Karlsruhe Institute of Technology

Card-Based Cryptography Meets Formal Verification

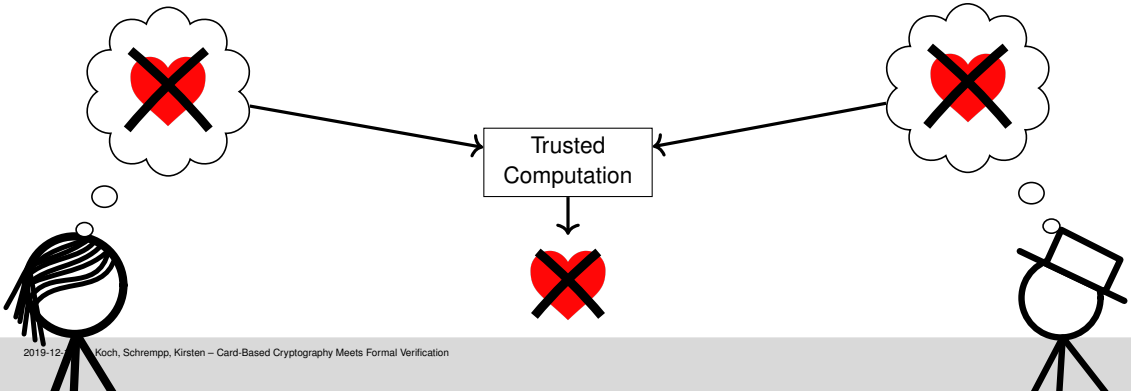
ASIACRYPT 2019 | Alexander Koch, Michael Schrempp, Michael Kirsten

KIT DEPARTMENT OF INFORMATICS, INSTITUTE OF THEORETICAL INFORMATICS

Motivating Scenario

Calculating Mutual Interest with Playing Cards

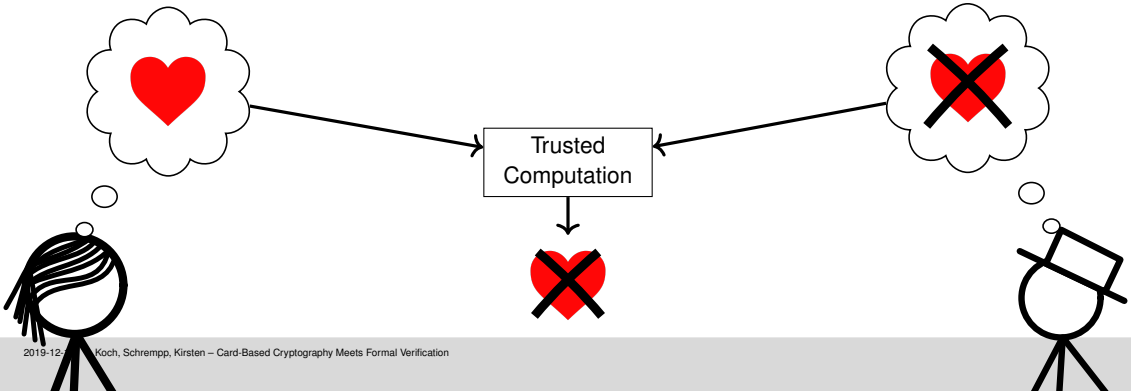
- Secrets: Do I fancy him/her?
- To compute: Is there mutual interest?
- ⇝ Secure 2-party AND without computers



Motivating Scenario

Calculating Mutual Interest with Playing Cards

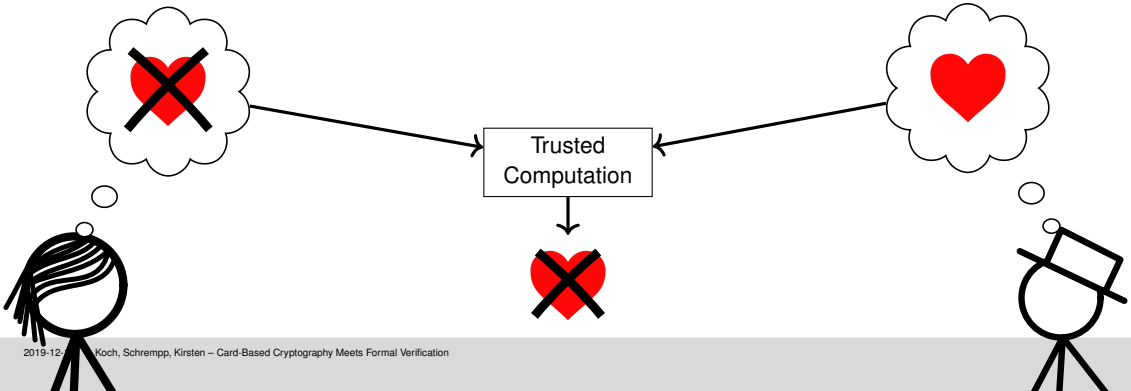
- Secrets: Do I fancy him/her?
- To compute: Is there mutual interest?
- ⇝ Secure 2-party AND without computers



