A real world application of secure multi-party computations Duplicate bridge for cheapskates

Matthew Johnson matthew.johnson@cl.cam.ac.uk Ralph Owen rho21@cam.ac.uk

University of Cambridge

The 16th International Workshop on Security Protocols

### Outline

Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

Future work

Conclusion

## Outline

#### Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Future work

Conclusion

### Example deal

#### 1st permutation

Order the suits: Clubs Hearts Spades Diamonds 1143 2323 4422 1143 2411 4143 1332 4344 1223 2433 1211 3242 4224

#### 2nd permutation

3231	1224	1243	4421	1233	4421
1311	1432	3332	2441	2244	3332
4141					

### Example deal

#### 1st permutation

Order the suits: Clubs Hearts Spades Diamonds 1143 2323 4422 1143 2411 4143 1332 4344 1223 2433 1211 3242 4224

#### 2nd permutation

3231	1224	1243	4421	1233	4421
1311	1432	3332	2441	2244	3332
4141					

▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ = 臣 = のへで

### Example deal

#### 1st permutation

Order the suits: Clubs Hearts Spades Diamonds 1143 2323 4422 1143 2411 4143 1332 4344 1223 2433 1211 3242 4224

#### 2nd permutation

3231	1224	1243	4421	1233	4421
1311	1432	3332	2441	2244	3332
4141					

▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ = 臣 = のへで

# Outline

Example deal

#### Bridge

Protocol

Flaws and corrections

Case study

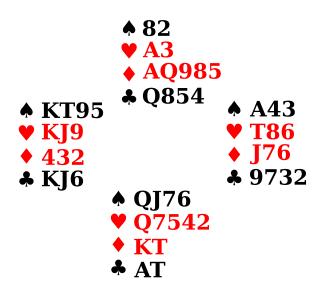
Error detection/correction

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Future work

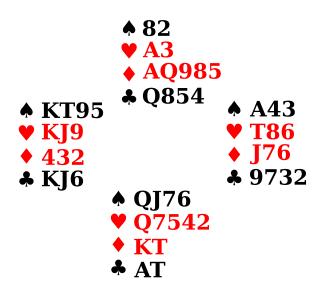
Conclusion

Bridge



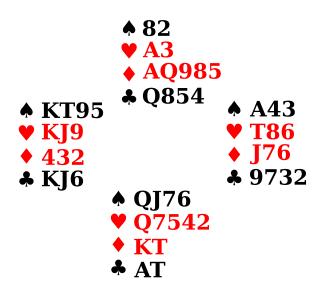
◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = のへで

Bridge



◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = のへで

Bridge



◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = のへで

# Outline

Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

Future work

Conclusion



### Traditionally

Secret inputs to each party generating a shared result

Computations done on computer

- Secret result, known inputs
- 'Computations' done by humans
- Intermediate state can be secret

### Traditionally

Secret inputs to each party generating a shared result

Computations done on computer

- Secret result, known inputs
- 'Computations' done by humans
- Intermediate state can be secret

### Traditionally

Secret inputs to each party generating a shared result

Computations done on computer

- Secret result, known inputs
- 'Computations' done by humans
- Intermediate state can be secret

### Traditionally

Secret inputs to each party generating a shared result

Computations done on computer

- Secret result, known inputs
- 'Computations' done by humans
- Intermediate state can be secret

### Traditionally

Secret inputs to each party generating a shared result

Computations done on computer

- Secret result, known inputs
- 'Computations' done by humans
- Intermediate state can be secret

### Attacker model

#### Assume the players are inherently trustworthy

- They can cheat anyway if not
- Most players are trustworthy
- Players are sufficiently intelligent to make use of small amounts of information
- Main security goals:
  - Ensure neither dealer can deduce much about the hands while dealing...

... and having seen one of the hands.

### Attacker model

#### Assume the players are inherently trustworthy

- They can cheat anyway if not
- Most players are trustworthy
- Players are sufficiently intelligent to make use of small amounts of information
- Main security goals:
  - Ensure neither dealer can deduce much about the hands while dealing...

... and having seen one of the hands.

### Attacker model

- Assume the players are inherently trustworthy
  - They can cheat anyway if not
  - Most players are trustworthy
- Players are sufficiently intelligent to make use of small amounts of information
- Main security goals:
  - Ensure neither dealer can deduce much about the hands while dealing...

