Students and Taxes: a Privacy-Preserving Social Study Using Secure Computation

Dan Bogdanov[1], Liina Kamm[1], Baldur Kubo[1], Reimo Rebane[1], Ville Sokk[1] and Riivo Talviste[1,2]

[1] Cybernetica, Tartu, Estonia
{dan.bogdanov, liina.kamm, baldur.kubo, reimo.rebane, ville.sokk, riivo.talviste}@cyber.ee
[2] University of Tartu, Institute of Computer Science, Tartu, Estonia

Abstract. We describe the use of secure multi-party computation for performing a large-scale privacy-preserving statistical study on real government data. In 2015, statisticians from the Estonian Center of Applied Research (CentAR) conducted a big data study to look for correlations between working during university studies and failing to graduate in time. The study was conducted by linking the database of individual tax payments from the Estonian Tax and Customs Board and the database of higher education events from the Ministry of Education and Research. Data collection, preparation and analysis were conducted using the Sharemind secure multi-party computation system that provided end-to-end cryptographic protection to the analysis. Using ten million tax records and half a million education records in the analysis, this is the largest cryptographically private statistical study ever conducted on real data.

Keywords: privacy, statistics, secure multi-party computation, case study

1 Introduction

Information and communication technology (ICT) is a growing industry where highly skilled specialists are in demand. This causes concern to both industry, where the wages keep rising, and the academia that cannot often match the pay grades offered by the industry. The universities in Estonia formed a hypothesis that students who work during their studies, do not graduate in the allotted time. Moreover, many students quit before graduation, thus, not acquiring the skills needed for building more complex ICT systems.

In this paper, we describe a big data study on Estonian government data that researches this topic and uses privacy-enhancing technologies to protect personal data. We collaborated with a team of social scientists who designed a statistical study that links tax and education records to determine the working habits of both ICT and non-ICT students. However, running the actual study would normally be impossible, as data protection and tax secrecy legislation

significantly hinder such studies. We explain the problem and legal situation in Section 2.

The main contribution of this paper is the procedural and technical description of a statistical study that used a combination of cryptographic secure multiparty computation (MPC) together with organisational measures and microdata release controls. We will now introduce our contributions in more detail. To our knowledge, none of these challenges have been addressed at the scale demonstrated in this paper.

First, we implemented data import and the full statistical study using the Sharemind MPC platform [4]. We were able to re-use some Sharemind functionalities, including the Rmind statistical analysis system [5]. However, we also implemented additional features such as data transformations, new attribute calculation, aggregations, custom merging procedures and visualisation using the programming tools provided by the Sharemind platform [15]. Sections 3 and 4 describe the technical solution.

Second, we describe how we convinced regulatory bodies and data owners to provide the data for analysis using MPC. For this, we prepared a detailed explanation of the security features of our solution and described it to data owners and regulatory bodies. Especially, we worked with the Data Protection Inspectorate who, after a lengthy review, accepted the privacy guarantees provided by our solution as going beyond the level of protection required by the Personal Data Protection Act. To satisfy tax secrecy requirements, we worked together with the Tax and Customs Board to perform a code audit and solution testing of the MPC-based analysis platform. Finally, we arranged for contracts between the data owners, Sharemind hosts and the statistician. Our work and methods towards a real-world deployment are detailed in Section 5.

Third, we supported the study when it was conducted. This included preparing installation manuals, performance profiling and technical support. Our study ensured that the security assumptions of the used MPC protocols were satisfied—the MPC platform was hosted and administered by three individual parties and we had no control over the whole system. This was a critical feature for acceptance of the technology, but also a key challenge, as it required MPC hosts to commit resources to the study. Furthermore, the statistical analysts used the MPC-based analytics tools on their own, without us overseeing their every step. We describe the execution of the study in Section 6.

