Zero-Knowledge Proofs - II

CS 601.642/442 Modern Cryptography

Fall 2022

Zero-Knowledge Proofs for **NP**

Theorem

If one-way permutations exist, then every language in **NP** *has a zero-knowledge interactive proof.*

Zero-Knowledge Proofs for **NP**

Theorem

If one-way permutations exist, then every language in **NP** *has a zero-knowledge interactive proof.*

- The assumption can in fact be relaxed to just one-way functions

Zero-Knowledge Proofs for **NP**

Theorem

If one-way permutations exist, then every language in **NP** *has a zero-knowledge interactive proof.*

- The assumption can in fact be relaxed to just one-way functions
- Think: How to prove the theorem?

Zero-Knowledge Proofs for **NP**

Theorem

If one-way permutations exist, then every language in **NP** *has a zero-knowledge interactive proof.*

- The assumption can in fact be relaxed to just one-way functions
- Think: How to prove the theorem?
- Construct ZK proof for every **NP** language?

Zero-Knowledge Proofs for **NP**

Theorem

If one-way permutations exist, then every language in **NP** *has a zero-knowledge interactive proof.*

- The assumption can in fact be relaxed to just one-way functions
- Think: How to prove the theorem?
- Construct ZK proof for every **NP** language?
- Not efficient!

Zero-Knowledge Proofs for **NP** (contd.)

Proof Strategy:

Zero-Knowledge Proofs for **NP** (contd.)

Proof Strategy:

Step 1: Construct a ZK proof for an **NP**-complete language. We will consider *Graph 3-Coloring*: language of all graphs whose vertices can be colored using only three colors s.t. no two connected vertices have the same color

Zero-Knowledge Proofs for **NP** (contd.)

Proof Strategy:

- Step 1: Construct a ZK proof for an **NP**-complete language. We will consider *Graph 3-Coloring*: language of all graphs whose vertices can be colored using only three colors s.t. no two connected vertices have the same color
- Step 2: To construct ZK proof for any **NP** language L, do the following:

Zero-Knowledge Proofs for **NP** (contd.)

Proof Strategy:

Step 1: Construct a ZK proof for an **NP**-complete language. We will consider *Graph 3-Coloring*: language of all graphs whose vertices can be colored using only three colors s.t. no two connected vertices have the same color

Step 2: To construct ZK proof for any **NP** language L, do the following:

- Given instance x and witness w, P and V reduce x into an instance x' of Graph 3-coloring using Cook's (deterministic) reduction

Zero-Knowledge Proofs for **NP** (contd.)

Proof Strategy:

Step 1: Construct a ZK proof for an **NP**-complete language. We will consider *Graph 3-Coloring*: language of all graphs whose vertices can be colored using only three colors s.t. no two connected vertices have the same color

Step 2: To construct ZK proof for any **NP** language L, do the following:

- Given instance x and witness w, P and V reduce x into an instance x' of Graph 3-coloring using Cook's (deterministic) reduction
- P also applies the reduction to witness w to obtain witness w' for x'

Zero-Knowledge Proofs for **NP** (contd.)

Proof Strategy:

Step 1: Construct a ZK proof for an **NP**-complete language. We will consider *Graph 3-Coloring*: language of all graphs whose vertices can be colored using only three colors s.t. no two connected vertices have the same color

Step 2: To construct ZK proof for any **NP** language L, do the following:

- Given instance x and witness w, P and V reduce x into an instance x' of Graph 3-coloring using Cook's (deterministic) reduction
- P also applies the reduction to witness w to obtain witness w' for x'
- Now, P and V can run the ZK proof from Step 1 on common input x'

Physical ZK Proof for Graph 3-Coloring

- Consider graph $G = (V, E)$. Let C be a 3-coloring of V given to P

Physical ZK Proof for Graph 3-Coloring

- Consider graph $G = (V, E)$. Let C be a 3-coloring of V given to P
- P picks a random permutation π over colors $\{1, 2, 3\}$ and colors G according to $\pi(C)$. It hides each vertex in V inside a locked box

