Zero-Knowledge Proofs

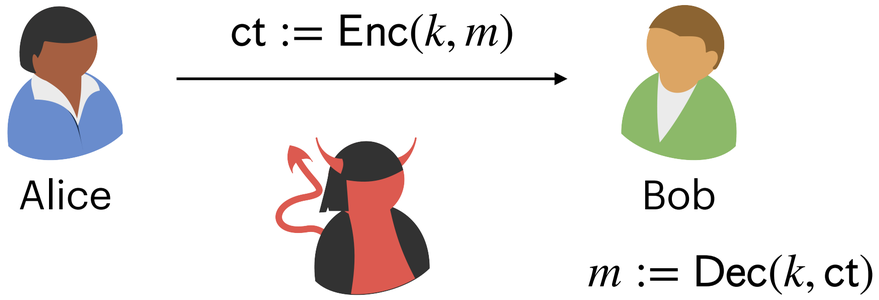
601.442/642 Modern Cryptography

24th March 2026

Recap

Recap

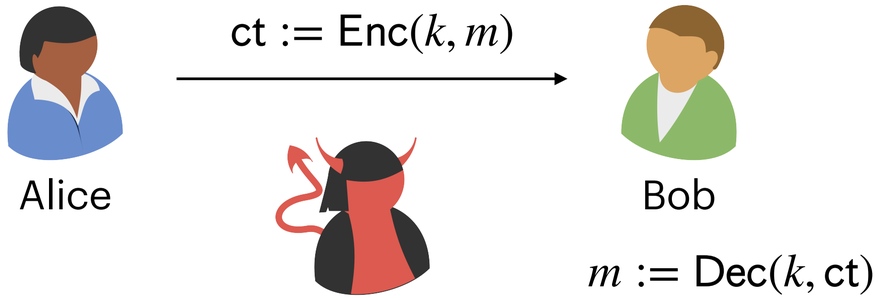
Encryption



Private communication in the presence of an eavesdropper.

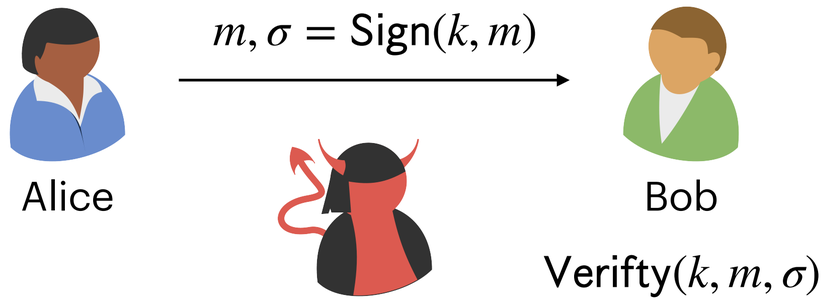
Recap

Encryption



Private communication in the presence of an eavesdropper.

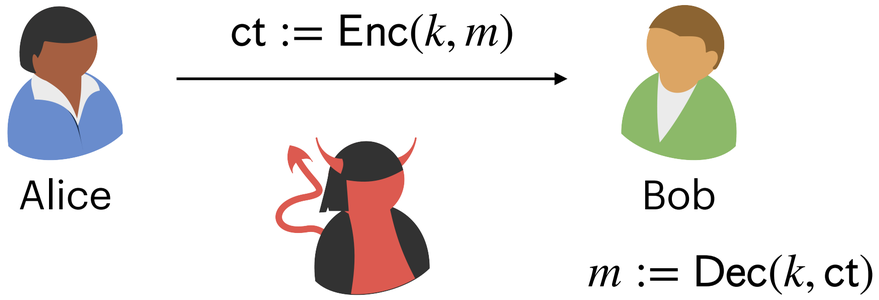
Authentication



Authentic communication in the presence of an impersonator.

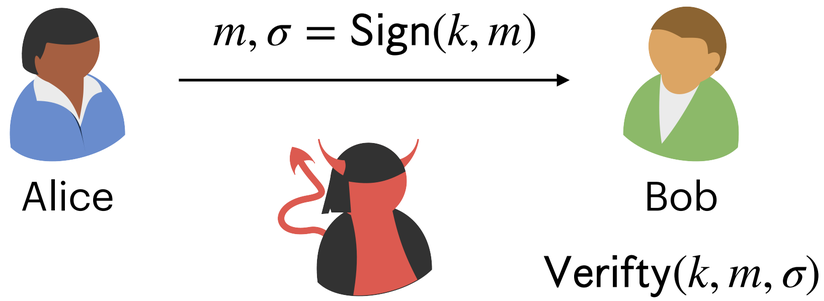
Recap

Encryption



Private communication in the presence of an eavesdropper.

