. It is one of $\{b_{2,2}\}, \{b_{2,3}\}$, $\{b_{3,2}\}$ or $\{b_{3,3}\}$. We assume, without loss of generality, $h_c = \{b_{2,3}\}$. According to the deleting rule, since $h_c \cap [R_2] \neq \phi$ and $h_c \cap [C_3] \neq \phi$, we have $H_c = \{[R_1], [R_2], [R_3], [C_2], [C_3]\} - \{[R_2], [C_3]\} = \{[R_1], [R_3], [C_2]\}$. Thus $H_s = \{b_{1,1}, b_{1,2}, b_{1,3}, b_{2,2}, b_{3,2}, b_{3,3}\} = S - h_c$. So formulas (2.2) and (2.3) hold.

column case: the proof is similar to *row case*. □

Back to the instance of solutions, Anne's hand 0-1-2 is one of the first three hands. So, Bill's hand contains three cards out of 3, 4, 5 and 6. There are $C_4^3 = 4$ possible hands 3-4-5, 3-4-6, 3-5-6 or 4-5-6 for Bill. Suppose his hand is 3-4-5. According to the deleting rule, Bill can remove four hands 0-3-4, 1-3-5, 2-4-6 and 0-5-6 from the hand set, and the left hand 0-1-2 should be Anne's hand. The intruder Crow holds card 6 and can remove hands 0-5-6 and 2-4-6 from the hand set, but the left hands 0-1-2, 0-3-4, 1-3-5 cover all the cards excluding card 6, so he cannot decide which cards Anne and Bill hold or do not hold. Therefore, the communication in this instance of solutions is safe.

3 Generalization of Russian Cards problem

The Russian Cards problem with 3 players and 7 cards, written as $R_{(3)}$, can be generalized to n players and $n \times (n-1) + 1$ cards, $\{0, 1, \ldots, n^2 - n\}$, written as $R_{(n)}$. The picking rule and deleting rule for $R_{(3)}$ can also be generalized for $R_{(n)}$. For convenience, we use the following notations. We call each player but the intruder a *party*, represented by P_i $(1 \leq i \leq n-1)$, and the party who announces his hand *announcer*. Similar to $R_{(3)}$, we assume there is a cards dealer who randomly dispatches n cards from $n(n-1)+1$ cards to each party as his hand, and leaves the remaining one card for intruder. The process for $R_{(n)}$ is much similar to $R_{(3)}$. We use h_{P_i} and h_{in} to denote hands of party P_i and the intruder respectively. With n $(n \geq 4)$ players, communication among parties is also based on announcements. With our approach, each announcement is actually a matrix (hand matrix for $R_{(n)}$) containing announcer's hand and other fake hands. Since it is hard to construct such matrix without knowledge of real card deal, only the cards dealer knows the card deal, so he is responsible for generating matrices for parties. In the communication, the first $n-2$ parties obtains their matrices from cards dealer then announce them one by one, namely ith matrix is announced after $(i-1)$th announcement; and $(n-1)$th party can figure out the intruder's card from those matrices. Further, $(n-1)$th announcement announces card of the intruder. After all of $n-1$ parties have made their announcements, they know each other's hand. In this way, the cards dealer only generates the matrix for each of the first $n-2$ parties. So, the communication is safe if, after all $n-1$ announcements, (i) all parties learn each other's hand; and (ii) the intruder knows nothing about any party's hand.

In the communication based on $R_{(n)}$, according to the picking rule, the cards dealer generates a hand matrix for each party and hiding hand of the announcer in it. Similar to $R_{(3)}$, announcer's hand can be placed in a row or column. So, we also consider picking rule for $R_{(n)}$ in row and column cases respectively. As a matter of fact, announcer's hand can be placed in any rows or columns of the matrix. However, for simplicity, we assume the announcer's hand is placed in the first row in row case and in the second column in column case. Note that this assumption is a secret to any players. In the communication protocol the picking rule is actually to encrypt the announcer's hand by means of hiding it in the matrix. In the following, we discuss the picking rule for $R_{(n)}$.

3.1 Picking rule

Each announcer announces his hand by means of a matrix, called hand matrix, so that his hand can be hidden in it. The cards dealer is responsible for constructing the hand matrix for each announcer. For convenience, we use $B^k = (b^k_{i,j})_{n\times n}$ $(1 \le k \le n-2)$ to denote the matrix for announcer P_k in kth announcement. In the following, we consider the construction of B^k in both row case and column case.

3.1.1 Construction of B^k in row case

In row case, the structure of B^k is as follows: the first row we call *hand row* holds all n cards of h_{P_k}. The first column which we call *sharing column* is filled with a card from h_{P_k} which we call *sharing card*. As shown in Fig. 1, the announcer's hand is $h_{P_k} = \{b^k_{1,1}, b^k_{1,2}, \ldots, b^k_{1,n}\}$, and the sharing card is $b^k_{1,1}$, where $b^k_{1,1} = b^k_{2,1} = \cdots = b^k_{n,1}$. The intruder's card can be placed in any of the remaining places. We assume that the intruder holds card $b^k_{p,q}$. We call the row and column containing this card respectively *redundant row* and *redundant column*. The remaining part of B^k holds the other $n^2 - 2n$ cards. However, how to place these cards into the matrix is a tricky job. In principle, to guarantee a successful matrix for the communication based on $R_{(n)}$, the above matrix has to satisfy the property (called *covering property* for row case): each row and each column, apart from the hand row and shar-

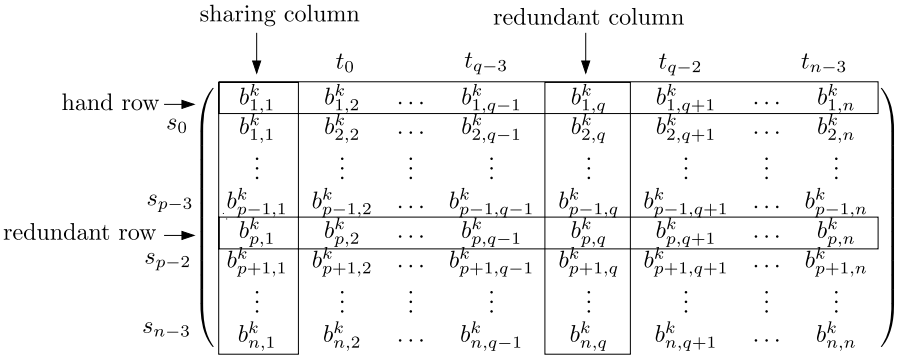

Fig. 1 Hand matrix of P_k in row case

ing column, contains a card from each hand of all parties except for the announcer. In this way, any row of B^k is a possible hand of the announcer, and apart from the sharing column, any column is also a possible hand of announcer. Let R_i^k (resp. $[R_i^k]$) represent the ith row (resp. set of cards from R_i^k), and C_j^k (resp. $[C_j^k]$) the jth column (resp. set of cards from C_j^k) of matrix B^k. Thus, the covering property can be given as follows

$$\forall i \forall j \big[(1 \le i \le n-1 \wedge i \ne k \wedge 2 \le j \le n) \rightarrow h_{P_i} \cap [R_j^k] \ne \phi \wedge h_{P_i} \cap [C_j^k] \ne \phi\big] \quad (3.1)$$

We call the procedure generating matrix B^k for P_k *picking rule*. Actually, the cards dealer first generates B^1 for P_1, then constructs B^h for P_h $(2 \le h \le n-2)$ based on B^1. The picking rule can be formalized as follows.

Constructing B^1

(a) Firstly, we choose a card from h_{P_1} as the *sharing card* to fill the *sharing column*. Then we place the remaining cards of h_{P_1} in the *hand row* in any order. Secondly we randomly place intruder's card in one of the remaining places of B^1. Suppose intruder's card is $b_{p,q}^1$, so pth row and qth column are respectively *redundant row* and *redundant column*. As shown in Fig. 1, apart from the sharing column, redundant column, hand row and redundant row, indices of the remaining rows from top to bottom are respectively denoted by $s_0, s_1, \ldots, s_{n-3}$, and indices of the remaining columns from left to right are respectively denoted by $t_0, t_1, \ldots, t_{n-3}$. For convenience, let $S := \{s_0, \ldots, s_{n-3}\}$ and $T := \{t_0, \ldots, t_{n-3}\}$. Thus, the indices can be generated in Algorithm 3.1. As a result, $(S, T) = \text{B1ra}\,(p, q, n)$.

Algorithm 3.1: B1ra (step (a) of generating B^1)

```
1  input: integer p, q, n;
2  output: set S, T;
3  temp variable integer i;
4  for i := 2 to n do
5      if i < p then
6          s_{i-2} := i;
7      else if i=p then                    /* skip the pth row */
8          skip;
9      else
10         s_{i-3} := i;
11     end
12     if i < q then
13         t_{i-2} := i;
14     else if i=q then                    /* skip the qth column */
15         skip;
16     else
17         t_{i-3} := i;
18     end
19 end
```

(b) We randomly divide each hand of the remaining parties P_k $(2 \le k \le n-1)$ into three parts such that $h_{P_k} = X[k] \cup Y[k] \cup Z[k]$, $|X[k]| = n-2$ and $|Y[k]| = |Z[k]| = 1$.

