Zero-Knowledge Proofs, with applications to Sudoku & Where’s Waldo?

Mike Rosulek
rosulek@illinois.edu

University of Montana
December 10, 2008
Scenario: Where’s Waldo?

A “Hey Bob, I found Waldo!”
Scenario: Where’s Waldo?

A  “Hey Bob, I found Waldo!”
B  “That was way too fast, I don’t believe you.”
Scenario: Sudoku

A “Hey Bob, check out this brutal Sudoku puzzle!”
Scenario: Sudoku

A “Hey Bob, check out this brutal Sudoku puzzle!”

B “Last week you gave me a puzzle with no solution. I wasted 3 hours.”
Scenario: Sudoku

A "Hey Bob, check out this brutal Sudoku puzzle!"

B "Last week you gave me a puzzle with no solution. I wasted 3 hours."

A "This one has a solution, trust me."
Scenario: Authentication

A “Can I have access to the database? It’s me, Alice.”
Scenario: Authentication

A  “Can I have access to the database? It’s me, Alice.”
B  “OK, send me your password so I know it’s you.”
A Problem of Trust and Information

Alice wants to convince Bob of something

- Waldo is in the picture
- Sudoku puzzle has a solution
- Alice is not an imposter
A Problem of Trust and Information

Alice wants to convince Bob of something

- Waldo is in the picture
- Sudoku puzzle has a solution
- Alice is not an imposter

Bob should not learn “too much”

- Waldo’s location
- Sudoku solution
- Alice’s password
A Problem of Trust and Information

Alice wants to convince Bob of something
- Waldo is in the picture
- Sudoku puzzle has a solution
- Alice is not an imposter

Bob should not learn “too much”
- Waldo’s location
- Sudoku solution
- Alice’s password

What might a possible solution look like?
Solution:

1. Alice places opaque cardboard with hole over picture, revealing Waldo.
2. Bob gets no information about Waldo's location within picture!
Where’s Waldo? Solution

Solution:

1. Alice places opaque cardboard with hole over picture, revealing Waldo
2. Bob gets no information about Waldo’s location within picture!
Solution:

1. Alice places opaque cardboard with hole over picture, revealing Waldo
Where’s Waldo? Solution

Solution:
1. Alice places opaque cardboard with hole over picture, revealing Waldo

Bob gets no information about Waldo’s location within picture!
Solution:

1. Alice places opaque cardboard with hole over picture, revealing Waldo

Bob gets no information about Waldo’s location within picture!
Where’s Waldo? Solution

Solution:

1. Alice places opaque cardboard with hole over picture, revealing Waldo

Bob gets no information about Waldo’s location within picture!
A **zero-knowledge proof** is a way to convince someone of a fact without giving out “any additional information”
A zero-knowledge proof is a way to convince someone of a fact without giving out “any additional information”

What does it mean to

- prove something?
- give out information?
Fuzzy Definition

A **zero-knowledge proof** is a way to convince someone of a fact without giving out “any additional information”

What does it mean to

- prove something?
- give out information?

Classical Definition

A proof is a list of logical steps. Something that Alice can write down and send to Bob.

M "Tea poured into milk tastes different than milk poured into tea."
R "Intriguing. Can you prove it?"
M "I'm just a tea connoisseur. You're the statistician."
R "..."

**M** “Tea poured into milk tastes different than milk poured into tea.”

M “Tea poured into milk tastes different than milk poured into tea.”

R “Intriguing. Can you prove it?”

M “Tea poured into milk tastes different than milk poured into tea.”

R “Intriguing. Can you prove it?”

M “I’m just a tea connoisseur. You’re the statistician.”
A Lady Testing Tea


M “Tea poured into milk tastes different than milk poured into tea.”

R “Intriguing. Can you prove it?”

M “I’m just a tea connoisseur. You’re the statistician.”

R “…”
Fisher’s Smart Idea: Interactive Proof

Random challenge In private, flip a coin to decide which to pour first (tea or milk).

Give cup to Muriel.

Response Muriel guesses.

▶ If Muriel can really tell, she gets it right.

▶ If no difference in two kinds of teas, she has 1/2 chance of guessing correctly.