Physical ZK Proof for Graph 3-Coloring

- Consider graph $G = (V, E)$. Let C be a 3-coloring of V given to P
- P picks a random permutation π over colors $\{1, 2, 3\}$ and colors G according to $\pi(C)$. It hides each vertex in V inside a locked box
- V picks a random edge (u, v) in E

Physical ZK Proof for Graph 3-Coloring

- Consider graph $G = (V, E)$. Let C be a 3-coloring of V given to P
- P picks a random permutation π over colors $\{1, 2, 3\}$ and colors G according to $\pi(C)$. It hides each vertex in V inside a locked box
- V picks a random edge (u, v) in E
- P opens the boxes corresponding to u, v. V accepts if u and v have different colors, and rejects otherwise

Physical ZK Proof for Graph 3-Coloring

- Consider graph $G = (V, E)$. Let C be a 3-coloring of V given to P
- P picks a random permutation π over colors $\{1, 2, 3\}$ and colors G according to $\pi(C)$. It hides each vertex in V inside a locked box
- V picks a random edge (u, v) in E
- P opens the boxes corresponding to u, v. V accepts if u and v have different colors, and rejects otherwise
- The above process is repeated $n|E|$ times

Physical ZK Proof for Graph 3-Coloring

- Consider graph $G = (V, E)$. Let C be a 3-coloring of V given to P
- P picks a random permutation π over colors $\{1, 2, 3\}$ and colors G according to $\pi(C)$. It hides each vertex in V inside a locked box
- V picks a random edge (u, v) in E
- P opens the boxes corresponding to u, v. V accepts if u and v have different colors, and rejects otherwise
- The above process is repeated $n|E|$ times
- **Intuition for Soundness:** In each iteration, cheating prover is caught with probability $\frac{1}{|E|}$

Physical ZK Proof for Graph 3-Coloring

- Consider graph $G = (V, E)$. Let C be a 3-coloring of V given to P
- P picks a random permutation π over colors $\{1, 2, 3\}$ and colors G according to $\pi(C)$. It hides each vertex in V inside a locked box
- V picks a random edge (u, v) in E
- P opens the boxes corresponding to u, v. V accepts if u and v have different colors, and rejects otherwise
- The above process is repeated $n|E|$ times
- **Intuition for Soundness:** In each iteration, cheating prover is caught with probability $\frac{1}{|E|}$
- **Intuition for ZK:** In each iteration, V only sees something it knew before – two random (but different) colors

Towards ZK Proof for Graph 3-Coloring

- To "digitze" the above proof, we need to implement locked boxes

Towards ZK Proof for Graph 3-Coloring

- To "digitze" the above proof, we need to implement locked boxes
- Need two properties from digital locked boxes:

Towards ZK Proof for Graph 3-Coloring

- To "digitze" the above proof, we need to implement locked boxes
- Need two properties from digital locked boxes:
  - **Hiding**: V should not be able to see the content inside a locked box

Towards ZK Proof for Graph 3-Coloring

- To "digitze" the above proof, we need to implement locked boxes
- Need two properties from digital locked boxes:
  - **Hiding**: V should not be able to see the content inside a locked box
  - **Binding**: P should not be able to modify the content inside a box once its locked

Commitment Schemes

- Digital analogue of locked boxes

Commitment Schemes

- Digital analogue of locked boxes
- Two phases:

Commitment Schemes

- Digital analogue of locked boxes
- Two phases:

  Commit phase: Sender locks a value v inside a box

Commitment Schemes

- Digital analogue of locked boxes
- Two phases:

  Commit phase: Sender locks a value v inside a box

  Open phase: Sender unlocks the box and reveals v

Commitment Schemes

- Digital analogue of locked boxes
- Two phases:
  Commit phase: Sender locks a value v inside a box
  Open phase: Sender unlocks the box and reveals v
- Can be implemented using interactive protocols, but we will consider non-interactive case. Both commit and reveal phases will consist of single messages

Commitment Schemes: Definition

Definition (Commitment)

A randomized polynomial-time algorithm Com is called a *commitment scheme* for n-bit strings if it satisfies the following properties:

- **Binding:** For all $v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^n$, it holds that $\mathsf{Com}(v_0; r_0) \neq \mathsf{Com}(v_1; r_1)$

