CPSC 420 Lecture 30: Today's announcements:

- Final Exam: Tue Apr 18, 2023 08:30am LSK 200.
  One 2-sided page of notes

Today's Plan

- Zero-knowledge Proofs

Zero-knowledge proofs

How to prove something in a way that prevents others from proving it, even though they believe it's true.

Why?

Foil impersonators Peggy (the Prover) can convince Victor (the Verifier) that she is Peggy in a way that prevents anyone listening in from later impersonating Peggy.

Verify bitcoin balance without revealing balance.

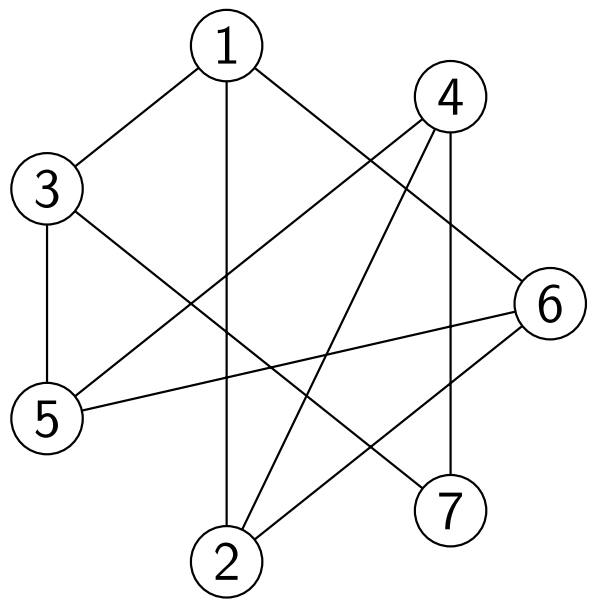
How?

Interactive proof

Zero-knowledge protocol

Peggy constructs a graph G so that she knows a Hamiltonian cycle in it but it's hard for other people to find a H.C. in G.
Everyone knows Peggy's graph G but only Peggy knows a Hamiltonian cycle in G.
Peggy and Victor participate in a **protocol** so that Peggy can convince Victor she knows a Ham. cycle in G.

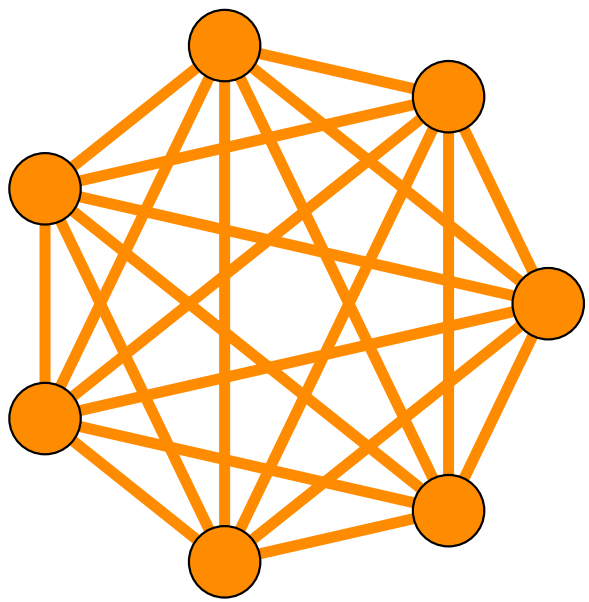
1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.



Zero-knowledge protocol

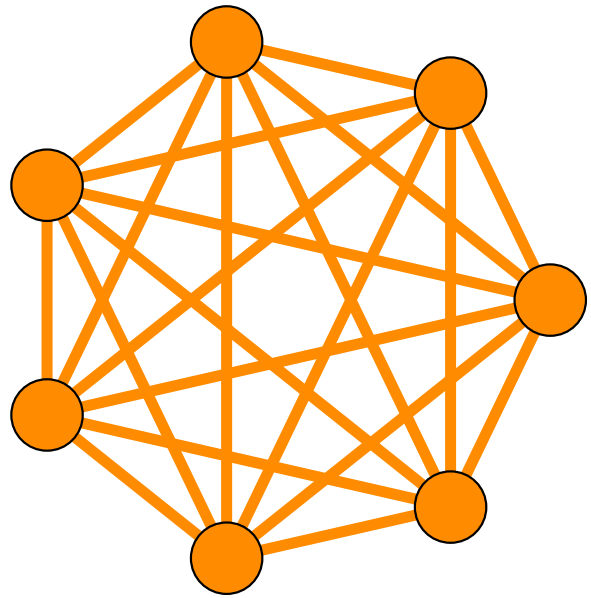
Peggy constructs a graph G so that she knows a Hamiltonian cycle in it but it's hard for other people to find a H.C. in G.
Everyone knows Peggy's graph G but only Peggy knows a Hamiltonian cycle in G.
Peggy and Victor participate in a **protocol** so that Peggy can convince Victor she knows a Ham. cycle in G.