To our knowledge, this is the largest real-world MPC application to date. Our analysis system processed over 600 000 education event records from the Ministry of Education and Research and over ten million tax payment records from the Tax and Customs Board on a deployment running over the public internet. Other successful real-world deployments include the Danisco sugar beet auction in Denmark that is the longest continuously running MPC application built, but processes significantly less data with much simpler functionality [8]. The financial reporting case study in [7] was also deployed on the public internet, but did not process more than a hundred records. There have also been attempts at MPC-based data analysis applications on real-world data that have reached the

necessary technology level, but met resistance during the deployment [14]. Recently, Damgård et al. demonstrated a financial benchmarking prototype jointly evaluated with banks that performed linear programming on 2,500 records [10] using MPC secure against an active adversary. While multiple other prototypes of privacy-preserving statistical analysis have been published, none of them have been validated in real-world use [13, 9].

We also validated our results by asking social scientists to run a parallel study with anonymisation technologies accepted by data owners today. We saw how the use of 3-anonymity caused 10%-30% of sample loss, depending on the demographic group. We show that secure multi-party computation can be used in the real-world to solve practical data protection problems, and can provide better privacy and accuracy than technologies deployed today.

2 Privacy-preserving analysis of government data

Modern governments are increasingly data-driven. Government agencies collect citizen data for their day-to-day operations. Some of this information is made freely available following the principles of *open data*. Such information is available to anyone, for use in innovative new services or analysis of public policy (e.g., use of national or local budgets). However, not all government data can be freely shared. Notably, personal data are often missing from open data services, as there are legal barriers preventing their use. This effectively prevents interested parties such as social scientists and economists from linking and analysing these databases.

The study in this paper is inspired by the following public policy concern. According to data from the Ministry of Education, 43% of students who enrolled in an ICT curriculum in years 2006–2012, quit their studies by December 2012. Universities in the Estonian Association of Information Technology and Telecommunications (ITL)—an organisation of companies and universities working in the field of ICT—hypothesize that the high drop-out rate is connected to students being hired as early as their first year and that the students favour their wages over a university degree. Others argued that the high drop-out might be related to the sudden increase in students enrolling in ICT subjects who find the subject too hard. Thus, a research problem was stated—is working during studies related to high drop-out rate?

Such studies can be conducted in two ways. First, one can conduct a survey and ask a number of students about their working habits and studying career. This way, the students will consent to the processing of their data individually. However, covering the majority of the students this way will be very expensive and the responses might be biased if the students are ashamed or angry over their academic achievements.

Alternatively, in today's age of big data, we should be able to tap into existing data stores that cover the entire population. We can get information about a person's employment from the payment records of social taxes. These can also include information about the kind of company (ICT or non-ICT) that

the student has been working in. The Ministry of Education keeps records on higher education—events like students enrolling and graduating—with the date, institution and curriculum related to the event. As another benefit of analysing the whole population, we can use simpler statistical methods that do not have to take into account the relation of the sample size to the whole population.

In their natural state, the education and tax records databases are not linked. The tax records are held by the Tax and Customs Board that operates under the Ministry of Finance. To study the described problem, the two databases must be joined and the data from both used in the analysis. However, these two institutions have to adhere to the same laws as companies do if a joint study is planned. The privacy issue, in this case, arises from the Estonian Personal Data Protection Act that regulates the use of education data [1, §4-§6], and the Taxation Act that regulates the use of tax data [2, §26-§30] which defines requirements for tax secrecy that can prevent such analyses from being conducted today.

Needless to say that these data are useful in different analyses, but accessing them is not an easy process. While the Ministry of Education and Research can give data out for analysis under non-disclosure agreements with the analyst, the Tax and Customs Board cannot. The latter's current policy requires that, before release, it pre-aggregates data into groups based on demographic attributes to achieve something similar to k-anonymisation [17]. Such a pre-aggregation for our study would be done in the following way.

1. The statistician signs a non-disclosure agreement to obtain pseudonymised education data.
2. The statistician forms groups of individuals based on demographic attributes and sends the groups to the Tax and Customs Board.
3. The Tax and Customs Board uses the pseudonyms to add income data to each demographic group in an order not related to the order of pseudonyms. Each group with less than three individuals will remain empty. The groups are returned to the statistician under a non-disclosure agreement.
4. The statistician completes the study on income data provided by demographic groups.