Commitment Schemes: Definition

Definition (Commitment)

A randomized polynomial-time algorithm Com is called a *commitment scheme* for n-bit strings if it satisfies the following properties:

- **Binding:** For all $v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^n$, it holds that $\mathsf{Com}(v_0; r_0) \neq \mathsf{Com}(v_1; r_1)$
- **Hiding:** For every non-uniform PPT distinguisher D, there exists a negligible function $\nu(\cdot)$ s.t. for every $v_0, v_1 \in \{0,1\}^n$, D distinguishes between the following distributions with probability at most $\nu(n)$

Commitment Schemes: Definition

Definition (Commitment)

A randomized polynomial-time algorithm Com is called a *commitment scheme* for n-bit strings if it satisfies the following properties:

- **Binding:** For all $v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^n$, it holds that $\mathsf{Com}(v_0; r_0) \neq \mathsf{Com}(v_1; r_1)$
- **Hiding:** For every non-uniform PPT distinguisher D, there exists a negligible function $\nu(\cdot)$ s.t. for every $v_0, v_1 \in \{0,1\}^n$, D distinguishes between the following distributions with probability at most $\nu(n)$
  - $\left\{ r \xleftarrow{\$} \{0,1\}^n \ : \ \mathsf{Com}(v_0; r) \right\}$

Commitment Schemes: Definition

Definition (Commitment)

A randomized polynomial-time algorithm Com is called a *commitment scheme* for n-bit strings if it satisfies the following properties:

- **Binding:** For all $v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^n$, it holds that $\mathsf{Com}(v_0; r_0) \neq \mathsf{Com}(v_1; r_1)$
- **Hiding:** For every non-uniform PPT distinguisher D, there exists a negligible function $\nu(\cdot)$ s.t. for every $v_0, v_1 \in \{0,1\}^n$, D distinguishes between the following distributions with probability at most $\nu(n)$
  - $\left\{r \xleftarrow{\$} \{0,1\}^n \ : \ \mathsf{Com}(v_0; r)\right\}$
  - $\left\{r \xleftarrow{\$} \{0,1\}^n \ : \ \mathsf{Com}(v_1; r)\right\}$

Commitment Schemes: Remarks

- The previous definition only guarantees hiding for one commitment

Commitment Schemes: Remarks

- The previous definition only guarantees hiding for one commitment
- **Multi-value Hiding:** Just like encryption, we can define multi-value hiding property for commitment schemes

Commitment Schemes: Remarks

- The previous definition only guarantees hiding for one commitment
- **Multi-value Hiding:** Just like encryption, we can define multi-value hiding property for commitment schemes
- Using hybrid argument (as for public-key encryption), we can prove that any commitment scheme satisfies multi-value hiding

Commitment Schemes: Remarks

- The previous definition only guarantees hiding for one commitment
- **Multi-value Hiding:** Just like encryption, we can define multi-value hiding property for commitment schemes
- Using hybrid argument (as for public-key encryption), we can prove that any commitment scheme satisfies multi-value hiding
- **Corollary:** One-bit commitment implies string commitment

Construction of Bit Commitments

Construction: Let f be a OWP, h be the hard core predicate for f

Construction of Bit Commitments

Construction: Let f be a OWP, h be the hard core predicate for f

Commit phase: Sender computes $\mathsf{Com}(b; r) = f(r), b \oplus h(r)$. Let C denote the commitment.

Construction of Bit Commitments

Construction: Let f be a OWP, h be the hard core predicate for f

Commit phase: Sender computes $\mathsf{Com}(b; r) = f(r), b \oplus h(r)$. Let C denote the commitment.

Open phase: Sender reveals (b, r). Receiver accepts if $C = (f(r), b \oplus h(r))$, and rejects otherwise

Construction: Let f be a OWP, h be the hard core predicate for f

Commit phase: Sender computes $\mathsf{Com}(b; r) = f(r), b \oplus h(r)$. Let C denote the commitment.