1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.



Zero-knowledge protocol

1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.

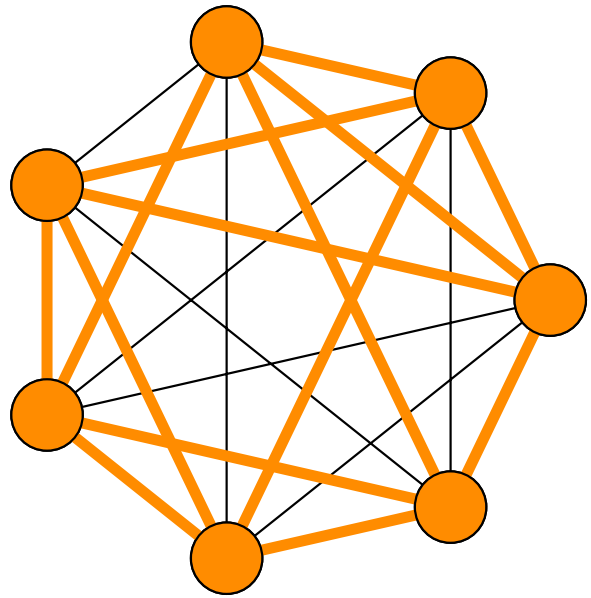


2. Victor asks to see either (a) Hamiltonian cycle or (b) the entire graph. (flips a fair coin to choose)
3. Peggy, if (a), scratches off paint on edges of H.C. or, if (b), scratches off all paint (Victor checks its Peggy's G)
4. Repeat k times

Zero-knowledge protocol

1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.

(a)

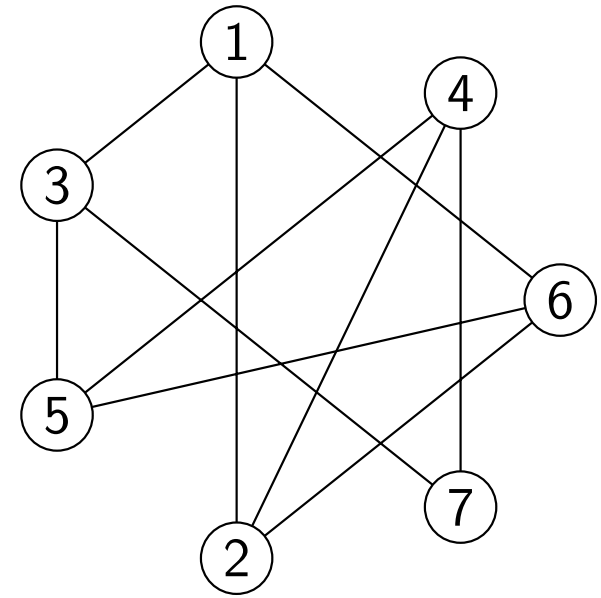


2. Victor asks to see either (a) Hamiltonian cycle or (b) the entire graph. (flips a fair coin to choose)
3. Peggy, if (a), scratches off paint on edges of H.C. or, if (b), scratches off all paint (Victor checks its Peggy's G)
4. Repeat k times

Zero-knowledge protocol

1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.

