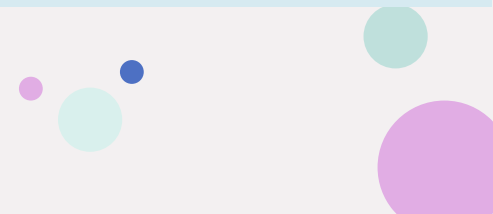




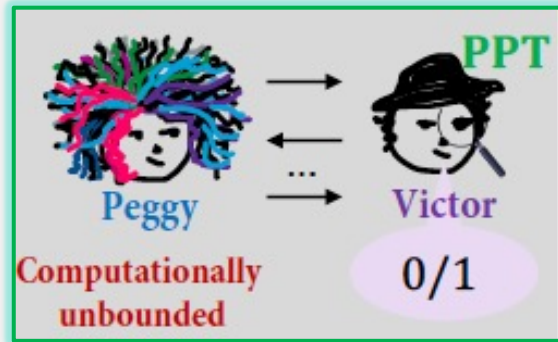
Lecture 19

Malicious Verifier ZKPs, Commitment Schemes, ZKP for NP

CS409: Introduction to Cryptography
Sruthi Sekar



Recap



Interactive Proof



Honest Verifier Perfect/Computational Zero-knowledge

Definition 1: Zero-knowledge (Interactive) Proof System

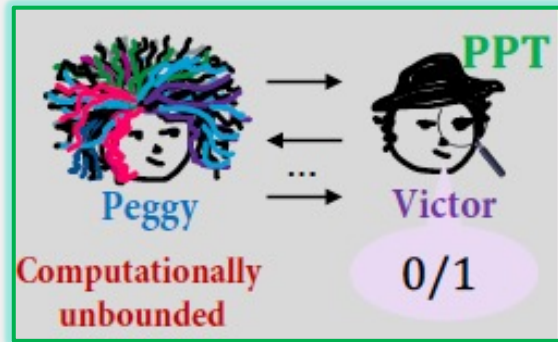
An IP protocol (P, V) for a language L is an interactive proof system if the following hold:

- **Completeness:** for every $x \in L$, $\Pr[1 \leftarrow \langle P, V \rangle(x)] = 1$
- **Soundness:** for every $x \notin L$, $\exists \text{negl}$ such that for every P^* , $\Pr[1 \leftarrow \langle P^*, V \rangle(x)] \leq \text{negl}(n)$
- **Honest Verifier Perfect ZK:** An IP protocol (P, V) for L is **honest-verifier ZK** if there exists PPT simulator **Sim** such that for all distinguishers **Spector** and $\forall x \in L$

$$\Pr[\text{Spector}(\text{View}_V(\langle P, V \rangle(x))) = 1] - \Pr[\text{Spector}(\text{Sim}(x)) = 1] = 0$$

****View_V(⟨P, V⟩(x)):** refers to the view seen by the Verifier (transcript along with verifier messages/randomness)

Recap



Interactive Proof



Honest Verifier Perfect/Computational Zero-knowledge

Definition 1: Zero-knowledge (Interactive) Proof System

An IP protocol (P, V) for a language L is an interactive proof system if the following hold:

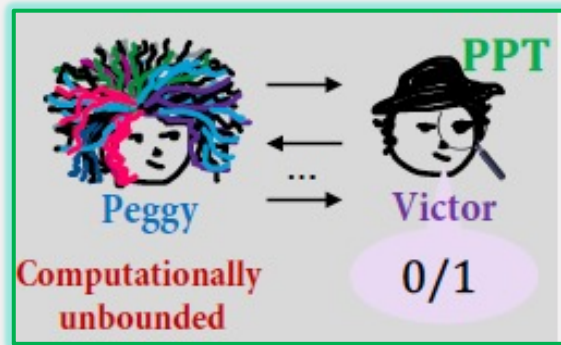
- **Completeness:** for every $x \in L$, $\Pr[1 \leftarrow \langle P, V \rangle(x)] = 1$
- **Soundness:** for every $x \notin L$, $\exists \text{negl}$ such that for every P^* , $\Pr[1 \leftarrow \langle P^*, V \rangle(x)] \leq \text{negl}(n)$

- **Honest Verifier ~~Perfect~~ ^{computational} ZK:** An IP protocol (P, V) for L is **honest-verifier ZK** if there exists PPT simulator **Sim** such that for all distinguishers **Spector** and $\forall x \in L$

$$\Pr[\text{Spector}(\text{View}_V(\langle P, V \rangle(x))) = 1] - \Pr[\text{Spector}(\text{Sim}(x)) = 1] \leq \text{negl}(n)$$

