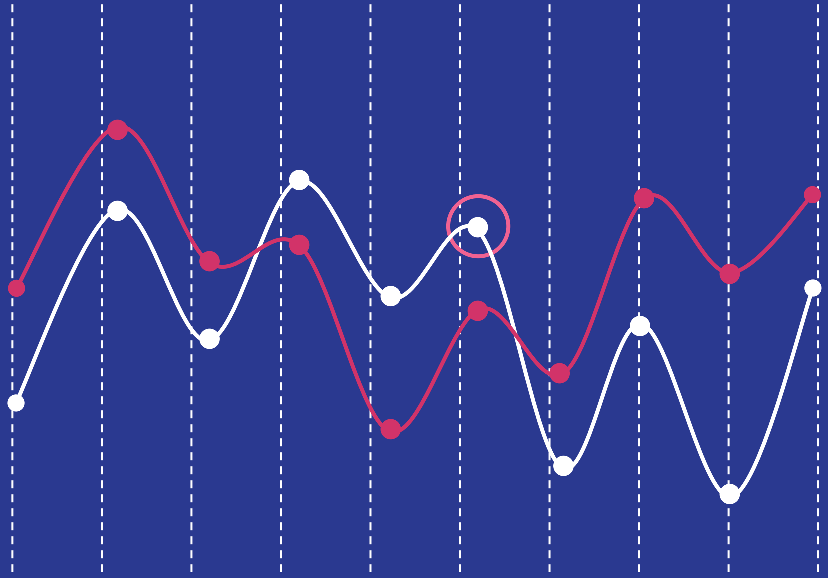
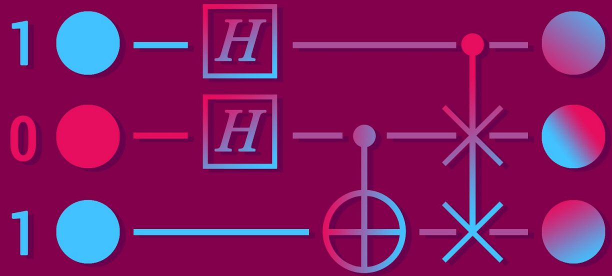
Theory Club