In parallel with our MPC-based study, the statisticians used this pre-aggregation method to perform a validation study. While this approach prevented the statisticians from learning the relation between an income and a pseudonymous individual, it also caused significant losses in the data. People who have a unique combination of attributes were left out from the analysis. Distinct groups can be rather small to begin with. Considering that outlier students might come from diverse social backgrounds, leaving them out of the study reduces its capability to explain the effects being analysed.

Thus, our goal with this line of research is to replace the anonymisation mechanism currently in use with a privacy-enhancing technology that has provable privacy guarantees and no loss of accuracy.

3 Tools for a privacy-preserving statistical study using MPC

3.1 The Sharemind MPC framework

Sharemind is a programmable distributed secure computation framework [4] supporting MPC. Sharemind is designed to be a database and application server that provides cryptographic protection for data during both storage and processing. The applications of Sharemind are implemented using the SecreC programming language supporting hybrid applications that allow public and encrypted operations to be performed in the same program. The programming and secure execution model of Sharemind supports different secure computing protocols abstracted as *protection domains* [6]. When the Sharemind servers execute compiled SecreC programs, they automatically run MPC protocols to process private data. Information is uploaded and queries are sent to a Sharemind installation using client applications. The import tools apply the relevant cryptographic protection mechanism to the input data and the data analysis tools recover the results from protected outputs.

At the time of this work, Sharemind's best-performing protection domain was the three-party protocol suite based on additive secret sharing [4]. Its protocols allow any number of *input parties* to use additive secret sharing on their private inputs and send them to three *computing parties*. These computing parties engage in MPC protocols to obtain secret-shared results from secret-shared inputs. The computations are *oblivious*, meaning that the parties learn neither the input nor the output values. In addition to MPC arithmetic, Sharemind supports a number of efficient data-oblivious algorithms such as sorting, shuffling and linking. The secret-shared results of operations can be sent to any of the *result parties* who can reconstruct the results. This model does not require each input party or result party to be included in the MPC protocols, saving on both performance and complexity.

From a privacy perspective, this protocol suite allows input data from any number of input parties to be processed without anyone but the data owner seeing the input values. The implementation of the protocol suite provides security against passive adversaries, which is sufficient for preserving privacy from computing parties and result parties.

3.2 The Rmind statistical analysis tool

The Rmind tool was developed to reduce the complexity of using MPC in statistical applications [5]. It is designed to mimic the scriptable command-line based system R[1]. Rmind is implemented in SecreC and it is installed into the Sharemind Application Server hosts as a package of compiled SecreC programs. These provide secure storage, computation, and statistical algorithms as a service. The analyst installs a client application that parses an R-like language

[1] The R Project for Statistical Computing. http://www.r-project.org

and uses the service for secure execution. The client application never receives private data, only aggregated results of the operations performed by the analyst. Before this study, Rmind supported data transformation (e.g., sorting and merging database tables), descriptive statistics (e.g., quantile estimation), null hypothesis significance testing (e.g., Student's t-test), outlier detection, linear and logistic regression and multiple testing correction (e.g., Benjamini-Hochberg false discovery rate).

While MPC with specially designed analysis algorithms protects the data of the input parties from everybody else, we still need to ensure that the outputs of the study do not leak information about the inputs. The Rmind tool enforces adherence to the study plan and deploys several microdata protection mechanisms. While statistical output privacy mechanisms such as differential privacy are supported on Rmind, they were not deployed for this study as the goal was to demonstrate results as accurate as possible. For more information on the other privacy mechanisms in Rmind, see [5].

For this study, we extended Rmind with several new features. We added a configurable aggregation procedure with support for multiple functions (count, sum, mean). We modified the existing database table join procedure to support outer left and outer right join in addition to inner join. While we also implemented a procedure for logistic regression, the tight schedule of the study did not allow us to use it in practice.