****View_V(⟨P, V⟩(x)):** refers to the view seen by the Verifier (transcript along with verifier messages/randomness)

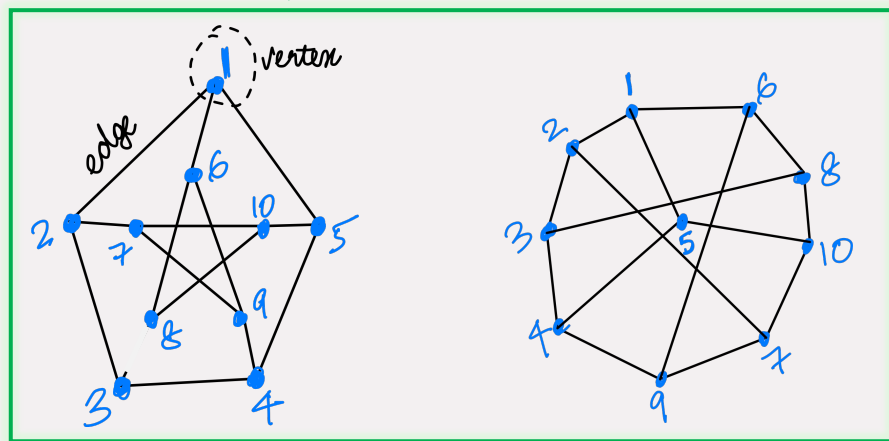
Recap



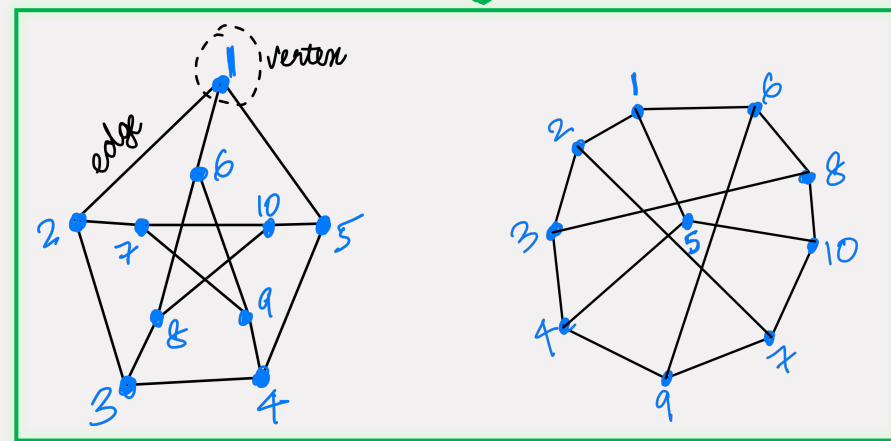
Interactive Proof



Honest Verifier Perfect/Computational Zero-knowledge

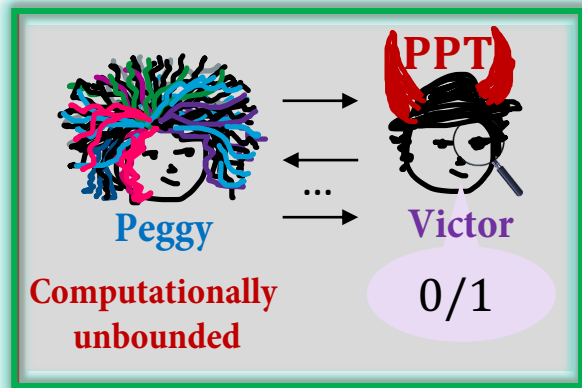


Graph Non-Isomorphism (GNI)



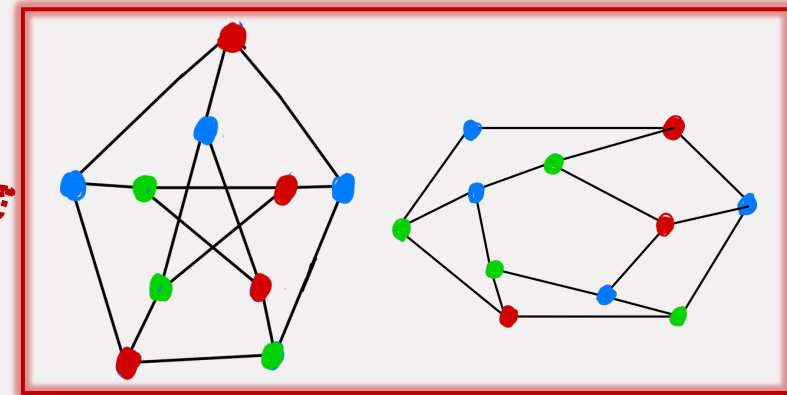
Graph Isomorphism (GI)

Today

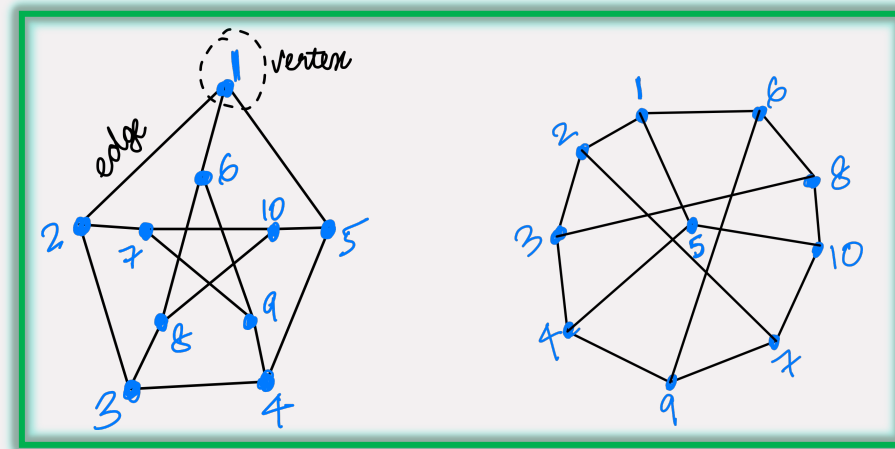


Malicious Verifier ZK Proof

For all $L \in \text{NP}$
Using
Commitment
Schemes

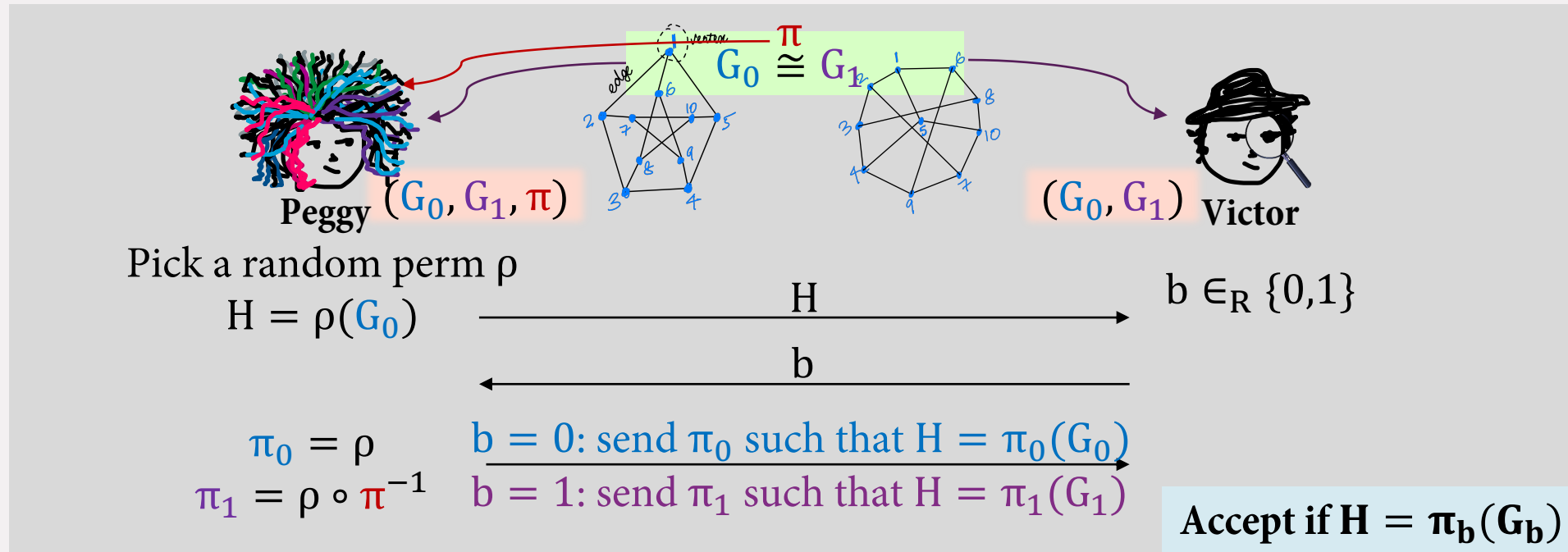


Graph 3-Coloring



Graph Isomorphism (GI)

Recall: Honest Verifier ZKP for GI



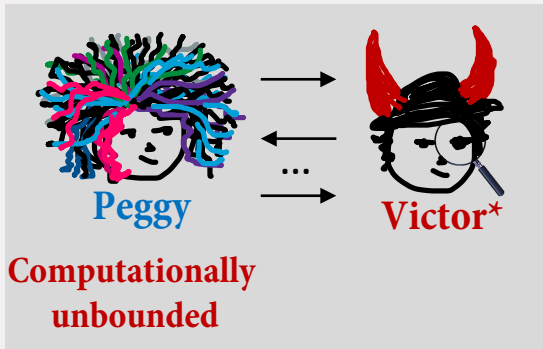
ZKP for Graph Isomorphism (GI): Π_{GI}

$$L_{GI} = \{(G_0, G_1): \exists \text{ permutation } \pi \text{ s.t. } G_1 = \pi(G_0)\}$$

Lecture 18, Theorem 2

Π_{GI} is honest-verifier perfect zero-knowledge proof for L_{GI} .