Authentication

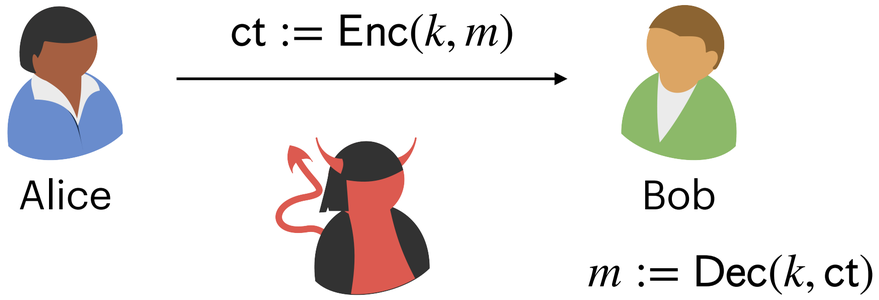


Authentic communication in the presence of an impersonator.

What if Alice and Bob do not trust each other?

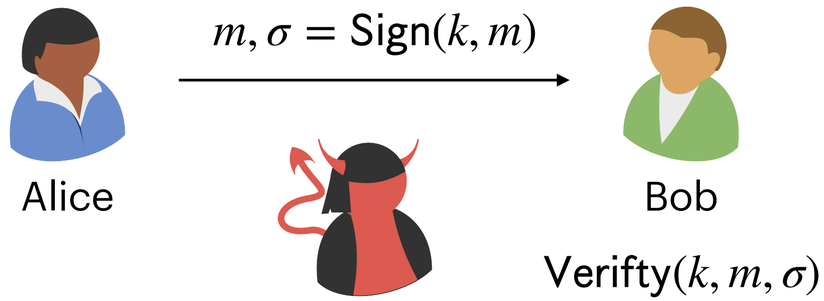
Recap

Encryption



Private communication in the presence of an eavesdropper.

Authentication

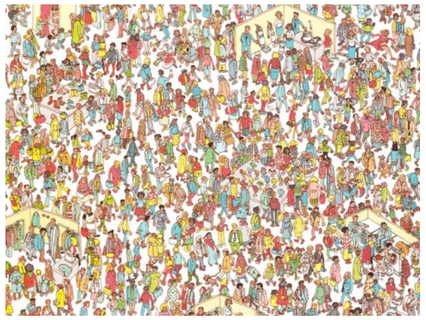


Authentic communication in the presence of an impersonator.

What if Alice and Bob do not trust each other?

What do they even want to achieve?

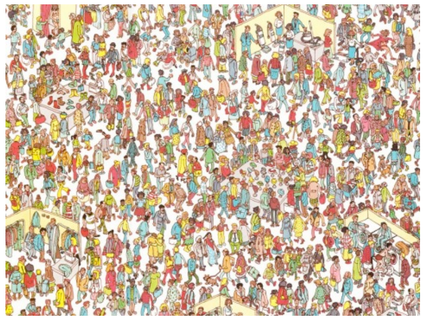
Example: Where’s Waldo?



Alice

Bob

Example: Where’s Waldo?

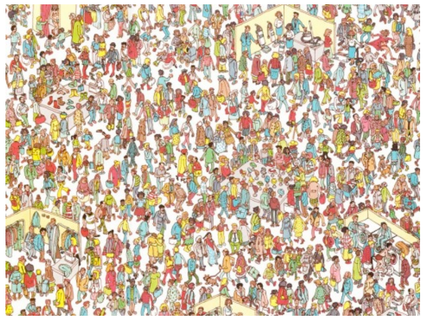


Alice

Bob

“I found Waldo.”

Example: Where’s Waldo?



Alice

“I found Waldo.”

Bob

“That was way too fast, I don’t believe you.
Show me where?”

Example: Authentication

Alice

Bob

Example: Authentication

Alice

Bob

“I’m Alice, can I access the database?”

Example: Authentication

Alice

Bob

“I’m Alice, can I access the database?”

“I don’t trust you. Give me Alice’s password.”

Example: Age Verification

Alice

Bob

Example: Age Verification

Alice

Bob

“I’m older than 21, can I get a beer?”