Motivating Scenario

Calculating Mutual Interest with Playing Cards

- Secrets: Do I fancy him/her?
- To compute: Is there mutual interest?
- ⇝ Secure 2-party AND without computers

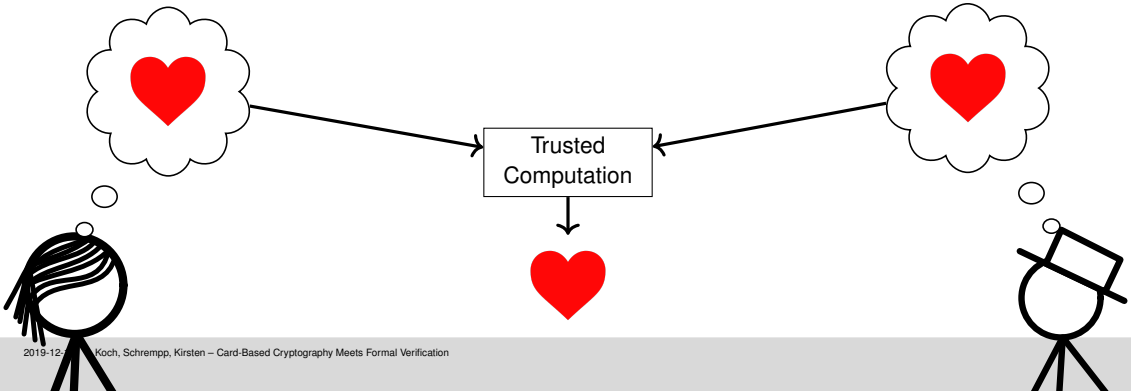


Motivating Scenario

Calculating Mutual Interest with Playing Cards

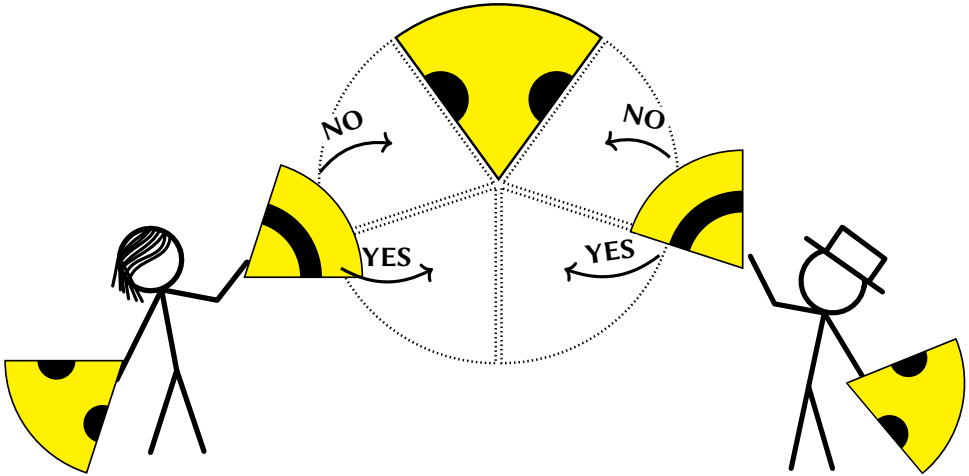
- Secrets: Do I fancy him/her?
- To compute: Is there mutual interest?