Open phase: Sender reveals (b, r). Receiver accepts if $C = (f(r), b \oplus h(r))$, and rejects otherwise

Security:

Construction of Bit Commitments

Construction: Let f be a OWP, h be the hard core predicate for f

Commit phase: Sender computes $\mathsf{Com}(b;r) = f(r), b \oplus h(r)$. Let C denote the commitment.

Open phase: Sender reveals (b, r). Receiver accepts if $C = (f(r), b \oplus h(r))$, and rejects otherwise

Security:

- Binding follows from construction since f is a permutation

Construction of Bit Commitments

Construction: Let f be a OWP, h be the hard core predicate for f

Commit phase: Sender computes $\mathsf{Com}(b;r) = f(r), b \oplus h(r)$. Let C denote the commitment.

Open phase: Sender reveals (b, r). Receiver accepts if $C = (f(r), b \oplus h(r))$, and rejects otherwise

Security:

- Binding follows from construction since f is a permutation
- Hiding follows in the same manner as IND-CPA security of public-key encryption scheme constructed from trapdoor permutations

ZK Proof for Graph 3-Coloring

Common Input: $G = (V, E)$**, where** $|V| = n$

P**'s witness: Colors** $\mathsf{color}_1, \ldots, \mathsf{color}_n \in \{1, 2, 3\}$

Protocol (P, V)**:** Repeat the following procedure $n|E|$ times *using fresh randomness*

$P \to V$: P chooses a random permutation π over $\{1, 2, 3\}$. For every $i \in [n]$, it computes $C_i = \mathsf{Com}(\widetilde{\mathsf{color}}_i)$ where $\widetilde{\mathsf{color}}_i = \pi(\mathsf{color}_i)$. It sends $(C_1, \ldots, C_n)$ to V

$V \to P$: V chooses a random edge $(i, j) \in E$ and sends it to P

$P \to V$: Prover opens C_i and C_j to reveal $(\widetilde{\mathsf{color}}_i, \widetilde{\mathsf{color}}_j)$

V: If the openings of C_i, C_j are valid and $\widetilde{\mathsf{color}}_i \neq \widetilde{\mathsf{color}}_j$, then V accepts the proof. Otherwise, it rejects.

Proof of Soundness

- If G is not 3-colorable, then for any coloring $\mathsf{color}_1, \ldots, \mathsf{color}_n$, there exists at least one edge which has the same colors on both endpoints

Proof of Soundness

- If G is not 3-colorable, then for any coloring $\mathsf{color}_1, \ldots, \mathsf{color}_n$, there exists at least one edge which has the same colors on both endpoints
- From the binding property of Com, it follows that $C_1, \ldots, C_n$ have unique openings $\widetilde{\mathsf{color}}_1, \ldots, \widetilde{\mathsf{color}}_n$

Proof of Soundness

- If G is not 3-colorable, then for any coloring $\mathsf{color}_1, \ldots, \mathsf{color}_n$, there exists at least one edge which has the same colors on both endpoints
- From the binding property of Com, it follows that $C_1, \ldots, C_n$ have unique openings $\widetilde{\mathsf{color}}_1, \ldots, \widetilde{\mathsf{color}}_n$
- Combining the above, let $(i^*, j^*) \in E$ be s.t. $\widetilde{\mathsf{color}}_{i^*} = \widetilde{\mathsf{color}}_{j^*}$

Proof of Soundness

- If G is not 3-colorable, then for any coloring $\mathsf{color}_1, \ldots, \mathsf{color}_n$, there exists at least one edge which has the same colors on both endpoints
- From the binding property of Com, it follows that $C_1, \ldots, C_n$ have unique openings $\widetilde{\mathsf{color}}_1, \ldots, \widetilde{\mathsf{color}}_n$
- Combining the above, let $(i^*, j^*) \in E$ be s.t. $\widetilde{\mathsf{color}}_{i^*} = \widetilde{\mathsf{color}}_{j^*}$
- Then, with probability $\frac{1}{|E|}$, V chooses $i = i^*, j = j^*$ and catches P