Example: Age Verification

Alice

“I’m older than 21, can I get a beer?”

Bob

“I don’t believe you. Give me your ID.”

A Problem of Trust and Knowledge

- **Proofs:** Alice wants to convince an untrusting Bob of something.
  - She found Waldo.
  - She is not an imposter.
  - She is over 21.

A Problem of Trust and Knowledge

- **Proofs:** Alice wants to convince an untrusting Bob of something.
  - She found Waldo.
  - She is not an imposter.
  - She is over 21.

A Problem of Trust and Knowledge

- **Proofs:** Alice wants to convince an untrusting Bob of something.
  - She found Waldo.
  - She is not an imposter.
  - She is over 21.

- **Zero-knowledge:** Alice does not trust Bob; Bob should not know anything beyond being convinced of her claim.
  - Waldo's location
  - Alice's password
  - ID (date of birth, address etc.,)

Zero-Knowledge Proofs: History

Invented by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in 1980s.

Zero-Knowledge Proofs: History

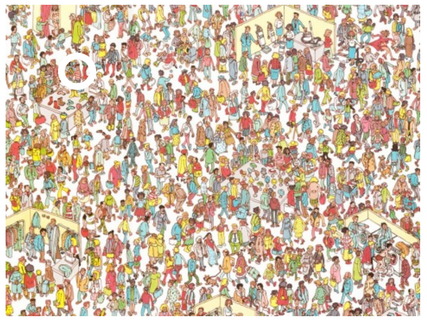
Invented by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in 1980s.



Paper rejected three times! Accepted the fourth time in 1985.