Repeat n times.

▶ If no difference in two kinds of teas, she has $(1/2)^n$ chance of guessing all correctly.
Fisher’s Smart Idea: Interactive Proof

Random challenge In private, flip a coin to decide which to pour first (tea or milk).
Fisher’s Smart Idea: Interactive Proof

**Random challenge** In private, flip a coin to decide which to pour first (tea or milk). Give cup to Muriel.
Fisher’s Smart Idea: Interactive Proof

Random challenge  In private, flip a coin to decide which to pour first (tea or milk). Give cup to Muriel.

Response  Muriel guesses.
  ▶ If Muriel can really tell, she gets it right.
  ▶ If no difference in two kinds of teas, she has 1/2 chance of guessing correctly.
Fisher’s Smart Idea: Interactive Proof

Random challenge In private, flip a coin to decide which to pour first (tea or milk). Give cup to Muriel.

Response Muriel guesses.

- If Muriel can really tell, she gets it right.
- If no difference in two kinds of teas, she has 1/2 chance of guessing correctly.

Repeat Repeat $n$ times.

- If no difference in two kinds of teas, she has $(1/2)^n$ chance of guessing all correctly.
Bad situation

\( x \) is true

Alice  Bob

I wonder what \( f(x) \) is

"Aha, \( f(x) \)"

This situation is bad if Bob couldn't have computed \( f(x) \) before the interaction.

Interaction transcript gives him computational power.

Want to say: Everything Bob can compute after seeing the transcript, he could have computed before seeing the transcript.
Bad situation

$x$ is true

I wonder what $f(x)$ is

Alice

Bob
**Bad situation**

\[ x \text{ is true} \]

Alice \rightleftharpoons Bob

I wonder what \( f(x) \) is

This situation is bad if Bob couldn't have computed \( f(x) \) before the interaction. Interaction transcript gives him computational power.

Want to say: Everything Bob can compute after seeing the transcript, he could have computed before seeing the transcript.
Bad situation

\( x \) is true

Alice \[\rightarrow\] Bob

I wonder what \( f(x) \) is

```
This situation is bad if Bob couldn’t have computed \( f(x) \) before the interaction.
```

Interaction transcript gives him computational power.

Want to say:

Everything Bob can compute after seeing the transcript, he could have computed before seeing the transcript.
Epistemology: What is Knowledge?

**Bad situation**

- $x$ is true
- Alice → Bob
- Bob: “I wonder what $f(x)$ is”
- Alice ← Bob

- “Aha, $f(x)$”

- This situation is bad if Bob couldn’t have computed $f(x)$ before the interaction.
Bad situation

\[ x \text{ is true} \]

Alice \quad \Rightarrow \quad \text{Bob}

I wonder what \( f(x) \) is

\[ \text{“Aha, } f(x) \text{”} \]

- This situation is bad if Bob couldn’t have computed \( f(x) \) before the interaction
- Interaction transcript gives him computational power
Epistemology: What is Knowledge?

Bad situation

\[ x \text{ is true} \]

Alice \quad \rightarrow \quad Bob

I wonder what \( f(x) \) is

"Aha, \( f(x) \)"

- This situation is bad if Bob couldn’t have computed \( f(x) \) before the interaction
- Interaction transcript gives him computational power

Want to say:

Everything Bob can compute after seeing the transcript, he could have computed before seeing the transcript.
Clever Definition

Interaction is zero-knowledge if Bob could generate transcripts without interacting with Alice:
Clever Definition

Interaction is **zero-knowledge** if Bob could generate transcripts without interacting with Alice:

\[
\text{Alice} \quad \leftrightarrow \quad \text{Bob} \quad f(x)
\]

Whatever Bob could compute *after* seeing the transcript ...
Clever Definition

Interaction is zero-knowledge if Bob could generate transcripts without interacting with Alice:

Whatever Bob could compute after seeing the transcript ...  

... there is a way to compute without interaction!
Apparent Paradox

Paradox?

- Transcript should *convince* Bob of something new
- Bob could have generated transcript himself
Apparent Paradox

Paradox?