August 26th, 2019



THEORY CLUB

QUANTUM COMPUTING ALGORITHMS



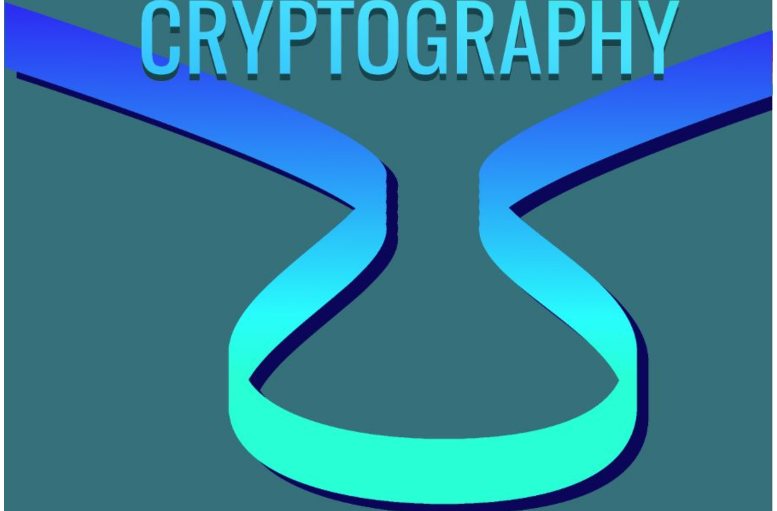
ES&T L1175: 02/27 @6PM

with

DEVON INGRAM

THEORY CLUB

ELLIPTIC CURVE CRYPTOGRAPHY



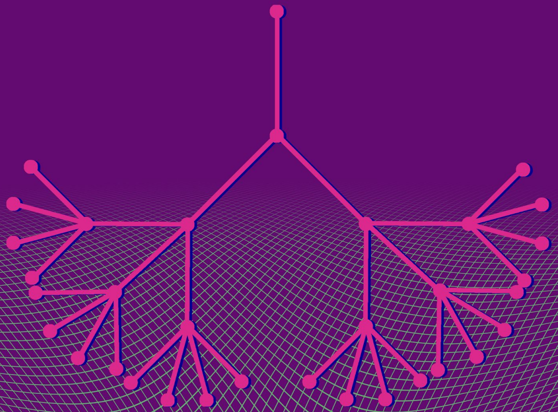
ES&T L1175: 04/03 @6PM

with

PROFESSOR MATTHEW BAKER

THEORY CLUB

PROBLEM SESSION



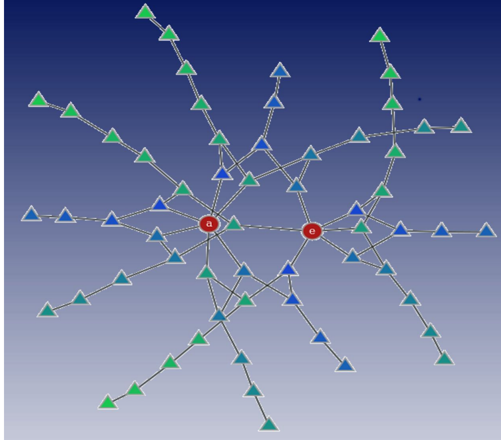
CCB 102: 02/26 @6PM

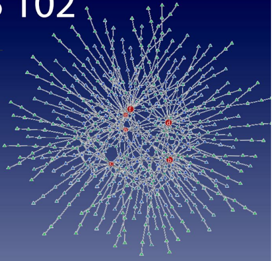
with

SHERRY SARKAR

THEORY CLUB

1/22 6:00 P.M CCB 102





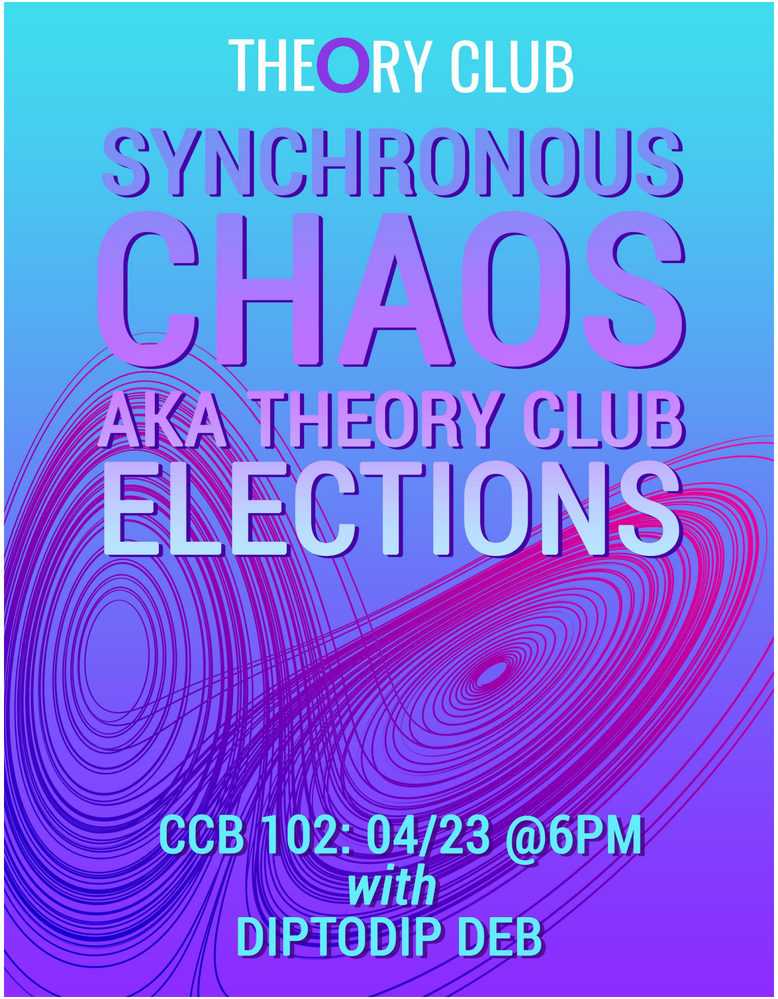
REDUCING THE

GROUP ISOMORPHISM PROBLEM

TO THE

GRAPH ISOMORPHISM PROBLEM

BY DANIEL HATHCOCK



THEORY CLUB

SYNCHRONOUS CHAOS

AKA THEORY CLUB ELECTIONS

CCB 102: 04/23 @6PM

with

DIPTODIP DEB

THEORY CLUB

SPRINGTIME

PROBLEM SESSION

CCB 102: 03/26 @6PM

with

SHYAMAL PATEL

General Information

General Meetings

- Professor Talks
- Student Talks
- Donut Problem Sessions
- ARC speakers

- Proof based
- No coding

Goal of Meetings

- See theory CS outside the GT curriculum

- Show what our faculty are researching

- Everyone leaves understanding something.