What about Malicious Verifiers?



Malicious Verifier ZK

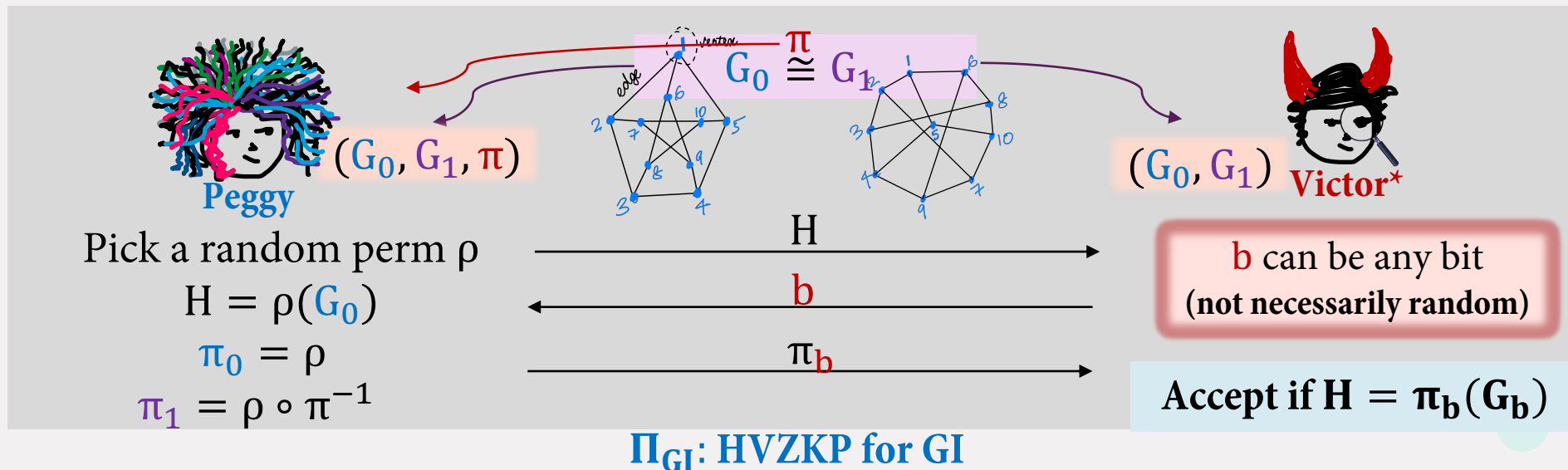
Definition 2: Malicious Verifier Perfect Zero-knowledge

An IP protocol (P, V) for L is **malicious-verifier perfect ZK** if for every PPT V^* , there exists a PPT simulator Sim^{V^*} such that for all distinguishers **Spector** and $\forall x \in L$

$$\Pr \left[\text{Spector} \left(\text{View}_{V^*}(\langle P, V^* \rangle(x)) \right) = 1 \right] - \Pr \left[\text{Spector} \left(\text{Sim}^{V^*}(x) \right) = 1 \right] = 0$$

V^* need not follow the protocol (e.g., need not pick a random challenge)

Sim^{V^*} gets oracle access to V^* as now the honest protocol need not be followed by V^* !



Π_{GI} is Malicious Verifiers ZKP

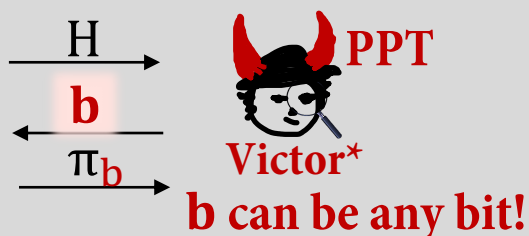
Theorem 1: Malicious Verifier ZKP *same protocol different simulator*

Π_{GI} is malicious-verifier perfect ZKP for L_{GI} .

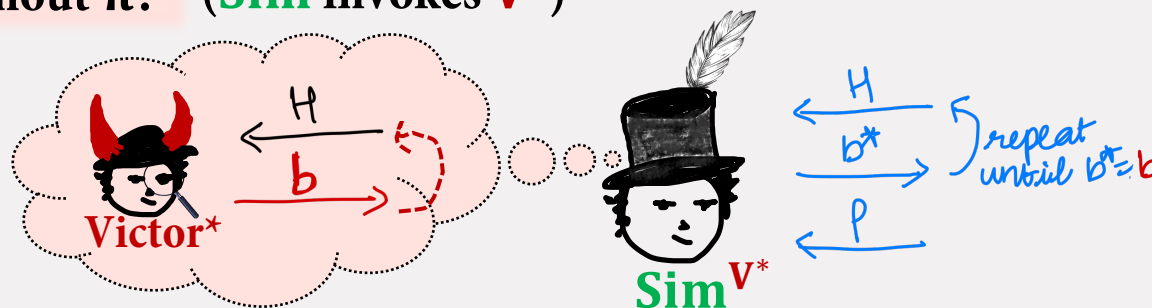
Proof Sketch (for soundness error 1/2)

****Completeness and Soundness (with error 1/2) is as before** (sequential or parallel repetition to reduce error?)

Malicious Verifier ZK How to efficiently simulate without π ? (**Sim** invokes V^*)



$\text{View}_{V^*}(\langle P, V^* \rangle(G_0, G_1):$
 $((G_0, G_1), (H, \mathbf{b}, \pi_b))$



1. Pick $b^* \in_R \{0,1\}$, perm. ρ at random, and compute $H = \rho(G_{b^*})$.
2. Invoke V^* on H to get $\mathbf{b}$ (using independent random coins for V^*).
3. If $b^* = \mathbf{b}$ then life is good so output $((G_0, G_1), (H, \mathbf{b}, \rho))$.
4. Else, repeat until needed (but you $\perp$ after polynomial steps)!