Solution for Where’s Waldo

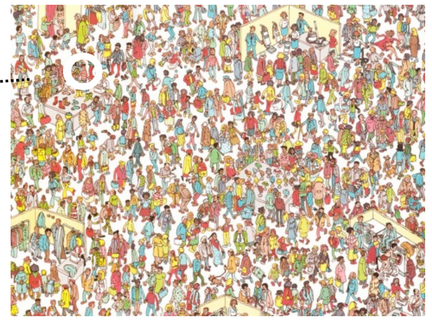
Alice



Bob

Solution for Where’s Waldo

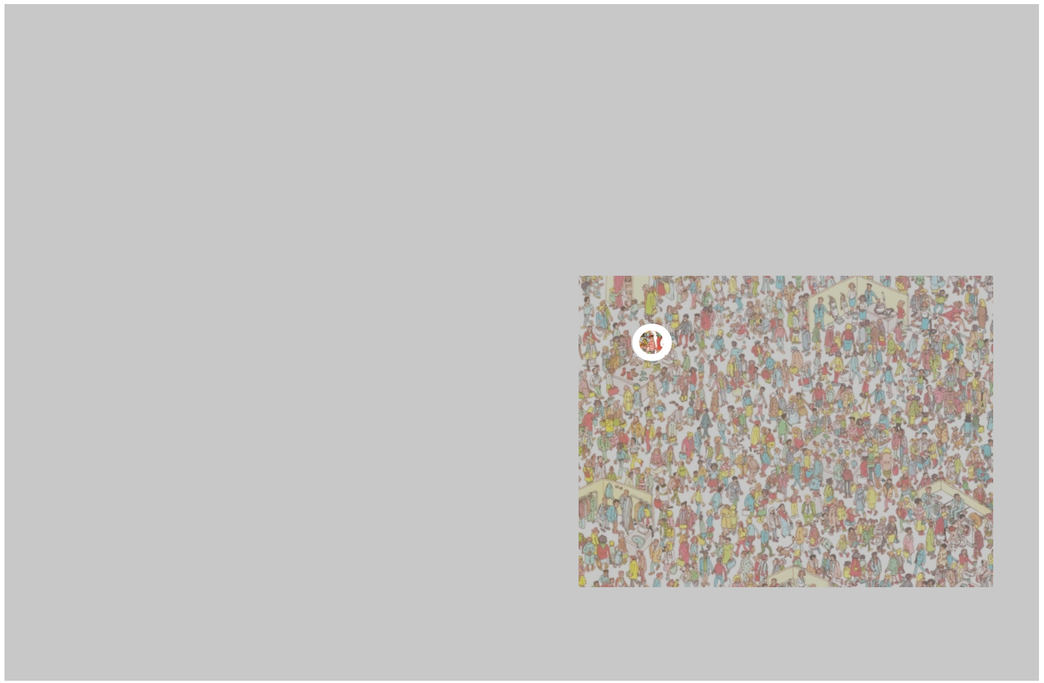
Alice



Bob

Solution for Where’s Waldo

Alice

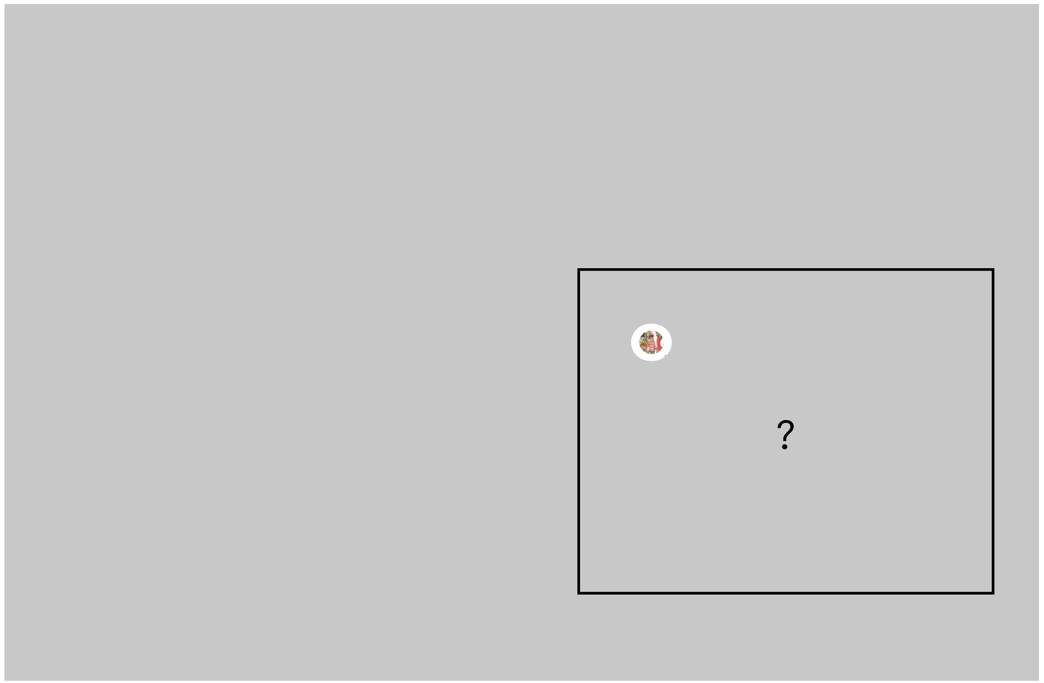


Bob

Alice places a large opaque screen with a hole and reveals Waldo’s location.