Prerequisites?

- Meetings are proof based.

- You don't need to be good at proofs or math to come to our meetings - you just have to be interested!

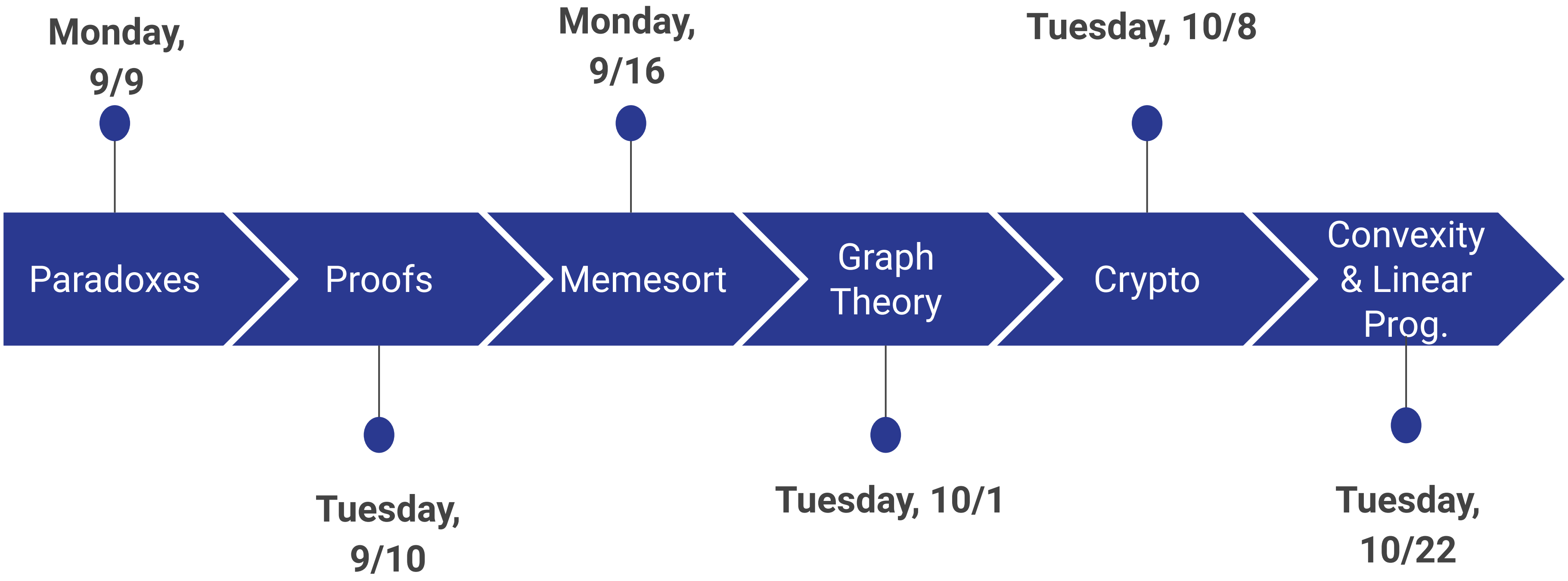
Introduction to Theory Workshops

- This year, we will be holding Intro to TCS workshops. There will be 6.

- The goal is to
    - Give a taste of popular TCS fields
    - An introduction to proof based thinking
    - Inspire you to take TCS classes

- Not meant to be classes

Introduction to Theory Workshops



Monday, 9/9

Paradoxes

Proofs

Tuesday, 9/10

Monday, 9/16

Memesort

Graph Theory

Tuesday, 10/1

Tuesday, 10/8

Crypto

Convexity & Linear Prog.