Run-time of Sim^{V^*} ? -in expectation: polynomial time

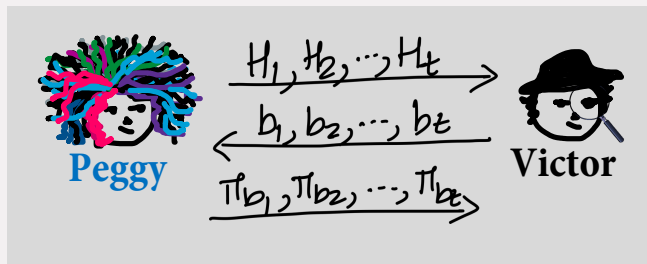
-if you do not $\perp$, worst case: exponential time!

As long as output is not $\perp$

$\text{View}_{V^*}(\langle P, V^* \rangle(G_0, G_1)) \equiv \text{Sim}^{V^*}(G_0, G_1)$

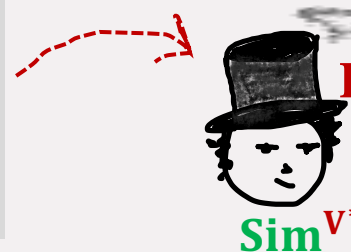
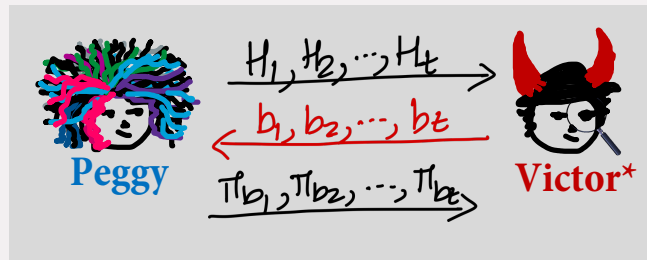
Sequential or Parallel Repetitions?

- Does repetition (t times parallel/sequential) of Π_{GI} reduce soundness error? YES, IT REDUCES TO $1/2^t$
- What about HVZK for this t -parallel repeated Π_{GI} ? YES, IT REMAINS HVZK! Why?



Does not need any input from V can perfectly simulate this using the out-of-order simulation for all t -runs!

- What about Malicious Verifier ZK for t -parallel repeated Π_{GI} ?



He is in trouble! Needs to guess $b_1^*, b_2^*, \dots, b_t^*$ that V^* can send! This only matches now with prob $1/2^t$!

Thm [Goldreich-Krawczyk'90]: $\exists$ ZKPs whose parallel repetition is not malicious verifier ZK!

Thm (Some good news): Repeating Π_{GI} t -times sequentially preserves malicious verifier ZK!

****Requires proof, a little advanced (if interested see Goldreich, Vol 1, Ch 4)**

Do all NP languages have Perfect ZKPs?

We showed perfect malicious verifier ZKP for GI. **Try thinking about GNI!** What about all languages in NP?



[Fortnow'89, Aiello-Hastad'87] showed that we cannot have perfect ZKPs for all NP unless something bizarre happens in the complexity world!

Definition 3: Malicious Verifier Computational Zero-knowledge

An IP protocol (P, V) for L is **malicious-verifier computational ZK** if, for every PPT V^* , there exists a (expected) PPT simulator Sim^{V^*} such that for all PPT distinguishers **Spector** and $\forall x \in L$

$$\left| \Pr \left[\text{Spector} \left(\text{View}_{V^*}(\langle P, V^* \rangle(x)) \right) = 1 \right] - \Pr \left[\text{Spector} \left(\text{Sim}^{V^*}(x) \right) = 1 \right] \right| \leq \text{negl}(n)$$

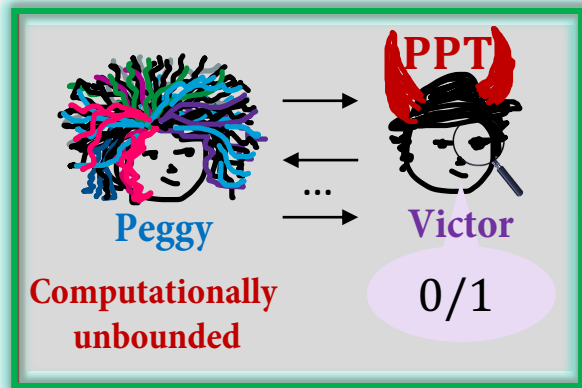


Thm[Goldreich-Micali-Wigderson'87]: If OWFs exist, then all of NP has computational verifier ZK!

What this is saying is anything you can think of proving regularly (like Euclid), can be proved in ZK!

Let's prove GMW!

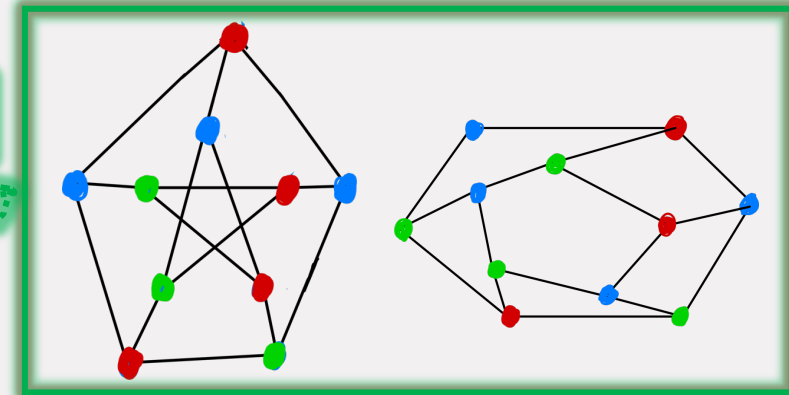
Today



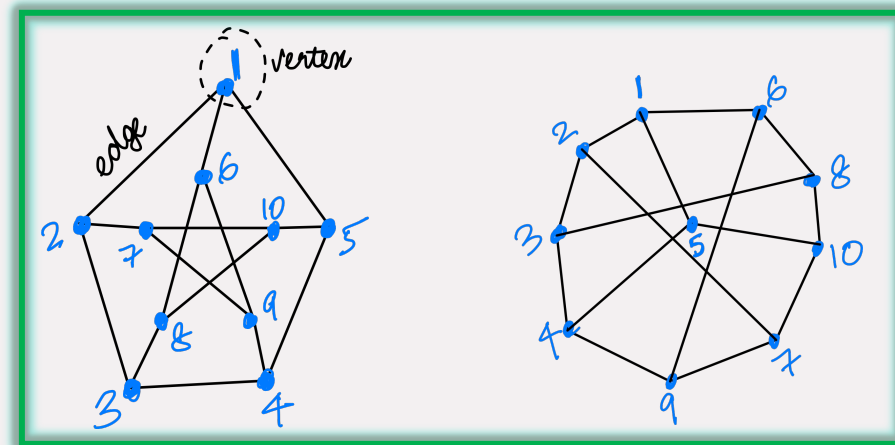
Malicious Verifier ZK Proof

For all $L \in \text{NP}$

Using
Commitment
Schemes



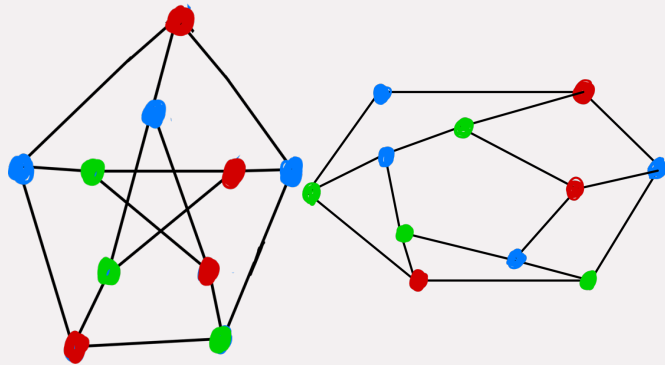
Graph 3-Coloring



Graph Isomorphism (GI)

ZKP for Graph 3-Coloring (G3C)

Graph 3-colorability: A graph is 3-colorable if there exists an assignment of colors for each vertex using only 3 different colors such that any two adjacent vertices have different colors!