⇝ Secure 2-party AND without computers



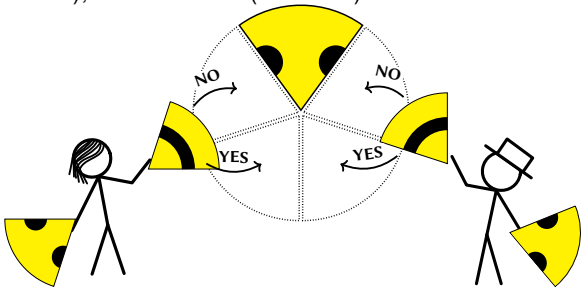
The “Five-Card Trick”

By den Boer (1989, with cards), Tom Verhoeff (with tiles)

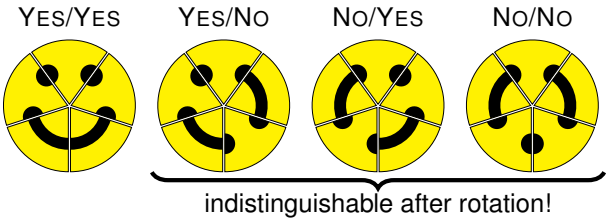


The “Five-Card Trick”

By den Boer (1989, with cards), Tom Verhoeff (with tiles)

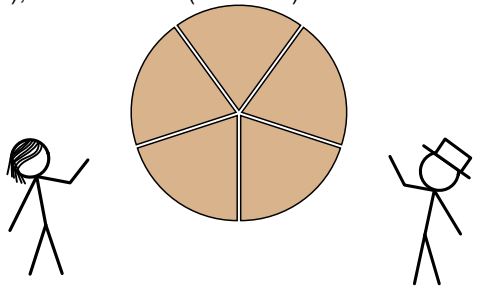


Configurations:

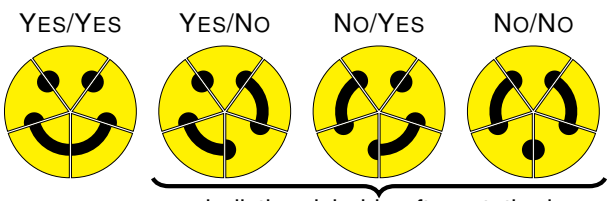


The “Five-Card Trick”

By den Boer (1989, with cards), Tom Verhoeff (with tiles)

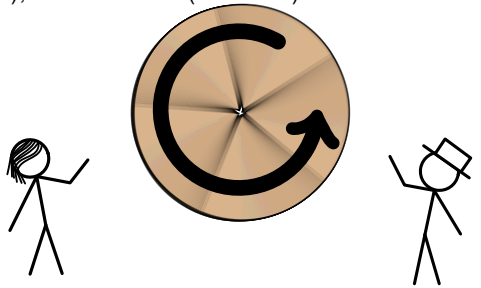


Configurations:

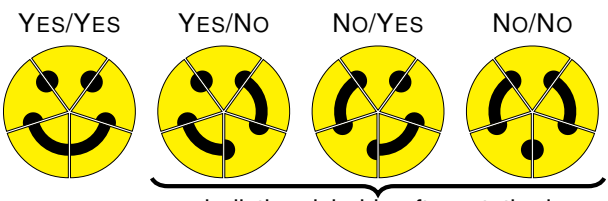


The “Five-Card Trick”

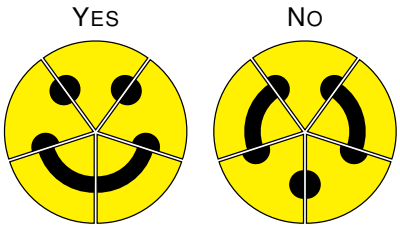
By den Boer (1989, with cards), Tom Verhoeff (with tiles)



Configurations:

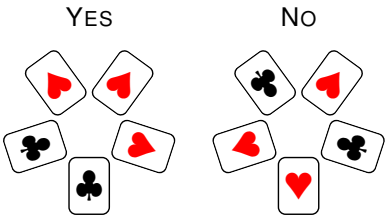


Reveal Tiles . . .



If you say **NO**, you do not learn anything about what the other person said!

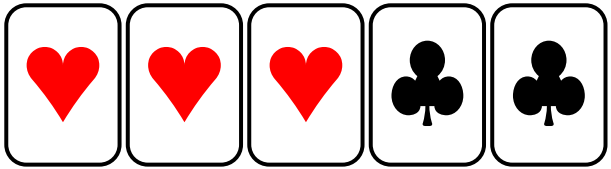
Reveal Cards . . . (equivalent)



If you say **NO**, you do not learn anything about what the other person said!