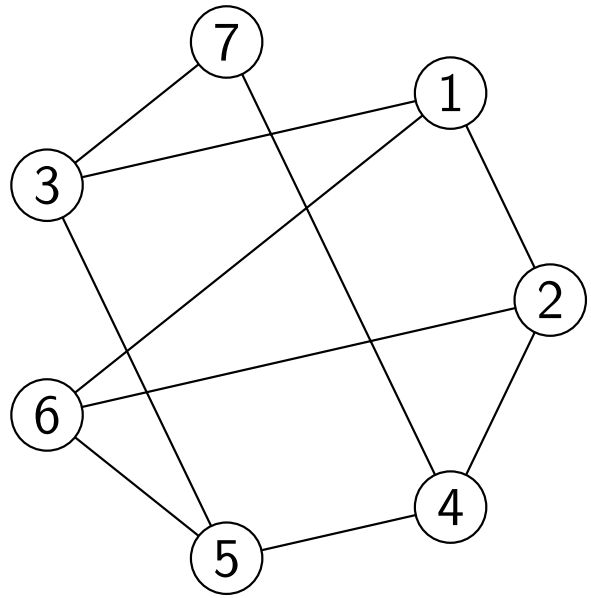
(b)



2. Victor asks to see either (a) Hamiltonian cycle or (b) the entire graph. (flips a fair coin to choose)
3. Peggy, if (a), scratches off paint on edges of H.C. or, if (b), scratches off all paint (Victor checks its Peggy's G)
4. Repeat k times

Zero-knowledge protocol

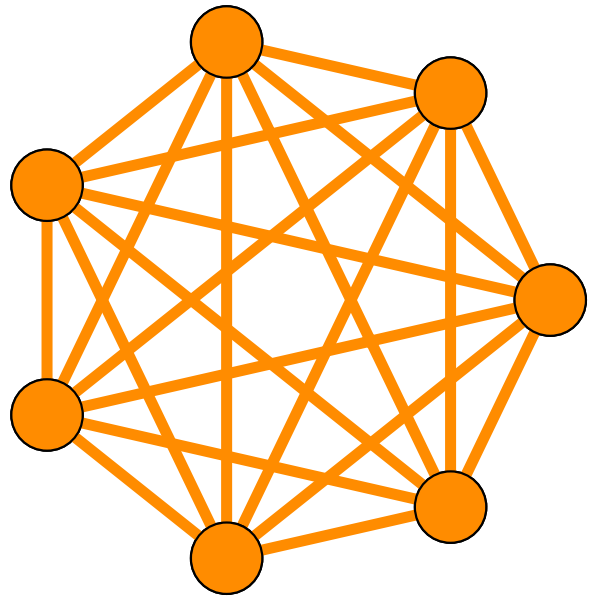
1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.



2. Victor asks to see either (a) Hamiltonian cycle or (b) the entire graph. (flips a fair coin to choose)
3. Peggy, if (a), scratches off paint on edges of H.C. or, if (b), scratches off all paint (Victor checks its Peggy's G)
4. Repeat k times

Zero-knowledge protocol

1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.

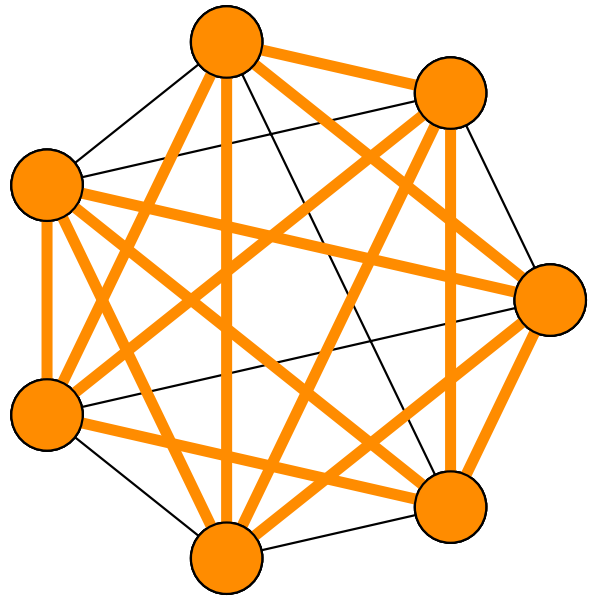


2. Victor asks to see either (a) Hamiltonian cycle or (b) the entire graph. (flips a fair coin to choose)
3. Peggy, if (a), scratches off paint on edges of H.C. or, if (b), scratches off all paint (Victor checks its Peggy's G)
4. Repeat k times

Zero-knowledge protocol

1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.