Proof of Soundness

- If G is not 3-colorable, then for any coloring $\mathsf{color}_1, \ldots, \mathsf{color}_n$, there exists at least one edge which has the same colors on both endpoints
- From the binding property of Com, it follows that $C_1, \ldots, C_n$ have unique openings $\widetilde{\mathsf{color}}_1, \ldots, \widetilde{\mathsf{color}}_n$
- Combining the above, let $(i^*, j^*) \in E$ be s.t. $\widetilde{\mathsf{color}}_{i^*} = \widetilde{\mathsf{color}}_{j^*}$
- Then, with probability $\frac{1}{|E|}$, V chooses $i = i^*, j = j^*$ and catches P
- In $n|E|$ independent repetitions, P successfully cheats in all repetitions with probability at most

$$\left(1 - \frac{1}{|E|}\right)^{n|E|} \approx e^{-n}$$

Proving Zero Knowledge: Strategy

- Will prove that a single iteration of (P, V) is zero knowledge
- For the full protocol, use the following (read proof online):

Theorem

Sequential repetition of any ZK protocol is also ZK

- To prove that a single iteration of (P, V) is ZK, we need to do the following:
  - Construct a Simulator S for every PPT V^*
  - Prove that expected runtime of S is polynomial
  - Prove that the output distribution of S is correct (i.e., indistinguishable from real execution)
- Intuition for proving ZK for a single iteration: V only sees two random colors. Hiding property of Com guarantees that everything else remains hidden from V.

Proving Zero Knowledge: Simulator

Simulator $S(x = G, z)$**:**

Proving Zero Knowledge: Simulator

Simulator $S(x = G, z)$:

- Choose a random edge $(i', j') \xleftarrow{\$} E$ and pick random colors $\mathsf{color}'_{i'}, \mathsf{color}'_{j'} \xleftarrow{\$} \{1, 2, 3\}$ s.t. $\mathsf{color}'_{i'} \neq \mathsf{color}'_{j'}$. For every other $k \in [n] \setminus \{i', j'\}$, set $\mathsf{color}'_k = 1$

Proving Zero Knowledge: Simulator

Simulator $S(x = G, z)$**:**

- Choose a random edge $(i', j') \xleftarrow{\$} E$ and pick random colors $\mathsf{color}'_{i'}, \mathsf{color}'_{j'} \xleftarrow{\$} \{1, 2, 3\}$ s.t. $\mathsf{color}'_{i'} \neq \mathsf{color}'_{j'}$. For every other $k \in [n] \setminus \{i', j'\}$, set $\mathsf{color}'_k = 1$
- For every $\ell \in [n]$, compute $C_\ell = \mathsf{Com}(\mathsf{color}'_\ell)$

Proving Zero Knowledge: Simulator

Simulator $S(x = G, z)$**:**

- Choose a random edge $(i', j') \xleftarrow{\$} E$ and pick random colors $\mathsf{color}'_{i'}, \mathsf{color}'_{j'} \xleftarrow{\$} \{1, 2, 3\}$ s.t. $\mathsf{color}'_{i'} \neq \mathsf{color}'_{j'}$. For every other $k \in [n] \setminus \{i', j'\}$, set $\mathsf{color}'_k = 1$
- For every $\ell \in [n]$, compute $C_\ell = \mathsf{Com}(\mathsf{color}'_\ell)$
- Emulate execution of $V^*(x, z)$ by feeding it $(C_1, \ldots, C_n)$. Let (i, j) denote its response

Proving Zero Knowledge: Simulator

Simulator $S(x = G, z)$**:**

- Choose a random edge $(i', j') \xleftarrow{\$} E$ and pick random colors $\mathsf{color}'_{i'}, \mathsf{color}'_{j'} \xleftarrow{\$} \{1, 2, 3\}$ s.t. $\mathsf{color}'_{i'} \neq \mathsf{color}'_{j'}$. For every other $k \in [n] \setminus \{i', j'\}$, set $\mathsf{color}'_k = 1$
- For every $\ell \in [n]$, compute $C_\ell = \mathsf{Com}(\mathsf{color}'_\ell)$
- Emulate execution of $V^*(x, z)$ by feeding it $(C_1, \ldots, C_n)$. Let (i, j) denote its response
- If $(i, j) = (i', j')$, then feed the openings of C_i, C_j to V^* and output its view. Otherwise, restart the above procedure, at most $n|E|$ times