... and having seen one of the hands.

- 1. Generate random  $P_T$ ;  $T = \{S\}_{E_{P_T}}$
- 2. Discard  $P_T$
- 3. Generate random  $P_1$  and  $P_1$
- 4. Calculate  $P_2$  s.t.  $T = \{\{S_{P_l}\}_{E_{P_1}}\}_{E_{P_2}}$

▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ = 臣 = のへで

- 5. Give  $P_1 \& P_1$  to dealer 1
- 6. Give  $P_2$  to dealer 2

- 1. Generate random  $P_T$ ;  $T = \{S\}_{E_{P_T}}$
- 2. Discard  $P_T$
- 3. Generate random  $P_1$  and  $P_1$
- 4. Calculate  $P_2$  s.t.  $T = \{\{S_{P_l}\}_{E_{P_1}}\}_{E_{P_2}}$

▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ = 臣 = のへで

- 5. Give  $P_1 \& P_1$  to dealer 1
- 6. Give  $P_2$  to dealer 2

- 1. Generate random  $P_T$ ;  $T = \{S\}_{E_{P_T}}$
- 2. Discard  $P_T$
- 3. Generate random  $P_1$  and  $P_1$
- 4. Calculate  $P_2$  s.t.  $T = \{\{S_{P_l}\}_{E_{P_1}}\}_{E_{P_2}}$

- 5. Give  $P_1 \& P_1$  to dealer 1
- 6. Give  $P_2$  to dealer 2

- 1. Generate random  $P_T$ ;  $T = \{S\}_{E_{P_T}}$
- 2. Discard  $P_T$
- 3. Generate random  $P_1$  and  $P_l$
- 4. Calculate  $P_2$  s.t.  $T = \{\{S_{P_l}\}_{E_{P_1}}\}_{E_{P_2}}$

- 5. Give  $P_I \& P_1$  to dealer 1
- 6. Give  $P_2$  to dealer 2

- 1. Generate random  $P_T$ ;  $T = \{S\}_{E_{P_T}}$
- 2. Discard  $P_T$
- 3. Generate random  $P_1$  and  $P_l$
- 4. Calculate  $P_2$  s.t.  $T = \{\{S_{P_l}\}_{E_{P_1}}\}_{E_{P_2}}$

- 5. Give  $P_1 \& P_1$  to dealer 1
- 6. Give  $P_2$  to dealer 2

# Outline

Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

Future work

Conclusion



#### Suit of the first card dealt

- ▶ Last thirteen cards in *P*<sup>1</sup> same suit.
- Likely that there will be a 1 in the last 13 numbers of  $P_1$ .
- ▶ Implies first card of *P*<sup>2</sup> is that suit.
- First hand dealt in  $P_2$  does not have a void in that suit.

#### Solution

Randomize the order of the suits in  $P_1$ .

#### But...

Hands must be shuffled before going into the boards, else the second dealer can infer the suit order from the order of the cards in their hands.

#### Suit of the first card dealt

- Last thirteen cards in P<sub>1</sub> same suit.
- Likely that there will be a 1 in the last 13 numbers of  $P_1$ .
- ▶ Implies first card of *P*<sup>2</sup> is that suit.
- ▶ First hand dealt in P<sub>2</sub> does not have a void in that suit.

#### Solution

Randomize the order of the suits in  $P_1$ .

#### But...

Hands must be shuffled before going into the boards, else the second dealer can infer the suit order from the order of the cards in their hands.

#### Suit of the first card dealt

- Last thirteen cards in P<sub>1</sub> same suit.
- Likely that there will be a 1 in the last 13 numbers of  $P_1$ .
- ▶ Implies first card of *P*<sup>2</sup> is that suit.
- First hand dealt in  $P_2$  does not have a void in that suit.

#### Solution

Randomize the order of the suits in  $P_1$ .

#### But. . .

Hands must be shuffled before going into the boards, else the second dealer can infer the suit order from the order of the cards in their hands.