(c) For convenience, we first define positive integers od_k and ed_k $(2 \le k \le n-1)$ as follows:

$$od_k = \begin{cases} 2k-4 & 2 \le k \le \frac{n+1}{2} \\ 2k-n-2 & \frac{n+1}{2}+1 \le k \le n-1 \end{cases}$$

$$ed_k = \begin{cases} 0 & k = 2 \\ n-k & 3 \le k \le \frac{n}{2} \\ n-1-k & \frac{n+2}{2} \le k \le n-2 \\ \frac{n-2}{2} & k = n-1 \end{cases}$$

we place all $n-2$ cards of $X[2]$ into the remaining places of B^1 so that any sth row, $s \in S$, and any tth column, $t \in T$, can be occupied, namely $[R_s^1] \cap X[2] \neq \phi$ and $[C_t^1] \cap X[2] \neq \phi$. Thus, in any tth column there exists a card from $X[2]$. Suppose the card is in s_ith row. For each of the remaining parties P_k $(3 \le k \le n-1)$ we randomly pick a card from $X[k]$ and place it in s_uth $(u = (i + od_k) \bmod (n-2))$ row if n is odd, or place it in s_vth $(v = (i + ed_k) \bmod (n-2))$ row if n is even. Formally, it is described in Algorithm 3.2. Thus, $B^0 = \mathrm{B1rc}\ (S, T, X, n)$.

(d) Since there exists only one card in $Y[2]$ (resp. $Z[2]$) we place it randomly in any column (resp. row) within the redundant row (resp. column). We assume the card is in t_ith column (resp. s_ith row), $t_i \in T$ (resp. $s_i \in S$) in the redundant row (resp. column). For each of the remaining parties P_k $(3 \le k \le n-1)$, if n is odd we place the only one card from $Y[k]$ (resp. $Z[k]$) into t_uth $(u = (i + od_k) \bmod (n-2))$ column (resp. s_uth row) in the redundant row (resp. column), if n is even we place the only one card from $Y[k]$ (resp. $Z[k]$) into t_vth $(v = (i + ed_k) \bmod (n-2))$ column (resp. s_vth row) in the redundant row (resp. column). Formally, it is described in Algorithm 3.3. Thus, $B^1 = \mathrm{B1rd}\ (B^0, Y, Z, n)$.

Constructing B^k, $2 \le k \le n-2$ For convenience, we need three auxiliary matrices $A^k = (a^k_{i,j})_{n\times n}$, $D^k = (d^k_{i,j})_{n\times n}$ and $F^k = (f^k_{i,j})_{n\times n}$. In the same way as B^k, indices of rows and columns of matrices A^k, D^k and F^k can respectively be represented by $s_0, s_1, \ldots, s_{n-3}$ and $t_0, t_1, \ldots, t_{n-3}$. Let $A^k(R_i)$, $D^k(R_i)$ and $F^k(R_i)$ (resp. $A^k(C_i)$, $D^k(C_i)$ and $F^k(C_i)$) be ith rows (resp. columns) of matrices A^k, D^k and F^k respectively. Let $[D^k(R_i)]$ represent the set of cards from $D^k(R_i)$, and $[D^k(C_j)]$ the set of cards from $D^k(C_j)$. The procedure for constructing B^k is as follows

(a) Constructing A^k from B^1 (see Algorithm 3.4)

We first copy the first row and pth row of B^1 to the first row and pth row of A^k respectively. Then, for the remaining rows, we copy s_ith row of B^1 to s_jth row of A^k, $j := (i - (k-1) + n - 2) \bmod (n-2)$. Thus, $A^k = \mathrm{B2Ar}\ (B^1, n, p, k)$.

(b) Constructing D^k from A^k (see Algorithm 3.5)

We first copy the first column and qth column of A^k to the first column and qth column of D^k respectively. Then, for the remaining columns, we copy t_ith

Algorithm 3.2: B1rc (step (c) of generating B^1)

```
1  input: set S, T, array X[2 : n − 1], integer n;
2  output: matrix B^0;
3  temp variable set H, integer c, d, array q[0 : n − 2];
4  Let H := {0, 1, ..., n − 3};
5  for i := 0 to n − 3 do        /* place all cards of X[2] in B^1 */
6      Let c ∈ X[2]; X[2] := X[2] − {c};
                                  /* pick a card out of X[2] */
7      Let d ∈ H; H := H − {d};
8      b^1_{s_d, t_i} := c;
         /* card c is placed in s_d th row and t_i th column */
9      q[i] := d;
        /* q[i] keeps the index of row card c appears in */
10 end
       /* for every other party P_k, all cards of X[k] are
          placed in B^1 */
11 for i := 0 to n − 3 do
12     for k := 3 to n − 1 do
13         Let c ∈ X[k]; X[k] := X[k] − {c};
14         if n is odd then
                 /* card c chosen from X[k] is placed in t_i th
                    column */
15             d := (q[i] + od_k) mod (n − 2);
16         else
17             d := (q[i] + ed_k) mod (n − 2);
18         end
19         b^1_{s_d,t_i} := c;
           /* card c is placed in s_d th row and t_i th column */
20     end
21 end
```

column of A^k to t_jth column of D^k, $j := (i - (k-1) + n - 2) \bmod (n-2)$. Thus, $D^k = \text{A2Dr}\,(A, n, q, k)$.

(c) Constructing F^k and B^k (see Algorithm 3.6)

We first swap cards of h_{P_1} with cards of h_{P_k} in F^k. Thus, $F^k = \text{D2BFr}\,(D, Y[k], h_{P_k}, n, p)$, then, $B^k := F^k$.

A hand matrix satisfies covering rule (3.1) if and only if each fake hand contained in the matrix shares cards with hand of the party who receives it. So, covering rule guarantees other parties can identify the hand of the announcer by removing fake hands. In the following, we prove that matrix B^h generated according to the *picking rule* in row case satisfies formula (3.1).

Algorithm 3.3: B1rd (step (d) of generating B^1)

input: matrix B^0, arrays $Y[2:n-1]$, $Z[2:n-1]$, integer n
output: matrix B^1;
temp variable: set H, integer c, d, k, i;
Let $H := \{0, 1, \ldots, n-3\}$; **Let** $c \in Y[2]$; **Let** $i \in H$;
$b^1_{p,t_i} := c$; /* in redundant row, card of $Y[2]$ is placed in t_ith column */
/* for every other party P_k, card of $Y[k]$ is placed into redundant row */
for $k := 3$ **to** $n-1$ **do**
  Let $c \in Y[k]$;
  if *n is odd* **then**
    /* card c of $Y[k]$ is placed in redundant row */
    $d := (i + od_k) \bmod (n-2)$;
  else
    $d := (i + ed_k) \bmod (n-2)$;
  end
  $b^1_{p,\,t_d} := c$; /* in redundant row, card c is placed in t_dth column */
end
Let $H := \{0, 1, \ldots, n-3\}$; **Let** $c \in Z[2]$; **Let** $i \in H$;
$b^1_{s_i,q} := c$; /* in redundant column, card of $Z[2]$ is placed in s_ith row */
/* for every other parity P_k, card of $Z[k]$ is placed into redundant column */
for $k := 3$ **to** $n-1$ **do**
  Let $c \in Z[k]$;
  if *n is odd* **then** /* card c of $Z[k]$ is placed in redundant column */
    $d := (i + od_k) \bmod (n-2)$;
  else
    $d := (i + ed_k) \bmod (n-2)$;
  end
  $b^1_{s_d,q} := c$; /* in redundant column, card c is placed in s_dth row */
end

Lemma 1 *In row case, matrix* B^h $(1 \le h \le n-2)$ *generated according to the picking rule satisfies formula* (3.1).

Proof *A*. Proof of B^1 satisfying formula (3.1)

Algorithm 3.4: B2Ar (constructing A^k from B^1)

input: matrix B^1, integer n, p, k;
output: matrix A;
temp variable: integer i, j;
$A(R_1) := B^1(R_1)$;
/* copy the first row of B^1 to the first row of A^k */
$A(R_p) := B^1(R_p)$;
/* copy pth row of B^1 to pth row of A^k */
for $i := 0$ **to** $n-3$ **do**
/* construct the remaining rows of A^k */
  $j := (i-(k-1)+n-2) \bmod (n-2)$;
  $A(R_{s_j}) := B^1(R_{s_i})$;
  /* copy s_ith row of B^1 to s_{i-k+1}th row of A^k */
end

Algorithm 3.5: A2Dr (constructing D^k from A^k)

input: matrix A, integer n, q, k;
output: matrix D;
$D(C_1) := A(C_1)$; /* copy the first column of A^k to the first column of D^k */
$D(C_q) := A(C_q)$;
/* copy qth column of A^k to qth column of D^k */
for $i := 0$ **to** $n-3$ **do**
/* construct the remaining columns of D^k */
  $j := (i-(k-1)+n-2) \bmod (n-2)$;
  $D(C_{t_j}) := A(C_{t_i})$;
  /* copy t_ith column of A^k to t_{i-k+1}th column of D^k */
end

1. By the picking rule, we placed cards of $X[i]$ $(2 \le i \le n-1)$ in B^1 so, apart from the sharing column, each remaining column holds one card of $X[i]$. We have

$$\forall i \forall j \big[(2 \le i \le n-1 \wedge 0 \le j \le n-3) \rightarrow X[i] \cap [C^1_{t_j}] \neq \phi\big]$$