- Transcript should *convince* Bob of something new
- Bob could have generated transcript himself

Bob “Alice can drink a gallon of milk in an hour!”

Charlie
Apparent Paradox

Paradox?

- Transcript should *convince* Bob of something new
- Bob could have generated transcript himself

Bob

Charlie

B “Alice can drink a gallon of milk in an hour!”

C “Oh really?”
Apparent Paradox

Paradox?

- Transcript should *convince* Bob of something new
- Bob could have generated transcript himself

Bob “Alice can drink a gallon of milk in an hour!”
Charlie “Oh really?”
Bob “Yes, see this empty milk jug and stopwatch?”
Apparent Paradox

Paradox?

- Transcript should *convince* Bob of something new
- Bob could have generated transcript himself

B  “Alice can drink a gallon of milk in an hour!”
C  “Oh really?”
B  “Yes, see this empty milk jug and stopwatch?”
C  “You dummy, anyone can find an empty milk jug and stopwatch!”
Apparent Paradox

Paradox?
- Transcript should *convince* Bob of something new
- Bob could have generated transcript himself

B “Alice can drink a gallon of milk in an hour!”
C “Oh really?”
B “Yes, see this empty milk jug and stopwatch?”
C “You dummy, anyone can find an empty milk jug and stopwatch!”
B “But I saw her drink it while I timed her!”
Apparent Paradox

Bob “Alice can tell whether tea is poured into milk or vice-versa!”

Charlie
Apparent Paradox

Bob

“Alice can tell whether tea is poured into milk or vice-versa!”

Charlie

“Oh really?”
Apparent Paradox

B “Alice can tell whether tea is poured into milk or vice-versa!”

C “Oh really?”

B “Yes, see all these correctly identified tea cups??”
Apparent Paradox

Bob “Alice can tell whether tea is poured into milk or vice-versa!”

Charlie

Bob already knew the correct responses to challenges

Convinced by how the transcript was generated (in response to his challenges)

C “Oh really?”

B “Yes, see all these correctly identified tea cups??”

C “You dummy, anyone can fill a tea cup and label it!”
Apparent Paradox

B “Alice can tell whether tea is poured into milk or vice-versa!”
C “Oh really?”
B “Yes, see all these correctly identified tea cups??”
C “You dummy, anyone can fill a tea cup and label it!”
B “But I picked the kind of pouring at random, and she was able to answer every time!”
Apparent Paradox

Bob “Alice can tell whether tea is poured into milk or vice-versa!”

Charlie “Oh really?”

Bob “Yes, see all these correctly identified tea cups??”

Charlie “You dummy, anyone can fill a tea cup and label it!”

Bob “But I picked the kind of pouring at random, and she was able to answer every time!”

Bob already knew the correct responses to challenges

- Convinced by how the transcript was generated (in response to his challenges)
Definition [GMR 1985]

A zero-knowledge proof is an interactive protocol satisfying:

- The prover can always convince the verifier of any true statement
Definition [GMR 1985]

A zero-knowledge proof is an interactive protocol satisfying:

- The prover can always convince the verifier of any true statement
- The verifier can’t be convinced of a false statement (even by a cheating prover), except with very low probability
Definition [GMR 1985]

A zero-knowledge proof is an interactive protocol satisfying:

- The prover can always convince the verifier of any true statement
- The verifier can’t be convinced of a false statement (even by a cheating prover), except with very low probability
- There is an efficient procedure to output “same-looking” protocol transcripts
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:

1. Alice randomly relabels {1, ..., 9}
2. Alice writes relabeled solution on scratch card, shows to Bob
3. Bob asks Alice to scratch off either:
   - A particular row
   - A particular column
   - A particular 3 × 3 block
   - Initial positions and checks consistency
4. Repeat n times
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:

1. Alice randomly relabels \{1, \ldots, 9\}
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:
1. Alice randomly relabels \{1, \ldots, 9\}
2. Alice writes relabeled solution on scratch card, shows to Bob
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:

1. Alice randomly relabels {1, \ldots, 9}

2. Alice writes relabeled solution on scratch card, shows to Bob

3. Bob asks Alice to scratch off either:
   - A particular row
   - A particular column
   - A particular 3 \times 3 block
   - Initial positions

and checks consistency
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:

1. Alice randomly relabels \{1, \ldots, 9\}

2. Alice writes relabeled solution on scratch card, shows to Bob

3. Bob asks Alice to scratch off either:
   - A particular row

and checks consistency
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:

1. Alice randomly relabels \{1, \ldots, 9\}
2. Alice writes relabeled solution on scratch card, shows to Bob
3. Bob asks Alice to scratch off either:
   - A particular row
   - A particular column

and checks consistency
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:

1. Alice randomly relabels \{1, \ldots, 9\}
2. Alice writes relabeled solution on scratch card, shows to Bob
3. Bob asks Alice to scratch off either:
   - A particular row
   - A particular column
   - A particular $3 \times 3$ block

and checks consistency
Zero-knowledge protocol:

1. Alice randomly relabels \{1, \ldots, 9\}
2. Alice writes relabeled solution on scratch card, shows to Bob
3. Bob asks Alice to scratch off either:
   - A particular row
   - A particular column
   - A particular 3 \times 3 block
   - Initial positions
   and checks consistency
Sudoku Zero-Knowledge Proof

Zero-knowledge protocol:

1. Alice randomly relabels \( \{1, \ldots, 9\} \)
2. Alice writes relabeled solution on scratch card, shows to Bob
3. Bob asks Alice to scratch off either:
   ▶ A particular row
   ▶ A particular column
   ▶ A particular \(3 \times 3\) block
   ▶ Initial positions
   and checks consistency
4. Repeat \( n \) times
Observation

If Alice can answer all challenges successfully, her scratch card satisfies:

- Every row, column, block is permutation of \{1, \ldots, 9\}
- Initial positions consistent with relabeling of \{1, \ldots, 9\} in original puzzle
Observation

If Alice can answer all challenges successfully, her scratch card satisfies:

- Every row, column, block is permutation of \{1, \ldots, 9\}
- Initial positions consistent with relabeling of \{1, \ldots, 9\} in original puzzle

Then the original puzzle has a solution.
Observation

If Alice can answer all challenges successfully, her scratch card satisfies:

- Every row, column, block is permutation of \( \{1, \ldots, 9\} \)
- Initial positions consistent with relabeling of \( \{1, \ldots, 9\} \) in original puzzle

Then the original puzzle has a solution.

What if Alice is cheating (there really is no solution)?
Observation

If Alice can answer all challenges successfully, her scratch card satisfies:

- Every row, column, block is permutation of \{1,\ldots,9\}
- Initial positions consistent with relabeling of \{1,\ldots,9\} in original puzzle

Then the original puzzle has a solution.

What if Alice is cheating (there really is no solution)?

\Rightarrow \text{ No scratch card can correctly answer all challenges.}
Suppose Alice tries to prove an incorrect statement. Let $c$ be a challenge that is bad for Alice’s scratch card.

- Bob picks random challenge (28 choices)
- With probability $1/28$, Bob chooses $c$ and Alice is caught!
- With probability $\leq 27/28$, Alice’s cheating undetected
Suppose Alice tries to prove an incorrect statement. Let $c$ be a challenge that is bad for Alice’s scratch card.

- Bob picks random challenge (28 choices)
- With probability $1/28$, Bob chooses $c$ and Alice is caught!
- With probability $\leq 27/28$, Alice’s cheating undetected

**Key Idea**

Repeat protocol $n$ times. Alice cheats undetected in all rounds with probability $(27/28)^n \approx (1/2)^{0.05n}$
Suppose Alice tries to prove an incorrect statement. Let $c$ be a challenge that is bad for Alice’s scratch card.

