LESSON TITLE: Secure Multi-Party Computation for Muggles

SUMMARY:

We don't need any magical powers like cryptography to do secure multi-party computation - only generating random numbers and adding!

In this lesson, you'll figure out how to compute an average of a value contributed by each member of the class, while keeping everyone's individual value secret. We'll discuss several potential ways to do this, and explore their privacy and security properties. The methods we'll be able to do by hand in class can provide strong privacy properties, but are only able to compute functions that can be implemented using only addition. Computing any function privately is possible, but doing it efficiently is an active computer science research area.

GRADE BAND: High School (can be adapted to any grade level that understands addition)

TIME REQUIRED: 60 minutes

LESSON LEARNING OUTCOMES: Upon completion, students will be able to:

- Define secure multi-party computation, and understand how it enables a group to compute something together while keeping everyone's input secret.
- Understand a simple secure multi-party computation protocol based on *masked addition*, and be able to evaluate its privacy and security properties.
- Understand a simple secure multi-party computation protocol based on *secret-sharing*, and be able to compare its privacy and security properties to the masked addition protocol.

Materials List (i.e., string, digital diary, raspberry pi, web link, drone):
index cards, something to write with

Describe any Previous Knowledge that may be Required:
ability to add multi-digit numbers, basic properties of addition, randomness

How will you facilitate the learning?
- Describe the Warm-up Activity:

What are the risks of data exposure?
Facebook/Cambridge Analytica incident
is there a way to allow computation on data without exposing the data?

What is something we want to compute that depends on private data from each of the participants?
(e.g., average number of years you've been teaching)

- Describe the Focused Activity:

Develop a protocol for computing the average of the group, while keeping everyone's individual value secret.

- First suggestions are often to use a trusted individual (this is called a "trusted third party")
    - what are the drawbacks of relying on a trusted third party?
        - all participants need to agree on the same trusted individual
        - the trusted individual learns everything! dangerous single point of failure/corruption; even if they are trustworthy, risk that the data could be exposed
        - if the trusted party is corrupted, they control everything about the result also
- Masked addition protocol:
    - First person selects a secret random number R, writes it on a card to remember
    - writes $Z_1 = R + x_1$ (their value on an index card)
    - hands to next person, who writes $Z_2 = Z_1 + x_2$ on a new card, hands it to the next person, and destroys the card they received.
    - this continues until the last person, who will write
      $$Z_n = Z_{n-1} + x_n$$
      on a card and hand it to the first person.
    - the first person receives
      $$Z_n = Z_{n-1} + x_n = R + x_1 + x_2 + \ldots + x_n$$
    - the first person subtracts R from the number received, to compute the actual sum.

- How private is the masked addition protocol?
    - if everyone follows the rules, and R is chosen well (how does it need to be chosen?) it is perfectly private!
    - but, if any two people collude, they can learn something (what can they learn, and how?)

- How *secure* is the masked addition protocol? (here, secure means produce the correct output)
    - not at all - any one person can cause the wrong result to be output, without detection
    - the first person, can output any result (and learn the actual result!)
- How can we make it better?
    - lots of potential ideas to discuss - but all the obvious modification either compromise privacy, or don't solve the security problem

If there's more time, or in a follow-on session can introduce a second protocol:

- Secret-sharing protocol:
  - Select two "semi-trusted" parties - they are trusted not to collude, and to follow the protocol, but don't get any sensitive data. The data must be sent secretly to each aggregator independently.
  - Each person selects their own random number R_i, and gives one of the aggregators R_i and the other one $x_i + R_i$.
  - The aggregators sum their numbers, one obtains $Y_A = R_1 + R_2 + \ldots + R_n$. The other obtains $Y_B = x_1 + R_1 + x_2 + R_2 + \ldots + x_n + R_n = x_1 + x_2 + \ldots + x_n + R_1 + R_2 + \ldots + R_n$. Then, both values are announced, and we can compute $Y_B - Y_A$ to get the result.

- How private is the secret-sharing protocol?
  - if the R's are chosen well (how does it need to be chosen?), the channels between the participants and the aggregators are secret, and the aggregators don't collude it is perfectly private!
  - but, if the aggregators collude, all is lost
  - if all but one of the participants collude, they can learn a target's secret value
  - …

- Describe the Teacher Instruction:

The main goal of this exercise is to get students thinking about how simple properties of addition combined with random numbers can enable secure multi-party computation. Don't ask students to computing anything too sensitive, though, since if there are mistakes the private information could be revealed. I think its best to encourage students to invent their own protocols before providing some hints to get to the masked addition protocol describe here. In an hour, they probably won't be time to get to other protocols, and it is probably best to spend more time discussing one protocol, but good to get to trade-offs between different protocols if there is time.

One important security goal for this exercise is to understand how things break if individual participants don't follow the protocol - and which kinds of misbehaviors break confidentiality, and which kinds break correctness.

Mapping to Cyber Security First Principles:

	Domain Separation		**Abstraction**
	Process Isolation	**X**	**Data Hiding**
	Resource Encapsulation		**Layering**
	Modularity	**X**	**Simplicity**
X	**Least Privilege**		**Minimization**

Assessment of Learning:

NAME/DESCRIPTION

Oral Questioning, Observation

Accommodations: (Examples may include closed captioning for hearing impaired students; accommodations for students with disabilities.)

I expect there could be an interesting version of this using sign language, but i don't know enough to explain how to adapt it.

Describe any Extension Activities (i.e., ideas for further work):

This paper provides some analysis of privacy issues in Facebook applications, and how most of them could operate fine with encrypted user data. the work was done by a uva undergraduate student (who now leads a security team for chrome):

> *Privacy Protection for Social Networking Platforms* (Adrienne Felt and David Evans)
> http://www.ieee-security.org/tc/w2sp/2008/papers/s3p1.pdf

Our research group works on making multi-party computation practical and easy to use in privacy-preserving applications. You can see some of our research work about this here: https://oblivc.org/. An example of the kind of application that is practical as a secure multi-party computation now is the national residency match for medical students (which is still done by a trusted third party organization).

An accessible introduction to secure multi-party computation is:

> Mike Rosulek, *Secure Your Data and Compute on It, Too*
> XRDS: Crossroads, The ACM Magazine for Students, Spring 2015
> https://dl.acm.org/citation.cfm?id=2730910

Acknowledgements:

This lesson was created by David Evans, University of Virginia, https://www.cs.virginia.edu/evans.

For more information, and any questions, feel free to contact me at evans@virginia.edu.