2. $\forall t_i, t_j \in T$, each of columns $C^1_{t_i}$ and $C^1_{t_j}$ holds exactly one card of $X[2]$. Suppose $b^1_{s_u,t_i} \in [C^1_{t_i}] \cap X[2]$ and $b^1_{s_v,t_j} \in [C^1_{t_j}] \cap X[2]$. By the picking rule, for each of the remaining parties P_k $(3 \le k \le n-1)$, there exist two cards, $b^1_{s_x,t_i}$ and $b^1_{s_y,t_j}$, of $X[k]$ such that $b^1_{s_x,t_i} \in [C^1_{t_i}]$ and $b^1_{s_y,t_j} \in [C^1_{t_j}]$; where x and y satisfy the following conditions:

Algorithm 3.6: D2BFr (constructing F^k and B^k)

```
1  input: matrix D, set Y, set h, integer n, p;
2  output: matrix F, B;
3  temp variable: i, j, g;
4  F := D;                                          /* copy D^k to F^k */
5  for j := 0 to n − 3 do
                  /* locate the card of Y[k] in redundant row */
6      if f_{p,t_j} ∈ Y then
7          break;
8      end
9  end
10 g := f_{p,t_j}; f_{p,t_j} := f_{1,1};
11 for i := 1 to n do /* fill the sharing column with card g */
12     f_{i,1} := g;
13 end
   /* swap the card of P_1 with card of P_k in every other
      column */
14 for j := 2 to n do
15     for i := 0 to n − 3 do
                      /* locate the card of h_{P_k} in jth column */
16         if f_{s_i,j} ∈ h then      /* card of P_k is in jth column */
17             break;
18         end
19     end
20     g := f_{s_i,j}; f_{s_i,j} := f_{1,j}; f_{1,j} := g;
                               /* swap card f_{s_i,j} with card f_{1,j} */
21 end
22 B := F;                                          /* copy F^k to B^k */
```

(a) if n is odd, $x = (u + od_k) \bmod (n-2)$ and $y = (v + od_k) \bmod (n-2)$.
(b) if n is even, $x = (u + ed_k) \bmod (n-2)$ and $y = (v + ed_k) \bmod (n-2)$.

Since $u \neq v$ so $x \neq y$. We have,

$$\forall i \forall j \big[(2 \leq i \leq n-1 \wedge 0 \leq j \leq n-3) \rightarrow X[i] \cap [R^1_{s_j}] \neq \phi\big]$$

3. Since $n-2$ cards in the redundant row (resp. column) respectively come from $Y[2], \ldots, Y[n-1]$ (resp. $Z[2], \ldots, Z[n-1]$). Thus we have

$$\forall i \big[(2 \leq i \leq n-1) \rightarrow Y[i] \cap [R^1_p] \neq \phi \wedge Z[i] \cap [C^1_q] \neq \phi\big]$$

By 1, 2, 3 above, apart from announcer P_1, the intersection of hand of any remaining party and the set of cards of any row (resp. any column), excluding the hand row (resp. the sharing column) of B^1 is not empty. Therefore, formula (3.1) holds for B^1.

B. Proof of B^k $(2 \leq k \leq n-2)$ satisfying formula (3.1)

$\forall t_i, t_j \in T$ (resp. $\forall s_u, s_v \in S$), apart from P_1 and P_k, each of remaining parties P_r ($2 \le r \le n-1$ and $r \ne k$) has two cards $b^1_{s_u,t_i} \in X[r]$ and $b^1_{s_v,t_j} \in X[r]$ ($s_u, s_v \in S$ and $u \ne v$ (resp. $t_i, t_j \in T$ and $i \ne j$)). According to the picking rule, they appear in D^k respectively as $d^k_{s_x,t_d}$ and $d^k_{s_y,t_e}$, where

(a) $x = (u-k+n-1) \bmod (n-2)$, $y = (v-k+n-1) \bmod (n-2)$,
(b) $d = (i-k+n-1) \bmod (n-2)$ and $e = (j-k+n-1) \bmod (n-2)$.

Since $e \ne d$ (resp. $x \ne y$), we have $\forall i \forall j[(2 \le i \le n \wedge 1 \le j \le n-1) \to [D^k(C_i)] \cap h_{P_j} \ne \phi]$ (resp. $\forall i \forall j[(2 \le i \le n \wedge 2 \le j \le n-1) \to [D^k(R_i)] \cap h_{P_j} \ne \phi]$). So, after h_{P_1} and h_{P_k} are swapped in F^k, $B^k (= F^k)$ satisfies

$$\forall i \forall j\big[(2 \le i \le n \wedge 1 \le j \le n-1 \wedge j \ne k) \to [C^k_i] \cap h_{P_j} \ne \phi \wedge [R^k_i] \cap h_{P_j} \ne \phi\big]$$

So, apart from announcer P_k, the intersection of hand of any remaining party and the set of cards of any row (resp. any column), excluding the hand row (resp. the sharing column) of B^k is not empty. Therefore, formula (3.1) holds for B^k. □

3.1.2 Construction of B^k in column case

In column case, the second column we call *hand column* holds all n cards of h_{P_k}. The first column we call *sharing column* is filled with a card from $h_{P_{k+1}}$ which we call *sharing card*. As shown in Fig. 2, the announcer's hand is $h_{P_k} = \{b^k_{1,2}, b^k_{2,2}, \ldots, b^k_{n,2}\}$, and the sharing card is $b^k_{1,1}$, where $b^k_{1,1} = b^k_{2,1} = \cdots = b^k_{n,1}$. The intruder's card can be placed in any of the remaining places. We assume that the intruder holds card $b^k_{p,q}$. Accordingly, we call the row and column containing this card respectively *redundant row* and *redundant column*. The remaining part of B^k holds the other $n^2 - 2n - 1$ cards. In principle, to guarantee a successful matrix for the communication based on $R_{(n)}$, the above matrix has to satisfy the property (called *covering property* for column case): each row and each column, apart from the sharing column and hand column, contains a card from each hand of all parties except for the announcer. In this way, any row of B^k is a possible hand of the announcer, and apart from the sharing column, any column is also a possible hand of announcer. Let R^k_i (resp. $[R^k_i]$) represent the

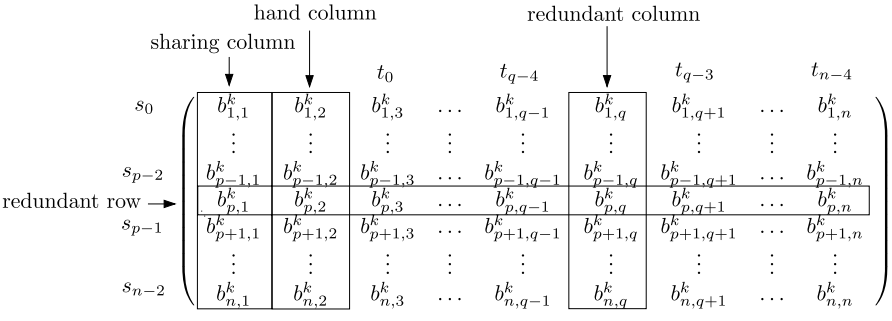

Fig. 2 Hand matrix of P_k in column case

ith row (resp. set of cards from R_i^k), and C_j^k (resp. $[C_j^k]$) the jth column (resp. set of cards from C_j^k) of matrix B^k. Thus, the covering property can be given as follows

$$\forall i \forall j \big[(1 \le i \le n-1 \wedge i \ne k \wedge 1 \le j \le n \wedge 3 \le l \le n) \\ \rightarrow h_{P_i} \cap [R_j^k] \ne \phi \wedge h_{P_i} \cap [C_l^k] \ne \phi\big] \tag{3.2}$$

Similar to row case, the cards dealer needs to be equipped with picking rule to obtain hand matrix in column case. The cards dealer first generates B^1 for P_1, then constructs B^h for P_h $(2 \le h \le n-2)$ based on B^{h-1}. Note that, in column case, construction of B^h depends on B^{h-1} not on B^1. The picking rule for column case can be formalized as follows.

A. Constructing B^1

(a) Firstly, we choose a card from h_{P_2} as the *sharing card* to fill the *sharing column*. Then we place the cards of h_{P_1} in the *hand column* in any order. Secondly we randomly place intruder's card in one of the remaining places of B^1. Suppose intruder's card is $b_{p,q}^1$, so pth row and qth column are respectively *redundant row* and *redundant column*. As shown in Fig. 2, apart from the sharing column, hand column and redundant row, indices of the remaining rows from top to bottom are respectively denoted by $s_0, s_1, \ldots, s_{n-2}$, and indices of the remaining columns from left to right are respectively denoted by $t_0, t_1, \ldots, t_{n-3}$. For convenience, let $S := \{s_0, \ldots, s_{n-2}\}$ and $T := \{t_0, \ldots, t_{n-4}\}$. The indices can be generated using Algorithm 3.7. Thus, $(S, T) = \mathrm{B1a}(p, q, n)$.

Algorithm 3.7: B1a (step (a) of generating B^1)

```
1  input: integer p, q, n;
2  output: set S, T;
3  temp variable integer i;
4  for i := 1 to n do
5      if i < p then
6          s_{i-1} := i;
7      else if i=p then                    /* skip the pth row */
8          skip;
9      else
10         s_{i-2} := i;
11     end
12     if i > 2 and i < q then
13         t_{i-3} := i;
14     else if i=q then                    /* skip the qth column */
15         skip;
16     else if i > 2 then
17         t_{i-4} := i;
18     end
19 end
```

(b) Apart from parties P_1 and P_2, we randomly divide each hand of the remaining parties P_k ($3 \le k \le n-1$) into two parts such that $h_{P_k} = X[k] \cup Y[k]$, $|X[k]| = n-1$ and $|Y[k]|$=1.