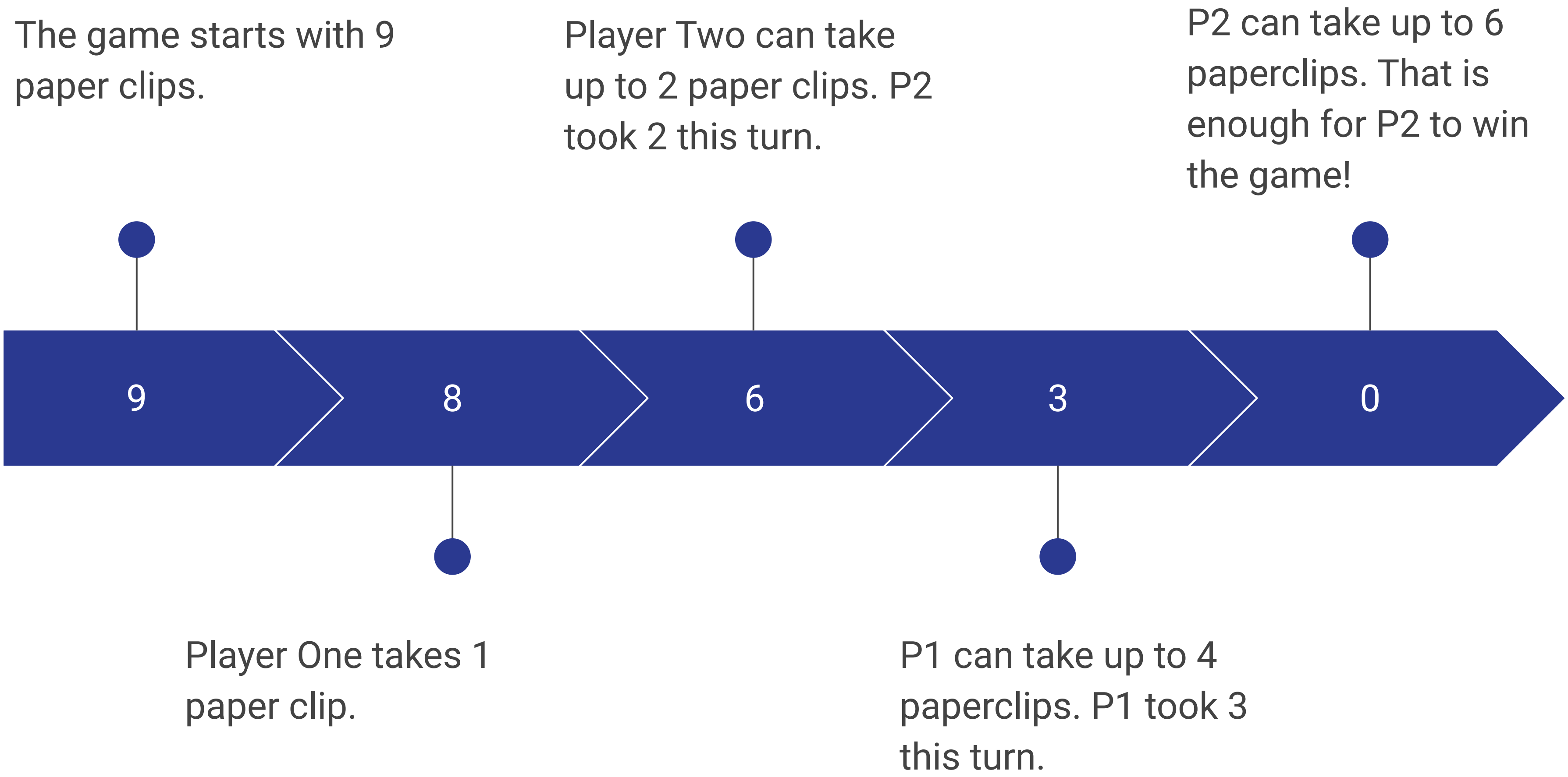
Tuesday, 10/22

Paperclip Game

Rules

- Two players
- N paper clips in a pile
- Player 1 on his/her first move can take up to N - 1 paper clips
- A player can take up to twice the number of paper clips the last player took

GOAL: Take the last paper clip!



The game starts with 9 paper clips.

Player One takes 1 paper clip.

Player Two can take up to 2 paper clips. P2 took 2 this turn.