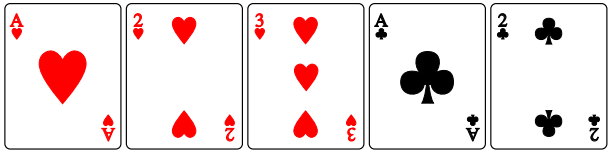
On Our Card Decks

Typical in Card-Based Crypto: Idealized Cards



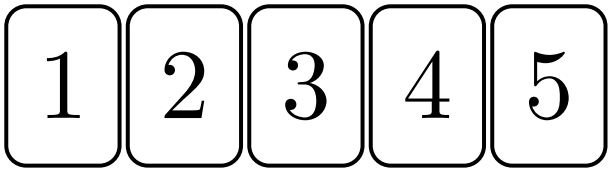
On Our Card Decks

In Reality: We often only have a Standard Deck of Real Cards



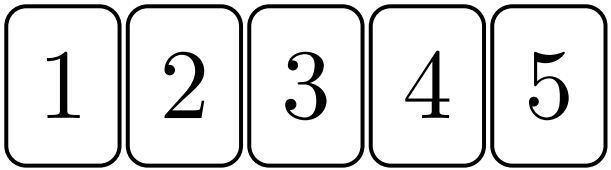
On Our Card Decks

In Reality: We often only have a Standard Deck of Real Cards



On Our Card Decks

In Reality: We often only have a Standard Deck of Real Cards



Disadvantage:

- Slightly more complex objects and bit encoding

Advantage:

- More readily available (does not need multiple decks/custom cards)

General Research Question

We want to compute arbitrary Boolean circuits
For this, we need protocols for AND, NOT and Bit COPY

Main Question: What are the best such protocols?

Criteria:

1. Number of cards used
2. Running time behavior (finite vs. Las Vegas)
3. Number of steps
4. Practicality of Shuffling steps

Our Question today: What are the best AND protocols with a standard deck?

Contribution:

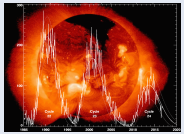
1. We give a new, shorter (Las Vegas) AND protocol with 4 cards (one less than before)
2. We prove by formal methods that it is of shortest length (6 steps)
3. We prove that finite-runtime AND protocols need at least 5 cards

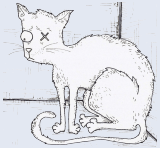
Context: Physical Assumptions in Cryptography

Idealized Hardware

Physical Objects

Physical Processes





Main advantages:

- Transparency and ease of understanding for the user
- Strong, otherwise unachievable, security guarantees

German ID card CC-0, TAN generator/Scratch-off cards copyrighted, Playing cards CC-BY-NC-2.0 by Philippa Willitts, Solar cycle CC-0 by NASA, Schroedinger's cat CC-BY-SA-3.0 Unported by ADA&Neagoe

Our Setting in Card-Based Cryptography

Standard-Deck Cards:

With indistinguishable back sides:  , each card is unique: $\boxed{1}\boxed{2}\boxed{3}$ ...

Two cards encode a bit:

$$\boxed{a}\boxed{b} \triangleq \begin{cases} 0 & \text{for } a < b \\ 1 & \text{for } a > b \end{cases}$$

Three main actions: Turning cards, Shuffling (e.g. card cutting), and Announcing output positions

Requirements for AND

- Input: face-down bits (a, b): 
- Output: face-down bit $(a \wedge b)$: 
- We do not learn anything non-trivial about input and output.

Our Setting in Card-Based Cryptography

Standard-Deck Cards:

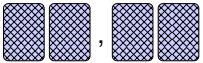
With indistinguishable back sides:  , each card is unique: $\boxed{1}\boxed{2}\boxed{3}$ ...

Two cards encode a bit:

$$\boxed{a}\boxed{b} \triangleq \begin{cases} 0 & \text{for } a < b \\ 1 & \text{for } a > b \end{cases}$$

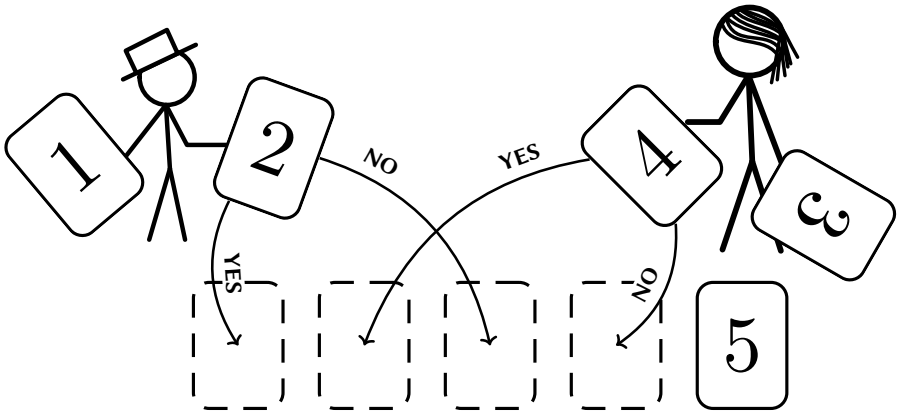
Three main actions: Turning cards, Shuffling (e.g. card cutting), and Announcing output positions