- Bob picks random challenge (28 choices)
- With probability $1/28$, Bob chooses $c$ and Alice is caught!
- With probability $\leq 27/28$, Alice’s cheating undetected

**Key Idea**

Repeat protocol $n$ times. Alice cheats undetected in all rounds with probability $(27/28)^n \approx (1/2)^{0.05n}$

When $n = 2500$, Alice caught with 99% probability.
If Alice follows protocol (there is a solution), then each round transcript is:
If Alice follows protocol (there is a solution), then each round transcript is:

- Random permutation of \{1, \ldots, 9\} in random row,
If Alice follows protocol (there is a solution), then each round transcript is:

- Random permutation of \( \{1, \ldots, 9\} \) in random row,
- Random permutation of \( \{1, \ldots, 9\} \) in random column,
If Alice follows protocol (there is a solution), then each round transcript is:

- Random permutation of \( \{1, \ldots, 9\} \) in random row,
- Random permutation of \( \{1, \ldots, 9\} \) in random column,
- Random permutation of \( \{1, \ldots, 9\} \) in random block, or
If Alice follows protocol (there is a solution), then each round transcript is:

- Random permutation of \( \{1, \ldots, 9\} \) in random row,
- Random permutation of \( \{1, \ldots, 9\} \) in random column,
- Random permutation of \( \{1, \ldots, 9\} \) in random block, or
- Random relabeling of original puzzle’s initial positions.
If Alice follows protocol (there is a solution), then each round transcript is:

- Random permutation of \( \{1, \ldots, 9\} \) in random row,
- Random permutation of \( \{1, \ldots, 9\} \) in random column,
- Random permutation of \( \{1, \ldots, 9\} \) in random block, or
- Random relabeling of original puzzle’s initial positions

Each of these Bob can generated himself (without the solution)!
Zero-Knowledge Proofs for *Everything*?

We have a zero-knowledge proof protocol for Sudoku, so what?
Zero-Knowledge Proofs for *Everything*?

We have a zero-knowledge proof protocol for Sudoku, so what?

**Theorem [Yato 2003]**

\( n \times n \) Sudoku is NP-complete. (Take cs332 and cs531!)

Zero-Knowledge Proofs for *Everything*?

We have a zero-knowledge proof protocol for Sudoku, so what?

**Theorem [Yato 2003]**

$n \times n$ Sudoku is NP-complete. (Take cs332 and cs531!)

Every (practical) statement can be expressed in terms of the solvability of a (generalized) Sudoku instance.
Zero-Knowledge Proofs for *Everything*?

We have a zero-knowledge proof protocol for Sudoku, so what?

**Theorem [Yato 2003]**

\[ n \times n \text{ Sudoku is NP-complete. (Take cs332 and cs531!)} \]

Every (practical) statement can be expressed in terms of the solvability of a (generalized) Sudoku instance.

- Given statement \( x \), can compute puzzle \( S(x) \)
We have a zero-knowledge proof protocol for Sudoku, so what?

**Theorem [Yato 2003]**

\[ n \times n \text{ Sudoku is NP-complete. (Take cs332 and cs531!)} \]

Every (practical) statement can be expressed in terms of the solvability of a (generalized) Sudoku instance.

- Given statement \( x \), can compute puzzle \( S(x) \)
- \( x \) is true \( \iff \) \( S(x) \) is a solvable Sudoku puzzle
We have a zero-knowledge proof protocol for Sudoku, so what?

**Theorem [Yato 2003]**

\[ n \times n \text{ Sudoku is NP-complete. (Take cs332 and cs531!)} \]

Every (practical) statement can be expressed in terms of the solvability of a (generalized) Sudoku instance.

- Given statement \( x \), can compute puzzle \( S(x) \)
- \( x \) is true \( \iff \) \( S(x) \) is a solvable Sudoku puzzle
- To prove \( x \), use Sudoku ZK on \( S(x) \)
We have a zero-knowledge proof protocol for Sudoku, so what?

Theorem [Yato 2003]

$n \times n$ Sudoku is NP-complete. (Take cs332 and cs531!)

Every (practical) statement can be expressed in terms of the solvability of a (generalized) Sudoku instance.

- Given statement $x$, can compute puzzle $S(x)$
- $x$ is true $\iff S(x)$ is a solvable Sudoku puzzle
- To prove $x$, use Sudoku ZK on $S(x)$

Theorem

Every NP statement can be proven in zero-knowledge.

- More efficient protocols for many classes of statements
What are They Good For?

Lots of things!
What are They Good For?