For example: used in scheduling (exams, classes, sports, etc.)

Why are we interested in this?

NP-complete problem: Every problem in NP can be reduced to this problem!

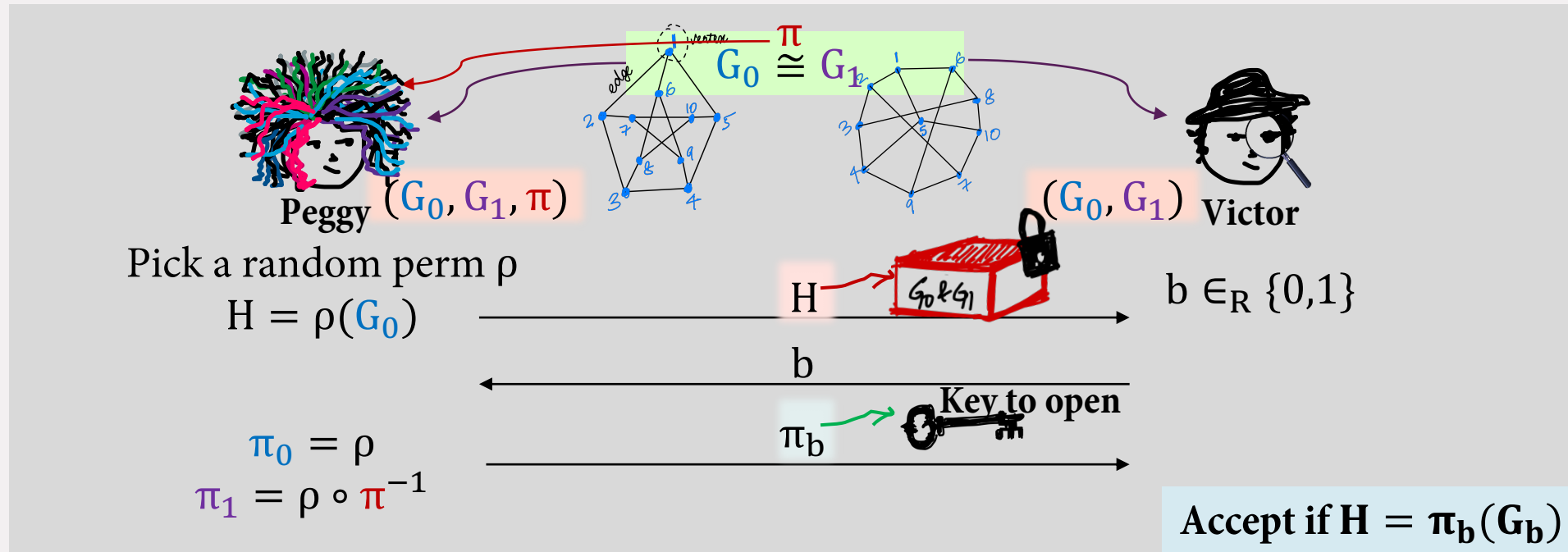
ZKP for an NP-complete language $\Rightarrow$ ZKP for all NP

***Proof of this requires some basic complexity theory!**

Examples of NP assertions:

- My public key is “well-formed.”
- Encrypted Bitcoin: “I have enough money to pay you.”
- Running programs on encrypted input: Given $\text{Enc}(x)$ and y , prove that $y = \text{PROG}(x)$.