Requirements for AND

- Input: face-down bits (a, b): 
- Output: face-down bit $(a \wedge b)$: 
- We do not learn anything non-trivial about input and output.

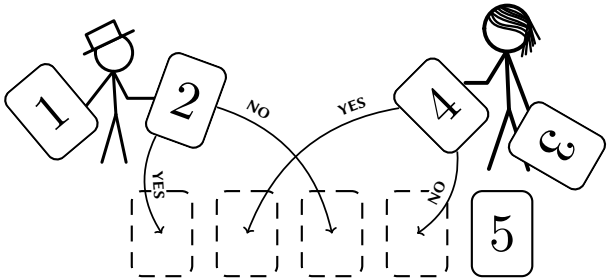
Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Placing Cards



Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Placing Cards



Configurations:

YES/YES (>)

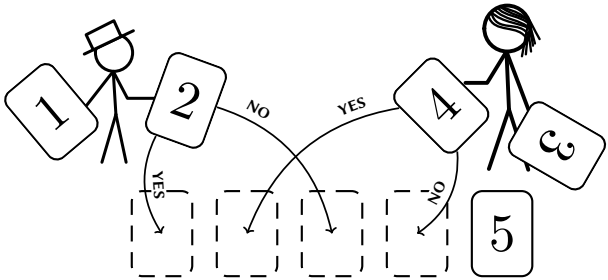
YES/NO (<)

NO/YES (<)

NO/NO (<)

Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Placing Cards



Configurations:

YES/YES (>)

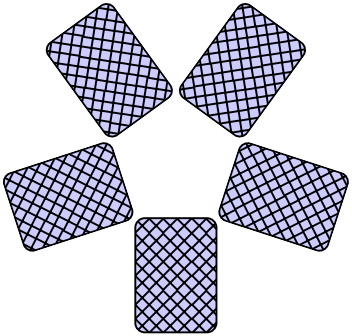
YES/NO (<)

NO/YES (<)

NO/NO (<)

Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Removing Cards (keeping cyclic order)

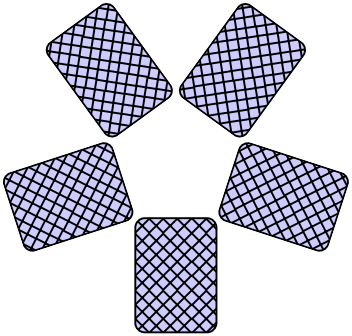


Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Removing Cards (keeping cyclic order)

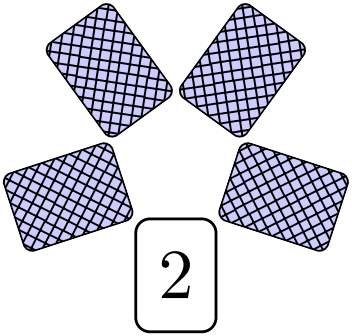
Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Removing Cards (keeping cyclic order)



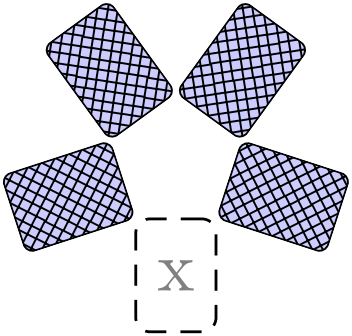
Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Removing Cards (keeping cyclic order)



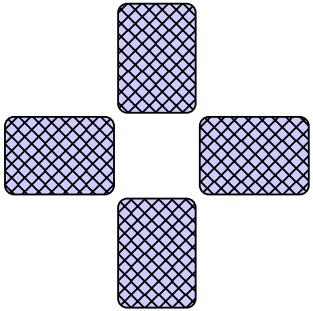
Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Removing Cards (keeping cyclic order)