Lots of things!

Disclaimer: ZK proofs very bad for teaching courses:
- Students convinced that professor knows a lot
- Students gained no additional knowledge
ZK Proof of Identity

Public Key DB:

Alice: 583ffb4b3..
Bob: 0fae6535d..
Charlie: 0dd1dd8de..
...

Please give access

Alice <-

Bob

Mike Rosulek (UIUC) Zero-Knowledge Proofs December 10, 2008
ZK Proof of Identity

Public Key DB:
- Alice: 583ffb4b3..
- Bob: 0fae6535d..
- Charlie: 0dd1dd8de..
  ...

ZK proof: I know the secret key corresponding to 583ffb4b3..
ZK Proof of Identity

Public Key DB:
Alice: 583ffb4b3..
Bob: 0fae6535d..
Charlie: 0dd1dd8de..

Ok, Alice

Alice

Bob
ZK Proof of Identity

Public Key DB:
- Alice: 583ff4b4b3...
- Bob: 0fae6535d...
- Charlie: 0dd1dd8de...
- ...
- ...

Alice has her PK published anyway
No one else knows/can compute corresponding secret key
Security “Compiler”

Problem

- Want protocols that give security guarantee, even against malicious parties who deviate from protocol
Problem
- Want protocols that give security guarantee, even against malicious parties who deviate from protocol
- This is hard!
Security “Compiler”

Problem

- Want protocols that give security guarantee, even against malicious parties who deviate from protocol
- This is hard!
- It’s easier to assume that all parties follow the protocol
Problem

- Want protocols that give security guarantee, even against malicious parties who deviate from protocol
- This is hard!
- It’s easier to assume that all parties follow the protocol


1. Design a protocol that is secure if everyone follows protocol (not too hard)
Problem

- Want protocols that give security guarantee, even against malicious parties who deviate from protocol
- This is hard!
- It’s easier to assume that all parties follow the protocol


1. Design a protocol that is secure if everyone follows protocol (not too hard)
2. Parties must prove that they follow protocol at each step
Security Compiler

protocol msg 1

Alice

Bob

ZK proofs leak no further information about secret inputs

If proofs succeed, then parties ran protocol honestly

Security is guaranteed

If proof fails, abort the protocol!
Security Compiler

protocol msg 1

ZK proof: msg 1 is consistent with my (secret) input and protocol

Alice

Bob
Zero-Knowledge Applications

Authentication Secure Protocols Extensions

Security Compiler

protocol msg 1

ZK proof: msg 1 is consistent with my (secret) input and protocol

protocol msg 2

Alice

Bob

Mike Rosulek (UIUC) Zero-Knowledge Proofs December 10, 2008 22 / 26
Security Compiler

protocol msg 1

ZK proof: msg 1 is consistent with my (secret) input and protocol

protocol msg 2

ZK proof: msg 2 is consistent with my (secret) input, protocol, msg 1

...
Zero-Knowledge Applications

Authentication Secure Protocols Extensions

Security Compiler

protocol msg 1

ZK proof: msg 1 is consistent with my (secret) input and protocol

protocol msg 2

ZK proof: msg 2 is consistent with my (secret) input, protocol, msg 1

... 

- ZK proofs leak no further information about secret inputs
intro Zero-Knowledge Applications Authentication Secure Protocols Extensions

Security Compiler

protocol msg 1

ZK proof: msg 1 is consistent with my (secret) input and protocol

protocol msg 2

ZK proof: msg 2 is consistent with my (secret) input, protocol, msg 1

- ▶ ZK proofs leak no further information about secret inputs
- ▶ If proofs succeed, then parties ran protocol honestly
  - ▶ Security is guaranteed

Mike Rosulek (UIUC) Zero-Knowledge Proofs December 10, 2008
ZK proofs leak no further information about secret inputs
- If proofs succeed, then parties ran protocol honestly
  - Security is guaranteed
- If proof fails, abort the protocol!
Conning the Chess Grandmasters

Bob: “Hey Alice, let’s play chess! If you win, I’ll give you $5. Otherwise, you give me $10. You play black.”
Conning the Chess Grandmasters

Bob says, "Hey Alice, let's play chess! If you win, I'll give you $5. Otherwise, you give me $10. You play black."