4 Design of the privacy-preserving study

4.1 Stakeholders and deployment

Figure 1 shows the stakeholders of the statistical study and the flow of data between them. The Estonian Association of Information and Communication Technology (ITL) was the customer for the study. They stated the study questions. Based on these questions, statisticians in the Estonian Center for Applied Research (CentAR) designed the statistical study. CentAR fulfilled the role of the result party, using the improved Rmind tool to send queries and prepare the final report.

The Sharemind MPC system was hosted by three computing parties, the Estonian Information System's Authority, the Ministry of Finance Information Technology Centre and Cybernetica. They provided the server and networking resources needed to run the study. Each used their own data centres for hosting and applied their information security controls. The Ministry of Education and Research and the Tax and Customs Board were the input parties who used the import tool to upload their database contents. This way, the private information never left the data owner unencrypted.

Only one step of the study was conducted without MPC. The Ministry of Education and Research determined the list of students to be included in the study and sent their IDs directly to the Tax and Customs Board. This way, we did not have to include the tax payments of all citizens in the study and reduced

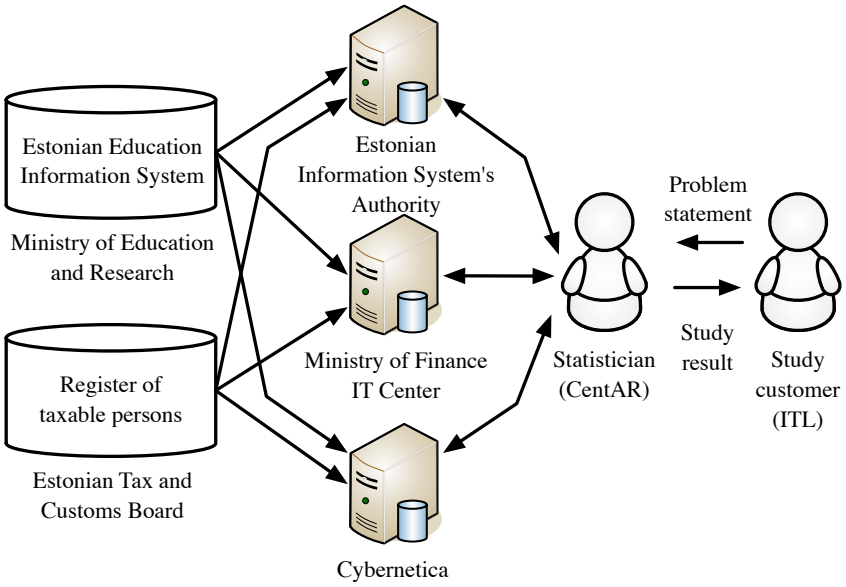

Fig. 1. Stakeholders of the privacy-preserving statistical study

the complexity of the study. We discuss the practical and legal implications of this further in Section 5.1.

4.2 Data import and pre-processing

The most challenging part of this project was the privacy-preserving transformation of the data into an analysable format. In real world studies, data owners collect information to support their own business process and, thus, it is not in a format that is suitable for statistical analysis. For this study, we had to extract the data from the input format, transform and merge the two data sources and load it into an *analysis table*. This *extract, transform, load* (ETL) process turned out to be the most time-consuming part of the analysis.

Hypothetically, transformations on the individual input tables could have been performed by the input parties. However, in our study, the data owners were explicitly interested in reducing their data analysis burden. We also considered adding pre-processing to our data import tool. However, this would have made the tool less universal. Moreover, if these aggregations are done before data sharing and import, we cannot later query more detailed data without a new data import stage, should the study plan change. Therefore, we carried out the aggregation process in the MPC setting.