(c) Let $X[2] = h_{P_2} - \{b^1_{1,1}\}$. We place all $n-1$ cards of $X[2]$ into the remaining places of B^1 so that any sth row, $s \in S$, and any tth column, $t \in T \cup \{q\}$, can be occupied, namely $[R^1_s] \cap X[2] \neq \phi$ and $[C^1_t] \cap X[2] \neq \phi$. Thus, in any sth row there exists a card from $X[2]$. Suppose the card is in uth column of sth row. For each of the remaining parties P_k ($3 \le k \le n-1$) we randomly pick a card from $X[k]$ and place it in vth ($v = (u-3+k-2) \bmod (n-2)+3$) column in sth row. Formally, it is described in Algorithm 3.8. Thus, $B^0 = \mathrm{B1c}\ (X, Y, n)$.

(d) Since there exists only one card in $Y[2]$, we place it randomly in any column within the redundant row. We assume the card is in t_ith column, $t_i \in T$, in the redun-

Algorithm 3.8: B1c (step (c) of generating B^1)

```
1  input: array X[3 : n − 1], Y[3 : n − 1], integer n;
2  output: matrix B^0;
3  temp variable set H, integer c, d, array q[0 : n − 2];
4  Let H := {3, . . . , n};
5  Let d ∈ H; c ∈ X[2];
     /* chose card c from X[2] and place it in s_{n−2}th row */
   X[2] := X[2] − {c}; b^1_{s_{n−2},d} := c; q[n − 2] = d;
6  for i := 0 to n − 3 do
                /* place n − 2 remaining cards of X[2] in B^1 */
7      Let c ∈ X[2]; X[2] := X[2] − {c};
                                    /* pick a card out of X[2] */
8      Let d ∈ H; H := H − {d};
9      b^1_{s_i,d} := c;
          /* card c is placed in s_ith row and dth column */
10     q[i] := d;
            /* q[i] keeps the index of column card c appears
               in */
11 end
   /* for every other party P_k, all cards of X[k] are
      placed in B^1 */
12 for i := 0 to n − 2 do
13     for k := 3 to n − 1 do
14         Let c ∈ X[k]; X[k] := X[k] − {c};
15         d := (q[i] − 3 + k − 2) mod (n − 2) + 3;
16         b^1_{s_i,d} := c;
           /* card c is placed in s_ith row and dth column */
17     end
18 end
```

Algorithm 3.9: B1d (step (d) of generating B^1)

input: matrix B^0, array $Y[3:n-1]$,integer n;
output: matrix B^1;
temp variable: set H, integer c, d, k, i;
Let $H := \{0, 1, \ldots, n-3\}$;
Let $c \in Y[2]$; **Let** $i \in H$;
$b^1_{p,\,t_i} := c$; /* in redundant row, card of $Y[2]$ is placed in t_ith column */
/* for every other party P_k, card of $Y[k]$ is placed into redundant row */
for $k := 3$ **to** $n-1$ **do**
  Let $c \in Y[k]$;
  $d := (i + k - 2) \bmod (n-3)$;
  $b^1_{p,\,t_d} := c$; /* in redundant row, card c is placed in t_dth column */
end

dant row. For each of the remaining parties P_k ($3 \le k \le n-1$), we place the only one card from $Y[k]$ into t_uth ($u = (i + k - 2) \bmod (n-3)$) column in the redundant row. Formally, it is described in Algorithm 3.9. Thus, $B^1 = \text{B1d}\ (B^0, Y, n)$.

B. Constructing B^k, $2 \le k \le n-2$.

Different from row case, the construction of B^k depends on B^{k-1} rather than B^1. So, the generation of hand matrix B^k is in a inductive way. The base condition is construction of B^1 which we have introduced. So, in the following, we give the induction method in which the existence of B^{k-1} is assumed for construction of B^k. For convenience, we need three auxiliary matrices $A^k = (a^k_{i,j})_{n\times n}$, $D^k = (d^k_{i,j})_{n\times n}$ and $F^k = (f^k_{i,j})_{n\times n}$. In the same way as B^k, indices of rows and columns of matrices A^k, D^k and F^k can respectively be represented by $s_0, s_1, \ldots, s_{n-2}$ and t_0, $t_1, \ldots, t_{n-3}$. Let $A^k(R_i)$, $D^k(R_i)$ and $F^k(R_i)$ (resp. $A^k(C_i)$, $D^k(C_i)$ and $F^k(C_i)$) be ith rows (resp. columns) of matrices A^k, D^k and F^k respectively. Let $[D^k(R_i)]$ represent the set of cards from $D^k(R_i)$, and $[D^k(C_j)]$ the set of cards from $D^k(C_j)$. The procedure for constructing B^k is as follows.

(a) Constructing A^{k-1} from B^{k-1} (see Algorithm 3.10). We first copy pth row of B^{k-1} to pth row of A^{k-1}. Then, for the remaining rows, we copy s_ith row of B^{k-1} to s_jth row of A^{k-1}, $j := (i - 1 + n - 1) \bmod (n-1)$. Thus, $A^{k-1} = \text{B2A}\ (B^{k-1}, n, p)$.

(b) Constructing D^{k-1} from A^{k-1} (see Algorithm 3.11). Firstly, we copy the first column and 2nd column of A^{k-1} to the first column and 2nd column of D^{k-1} respectively. Then, for pth row of D^{k-1}, we first copy card $a^{k-1}_{p,q}$ to D^{k-1} as card $d^{k-1}_{p,q}$, then for the remaining $n-3$ places in pth row of D^{k-1}, we copy card a^{k-1}_{p,t_i} to D^{k-1} as card d^{k-1}_{p,t_j}, $j := (i - 1 + n - 3) \bmod (n-3)$. Finally, for each s_ith row ($s_i \in S$) of D^{k-1},

Algorithm 3.10: B2A (constructing A^{k-1} from B^{k-1})

input: matrix B, integer n, p;
output: matrix A;
temp variable: i, j;
$A(R_p) := B(R_p)$;
/* copy pth row of B^{k-1} to pth row of A^{k-1} */
for $i := 0$ **to** $n-2$ **do**
/* construct the remaining rows of A^{k-1} */
  $j := (i-1+n-1) \bmod (n-1)$;
  $A(R_{s_j}) := B(R_{s_i})$;
  /* copy s_ith row of B^{k-1} to s_{i-1}th row of A^{k-1} */
end

Algorithm 3.11: A2D (constructing D^{k-1} from A^{k-1})

input: matrix A, integer n, p, q;
output: matrix D;
temp variable: i, j;
$D(C_1) := A(C_1)$;
/* copy 1st column of A^{k-1} to 1st column of D^{k-1} */
$D(C_2) := A(C_2)$;
/* copy 2nd column of A^{k-1} to 2nd column of D^{k-1} */
$d_{p,q} := a_{p,q}$;
/* copy card $a_{p,q}^{k-1}$ of A^{K-1} to D^{k-1} as card $d_{p,q}^{k-1}$ */
/* place the remaining $n-3$ cards in pth row of A^{k-1} to pth row of D^{k-1} */
for $i := 0$ **to** $n-4$ **do**
  $j := (i-1+n-3) \bmod (n-3)$;
  $d_{p,t_j} := a_{p,t_i}$;
  /* copy card a_{p,t_i}^{k-1} of A^{k-1} to D^{k-1} as card $a_{p,t_{i-1}}^{k-1}$ */
end
for $i := 0$ **to** $n-2$ **do**
/* place cards for $n-1$ remaining rows of D^{k-1} */
  for $j := 3$ **to** n **do**
    $u := (j-3-1+n-2) \bmod (n-2)+3$;
    $d_{s_i,u} := a_{s_i,j}$;
    /* copy card $a_{s_i,j}^{k-1}$ of A^{k-1} to D^{k-1} as card $d_{s_i,j-1}^{k-1}$ */
  end
end

Algorithm 3.12: D2BF (constructing F^{k-1} and B^k)

```
1  input: matrix D, set Y, h, integer n, k;
2  output: matrix B, F;
3  temp variable: integer i, j, g;
4  F := D;                                   /* copy D^{k-1} to F^{k-1} */
5  for j := 0 to n − 4 do
                /* locate the card of Y[k] in redundant row */
6      if f_{p,t_j} ∈ Y then
7          break;
8      end
9  end
10 g := f_{p,t_j}; f_{p,t_j} := f_{p,2}; f_{p,2} := f_{1,1};
11 for i := 1 to n do /* fill the sharing column with card g */
12     f_{i,1} := g;
13 end
          /* swap the card of P_{k−1} with card of P_k in every
             other row */ for i := 0 to n − 2 do
14     for j := 3 to n do
                        /* locate the card of h_{P_k} in s_ith row */
15         if f^{k−1}_{s_i,j} ∈ h then      /* card of P_k is in jth column */
16             break;
17         end
18     end
19     g := f_{s_i,j}; f_{s_i,j} := f_{s_i,2}; f_{s_i,2} := g;
                              /* swap card f^{k−1}_{s_i,j} with card f^{k−1}_{s_i,2} */
20 end
21 B^k := F^{k−1};                                /* copy F^k to B^k */
```

we copy card $a^{k-1}_{s_i,j}$ to D^{k-1} as card $d^{k-1}_{s_i,u}$, $u := (j-3-1+n-2) \bmod (n-2)+3$. Thus, $D^{k-1} = \mathrm{A2D}(A^{k-1}, n, p, q)$.

