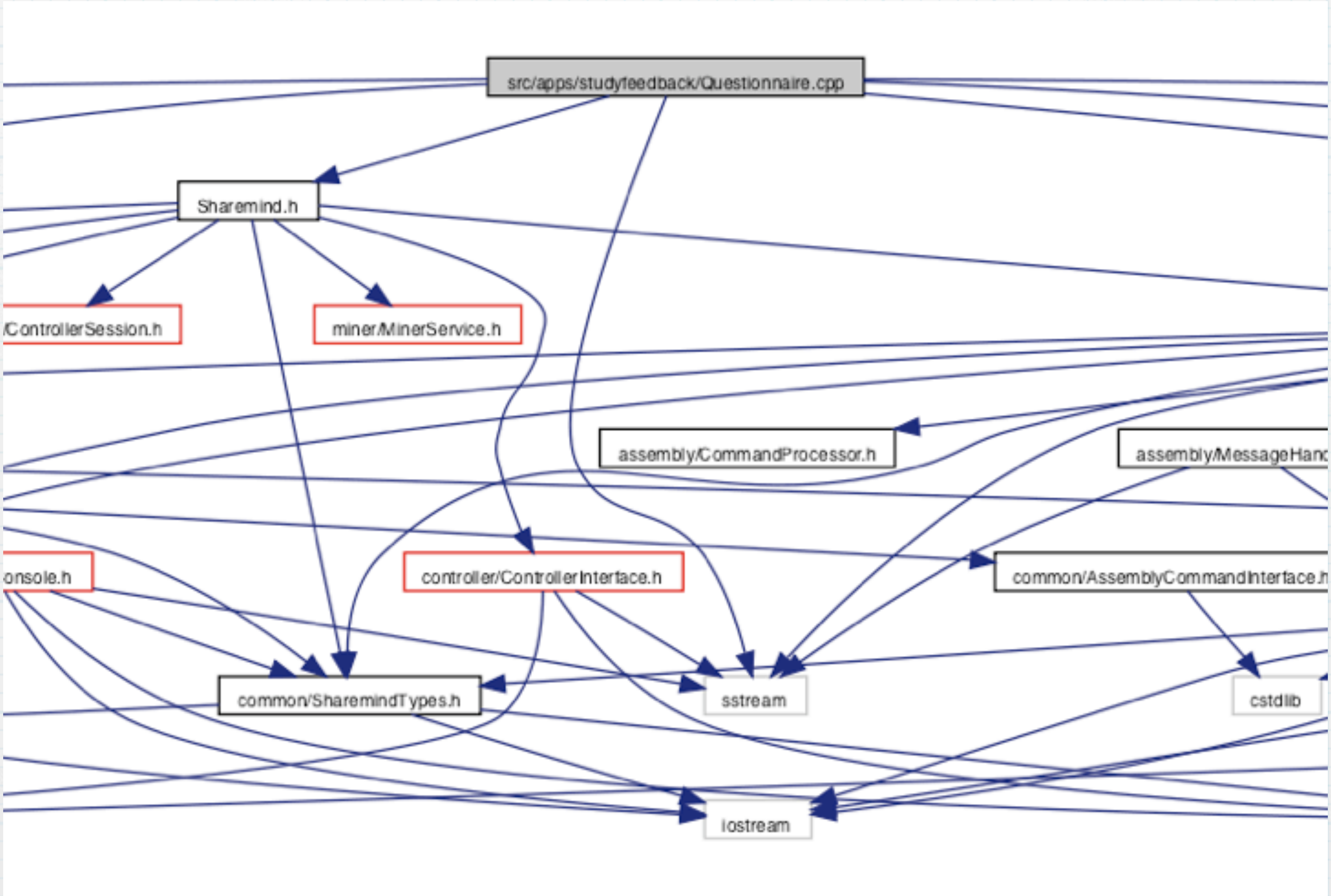
Private computations on humans

Interactive seminar on privacy-preserving data mining

Dan Bogdanov

Mandatory graph



The plans for tonight

- First, we gather some data while preserving everyones' privacy.
- Then we will process the data with privacy-preserving methods.
- Finally, we analyze the method and look at more interesting cases.

Privacy-preserving data collection

We need volunteers

- We will have three data miners.
- Three people will make one miner.
- Please, nine volunteers.
    - Two pens or pencils per team.
    - One (or two) calculators per team.

Role distribution

- Distribute the roles in the miners.
  - The Database stores incoming data.
  - The Processor performs computations.
  - The Network does message exchange.

One slide of theory

- Assume that you have a value s.
- Generate values $s_1, s_2 \leftarrow \mathbb{Z}_n$.
- Find $s_3 = s - s_1 - s_2 \bmod n$.
- Learning up to two of the values s_i will reveal nothing about s.
- This is a secret sharing scheme.

Collecting private data

1. Choose your private value s.
2. Divide it into shares s_1, s_2 and s_3.
3. Securely send the three shares to three separate data miners.