The data was imported as two separate sets—the education information of students who finished or started their studies between 2006 and 2012, and their

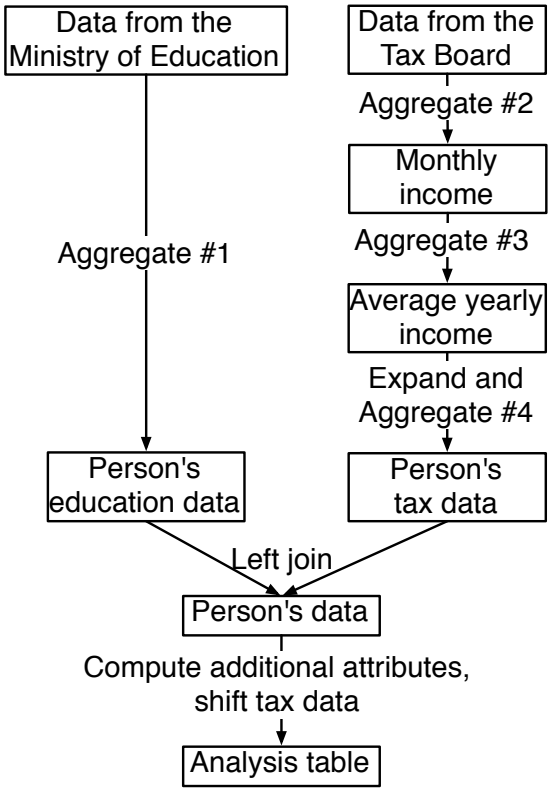

Fig. 2. The privacy-preserving ETL process of the study

income from 2004 to 2013 to also evaluate the potential salary prior to studies. Figure 2 shows the privacy-preserving ETL process for the the study.

The ETL process for our study can be roughly divided into three subtasks, all of which were performed using MPC. First, we needed to extract the education data and transform them into a format where each row corresponds to one person's studies in one curriculum. Second, we wanted to extract the salary data and transform them into a format where each row corresponds to one person's salary information for all ten years. And third, we need to join the two tables and transform the obtained table into the final analysis format with the following three types of attributes.

- Fixed attributes, (e.g., whether the student was working in an ICT company during their studies).
- Attributes ranging over years of study (e.g., whether a person was working during study year i).
- Attributes ranging over years after graduation (e.g., whether a person was working during year j after graduation).

We now describe the additional features we implemented into RMIND to complete the ETL process.

4.3 Privacy-preserving aggregation

Aggregation is a standard database operation that groups items based on a chosen attribute the values of which are equal in all rows within the group. Then an aggregation function is applied to the columns in each group and, as a result, we receive a dataset with one aggregated row for each group. Consider as an example the GROUP BY operation used in the popular database query language SQL.

In this section a key is an attribute by which a row is queried. A *composite key* is a set of attributes (or keys) by which we perform queries. For example, if we want to get a set of all women in the dataset, we use the gender column as the key column. If we want all women in an age group, we use the gender and age columns as the composite key.

More formally, let $\mathbf{A} = a_{s,t}$ be a dataset with m attributes and N records, $s \in \{1, \ldots, N\}$ and $t \in \{1, \ldots, m\}$. We denote rows of this dataset as $\boldsymbol{a}_s$. Let $C \subseteq \{1, \ldots, m\}$ be the set containing the indices of the columns by which the rows in the table will be grouped. Together the columns in this set C constitute a composite key. Then $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ is a set of n matrices ($1 \leq n \leq N$) with m columns where each matrix $\mathbf{X}$ is composed of rows $\boldsymbol{a}_g$ such that for each $c \in C$, the elements of attribute c are equal in all rows.

Let $j \in \{1, \ldots, v\}, 1 \leq v$ and let $\boldsymbol{b}$ be a tuple of indices of attributes, such that $b_j \in \{1, \ldots, m\}$. In addition, let $\boldsymbol{op}$ be a tuple of identifiers of aggregation operations so that $op_j \in \{\mathbf{random}, \mathbf{max}, \mathbf{min}, \mathbf{sum}, \mathbf{avg}, \mathbf{count}\}$. Let q_i be the number of rows in the grouped matrix $\mathbf{X}_i, i \in \{1, \ldots, n\}$