(c) Constructing F^{k-1} and B^k (see Algorithm 3.12). We first swap cards of $h_{P_{k-1}}$ with cards of h_{P_k} in F^{k-1}. Thus, $F^{k-1} = \mathrm{D2BF}\ (D^k, Y[k], h_{P_k}, n, k)$, then $B^k := F^{k-1}$.

We prove that hand matrix B^h $(1 \le h \le n-2)$ generated by the picking rule in column case satisfies formula (3.2).

Lemma 2 *In column case, matrix* B^h $(1 \le h \le n-2)$ *generated by the picking rule satisfies formula* (3.2).

Proof The proof is similar to row case. □

3.2 Deleting rule

After the cards dealer generates $n-2$ hand matrices, he dispatches these matrices to corresponding parties as their announcements, namely B^k to party P_k. According to the rule of $R_{(n)}$, $n-1$ parties make announcement one by one. And after each announcement B^h $(1 \le h \le n-2)$, apart from announcer P_h, any other party P_k $(1 \le k \le n-1,\ k \ne h)$ compares his hand with the announcement to determine the announcer's hand. Similarly, the intruder also compares his hand with the announcement to probe announcer's hand. In the communication protocol, a *deleting rule* for parties is required to decrypt the announcer's hand from matrix B^h, and might be used for the intruder to detect the announcer's hand.

Suppose the current announcer is P_h $(1 \le h \le n-2)$. The idea behind the deleting for P_k is that party P_k compares his hand with each row and each column (excluding the sharing column) of B^h, and records them into set H_{P_k} if the row or column does not intersect with his hand. The formal description of the deleting rule is similar to Algorithm 2.3. Note that if

$$|H_{P_k}| = 1 \tag{3.3}$$

then P_k is aware of announcer's hand.

Similarly, the intruder can also compare his hand with each row and each column (excluding the sharing column) of B^h, and records them into set H_{in} if the row or column does not intersect with his hand. However, different from the deleting rule for parties, the intruder further obtains set H_{intr} of cards extracted from members of H_{in}. Subsequently, intruder checks if

$$|H_{in}| > 1 \tag{3.4}$$

and

$$h_{in} = \{0, 1, \ldots, n^2 - n\} - H_{intr} \tag{3.5}$$

If so, the intruder may learn nothing about announcer's hand.

As a matter of fact, $R_{(3)}$ is a trivial problem of $R_{(n)}$. Because once Anne sent the matrix to Bill by means of announcement, no matter what hand Bill holds, the hand matrix features that apart from the first column and the row or column containing Anne's hand, every other row and column contain at least a card of Bill. This is much like what we call covering property for $R_{(n)}$. The covering property is important for hand matrix to guarantee that other parties are aware of hand of the announcer by the deleting rule. After hth announcement, by the deleting rule, the intruder can only remove the redundant row and redundant column from B^h. So, from the intruder's view, other hands contained in the hand matrix are actually candidates of hand of party P_h. For convenience, we call the set candidate set. We use HS^h to denote candidate set for party P_h. So, in row case we have $HS^h = \{[R^h_1], [R^h_{s_0}], \ldots, [R^h_{s_{n-3}}], [C^h_{t_0}], \ldots, [C^h_{t_{n-3}}]\}$, in column case we have $HS^h = \{[R^h_{s_0}], \ldots, [R^h_{s_{n-2}}], [C^h_2], [C^h_{t_0}], \ldots, [C^h_{t_{n-4}}]\}$. So, after $(n-2)$th announcement, the intruder can obtain $n-2$ hand sets: HS^k $(1 \le k \le n-2)$. Since any two parties do not allow to share a card, the intruder may obtain a group of hand sets,

$G = \{\{h_1, h_2, \ldots, h_{n-2}\} \in HS^1 \times \cdots \times HS^{n-2} | \forall i \forall j h_i \cap h_j = \phi, i \neq j \text{ and } 1 \leq i, j \leq n-2\}$. Note that any hand set from G can be regarded as a candidate of card deal.

Note that if $|G| = 1$ the intruder knows the card deal, and further learns the hand of any party. Even $|G| > 1$, the intruder may also know these cards a party does not hold though the intruder is not aware of the real card deal. For example, $1 < |G| < 2n-3$ means for each HS^k there are some candidates which are excluded by the intruder. And the remaining candidates cannot cover all cards excluding the intruder's card. So, the intruder knows what cards are not in h_{P_k}. Based on the above discussion, we know if

$$|G| = 2n - 3 \tag{3.6}$$

the intruder is not aware of the real card deal and hence of any information about hand of each party.

Although $R_{(3)}$ is a trivial problem of $R_{(n)}$, we do not consider the candidate sets in the safety definition. Because in $R_{(3)}$, there is only one hand matrix (announced by Anne) communicated among parties. The intruder (Crow) can obtain just one candidate set HS^1. Thus, the intruder figures out $G = HS^1$ and $|G| = 3$. So, he does not know the real card deal. Regarding safety for $R_{(n)}$, we must take candidate set into account.

Definition 2 (Safe communication) *A communication for $R_{(n)}$, with the first $n-2$ announcements generated by picking rule, is safe if formulas* (3.3), (3.4), (3.5) *and* (3.6) *are satisfied.*

Actually, the picking rule for choosing sharing card is merely for simplicity. Of course, the sharing card can be picked out from hand of any party. If we go this way, the order of swap of cards in the redundant row of F^{k-1} must be accordingly modified in order to produce qualified hand matrix B^k. Furthermore, we use hand matrix just for clearly presenting the construction of hand set in each announcement. In fact, hand matrix can be smoothly converted back to hand set for each party to announce. It is harder for the intruder to obtain any information about hand of any party from hand set than hand matrix. We believe there may exist other solutions for Generalized Russian Cards problem. However, using hand matrix approach we gave, the communication is safe. In the following, we prove the safety for Generalized Russian Cards problem in row case and column case respectively.

3.3 Safety proof for $R_{(n)}$ in row case

We first prove the following lemmas.

Lemma 3 $\forall s \forall t [(2 \leq s, t \leq n-2 \wedge s \neq t) \rightarrow (od_s - s + n - 1) \bmod (n-2) \neq (od_t - t + n - 1) \bmod (n-2) \wedge (ed_s - s + n - 1) \bmod (n-2) \neq (ed_t - t + n - 1) \bmod (n-2)]$.

Proof Let $og_i = (od_i - i + n - 1) \bmod (n-2)$ and $eg_i = (ed_i - i + n - 1) \bmod (n-2)$.

(1) if n is odd, og_i is defined as follows:

$$og_i = \begin{cases} n-3 & i=2 \\ i-3 & 3 \le i \le \frac{n+1}{2} \\ i-3 & \frac{n+1}{2}+1 \le i \le n-2 \end{cases}$$

When $i=2$, $og_i = n-3$; when $i \in [3, \frac{n+1}{2}]$, $og_i \in [0, \frac{n-5}{2}]$; when $i \in [\frac{n+1}{2}+1, n-2]$, $og_i \in [\frac{n-3}{2}, n-5]$. Note that $[0, \frac{n-5}{2}]$, $[\frac{n-3}{2}, n-5]$, $\{n-4\}$ and $\{n-3\}$ are a partition of $[0, n-3]$. Thus, in the case in which n is odd, $\forall s\ \forall t\ [(2 \le s,t \le n-2 \wedge s \ne t) \to (od_s - s + n - 1) \bmod (n-2) \ne (od_t - t + n - 1) \bmod (n-2)]$.

(2) if n is even, eg_i is defined as follows:

$$eg_i = \begin{cases} n-3 & i=2 \\ n-2i+1 & 3 \le i \le \frac{n}{2} \\ 2n-2i-2 & \frac{n+2}{2} \le i \le n-2 \end{cases}$$

When $i=2$, $eg_i = n-3$; when $i \in [3, \frac{n}{2}]$, $eg_i \in [1, n-5]$; when $i \in [\frac{n+2}{2}, n-2]$, $eg_i \in [2, n-4]$. Let $W_1 = \{k \mid 2 \le k \le n-4$ and k is even$\}$, $W_2 = \{k \mid 1 \le k \le n-5$ and k is odd$\}$ and $W_3 = \{n-3\}$. We have $W_i \cap W_j = \phi (1 \le i, j \le 3, i \ne j)$. Thus, in the case in which n is even, $\forall s \forall t [(2 \le s,t \le n-2 \wedge s \ne t) \to (ed_s - s + n - 1) \bmod (n-2) \ne (ed_t - t + n - 1) \bmod (n-2)]$. □

Armed with Lemma 3, we can prove that for any pair of two hand matrixes, s_ith $(0 \le i \le n-3)$ rows share no cards as well as t_jth $(0 \le j \le n-3)$ columns share no cards.

Lemma 4 *In row case*, $\forall h \forall k \forall i [(1 \le h,k \le n-2 \wedge h \ne k \wedge 0 \le i \le n-3) \to [R^h_{s_i}] \cap [R^k_{s_i}] = \phi \wedge [C^h_{t_i}] \cap [C^k_{t_i}] = \phi]$.

Proof 1. $\forall h \forall k \forall i [(1 \le h,k \le n-2 \wedge h \ne k \wedge 0 \le i \le n-3) \to [C^h_{s_i}] \cap [C^k_{s_i}] = \phi]$.