P1 can take up to 4 paperclips. P1 took 3 this turn.

P2 can take up to 6 paperclips. That is enough for P2 to win the game!

Solution

The losing positions are the **Fibonacci Numbers**

The Winning strategy:

- Find the Zeckendorf decomposition of n (found using the Greedy Algorithm)
- Remove the **smallest part** of the decomposition.

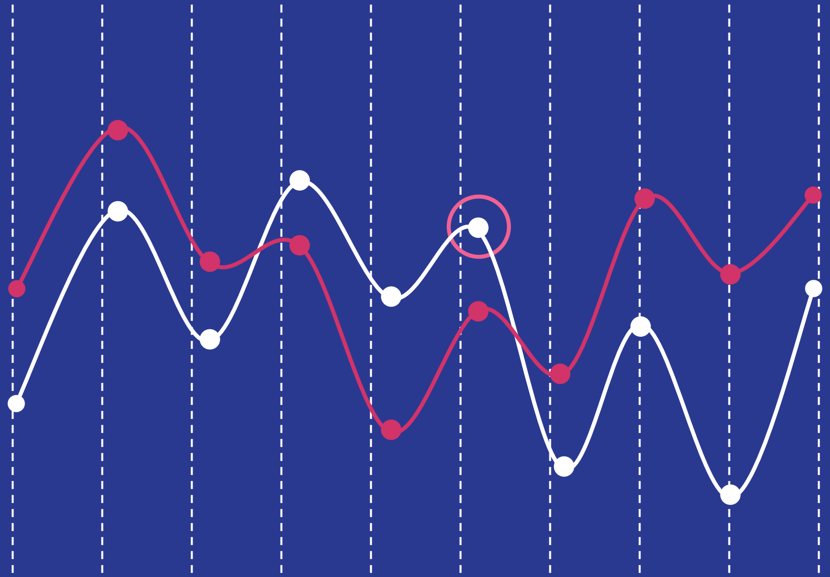
Why is it the Fibonacci Numbers?

- **Lemma: $2 * F_i < F_{i + 2}$**

- This implies that...
  - removing the smallest Zeckendorf part **will never allow the other player** to remove the next smallest Zeckendorf part
- For example,
  - 19 = 13 + 5 + 1
  - The first player removes 1 paper clip.
  - The next player is **forced to play the losing position of 5** paper clips!
  - The second player starts the game of 5 paper clips (and inevitably loses). That means he/she **has to start the next losing position of 13 paper clips!**
- And **Player 2 loses!!**

So does Player 1 always win?

An Introduction To CS Theory



What Questions Does CS Theory Consider?

Algorithms

How fast can you compute the volume of a shape?

How can you quickly compute a close to optimal route to visit a set of cities?

How can you quickly sample a random schedule?

Given a black box function, how many values do you need to know to be reasonably convinced that it is linear (or close to linear)?

Limits of Computation

How many comparisons do you need to sort a list?

Suppose you and a friend are given numbers, how many bits do you need to exchange to know if they are the same?

Does randomness allow us to compute functions faster?

Suppose we know that a solving a problem takes a long time, what other problems does this imply are slow?

What tools are used?

Discrete Math:

- Combinatorics
- Graph Theory

Continuous Math

- Geometry and Calculus
- Linear Algebra

Algorithmic Ideas:

- Binary Search
- Data Structures
- Dynamic Programming

Interesting Tidbits

Can the existence of an algorithm for a problem be used to prove that other algorithms **cannot** exist for other problems? Can the nonexistence of algorithms be used to prove that another algorithm correctly solves a problem? (In fact, there are "yes" answers to both questions!)

(Stolen from Ryan Williams)

There exist problems that cannot be computed.