#### Suit of the first card dealt

- Last thirteen cards in P<sub>1</sub> same suit.
- Likely that there will be a 1 in the last 13 numbers of  $P_1$ .
- Implies first card of P<sub>2</sub> is that suit.
- First hand dealt in  $P_2$  does not have a void in that suit.

#### Solution

Randomize the order of the suits in  $P_1$ .

#### But...

Hands must be shuffled before going into the boards, else the second dealer can infer the suit order from the order of the cards in their hands.

### Locating high cards

 High cards from first suit will be at the bottom of some of the piles

• One of positions  $\{13, 26, 39, 52\}$  in  $P_2$  will hold an ace.

#### Solution

Randomize the number of cards in each pile at the end of  $P_1$ .

### Locating high cards

 High cards from first suit will be at the bottom of some of the piles

• One of positions  $\{13, 26, 39, 52\}$  in  $P_2$  will hold an ace.

#### Solution

Randomize the number of cards in each pile at the end of  $P_1$ .

### Locating high cards

 High cards from first suit will be at the bottom of some of the piles

• One of positions  $\{13, 26, 39, 52\}$  in  $P_2$  will hold an ace.

#### Solution

Randomize the number of cards in each pile at the end of  $P_1$ .

# Outline

Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

Future work

Conclusion



# Case study

- Two trials, 3 sessions in November–December 2007, 6 sessions in January–March 2008.
- Approximately six dealers in total, three pairs.
- ▶ Time to deal 28 boards consistently 10–15 minutes.
- ▶ Observed error rate 4–6 boards, with one perfect result.

# Case study

- Two trials, 3 sessions in November–December 2007, 6 sessions in January–March 2008.
- Approximately six dealers in total, three pairs.
- ► Time to deal 28 boards consistently 10–15 minutes.
- Observed error rate 4–6 boards, with one perfect result.

# Case study

- Two trials, 3 sessions in November–December 2007, 6 sessions in January–March 2008.
- Approximately six dealers in total, three pairs.
- ► Time to deal 28 boards consistently 10–15 minutes.
- ▶ Observed error rate 4–6 boards, with one perfect result.

## Case study

- Two trials, 3 sessions in November–December 2007, 6 sessions in January–March 2008.
- Approximately six dealers in total, three pairs.
- ► Time to deal 28 boards consistently 10–15 minutes.
- Observed error rate 4–6 boards, with one perfect result.

## Outline

Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Future work

Conclusion

### No detection

Ignore errors.

#### Detection only

Check at some point during the play against the hand record for that board.

#### Detection and correction

• Check the first time the board is played using curtain cards.

### No detection

Ignore errors.

#### Detection only

 Check at some point during the play against the hand record for that board.

### Detection and correction

• Check the first time the board is played using curtain cards.

### No detection

Ignore errors.

#### Detection only

 Check at some point during the play against the hand record for that board.

#### Detection and correction

Check the first time the board is played using curtain cards.

### No detection

Ignore errors.

#### Detection only

 Check at some point during the play against the hand record for that board.

#### Detection and correction

Check the first time the board is played using curtain cards.

## Outline

Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Future work

Conclusion

### Future work

#### More rigorous trials

- Montecarlo simulations
- Alternative primitives



### Future work

- More rigorous trials
- Montecarlo simulations

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Alternative primitives

### Future work

- More rigorous trials
- Montecarlo simulations

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Alternative primitives

## Outline

Example deal

Bridge

Protocol

Flaws and corrections

Case study

Error detection/correction

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Future work

Conclusion

## Conclusion

#### Security is sufficient

- Doesn't take too long
- Error rate is not zero, but can be worked around

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

### Conclusion

- Security is sufficient
- Doesn't take too long
- Error rate is not zero, but can be worked around

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

## Conclusion

- Security is sufficient
- Doesn't take too long
- Error rate is not zero, but can be worked around

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

▲□ > ▲□ > ▲目 > ▲目 > ▲□ > ▲□ >

▲□ > ▲□ > ▲目 > ▲目 > ▲□ > ▲□ >

### Trial error results

Session	Failures	Recoverable Errors
14/03/08	1	3
07/03/08	2	3
22/02/08	0	0
15/02/08	2	2
07/02/08	3	4
31/02/08	4	2
30/11/07	5	2
16/11/07	7	1
01/11/07	4	1

Table: Errors in each session