(a)

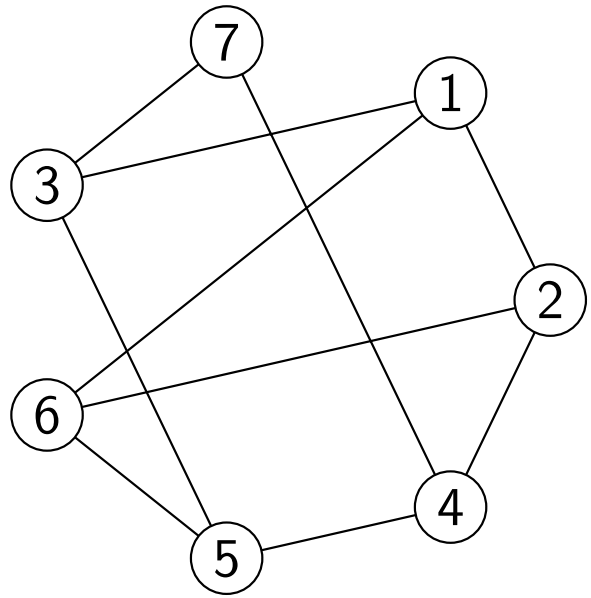


2. Victor asks to see either (a) Hamiltonian cycle or (b) the entire graph. (flips a fair coin to choose)
3. Peggy, if (a), scratches off paint on edges of H.C. or, if (b), scratches off all paint (Victor checks its Peggy's G)
4. Repeat k times

Zero-knowledge protocol

1. Peggy writes vertices around a circle in random order. Adds edges. Covers all edge slots and vertex names with scratch-off paint.

(b)



2. Victor asks to see either (a) Hamiltonian cycle or (b) the entire graph. (flips a fair coin to choose)
3. Peggy, if (a), scratches off paint on edges of H.C. or, if (b), scratches off all paint (Victor checks its Peggy's G)
4. Repeat k times

What we need to prove

1. Protocol convinces Victor that Peggy knows Ham. cycle in G.
2. Protocol gives no other knowledge to Victor (even if he doesn't follow the rules).

Claim 1: If Charlie, who doesn't know H.C. in G, impersonates Peggy then Charlie will be caught with probability $\geq 1 - \frac{1}{2^k}$.

Proof: At each round Charlie is caught with prob. $\geq 1/2$. Either Charlie draws Peggy's graph G (and Victor asks for H.C.) or Charlie draws a different graph (and Victor asks to see G).
In either case, Charlie is caught with prob. $\geq 1/2$.

Note: It might be possible for Charlie to do some work and have a non-zero probability of guessing a H.C. in G. We assume this can't happen.

What we need to prove

1. Protocol convinces Victor that Peggy knows Ham. cycle in G.
2. Protocol gives no other knowledge to Victor (even if he doesn't follow the rules).

Claim 2: Any scheme to extract information from Peggy can be simulated without Peggy.

Proof: "Scheme" means polynomial time probabilistic algorithm V. "Simulate" means produce the same distribution of conversations in about the same time.

What we need to prove

1. Protocol convinces Victor that Peggy knows Ham. cycle in G.
2. Protocol gives no other knowledge to Victor (even if he doesn't follow the rules).

Claim 2: Any scheme to extract information from Peggy can be simulated without Peggy.

One simulation step

1. Simulate Peggy:
   Flip coin to pick either
   (A) draw an n-cycle in random order or
   (B) draw Peggy's graph in random order.
2. Simulate V:
   If V asks to see
   (a) Peggy's G and we picked (A) or
   (b) Ham. Cycle and we picked (B) then done.
   Otherwise, rewind V to previous state and goto 1.

Expected number of rewinds is two.

V sees same distribution of conversations as with Peggy.

Course Topics

- Geometry: convex hull, Voronoi diagram
- Linear programming: duality
- Network flow: Ford-Fulkerson, Edmonds-Karp, bipartite matching, pennant race, open-pit mining
- Dynamic programming: LIS, LCS, edit distance
- Compression: Shannon information, Huffman, Lempel-Ziv
- NP: completeness, hardness, reduction
- Approximation algs: vertex cover, list sched, hardness
- Online algorithms: hiring, paging, move-to-front, experts
- Universal+Cuckoo hashing, Fast Fourier Transform, RSA cryptosystem, Quantum computing, Zero-knowledge proof

Thank you!

Please fill out course evaluations for this course.