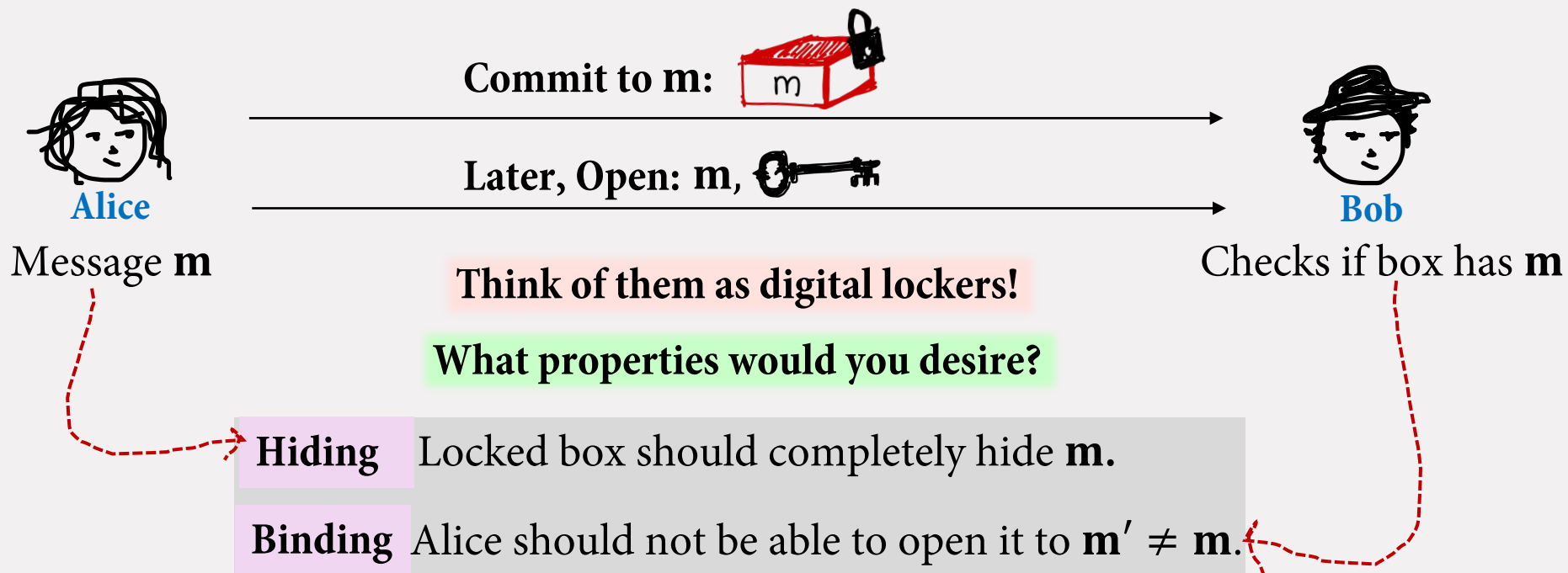
Reflecting on ZKP for GI



Π_{GI} : ZKP for GI

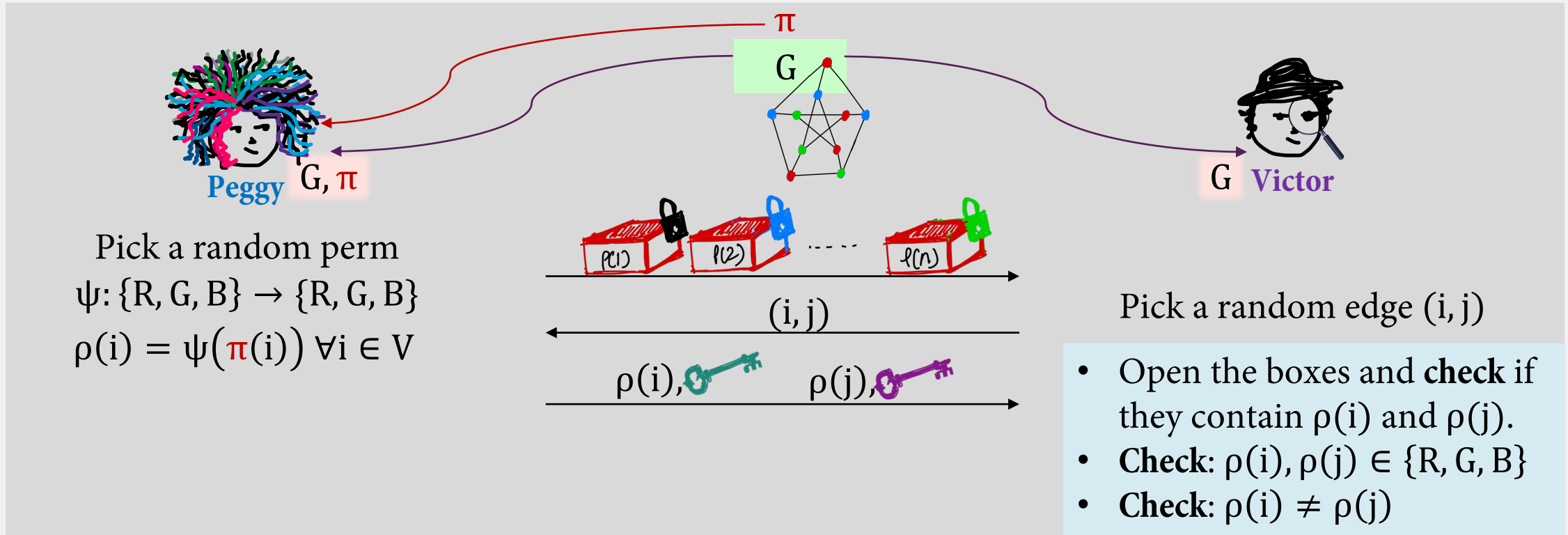
- Honest Peggy “*commits*” to G_0 and G_1 by sending the “*commitment*” H to π .
- Soundness:** The commitment H is “*perfectly binding*”: if $G_0 \not\cong G_1$, then malicious P^* cannot fool Victor, as she can commit to only one of G_0 or G_1 in advance!
- ZK:** H *hides* information about π from a malicious V^*

Commitment Schemes



We will formalize this later but say we had this digital locker, let's build a ZKP for G3C

ZKP for G3C



ZKP for Graph 3-Coloring (G3C): Π_{G3C}

$L_{G3C} = \{G: \exists \text{ coloring assignment } \pi: V \rightarrow \{R, G, B\} \text{ to } G \text{ such that neighbors have distinct colors}\}$

****** $V = \{1, 2, \dots, n\}$

ZKP for G3C

Theorem 1

If the *digital lockers* have *perfect binding and computational hiding*, then Π_{G3C} is a **malicious-verifier computational ZKP** for L_{G3C} .

Proof Sketch

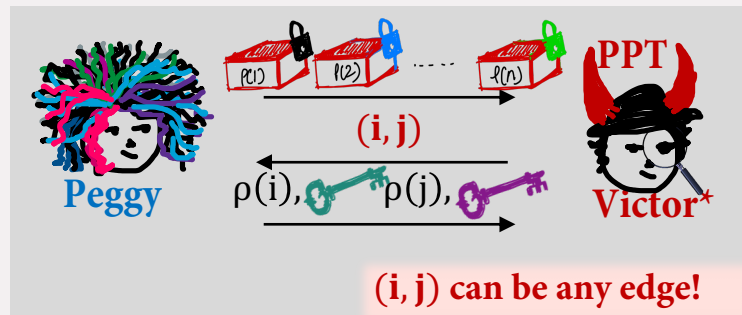
Completeness H.W.

Soundness G not 3-colorable $\Rightarrow \exists (i, j)$ (an edge) such that $\pi(i)$ and $\pi(j)$ are the same color!

Binding? V will catch this edge and reject with probability $\geq 1/|E|$ (E is set of edges).

(*Sequentially*)^{*} repeat $|E| \cdot t$ times so that V accepts with probability $\leq (1 - 1/|E|)^{|E| \cdot t} \leq 2^{-t}$

Malicious Verifier Comp ZK



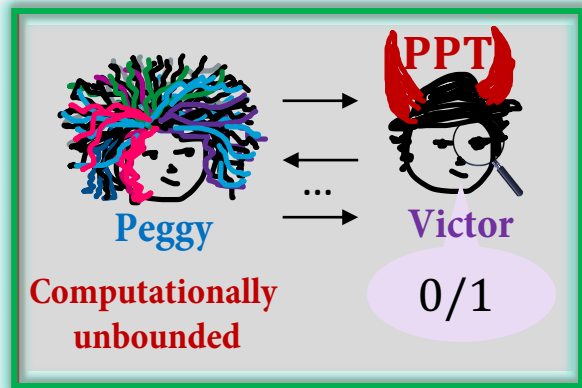
****For one run (not the sequential repeat setting)**

1. Pick random (i^*, j^*) . Color these with different colors and the rest of the vertices red (ρ be this map).
2. Send $\rho(1)$, $\rho(2)$, ..., $\rho(n)$ to V^* and get the edge (i, j) .
3. If $(i^*, j^*) = (i, j)$ life is good, **output**: $\rho(i)$, $\rho(j)$.
4. Else, repeat!