Alice responds, "OK. I'm a grandmaster. This will be an easy $5."

Bob relays moves to synchronize both chess games.
Conning the Chess Grandmasters

B “Hey Alice, let’s play chess! If you win, I’ll give you $5. Otherwise, you give me $10. You play black.”

A “OK. I’m a grandmaster. This will be an easy $5.”

B “Hey Charlie, let’s play chess! If you win, I’ll give you $5. Otherwise, you give me $10. You play white.”
Conning the Chess Grandmasters

B “Hey Alice, let’s play chess! If you win, I’ll give you $5. Otherwise, you give me $10. You play black.”

A “OK. I’m a grandmaster. This will be an easy $5.”

B “Hey Charlie, let’s play chess! If you win, I’ll give you $5. Otherwise, you give me $10. You play white.”

C “OK. I’m a grandmaster. This will be an easy $5.”
Conning the Chess Grandmasters

B “Hey Alice, let’s play chess! If you win, I’ll give you $5. Otherwise, you give me $10. You play black.”

A “OK. I’m a grandmaster. This will be an easy $5.”

B “Hey Charlie, let’s play chess! If you win, I’ll give you $5. Otherwise, you give me $10. You play white.”

C “OK. I’m a grandmaster. This will be an easy $5.”

Bob relays moves to synchronize both chess games.
Conning the Chess Grandmasters

Bob relays moves to synchronize both chess games.

▶ “Man-in-the-middle attack”
Key Idea:

“Weird things” can happen when multiple protocol instances run concurrently.
Key Idea:

“Weird things” can happen when multiple protocol instances run concurrently.

Chess:

- Bob will always lose to Alice or Charlie 1-on-1.
- Bob will lose to at most one, if playing concurrently.
Chess Grandmasters: Lesson

**Key Idea:**

“Weird things” can happen when multiple protocol instances run concurrently.

**Chess:**

- Bob will always lose to Alice or Charlie 1-on-1.
- Bob will lose to at most one, if playing concurrently.

**Zero-Knowledge:**

- Bob gets no information from Alice’s ZK proof 1-on-1.
- Bob can prove the same statement to Charlie concurrently.
Research into ZK has led to better models:
Better Network Security Models

Research into ZK has led to better models:

Is the ZK proof still zero-knowledge if a player participates in:

- two sessions as verifier?
- one session as verifier, another as prover?
- \( n \) sessions as verifier, \( m \) as prover?
- \( n \) sessions as verifier/prover, and \( m \) arbitrary other protocols?
Better Network Security Models

Research into ZK has led to better models:

Is the ZK proof still zero-knowledge if a player participates in:

- two sessions as verifier?
- one session as verifier, another as prover?
- \( n \) sessions as verifier, \( m \) as prover?
- \( n \) sessions as verifier/prover, and \( m \) arbitrary other protocols?

ZK proofs possible in some of these situations, impossible in others.
Better Network Security Models

Research into ZK has led to better models:

Is the ZK proof still zero-knowledge if a player participates in:

- two sessions as verifier?
- one session as verifier, another as prover?
- $n$ sessions as verifier, $m$ as prover?
- $n$ sessions as verifier/prover, and $m$ arbitrary other protocols?

ZK proofs possible in some of these situations, impossible in others.

Other concerns for ZK definitions:

- “This Sudoku puzzle has a solution”, vs.
- “I know a solution to this Sudoku puzzle”
Conclusions

Defining zero-knowledge:

▶ A proof can be randomized, interactive, have small error probability
▶ Bob “learns nothing” from an interaction if he could have generated transcripts himself

Achieving zero-knowledge:

▶ Scratch-off card protocol for Sudoku (inefficient)
▶ Every statement can be expressed in terms of Sudoku

Using zero-knowledge:

▶ Authenticate without a password
▶ “Compile” any simple protocol (secure when players are honest) into a robust one
Thanks for your attention!

fin.

I hope this was a “talk about zero-knowledge,” not a “zero-knowledge talk.”