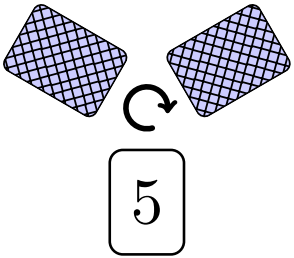
Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Removing Cards (keeping cyclic order)



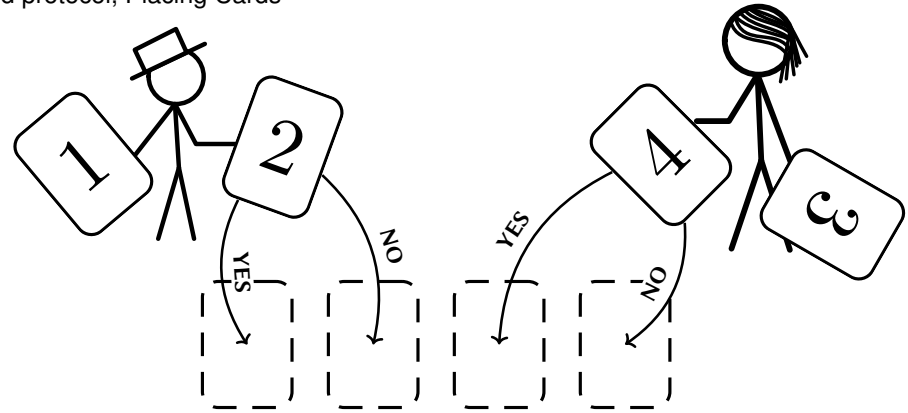
Computing AND with a Standard Deck

5-Card protocol of Niemi and Renvall [NR99], Removing Cards (keeping cyclic order)



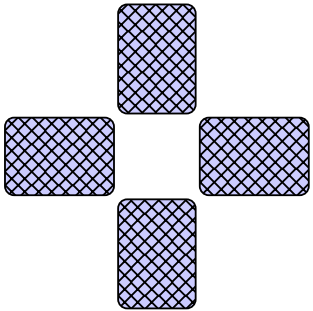
Computing AND with a Standard Deck

Our 4-Card protocol, Placing Cards



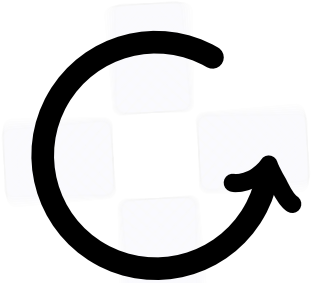
Computing AND with a Standard Deck

Our 4-Card protocol



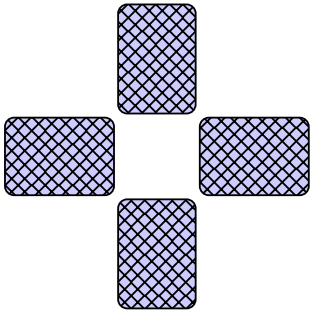
Computing AND with a Standard Deck

Our 4-Card protocol



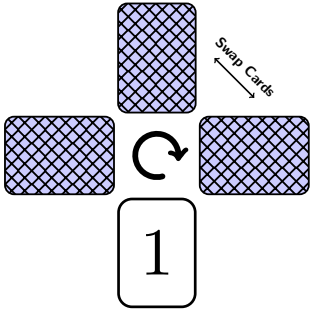
Computing AND with a Standard Deck

Our 4-Card protocol



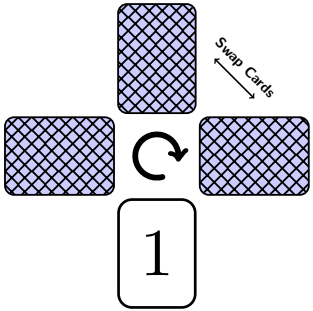
Computing AND with a Standard Deck

Our 4-Card protocol: Case of 1 as separator, others similar



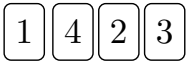
Computing AND with a Standard Deck

Our 4-Card protocol: Case of 1 as separator, others similar

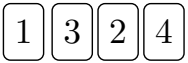


Configurations:

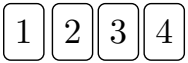
YES/YES (>)



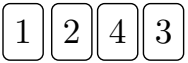
YES/NO (<)



NO/YES (<)