Hiding? ****Use hiding to show that the simulated transcript is indistinguishable from the real transcript (only for single run!)**

^{*}Parallel repetition of this protocol is not zero-knowledge! –[Holmgren, Lombardi, Rothblum, 21]

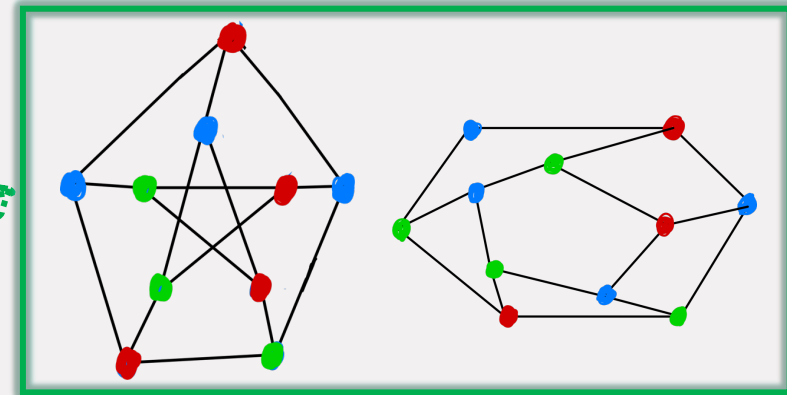
Today



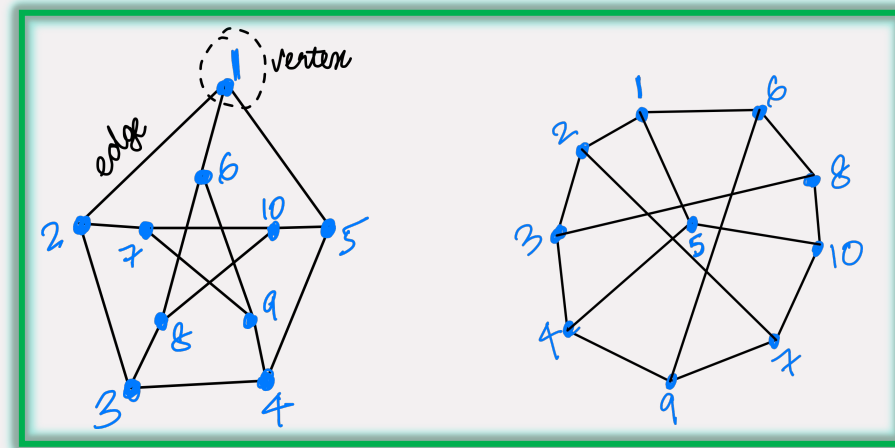
Malicious Verifier ZK Proof

For all $L \in \text{NP}$

Using
**Commitment
Schemes**



Graph 3-Coloring



Graph Isomorphism (GI)

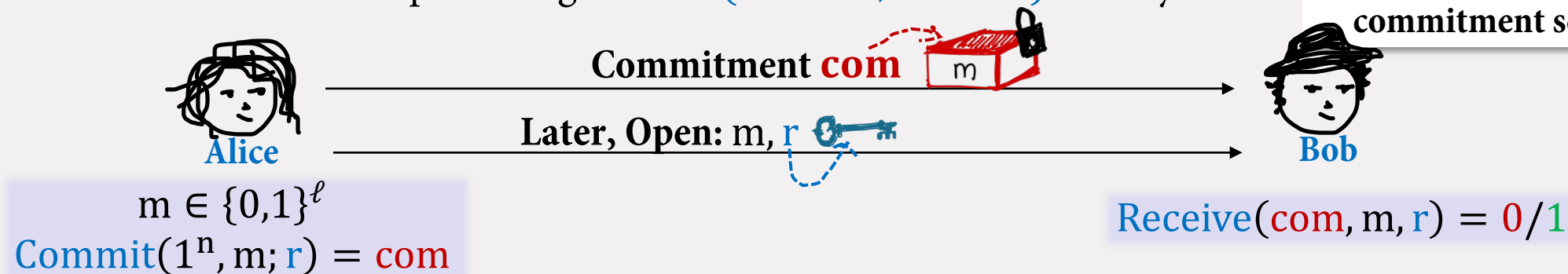
Commitment Schemes

HW 1

Can you get both perfect hiding and perfect binding simultaneously from a commitment scheme?

Definition 3: Commitment Scheme

A commitment scheme is a pair of algorithms (**Commit**, **Receive**) with syntax:



Correctness: For all $n \in \mathbb{N}$ and inputs $m \in \{0,1\}^\ell$: $\Pr_r[\text{Receive}(\text{Commit}(1^n, m; r), m, r) = 1] = 1$

Computational Hiding: For any $m_0 \neq m_1 \in \{0,1\}^\ell$: $\text{Commit}(1^n, m_0; r_0) \approx_c \text{Commit}(1^n, m_1; r_1)$

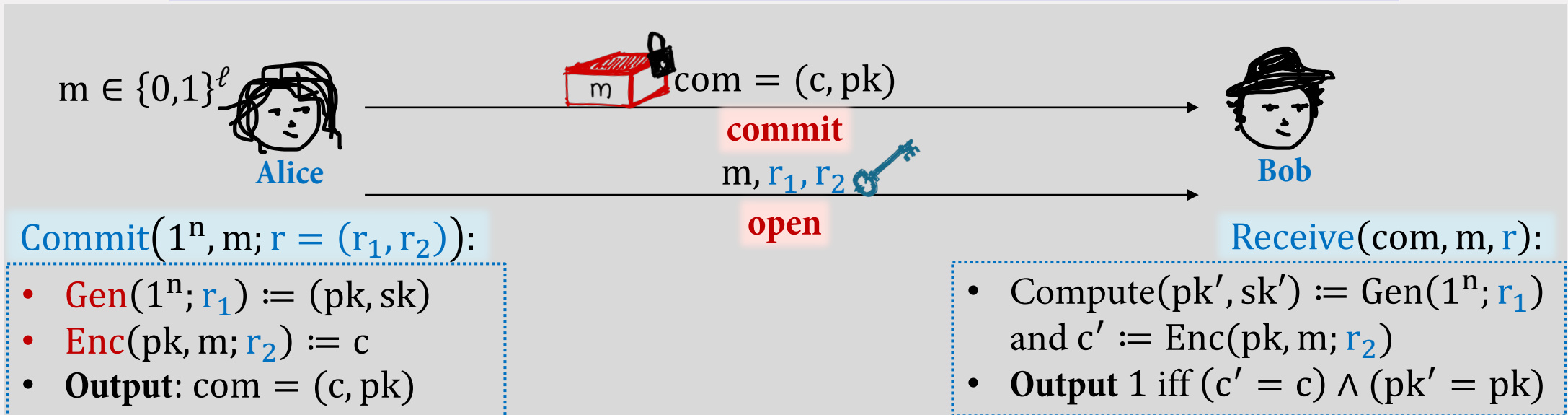
Perfect Binding: For any **com** $\in \{0,1\}^*$, there **do not exist** two openings (keys) r_1 and r_2 such that $\text{Receive}(\text{com}, m_1, r_1) = 1 = \text{Receive}(\text{com}, m_2, r_2)$ for distinct m_1 and m_2