A card $b^1_{s_u,t_v}$ in B^1, appears as card $d^h_{s_x,t_y}$ in D^h, and card $d^k_{s_i,t_j}$ in D^k ($s_u \in S$, $t_v \in T$, $2 \le k, h \le n-2$ and $h \ne k$), where $x = (u - h + n - 1) \bmod (n-2)$, $y = (v - h + n - 1) \bmod (n-2)$, $i = (u - k + n - 1) \bmod (n-2)$ and $j = (v - k + n - 1) \bmod (n-2)$. Since $u \ne x \ne i$, $u \ne i$, $v \ne y \ne j$ and $v \ne j$ we have

$$\begin{aligned} &\forall i \forall h \forall k \big[(0 \le i \le n-3 \wedge 1 \le h,k \le n-2 \wedge h \ne k) \\ &\quad \to [D^h(C_{t_i})] \cap [D^k(C_{t_i})] = \phi\big] \end{aligned} \tag{3.7}$$

According to the picking rule, there exists a card $b^1_{p,t_r} \in Y[2]$ ($t_r \in T$) in redundant row. And for any pair of other parties (excluding P_1), P_h and P_k, there exist two cards in the redundant row: $b^1_{p,t_x} \in Y[h]$ and $b^1_{p,t_y} \in Y[k]$. If n is odd $x = (r + od_h) \bmod (n-2)$ and $y = (r + od_k) \bmod (n-2)$, and if n is even $x = (r + ed_h) \bmod (n-2)$ and $y = (r + ed_k) \bmod (n-2)$. In addition, the two cards appear respectively as cards f^h_{p,t_i} in F^h and f^k_{p,t_j} in F^k, where

$i = (x - h + n - 1) \bmod (n-2)$ and $j = (y - k + n - 1) \bmod (n-2)$. According to Lemma 3 we have $i \neq j$. So, after h_{P_1} and h_{P_h} are swapped in F^h, and h_{P_1} and h_{P_k} in F^k, we have $\forall i[(0 \leq i \leq n-3) \rightarrow [C^h_{t_i}] \cap [C^k_{t_i}] = \phi]$. Thus,

$$\forall i \forall h \forall k[(0 \leq i \leq n-3 \wedge 1 \leq h, k \leq n-2 \wedge h \neq k) \rightarrow [C^h_{t_i}] \cap [C^k_{t_i}] = \phi]$$

2. $\forall h \forall k \forall i[(1 \leq h, k \leq n-2 \wedge h \neq k \wedge 0 \leq i \leq n-3) \rightarrow [R^h_{s_i}] \cap [R^k_{s_i}] = \phi]$.
Same to the way in which we obtain formula (3.7), we have

$$\forall i \forall h \forall k\big[(0 \leq i \leq n-3 \wedge 1 \leq h, k \leq n-2 \wedge h \neq k) \rightarrow [D^h(R_{s_i})] \cap [D^k(R_{s_i})] = \{b^1_{1,1}\}\big]$$

According to the picking rule, apart from the sharing column, in each of remaining columns C^1_e $(2 \leq e \leq n)$ of B^1, there exists a card of P_2 such that $b^1_{s_r,e} \in X[2]$. In addition, for any pair of other parties (excluding P_1) P_h and P_k, there exist two cards in C^1_e such that $b^1_{s_u,e} \in X[h]$ and $b^1_{s_v,e} \in X[k]$. If n is odd $u = (r + od_h) \bmod (n-2)$ and $v = (r + od_k) \bmod (n-2)$, and if n is even $u = (r + ed_h) \bmod (n-2)$ and $v = (r + ed_k) \bmod (n-2)$. The two cards appear respectively as cards $f^h_{s_i,x}$ in F^h and $f^k_{s_j,y}$ in F^k, where $i = (u - h + n - 1) \bmod (n-2)$ and $j = (v - k + n - 1) \bmod (n-2)$. According to Lemma 3 we have $i \neq j$. So, h_{P_1} and h_{P_h} are swapped in F^h, and h_{P_1} and h_{P_k} in F^k, we have $\forall i$ $[(0 \leq i \leq n-3) \rightarrow [R^h_{s_i}] \cap [R^k_{s_i}] = \phi]$. Thus,

$$\forall i \forall h \forall k\big[(0 \leq i \leq n-3 \wedge 1 \leq h, k \leq n-2 \wedge h \neq k) \rightarrow [R^h_{s_i}] \cap [R^k_{s_i}] = \phi\big]$$ □

With Lemmas 1 and 4 we can prove the safety for $R_{(n)}$ in row case.

Theorem 2 (Safe communication for $R_{(n)}$) *Any card deal for $R_{(n)}$, the communication based on B^k $(1 \leq k \leq n-2)$ generated by the picking rule in row case is safe.*

Proof Let $S = \{s_0, \ldots, s_{n-3}\}$ and $T = \{t_0, \ldots, t_{n-3}\}$.

1. After all $n-1$ announcements, all parties learn each other's hand.
   According to Lemma 1, B^k $(1 \leq k \leq n-2)$ satisfies formula (3.1). So, after kth announcement, for each of other parties P_h $(1 \leq h \leq n-1$ and $k \neq h)$, we have $h_{P_h} \cap [R^k_s] \neq \phi$ and $h_{P_h} \cap [C^k_t] \neq \phi$, $s \in S \cup \{p\}$ and $t \in T \cup \{q\}$. By the deleting rule, party P_h obtains $H_{P_h} = \{[R^k_1]\}$, so $|H_{P_h}| = 1$ (3.3). Since $[R^k_1] = h_{P_k}$, P_h learns h_{P_k}. In the same way, after all $n-2$ announcements, apart from $h_{P_{n-1}}$ all other hands $h_{P_1}, h_{P_2}, \ldots$ and $h_{P_{n-2}}$ are known by $P_1, P_2, \ldots$ and P_{n-1}. Since P_{n-1} knows $h_{P_1}, h_{P_2}, \ldots, h_{P_{n-2}}$ and $h_{P_{n-1}}$, he can figure out intruder's hand $h_{in} = \{0, 1, \ldots, n^2 - n\} - \bigcup_{i=1}^{n-1} h_{p_i}$. Further, in $(n-1)$th announcement, P_{n-1} announces intruder's card. Thus, the other parties can learn h_{n-1}. Therefore, after all $n-1$ announcements, all parties learn each other's hand.

2. After all $n-1$ announcements, the intruder knows nothing about any party's hand.
After each announcement B^k $(1 \le k \le n-2)$, since $h_{in} \cap [R_p^k] \ne \phi$ and $h_{in} \cap [C_q^k] \ne \phi$, by the deleting rule, the intruder obtains $H_{in} = \{[R_1^k], [R_{s_0}^k], \ldots, [R_{s_{n-3}}^k], [C_{t_0}^k], \ldots, [C_{t_{n-3}}^k]\}$. However, he is unable to learn any card the announcer holds because of $|H_{in}| > 1$ (3.4). Further, he is also unable to learn any cards the announcer does not hold, because $H_{intr} = [R_1^k] \cup [R_{s_0}^k] \cup \cdots \cup [R_{s_{n-3}}^k] \cup [C_{t_0}^k] \cup \cdots \cup [C_{t_{n-3}}^k] = \{0, \ldots, n^2 - n\} - h_{in}$. So formula (3.5) holds. In addition, after each announcement B^k, the intruder obtains set $HS^k = \{[R_1^k], [R_{s_0}^k], \ldots, [R_{s_{n-3}}^k], [C_{t_0}^k], \ldots, [C_{t_{n-3}}^k]\}(|HS^k| = 2n - 3)$ of candidates for h_{P_k}. So, after $(n-2)$th announcement, the intruder can obtain $n-2$ hand sets: $HS^h (1 \le h \le n-2)$. According to Lemma 3, $[R_s^i] \cap [R_s^j] = \phi \wedge [C_t^i] \cap [C_t^j] = \phi (1 \le i, j \le n-2, i \ne j, s \in S$ and $t \in T)$. The intruder figures out a set $G = \{\{[R_1^1], \ldots, [R_1^{n-2}]\}, \{[R_{s_0}^1], \ldots, [R_{s_0}^{n-2}]\}, \ldots, \{[R_{s_{n-3}}^1], \ldots, [R_{s_{n-3}}^{n-2}]\}, \{[C_{t_0}^1], \ldots, [C_{t_0}^{n-2}]\}, \ldots, \{[C_{t_{n-3}}^1], \ldots, [C_{t_{n-3}}^{n-2}]\}\}$ of candidates for card deal. Since $|G| = 2n - 3$ (3.6), the intruder is unable to learn the card deal. Therefore, the intruder knows nothing about any party's hand.

Therefore, the communication in row case for $R_{(n)}$ is safe. □

3.4 Safety proof for $R_{(n)}$ in column case

Similar to the safety proof for row case, we fist prove that for any pair of two hand matrixes, both their s_ith $(0 \le i \le n-3)$ rows and their t_jth $(0 \le j \le n-4)$ columns share no cards.

Lemma 5 *In column case*, $\forall h \forall k \forall i \forall j [(1 \le h, k \le n-2 \wedge h \ne k \wedge 0 \le i \le n-2 \wedge 0 \le j \le n-4) \rightarrow [R_{s_i}^h] \cap [R_{s_i}^k] = \phi \wedge [C_{t_j}^h] \cap [C_{t_j}^k] = \phi]$.

Proof The proof is similar to Lemma 4. □

With Lemmas 2 and 5 we can prove the safety for $R_{(n)}$ in column case.