Proving Zero Knowledge: Simulator

Simulator $S(x = G, z)$**:**

- Choose a random edge $(i', j') \xleftarrow{\$} E$ and pick random colors $\mathsf{color}'_{i'}, \mathsf{color}'_{j'} \xleftarrow{\$} \{1, 2, 3\}$ s.t. $\mathsf{color}'_{i'} \neq \mathsf{color}'_{j'}$. For every other $k \in [n] \setminus \{i', j'\}$, set $\mathsf{color}'_k = 1$
- For every $\ell \in [n]$, compute $C_\ell = \mathsf{Com}(\mathsf{color}'_\ell)$
- Emulate execution of $V^*(x, z)$ by feeding it $(C_1, \ldots, C_n)$. Let (i, j) denote its response
- If $(i, j) = (i', j')$, then feed the openings of C_i, C_j to V^* and output its view. Otherwise, restart the above procedure, at most $n|E|$ times
- If simulation has not succeeded after $n|E|$ attempts, then output `fail`

Correctness of Simulation

Hybrid Experiments:

- H_0: Real execution

Hybrid Experiments:

- H_0: Real execution
- H_1: Hybrid simulator S' that acts like the real prover (using witness $\mathsf{color}_1, \ldots, \mathsf{color}_n$), except that it also chooses $(i', j') \xleftarrow{\$} E$ at random. If $(i', j') \neq (i, j)$, then try again. After $n|E|$ attempts, output `fail`

Correctness of Simulation

Hybrid Experiments:

- H_0: Real execution
- H_1: Hybrid simulator S' that acts like the real prover (using witness $\mathsf{color}_1, \ldots, \mathsf{color}_n$), except that it also chooses $(i', j') \xleftarrow{\$} E$ at random. If $(i', j') \neq (i, j)$, then try again. After $n|E|$ attempts, output `fail`
- H_2: Simulator S

Correctness of Simulation (contd.)

- $H_0 \approx H_1$: If S' does not output `fail`, then H_0 and H_1 are identical. Since (i, j) and (i', j') are independently chosen, S' fails with probability at most:

$$\left(1 - \frac{1}{|E|}\right)^{n|E|} \approx e^{-n}$$

  Therefore, H_0 and H_1 are statistically indistinguishable

Correctness of Simulation (contd.)

- $H_0 \approx H_1$: If S' does not output `fail`, then H_0 and H_1 are identical. Since (i, j) and (i', j') are independently chosen, S' fails with probability at most:

$$\left(1 - \frac{1}{|E|}\right)^{n|E|} \approx e^{-n}$$

  Therefore, H_0 and H_1 are statistically indistinguishable

- $H_1 \approx H_2$: The only difference between H_1 and H_2 is that for all $k \in [n] \setminus \{i', j'\}$, C_k is a commitment to $\pi(\mathsf{color}_k)$ in H_1 and a commitment to 1 in H_2. Then, from the multi-value hiding property of Com, it follows that $H_1 \approx H_2$

Additional Reading

- Zero-knowledge Proofs for Nuclear Disarmament [Glaser-Barak-Goldston'14]

Additional Reading

- Zero-knowledge Proofs for Nuclear Disarmament [Glaser-Barak-Goldston'14]
- Non-black-box Simulation [Barak'01]

Additional Reading

- Zero-knowledge Proofs for Nuclear Disarmament [Glaser-Barak-Goldston'14]
- Non-black-box Simulation [Barak'01]
- Concurrent Composition of Zero-Knowledge Proofs [Dwork-Naor-Sahai'98, Richardson-Kilian'99, Kilian-Petrank'01,Prabhakaran-Rosen-Sahai'02]

Additional Reading

- Zero-knowledge Proofs for Nuclear Disarmament [Glaser-Barak-Goldston'14]
- Non-black-box Simulation [Barak'01]
- Concurrent Composition of Zero-Knowledge Proofs [Dwork-Naor-Sahai'98, Richardson-Kilian'99, Kilian-Petrank'01,Prabhakaran-Rosen-Sahai'02]
- Non-malleable Commitments and ZK Proofs [Dolev-Dwork-Naor'91]