Solution for Where’s Waldo

Alice



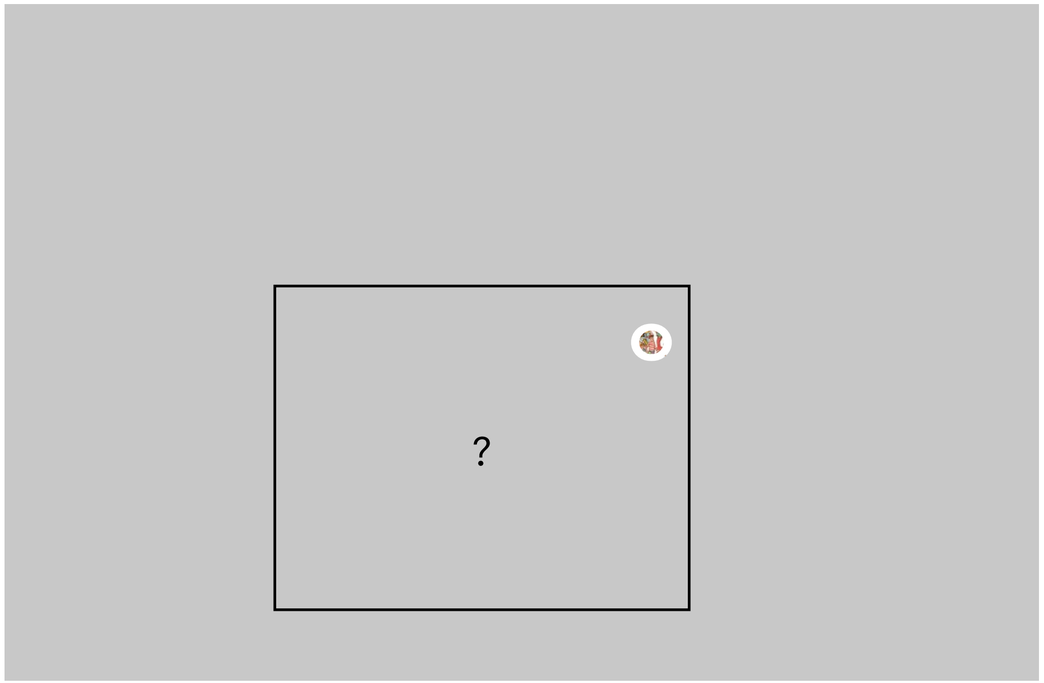
Bob

Alice places a large opaque screen with a hole and reveals Waldo’s location.

Bob does not learn Waldo’s location within the picture.

Solution for Where’s Waldo

Alice



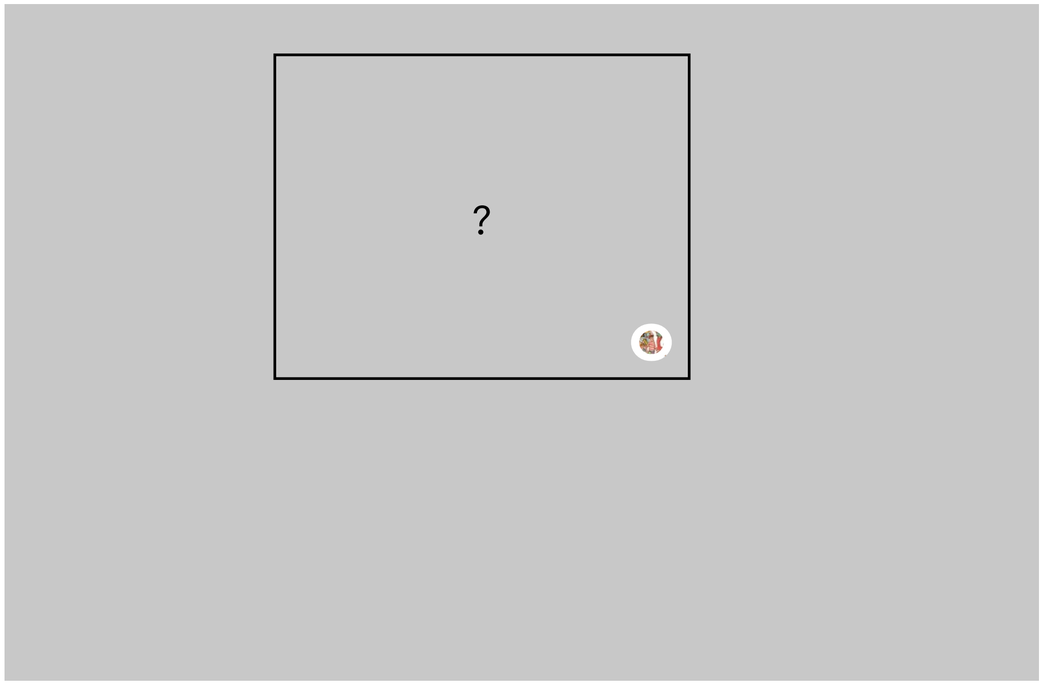
Bob

Alice places a large opaque screen with a hole and reveals Waldo’s location.

Bob does not learn Waldo’s location within the picture.

Solution for Where’s Waldo

Alice



Bob

Alice places a large opaque screen with a hole and reveals Waldo’s location.

Bob does not learn Waldo’s location within the picture.

A Problem of Trust and Knowledge

- **Proofs:** Alice wants to convince an untrusting Bob of something.
  - She found Waldo.
  - She is not an imposter.
  - She is over 21.

- **Zero-knowledge:** Alice does not trust Bob; Bob should not know anything beyond being convinced of her claim.
  - Waldo's location
  - Alice's password
  - ID (date of birth, address etc.,)

A Problem of Trust and Knowledge

- **Proofs:** Alice wants to convince an untrusting Bob of something.
  - She found Waldo.
  - She is not an imposter.
  - She is over 21.

What is a proof?