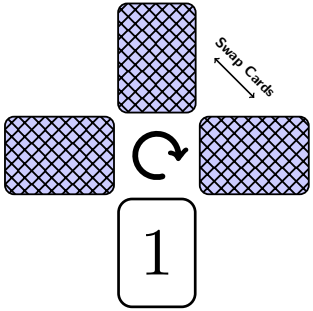


NO/NO (<)



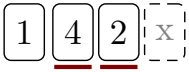
Computing AND with a Standard Deck

Our 4-Card protocol: Case of 1 as separator, others similar

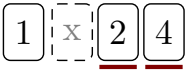


Configurations:

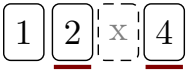
YES/YES (>)



YES/NO (<)



NO/YES (<)

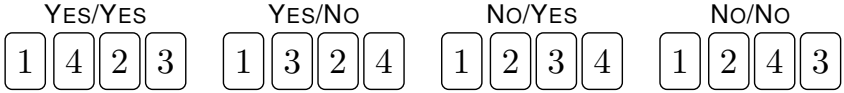


NO/NO (<)

States of a Protocol

A combinatorial way to express the states in which the protocol is in

Previous Way to Speak about Possible Configurations:



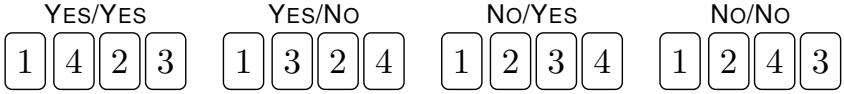
Combinatorial Format:

1423	YES/YES
1324	YES/NO
1234	NO/YES
1243	NO/NO

States of a Protocol

A combinatorial way to express the states in which the protocol is in

Previous Way to Speak about Possible Configurations:

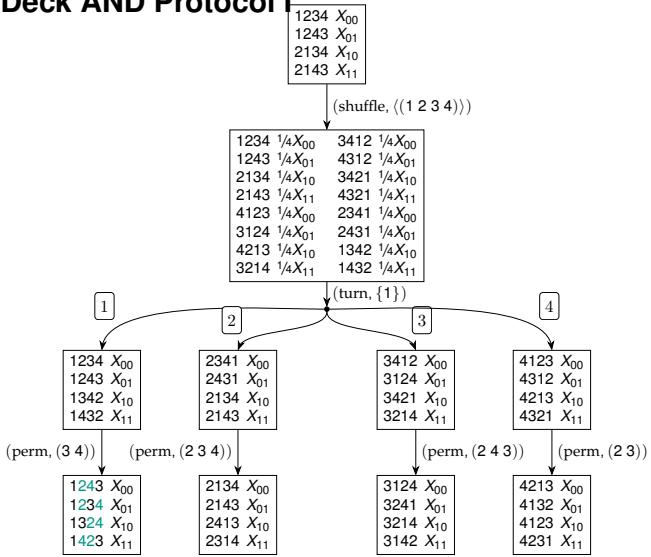


Combinatorial Format (reduced to outputs):

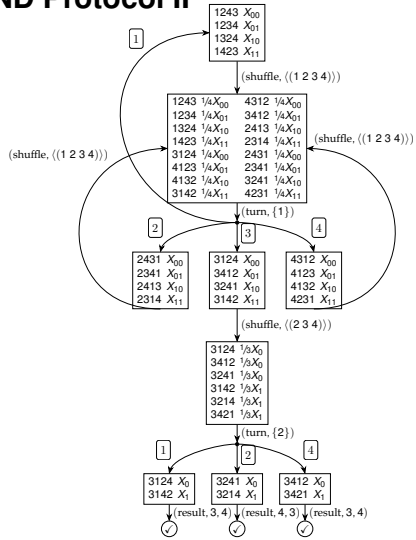
1423	YES
1324	NO
1234	NO
1243	NO

We use “calculus of states” to derive new states from states via turn, and shuffle.