(See undecidability of the halting problem)

Interesting Tidbits

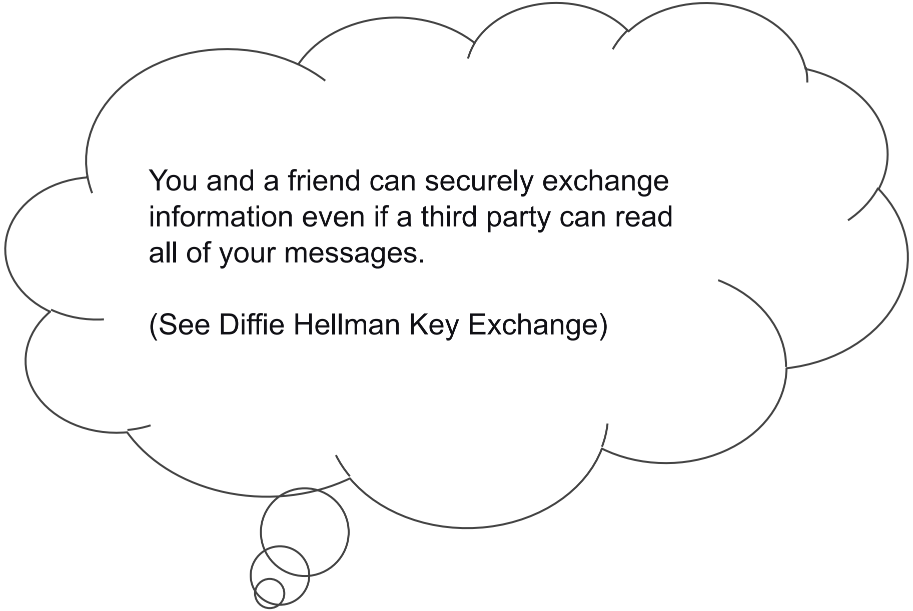
Every problem whose solution can be checked quickly is no harder to solve than SAT.

(See Cook-Levin Theorem)

Any set of m points in R^m can be embedded in R^d where d = O(ln(m)) and in such a way to approximately preserve distances.

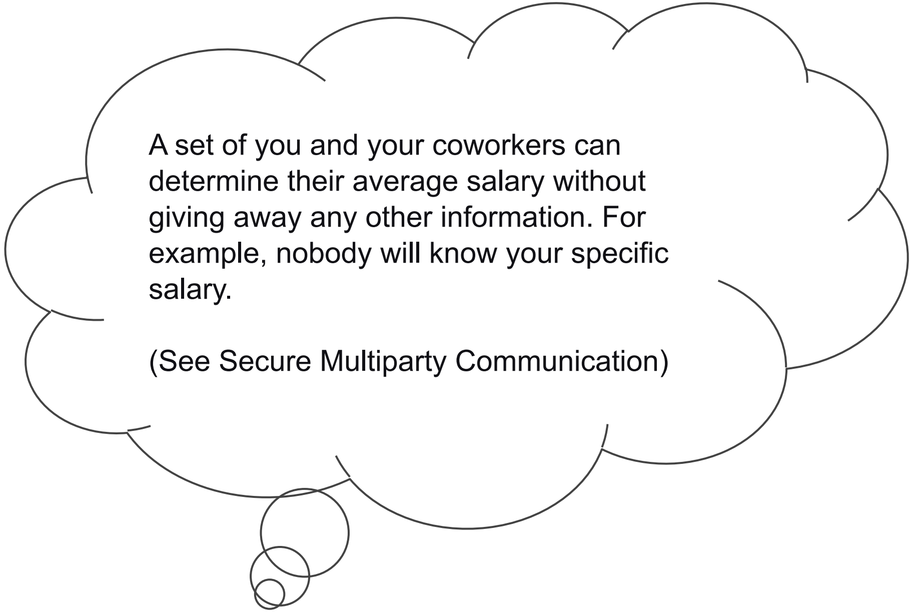
(See Johnson Lindenstrauss Theorem)

Interesting Tidbits



You and a friend can securely exchange information even if a third party can read all of your messages.

(See Diffie Hellman Key Exchange)



A set of you and your coworkers can determine their average salary without giving away any other information. For example, nobody will know your specific salary.

(See Secure Multiparty Communication)

Open Questions

Open Problem 1

Sum of Square Roots

Given a list of integers x_1, x_2, …, x_n and k can you determine if

$$\sqrt{x}_1 + \sqrt{x}_2 \cdots \sqrt{x}_n \leq k?$$

Can this be done in polynomial time?

Open Problem 2

All Pairs Shortest Paths

Given a weighted graph G=(V, E,w) does there exist a truly subcubic algorithm to find the distance between every pair of vertices?

Open Problem 3

P = NP?

For any problem whose solution can be checked in polynomial time, can we compute its solution in polynomial time?