- **Zero-knowledge:** Alice does not trust Bob; Bob should not know anything beyond being convinced of her claim.
  - Waldo's location
  - Alice's password
  - ID (date of birth, address etc.,)

A Problem of Trust and Knowledge

- **Proofs:** Alice wants to convince an untrusting Bob of something.
  - She found Waldo.
  - She is not an imposter.
  - She is over 21.

What is a proof?

- **Zero-knowledge:** Alice does not trust Bob; Bob should not know anything beyond being convinced of her claim.
  - Waldo's location
  - Alice's password
  - ID (date of birth, address etc.,)

How to define zero-knowledge?

Proofs

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Proof π :

1	$(P \vee Q) \wedge (P \vee R)$	
2	$P \vee Q$	∧E 1
3	$P \vee R$	∧E 1
4	$\quad P$	
5	$\quad P \vee (Q \wedge R)$	∨I 4
6	$\quad Q$	
7	$\quad\quad P$	
8	$\quad\quad P \vee (Q \wedge R)$	∨I 7
9	$\quad\quad R$	
10	$\quad\quad Q \wedge R$	∧I 6, 9
11	$\quad\quad P \vee (Q \wedge R)$	∨I 10
12	$\quad P \vee (Q \wedge R)$	∨E 3, 7–8, 9–11
13	$P \vee (Q \wedge R)$	∨E 2, 4–5, 6–12

Proofs

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Proof π :

1	$(P \vee Q) \wedge (P \vee R)$	
2	$P \vee Q$	∧E 1
3	$P \vee R$	∧E 1
4	$\quad P$	
5	$\quad P \vee (Q \wedge R)$	∨I 4
6	$\quad Q$	
7	$\quad\quad P$	
8	$\quad\quad P \vee (Q \wedge R)$	∨I 7
9	$\quad\quad R$	
10	$\quad\quad Q \wedge R$	∧I 6, 9
11	$\quad\quad P \vee (Q \wedge R)$	∨I 10
12	$\quad P \vee (Q \wedge R)$	∨E 3, 7–8, 9–11
13	$P \vee (Q \wedge R)$	∨E 2, 4–5, 6–12

Showing the proof π convinces a verifier that the statement x is true.

Proofs

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Proof π :

1	$(P \vee Q) \wedge (P \vee R)$	
2	$P \vee Q$	∧E 1
3	$P \vee R$	∧E 1
4	$\quad P$	
5	$\quad P \vee (Q \wedge R)$	∨I 4
6	$\quad Q$	
7	$\quad\quad P$	
8	$\quad\quad P \vee (Q \wedge R)$	∨I 7
9	$\quad\quad R$	
10	$\quad\quad Q \wedge R$	∧I 6, 9
11	$\quad\quad P \vee (Q \wedge R)$	∨I 10
12	$\quad P \vee (Q \wedge R)$	∨E 3, 7–8, 9–11
13	$P \vee (Q \wedge R)$	∨E 2, 4–5, 6–12

Showing the proof π convinces a verifier that the statement x is true.

Verifier should accept the proof if x is true.

Proofs

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Proof π :

1	$(P \vee Q) \wedge (P \vee R)$	
2	$P \vee Q$	∧E 1
3	$P \vee R$	∧E 1
4	$\quad P$	
5	$\quad P \vee (Q \wedge R)$	∨I 4
6	$\quad Q$	
7	$\quad\quad P$	
8	$\quad\quad P \vee (Q \wedge R)$	∨I 7
9	$\quad\quad R$	
10	$\quad\quad Q \wedge R$	∧I 6, 9
11	$\quad\quad P \vee (Q \wedge R)$	∨I 10
12	$\quad P \vee (Q \wedge R)$	∨E 3, 7–8, 9–11
13	$P \vee (Q \wedge R)$	∨E 2, 4–5, 6–12

Showing the proof π convinces a verifier that the statement x is true.

Verifier should accept the proof if x is true.

Verifier should reject the proof if x is false.

Proof System

Helpful to think about a prover trying to convince a verifier that the statement is true.

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Prover

Verifier

Proof System

Helpful to think about a prover trying to convince a verifier that the statement is true.

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Prover

Verifier

"x is true."

Proof System

Helpful to think about a prover trying to convince a verifier that the statement is true.

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Prover

"x is true."

Verifier

"I don't trust you."

Proof System

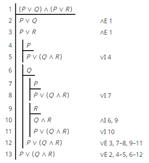
Helpful to think about a prover trying to convince a verifier that the statement is true.

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Prover

"x is true."

Proof π :



Verifier

"I don't trust you."