Theorem 3 (Safe communication for $R_{(n)}$) *Any card deal for $R_{(n)}$, the communication based on B^k $(1 \le k \le n-2)$ generated by the picking rule in column case is safe.*

Proof The proof is similar to Theorem 2. □

Time complexity The cards dealer constructs a matrix for each of $n-2$ parties in $\mathcal{O}(n^2)$ time units, so, generation of announcements is at most in $\mathcal{O}(n^3)$ time units. With each announcement, $n-2$ parties identify hand of the announcer from an $n \times n$ hand matrix using deleting rule. There are $n-2$ possible announcers in total, so, the time complexity of this process is at most $\mathcal{O}(n^3)$. The intruder constructs a candidates set for each announcer using deleting rule. This process is in $\mathcal{O}(n^3)$ time units. Then, the intruder compares these $n-2$ candidates set to form a group G of hand sets in $\mathcal{O}(n^3)$ time units. Therefore, the time complexity of communication for $R(n)$ is $\mathcal{O}(n^3)$.

4 Example

To illustrate how our approach works for safe communication, we study $R_{(5)}$. The procedure for constructing B^h $(1 \le h \le 3)$ in row case is presented in detail. Twenty-one cards are randomly dispatched by the cards dealer to five players so that each party holds five cards and the intruder holds only one card. Without loss of generality, we assume the card deal is $h_{P_1} = \{3, 4, 8, 10, 12\}$, $h_{P_2} = \{0, 9, 11, 14, 17\}$, $h_{P_3} = \{1, 5, 15, 16, 20\}$, $h_{P_4} = \{2, 6, 7, 13, 18\}$ and $h_{in} = \{19\}$. By using the picking rule, the card dealer can construct $B^h (1 \le h \le 3)$ step by step. We only consider $R_{(5)}$ in Row Case.

1. Constructing B^1

(a) As shown in Fig. 3 (1), we choose card 8 from h_{P_1} as sharing card to fill the first column. And we place the remaining cards 10, 3, 4, 12 in the first row. Then, we randomly place intruder's card 19 as $b^1_{4,3}$, so 4th row and 3rd column are redundant row and redundant column respectively. In addition, we obtain indices $s_0 = 2$, $s_1 = 3$, $s_2 = 5$, $t_0 = 2$, $t_1 = 4$ and $t_2 = 5$.

(b) We randomly divide each hand of parties P_2, P_3, P_4 into three parts: $h_{P_2} = X[2] \cup Y[2] \cup Z[2] = \{9, 11, 14\} \cup \{17\} \cup \{0\}$, $h_{P_3} = X[3] \cup Y[3] \cup Z[3] = \{1, 15, 16\} \cup \{20\} \cup \{5\}$ and $h_{P_4} = X[4] \cup Y[4] \cup Z[4] = \{2, 6, 13\} \cup \{18\} \cup \{7\}$.

(c) We first place cards of $X[2] = \{9, 11, 14\}$ into B^1. As shown in Fig. 3 (2), 9 is placed as $b^1_{3,2}$, 11 as $b^1_{2,4}$ and 14 as $b^1_{5,5}$, so that each of rows s_0, s_1 and s_2 as well as each column of t_0, t_1 and t_2 corresponds to one card from $X[2]$. According to the placement of $X[2]$, we then place cards of $X[3] = \{1, 15, 16\}$ and $X[4] = \{2, 6, 13\}$ in B^1 as shown in Fig. 3 (3). For instance, in t_0th (=2nd) column, since card 9 from $X[2]$ is in s_1th (=3rd) row and $(1 + od_3)$ mod $(5 - 2) = 0$ $(od_3 = (2 \times 3 - 4) = 2)$, we choose card 1 from $X[3]$ and place it in s_0th (=2nd) row and t_0th column.

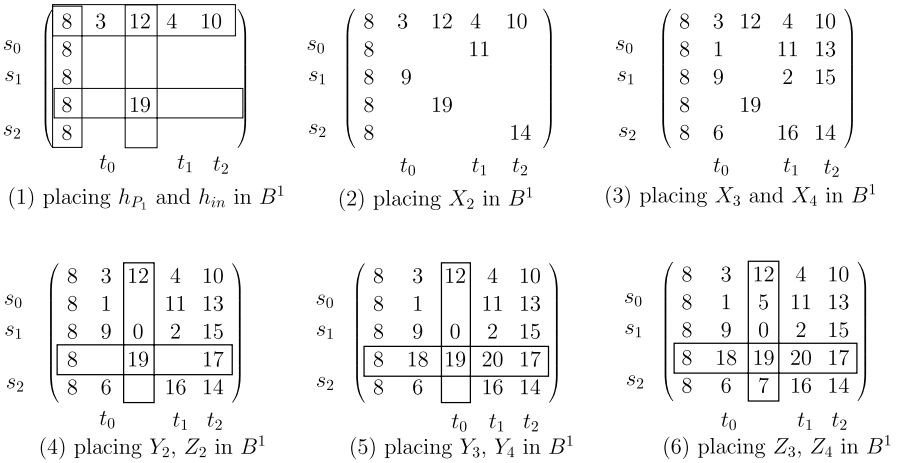

Fig. 3 Construction of B^1

(d) As shown in Fig. 3 (4), we randomly place card of $Y[2] = \{17\}$ in the redundant row (4th row) and t_2th column, and card of $Z[2] = \{0\}$ in the redundant column (3th column) and s_1th row. According to the placement of $Y[2]$, we then place $Y[3] = \{20\}$ and $Y[4] = \{18\}$ in B^1 as shown in Fig. 3 (5). For example, since card 17 is in t_2th column and $(2 + od_3)$ mod $(5 - 2) = 1$, we place the only card 20 from $Y[3]$ in t_1th column within the redundant row. Further, as shown in Fig. 3 (6), according to the placement of $Z[2]$, we place $Z[3] = \{5\}$ and $Z[4] = \{7\}$ in B^1. For instance, since card 0 is in s_1th row and $(1 + od_3)$ mod $(5 - 2) = 0$, we place the only card 5 from $Z[3]$ in s_0th row within the redundant column.

2. Constructing B^2 and B^3

(a) As shown in Fig. 4 (1) (2), since $(0 - (2 - 1) + 5 - 2)$ mod $(5 - 2) = 2$, we copy s_0th row of B^1 to A^2 as its s_2th row. In the same way, we copy s_1th row of B^1 to A^2 as its s_0th row, and s_2th row of B^1 to A^2 as its s_1th row. In addition, we copy 1st and 4th rows of B^1 to A^2 as its 1st and 4th rows respectively. As shown in Fig. 4 (4) (5), since $(0 - (3 - 1) + 5 - 2)$ mod $(5 - 2) = 1$, we copy s_0th row of B^1 to A^3 as its s_1th row. In the same way, we copy s_1th row of B^1 to A^3 as its s_2th row, and s_2th row of B^1 to A^3 as its s_0th row. Besides, we copy 1st and 4th rows of B^1 to A^3 as its 1st and 4th rows respectively.

(b) As shown in Fig. 4 (2) (3), in a similar way as (a), we copy t_0th column of A^2 to D^2 as its t_2th column, t_1th column of A^2 to D^2 as its t_0th column and t_2th column of A^2 to D^2 as its t_1th column. Moreover, we copy 1st and 3rd columns of A^2 to D^2 as its 1st and 3rd column respectively. As shown in Fig. 4 (5) (6), we copy t_0th column of A^3 to D^3 as its t_1th column, t_1th column of A^3 to D^3 as its t_2th column and t_2th column of A^3 to D^3 as its t_0th column. Moreover, we copy 1st and 3rd columns of A^2 to D^3 as its 1st and 3rd columns respectively.