****Remark:** The commit phase can be interactive as well!

****Commit** $(1^n, m; r)$: denotes the algorithm **Commit** taking input $(1^n, m)$ as input, using randomness r to output **com**

Commitment Schemes: Construction 1

PKE $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) \Rightarrow$ Commitment Scheme $\Sigma = (\text{Commit}, \text{Reveal})$



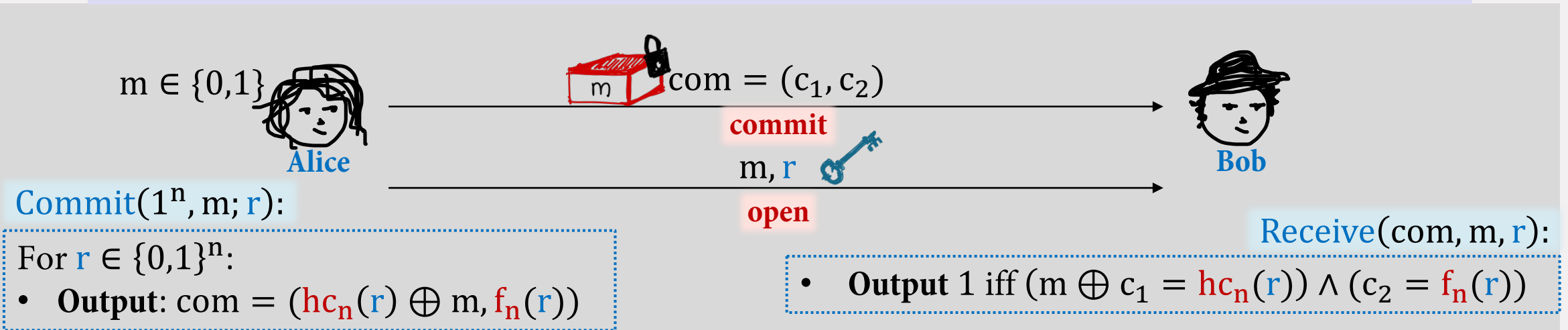
Properties from Π ?

- Should be able to recognize honestly sampled pk .
- IND-CPA security $\Rightarrow$ Computational hiding
- Perfect Correctness of decryption $\Rightarrow$ Perfect binding

(Proof discuss)

Commitment Schemes: Construction 2

OWP family $\{f_n: \{0, 1\}^n \rightarrow \{0, 1\}^n\}$ with hard-core predicate $hc \Rightarrow$ Bit-commitment Scheme Σ

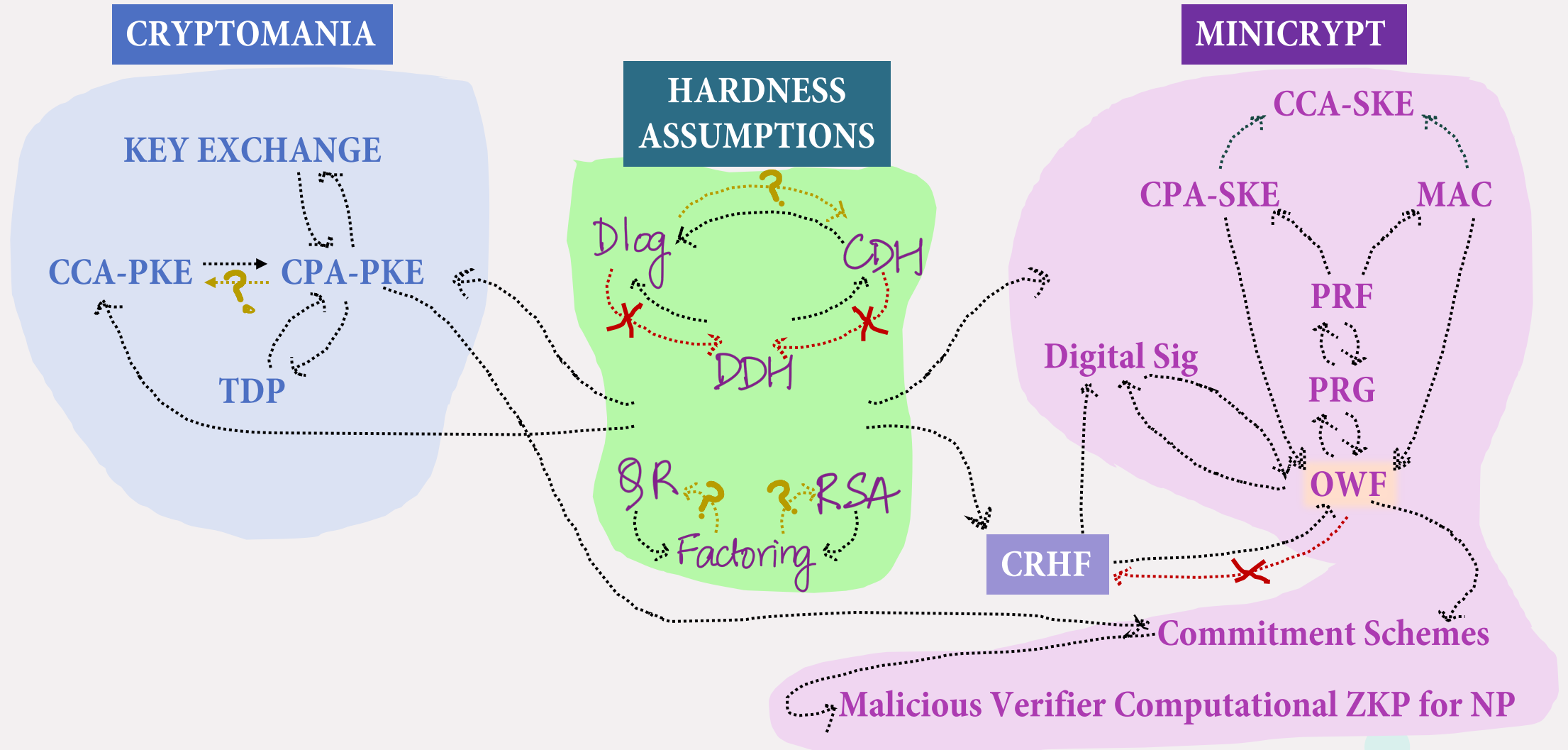


- Security of hard-core predicate $hc \Rightarrow$ Computational hiding [\[Lec15, Construction 1\]](#)
- f_n permutation $\Rightarrow$ Perfect binding (Why?)

HW 2

1. Formalize the proof!
2. How to extend this bit-commitment to string-commitment?

Big Picture so far



1. Explain Newton's First Law of Motion in your own words.



!



Yakka Foob Mog. Grug
Pubbawup Zink Wattoom
Gazork. Chumble Spuzz.



I LOVE
LOOPHOLES.



THANK YOU