Proof System

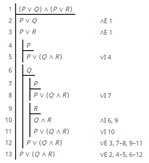
Helpful to think about a prover trying to convince a verifier that the statement is true.

Statement x : $(P \vee Q) \wedge (P \vee R) \implies P \vee (Q \wedge R)$

Prover

"x is true."

Proof π :



Verifier

"I don't trust you."

A proof is needed to convince a verifier that does not trust the prover.

Generalizing to Decision Problems

Statement x

Prover

Verifier

Generalizing to Decision Problems

Statement x

Prover

Verifier

"$x \in L$"

Generalizing to Decision Problems

Statement x

Prover

Language: Set of true statements.

$"x \in L"$

Verifier

Generalizing to Decision Problems

Statement x

Prover

Language: Set of true statements.

Verifier

"$x \in L$"

"I don't trust you."

Generalizing to Decision Problems

Statement x

Prover

Language: Set of true statements.

Verifier

"$x \in L$"

"I don't trust you."

Proof π

Generalizing to Decision Problems

Statement x

Prover

Verifier

"$x \in L$"

Proof π

"I don't trust you."

NP: The verifier can check π in poly($|x|$) time.

Generalizing to Decision Problems

Statement x

Prover

Verifier

"$x \in L$"

"I don't trust you."

Proof π

NP: The verifier can check π in poly($|x|$) time.

Should a proof system necessarily be non-interactive?

Generalizing to Decision Problems

Statement x

Prover

Verifier

"$x \in L$"

"I don't trust you."

Proof π

NP: The verifier can check π in poly($|x|$) time.

Should a proof system necessarily be non-interactive?

Think written proof vs seminar/talk

Proof Systems

- **Proof Systems in CS:** A pair of interactive algorithms (P, V).

Proof Systems

- **Proof Systems in CS:** A pair of interactive algorithms (P, V).
  - **Completeness:** P can be used to convince V to accept a true statement.

Proof Systems

- **Proof Systems in CS:** A pair of interactive algorithms (P, V).
  - **Completeness:** P can be used to convince V to accept a true statement.
  - **Soundness:** V will always reject a false statement.

Proof Systems

- **Proof Systems in CS:** A pair of interactive algorithms (P, V).
  - **Completeness:** P can be used to convince V to accept a true statement.
  - **Soundness:** V will always reject a false statement.
  - **Efficient Verification**: V must be polynomial time in the length of the statement.

Proof Systems

Proof System

A proof system for a language $L \subset \{0,1\}^*$ is a pair of interactive algorithms (P, V), where V is a PPT algorithm, if it satisfies the following properties.

Proof Systems

Proof System

A proof system for a language $L \subset \{0,1\}^*$ is a pair of interactive algorithms (P, V), where V is a PPT algorithm, if it satisfies the following properties.

- **Completeness:** For every $x \in L$,

$$\Pr\left[\mathsf{Out}_V\left[P(x) \leftrightarrow V(x)\right] = 1\right] = 1.$$

Proof Systems

Proof System

A proof system for a language $L \subset \{0,1\}^*$ is a pair of interactive algorithms (P, V), where V is a PPT algorithm, if it satisfies the following properties.

- **Completeness:** For every $x \in L$,

Verifier's final output

$$\Pr\left[\mathsf{Out}_V\left[P(x) \leftrightarrow V(x)\right] = 1\right] = 1.$$

Proof Systems

Proof System

A proof system for a language $L \subset \{0,1\}^*$ is a pair of interactive algorithms (P, V), where V is a PPT algorithm, if it satisfies the following properties.

- **Completeness:** For every $x \in L$,

Verifier's final output

$$\Pr\left[\mathsf{Out}_V\left[P(x) \leftrightarrow V(x)\right] = 1\right] = 1.$$

- **Soundness:** There exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $x \notin L$ and all $\widehat{P}$

$$\Pr\left[\mathsf{Out}_V\left[\widehat{P}(x) \leftrightarrow V(x)\right] = 1\right] \leq \mathsf{negl}(|x|).$$

Proof Systems

Proof System

A proof system for a language $L \subset \{0,1\}^*$ is a pair of interactive algorithms (P, V), where V is a PPT algorithm, if it satisfies the following properties.