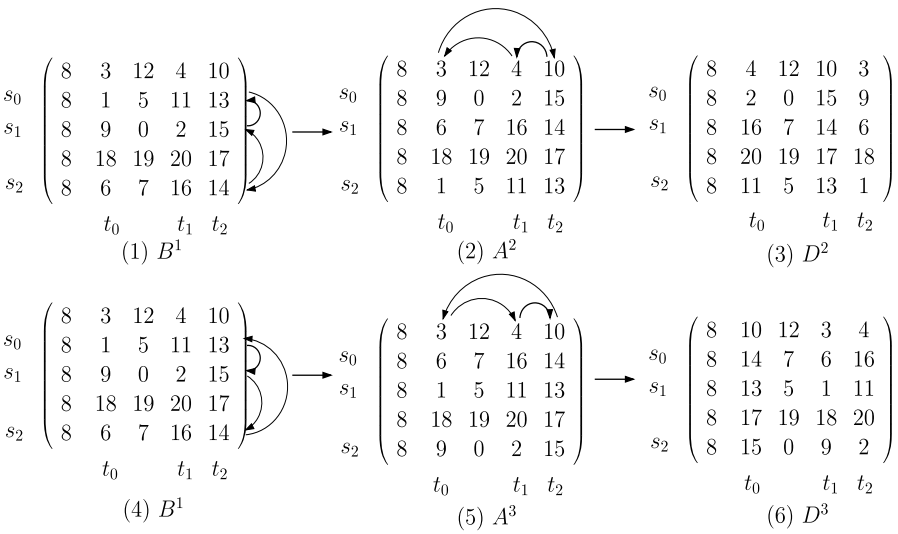

Fig. 4 Generation of A^2, A^3, D^2 and D^3

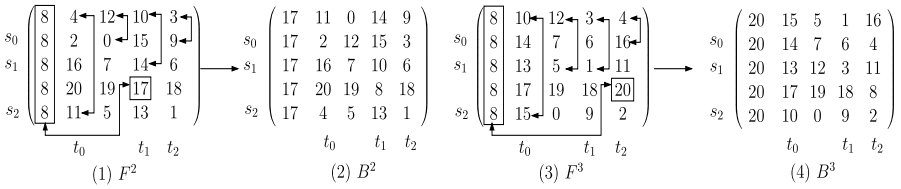

Fig. 5 Generation of B^2 and B^3 from F^2 and F^3

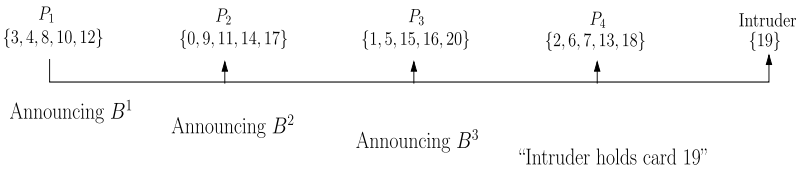

Fig. 6 Safe communication for $R_{(5)}$

(c) Let $F^2 := D^2$. As shown in Fig. 5 (1) (2), we try to swap all cards in h_{P_1} with cards in h_{P_2} in F^2. To do so, we first swap card 8 in the first column with card 17 in the fourth row. Then, we swap card 4 with card 11 in 2nd column, card 12 with card 0 in 3rd column, card 10 with card 14 in 4th column, and card 3 with card 9 in 5th column. Finally, let $B^2 := F^2$. In the same way, we obtain $B^3 := F^3$ as shown in Fig. 5 (3) (4).

After B^1, B^2 and B^3 are generated, P_1 firstly announces B^1, as shown in Fig. 6. On receiving it, according to the deleting rule P_2 knows announcer's hand h_{P_1}. In the same way, P_3 and P_4 are aware of h_{P_1} from B^1. However, for the intruder, he can only remove the redundant row and redundant column. Thus, the intruder can create a set for P_1 as candidates: $HS^1 = \{\{8, 3, 12, 4, 10\}, \{8, 1, 5, 11, 13\}, \{8, 9, 0, 2, 15\}, \{8, 6, 7, 16, 14\}, \{3, 1, 9, 18, 6\}, \{4, 11, 2, 20, 16\}, \{10, 13, 15, 17, 14\}\}$. Since the union of all hands from the set covers all cards but card 19, so the intruder cannot learn which card does not belong to announcer's hand h_{P_1}. Then P_2 announces B^2. In the same way, P_1, P_3 and P_4 are aware of h_{P_2} from B^2; and the intruder can create a set for P_2: $HS^2 = \{\{17, 11, 0, 14, 9\}, \{17, 2, 12, 15, 3\}, \{17, 16, 7, 10, 6\}, \{17, 4, 5, 13, 1\}, \{11, 2, 16, 20, 4\}, \{14, 15, 10, 8, 13\}, \{9, 3, 6, 18, 1\}\}$ without learning anything about the announcer's hand h_{P_2}. Further, P_3 announces B^3. Similarly, P_1, P_2 and P_4 are aware of h_{P_3} from B^3; and the intruder can create a set for P_3: $HS^3 = \{\{20, 15, 5, 1, 16\}, \{20, 14, 7, 6, 4\}, \{20, 13, 12, 3, 11\}, \{20, 10, 0, 9, 2\}, \{15, 14, 13, 17, 10\}, \{1, 6, 3, 18, 9\}, \{16, 4, 11, 8, 2\}\}$ without learning which card does not belong to announcer's hand h_{P_3}. Finally, P_4 figures out the intruder's hand, since he has known other parties' hands h_{P_1}, h_{P_2} and h_{P_3}. He announces that intruder's card is 19. Then, on receiving the announcement, P_1, P_2 and P_3 can figure out h_{P_4} because among them they have known each other's hand and intruder's hand. Therefore, all the parties learn each other's hand. However, for the intruder, from HS^1, HS^2 and HS^3 he obtains seven candidates of

card deal:

$$
\begin{aligned}
G = \{&\{\{8,3,12,4,10\},\{17,11,0,14,9\},\{20,15,5,1,16\},\{2,6,7,13,18\}\},\\
&\{\{8,1,5,11,13\},\{17,2,12,15,3\},\{20,14,7,6,4\},\{0,9,10,16,18\}\},\\
&\{\{8,9,0,2,15\},\{17,16,7,10,6\},\{20,13,12,3,11\},\{1,4,5,14,18\}\},\\
&\{\{8,6,7,16,14\},\{17,4,5,13,1\},\{20,10,0,9,2\},\{3,11,12,15,18\}\},\\
&\{\{3,1,9,18,6\},\{11,2,16,20,4\},\{15,14,13,17,10\},\{0,5,7,8,12\}\},\\
&\{\{4,11,2,20,16\},\{14,15,10,8,13\},\{1,6,3,18,9\},\{0,5,7,12,17\}\},\\
&\{\{10,13,15,17,14\},\{9,3,6,18,1\},\{16,4,11,8,2\},\{0,5,7,12,20\}\}\}
\end{aligned}
$$

Since $|G| = 2 \times 5 - 3 = 7$, the intruder cannot learn the card deal. Moreover, he can figure out all candidates of h_{P_4} being $\{2,6,7,13,18\}$, $\{0,9,10,16,18\}$, $\{1,4,5,14,18\}$, $\{3,11,12,15,18\}$, $\{0,5,7,8,12\}$, $\{0,5,7,12,17\}$ and $\{0,5,7,12,20\}$. Since the union of these candidates covers all cards but card 19, the intruder cannot learn which card does not belong to P_4. For the same reason the intruder has no idea about which card does not belong to P_1, P_2 or P_3 as well. After the communication, each party knows each other's hand, but the intruder learns nothing about any party's hand. Therefore, the communication for $R_{(5)}$ illustrated above is safe.

5 Conclusion

We discussed original Russia Cards problem $R_{(3)}$ with 3 players and 7 cards and generated it to $R_{(n)}$ with n players and $n(n-1)+1$ cards. The picking rule was developed to construct hand set while the deleting rule was designed to decide card deal. Based on $R_{(n)}$, an unconditionally secure protocol further be developed to tackle n parties communication without public keys. However how to work out a paradigm for using the protocol for communication is a challenge for us in the future. In addition, the verification of the paradigm with the protocol based on $R_{(n)}$ is also required. To do so, we will employ Propositional Projection Temporal Logic (Cong and Zhenhua 2007; Zhenhua et al. 2008a, 2008b; Zhenhua and Koutny 2004) to express the properties and use Promela (Holzmann 2003) to describe the behavior of the protocol. Thus, the model checker SPIN (Holzmann 2003) can be used to check the properties.

References

Albert MH, Aldred REL, Atkinson MD, van Ditmarsch HP, Handley CC (2005) Safe communication for card players by combinatorial designs for two-step protocols. Australasian J Comb 33:33–46

Atkinson MD, van Ditmarsch HP, Roehling S (2007) Avoiding bias in cards cryptography. CoRR. cs/0702097

Cyriac A, Krishnan KM (2008) Lower Bound for the Communication Complexity of the Russian Cards Problem. CoRR. 0805.1974

Cong T, Zhenhua D (2007) Model checking propositional projection temporal logic based on SPIN. In: Proceedings of ICFEM07. Lecture Notes in Computer Science, vol 4789. Springer, Berlin, pp 246–265

Fischer MJ, Wright RN (1996) Bounds on secret key exchange using a random deal of cards. J Cryptol 9(2):71–99

Holzmann GJ (2003) The spin model checker: Primer and reference manual. Addison-Wesley, Berlin

Koichi K, Takaaki M, Takao N (2004) Necessary and sufficient numbers of cards for the transformation protocol. Lecture Notes in Computer Science, vol 3106. Springer, Berlin, pp 92–101

Ramanujam R, Suresh SP (2001) Information based reasoning about security protocols. Electron Notes Theor Comput Sci 55(1):89–104

Rivest R, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126

Roehling S (2005) Cards and cryptography. Report in partial fulfilment of MSc in Computer Science, University of Otago

Makarychev K (2001) Logicheskie voprosy peredachi informacii (logical issues of information transmission). Master's thesis, Moscow State University, Diplomnaja rabota, Part 1

Stiglic A (2001) Computations with a deck of cards. Theor Comput Sci 259(1–2):671–678

van Ditmarsch HP (2003) The Russian Cards problem. Stud Log 75:31–62

van Ditmarsch HP (2005) The case of the hidden hand. J Appl Non-Class Log 15(4):437–452

van Ditmarsch HP, van der Hoek W, van der Meyden R, Ruan J (2006) Model checking Russian Cards. Electron Notes Theor Comput Sci 149:105–123

Vasilenko O (2006) Number-theoretic algorithms in cryptography. American Mathematical Society, Boston. (Translations of Mathematical Monographs)

Zhenhua D, Koutny M (2004) A framed temporal logic programming language. J Comput Sci Technol 19:333–344

Zhenhua D, Cong T, Li Z (2008a) A decision procedure for propositional projection temporal logic with infinite models. Acta Inform 45(1):43–78

Zhenhua D, Xiaoxiao Y, Koutny M (2008b) Framed temporal logic programming. Sci Comput Program 70:31–61