Today’s example

- The miners are going to gather and process the data.
- Everyone else will be clients.
- We will find the average age and income of the people in this room.
- Don’t worry - privacy is preserved :)

The task

- Use secret sharing to hide your age and monthly income.
  - The age is in years, the income in Estonian kroons, after tax deductions. Use integers.
- On each of the three slips write:
  - your first name and initial of the surname
  - one share of both the age and income

Cheat sheet

- Let s be your secret value.
- Generate values $s_1, s_2 \leftarrow \mathbb{Z}_{1000000}$.
- Compute $s_3 = s - s_1, s_2 \bmod 1000000$.

First name and initial:
Dan B
Age (in years):
834756
Monthly income (EEK):
65783

First name and initial:
Dan B
Age (in years):
234993
Monthly income (EEK):
340832

First name and initial:
Dan B
Age (in years):
930276
Monthly income (EEK):
[illegible]

Cheat sheet

First name and initial:	First name and initial:	First name and initial:
Dan B	Dan B	Dan B
Age (in years):	Age (in years):	Age (in years):
834756	234993	930276
Monthly income (EEK):	Monthly income (EEK):	Monthly income (EEK):
65783	340832	3rd share

- Clients - send each piece to a separate miner by raising your hand when you're ready.
- Miners - the Network will collect the inputs and the Database will write them in the table.

The interactive part

- Why was the privacy preserved?
- What kinds of attacks are there?
  1. How could the data miners attack?
  2. How could the clients attack?
  3. How could the outsiders attack?
- Is it better than standard systems?

Processing the data

First slide of theory

- We have shared values u and v.
- We want to compute $u \bigoplus v$.
- This is called share computing.
- It is usually achieved with secure multi-party computation protocols.

Addition is easy

- We are using the additive secret sharing scheme.
- The scheme is (+,+)-homomorphic.
- When we add the shares, we get the shares of the sum.
- Let’s do this now.

The task

- In each data miner the Processor should add all the shares together.
- The addition should be $\bmod 10^6$
- The Database should verify the computations.
- Clients, try to predict the results.

The interactive part

- Why was the privacy preserved?
- What kinds of attacks are there?

1. How could the data miners attack?

Publishing the results

Straight to the point

- We need three volunteers from the clients. You will be data analysts.
- Request computation results from the data miners via the Network

Number of people: M_1 #people	Number of people: M_2 #people	Number of people: M_3 #people
Sum of ages: sum_age_1	Sum of ages: sum_age_2	Sum of ages: sum_age_3
Sum of incomes: sum_income_1	Sum of incomes: sum_income_2	Sum of incomes: sum_income_3

The reconstruction

- Analysts should verify that the number of people match.
- Now separately add together the shares of age and income.
- Divide both results by the number of people.
- Say the results out loud.

The interactive part

- Why was the privacy preserved?
- Do the results seem to be correct?
- What can go wrong during the reconstruction process?

Private multiplication

One slide of theory

- Multiplication can’t be done locally, given the secret sharing we used.
- The miners have to exchange information about inputs.
- How to do that without losing the privacy guarantees?

Protocol setup

- The data miners will be the same.
- We need two factors as inputs.
- We need two volunteers.
- Both volunteers pick an input factor and divide it into shares.
- Send it to the data miners.

Round 1, generate!

- We start by creating randomness.
- Miner i will choose random values $r_{ij}, r_{ik}, s_{ij}, s_{ik}, t_{ij}$ where j is the number of the next miner and k is the previous miner.
- Send each value m_{ij} to miner j.

Round 2, hide values!

- Hide the shares with randomness:

$$\hat{a}_{ij} \leftarrow u_i + r_{ki}$$

$$\hat{b}_{ij} \leftarrow v_i + s_{ki}$$

$$\hat{a}_{ik} \leftarrow u_i + r_{ji}$$

$$\hat{b}_{ik} \leftarrow v_i + s_{ji}$$

- Send each value m_{ij} to miner j.

Round 3, compute!

- Compute shares of the product:

$$
\begin{aligned}
w_i \quad \leftarrow \quad & u_i v_i + u_i \hat{b}_{ji} + u_i \hat{b}_{ki} \\
+ \ & v_i \hat{a}_{ji} + v_1 \hat{a}_{ki} - \hat{a}_{ij} \hat{b}_{ji} \\
- \ & \hat{b}_{ij} \hat{a}_{ji} + r_{ij} s_{ik} + s_{ij} r_{ik} \\
- \ & t_{ij} + t_{ki}.
\end{aligned}
$$

Publish results

- Publish the shares of w as before.
- The volunteers can verify, if their factors exist in the product.

The interactive part

- Why was the privacy preserved?
- Is this protocol easier to attack than the addition protocol?
- Why?
- Where is the performance bottleneck in such protocols?

More multiplications

- Can we send the values for more than one multiplication in a single message?
- Do we get the same security?
- How would you multiply 1000 values together?

1000 multiplications

- How would you usually do this?
- Is this optimal in share computing?
- How many protocol executions?
- How could we decrease the number of protocol executions?

If you want more, find out about

sharemind

http://sharemind.cs.ut.ee/