4-Card Standard-Deck AND Protocol I



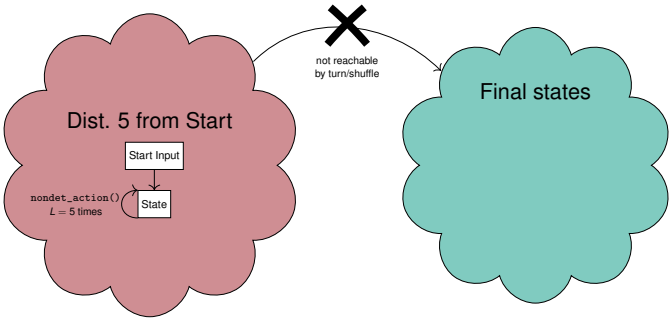
4-Card Standard-Deck AND Protocol II



Example Impossibility Result

Theorem
There is no secure 4-Card AND protocol with a run shorter than 6 steps and standard decks

Proof Idea



Method: Bounded Model Checking

- We used Software Bounded Model Checking: CBMC
- In C code: What does it mean to execute a protocol of length L steps
- Protocol arbitrary via non-deterministic choice: `nondet_action()`
- Assert: We do not reach a final state (i.e. we search a model that violates the assertion)

Results

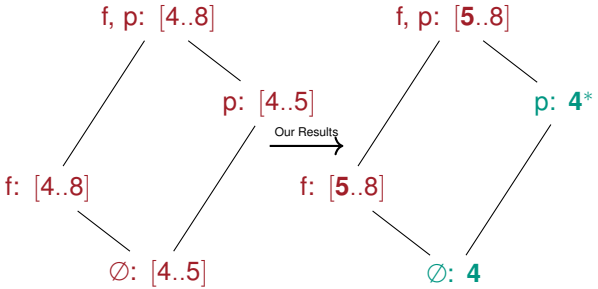
→ Every 4-card standard deck AND protocol needs $\geq$ 6 steps

→ Our protocol (with 6 actions) is optimal

Running times

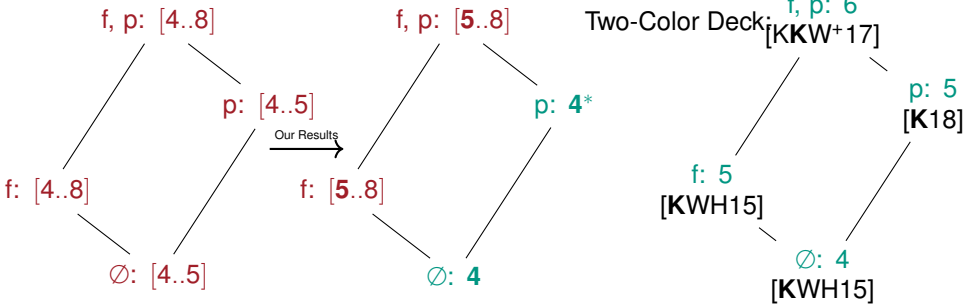
- proof: 57 h on a 6-Core AMD Opteron with 2.40 GHz, 32 GB RAM
- SAT formula contains 150 Mio. clauses

Numbers of Cards for AND with a Standard Deck



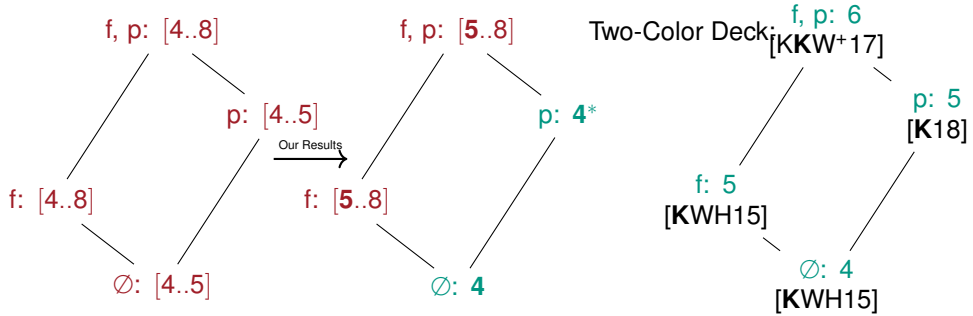
f: finite running time, **p**: practicable shuffling (drawn uniformly random from a permutation group)
* Formal proof of protocol-length minimality with Software Bounded Model Checking

Numbers of Cards for AND with a Standard Deck



f: finite running time, **p**: practicable shuffling (drawn uniformly random from a permutation group)
* Formal proof of protocol-length minimality with Software Bounded Model Checking

Numbers of Cards for AND with a Standard Deck



f: finite running time, **p**: practicable shuffling (drawn uniformly random from a permutation group)
* Formal proof of protocol-length minimality with Software Bounded Model Checking

Future Work:

- Refine CBMC method to work without length bound and with arbitrary decks
- Do finite-runtime standard-deck AND protocols exist with less than 8 cards?