- **Completeness:** For every $x \in L$,

Verifier's final output

$$\Pr\left[\mathsf{Out}_V\left[P(x) \leftrightarrow V(x)\right] = 1\right] = 1.$$

- **Soundness:** There exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $x \notin L$ and all $\widehat{P}$

$$\Pr\left[\mathsf{Out}_V\left[\widehat{P}(x) \leftrightarrow V(x)\right] = 1\right] \leq \mathsf{negl}(|x|).$$

Prover is not required to be efficient (similar to definition of NP).

Proof Systems

Proof System

A proof system for a language $L \subset \{0,1\}^*$ is a pair of interactive algorithms (P, V), where V is a PPT algorithm, if it satisfies the following properties.

- **Completeness:** For every $x \in L$,

Verifier's final output

$$\Pr\left[\mathrm{Out}_V\left[P(x) \leftrightarrow V(x)\right] = 1\right] = 1.$$

- **Soundness:** There exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $x \notin L$ and all $\widehat{P}$

$$\Pr\left[\mathrm{Out}_V\left[\widehat{P}(x) \leftrightarrow V(x)\right] = 1\right] \leq \mathsf{negl}(|x|).$$

Prover is not required to be efficient (similar to definition of NP).

Prover and verifier can use randomness.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

Verifier

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

Verifier

“I can tell then apart with a glance.”

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

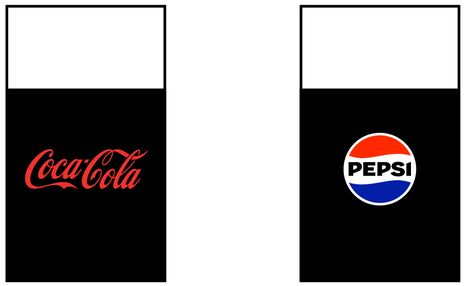
Verifier

“I can tell then apart with a glance.”

“I can’t tell them apart even after tasting them.”

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.



Glass A Glass B

Prover

Verifier

“I can tell then apart with a glance.”

“I can’t tell them apart even after tasting them.”

How do we convince the Verifier?

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

Verifier

“I can tell then apart with a glance.”

“I can’t tell them apart even after tasting them.”

Solution:

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

Verifier

“I can tell then apart with a glance.”

“I can’t tell them apart even after tasting them.”

Solution: The verifier swaps the glasses with probability 1/2.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

Verifier

"I can tell then apart with a glance."

"I can't tell them apart even after tasting them."

Solution: The verifier swaps the glasses with probability 1/2.

The prover guesses if the verifier swapped the glasses.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

Verifier

"I can tell then apart with a glance."

"I can't tell them apart even after tasting them."

Solution: The verifier swaps the glasses with probability 1/2.

The prover guesses if the verifier swapped the glasses.

If the prover can tell them apart with a glance, it always wins.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

Prover

Glass A

Glass B

Verifier

"I can tell then apart with a glance."

"I can't tell them apart even after tasting them."

Solution: The verifier swaps the glasses with probability 1/2.

The prover guesses if the verifier swapped the glasses.

If the prover can tell them apart with a glance, it always wins.

Else, it wins with probability 1/2. Repeat to amplify soundness.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.
    - Languages in NP have a non-interactive proof.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.
  - Languages in NP have a non-interactive proof.
  - Interactive proofs can prove statements in languages not known to be in NP.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.
  - Languages in NP have a non-interactive proof.
  - Interactive proofs can prove statements in languages not known to be in NP.
    - Single prover [Shamir]: IP = PSPACE.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.
  - Languages in NP have a non-interactive proof.
  - Interactive proofs can prove statements in languages not known to be in NP.
    - Single prover [Shamir]: IP = PSPACE.
    - Multiple provers [Babai-Fortnow-Lund]: MIP = NEXP.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.
  - Languages in NP have a non-interactive proof.
  - Interactive proofs can prove statements in languages not known to be in NP.
    - Single prover [Shamir]: IP = PSPACE.
    - Multiple provers [Babai-Fortnow-Lund]: MIP = NEXP.
- Achieving privacy guarantee for the prover.

The Power of Interaction and Randomness

- Proof system for languages beyond what is captured by non-interactive proofs.
  - Languages in NP have a non-interactive proof.
  - Interactive proofs can prove statements in languages not known to be in NP.
    - Single prover [Shamir]: IP = PSPACE.
    - Multiple provers [Babai-Fortnow-Lund]: MIP = NEXP.
- Achieving privacy guarantee for the prover.
  - **Zero knowledge:** Verifier learns nothing from the proof beyond the validity of the statement.