The aggregated dataset $\mathbf{D} = d_{i,b_j}$ has v columns and n rows, so that

$$\begin{aligned} \mathbf{D} = \{d_{i,b_j} \mid\ & op_j(y_j) \wedge y_j = (x_{g,b_j}) \wedge \\ & x_{g,b_j} \in \mathbf{X}_i \wedge g \in \{1, \ldots, q_i\}\}. \end{aligned} \tag{1}$$

Equation (1) describes the resulting aggregated dataset $\mathbf{D}$ where the element d in the i-th row and b_j-th column (denoted in the definition as d_{i,b_j}) is obtained in the following way. The j-th operation from the set of operations is applied to the b_j-th column of each grouped matrix $\mathbf{X}_i$.

Note that the input identifiers of attributes $b_1, \ldots, b_v$ can have recurring elements, as the analyst might need to perform several operations on elements of one attribute. The possible aggregation operations *op* are the following:

- **random** - taking the first available element, as all elements are assumed to be equal (equality will not be checked),
- **max** - taking the maximal element (this also serves as the disjunction operation for Booleans),
- **min** - taking the minimal element (this also serves as the conjunction operation for Booleans),

Algorithm 1: Privacy-preserving aggregation

Data: Dataset $[\![\mathbf{A}]\!]$ with m attributes and N records, indices of attributes $C \subseteq \{1, \ldots, m\}$ by which to group the dataset, tuple of indices $(b_1, \ldots, b_v), b_i \in \{1, \ldots, m\}, (1 \leq v)$ of attributes that will be included in the resulting dataset, tuple of identifiers $(op_1, \ldots, op_v)$ of aggregation operations

Result: Dataset $[\![\mathbf{D}]\!]$ with v attributes and n records $(1 \leq n \leq N)$

```
Obliviously shuffle rows of [[A]]
Combine the values of attributes c ∈ C into composite keys [[k_1]], ..., [[k_N]]
Use oblivious AES to encrypt the composite keys, denote them as [[k'_1]], ..., [[k'_N]]
(k'_1, ..., k'_N) ← declassify([[k'_1]], ..., [[k'_N]])
Let n be the number of unique groups in (k'_1, ..., k'_N)
for i ∈ {1, ..., n} do
    for j ∈ {1, ..., v} do
        Obliviously apply operation op_j to elements of attribute b_j within group i
    end
    Write the result into [[D]]
end
return [[D]]
```

- **sum** - summing the elements,
- **avg** - computing the average of elements,
- **count** - counting the available elements.

The privacy-preserving aggregation procedure is shown as Algorithm 1, where a secret-shared value x is denoted as $[\![x]\!]$. The algorithm uses a number of oblivious operations that are performed using MPC to prevent any information leakage. As seen, the groups are formed according to equal declassified (i.e., recombined) ciphertext values. This algorithm works similarly to the join operation discussed in [16].

This algorithm leaks the number of groups and the number of elements in each group. The former is a desired result. As for the latter, we use shuffle at the beginning of the algorithm so the number of elements in each group cannot be linked to the original dataset. There is a possibility of performing aggregation without leaking the number of elements in each group. This can be done by obliviously adding dummy elements into the set to compensate for the groups that have fewer elements. For details, see the size unification protocol from [16] where the same idea is discussed for the database merging operation.

4.4 Transforming education data

The Ministry of Education and Research imported values for the following attributes: person ID, gender, year of birth, year of observation, level of study (Bachelor's, Master's, PhD, professional higher education), curriculum, length

of nominal period of curriculum, school, date of admission, status of studies (in progress, quit, graduated), date of graduation/termination.

This dataset was too detailed for our needs. We wanted the data to be in the format where each row corresponds to one person's studies in one curriculum. The original records had the following structure.

1. One record for each person's each study position for each year from 2006 to 2012.
2. One record for the enrolment of a person in a study position prior to 2006.

We needed to aggregate the records based on unique people. We also needed to keep separate records for a person's studies in different curricula, so the unique identifier in this case was the person ID combined with the curriculum ID. This is depicted as Aggregation 1 in Figure 2. We used the aggregation procedure from Algorithm 1 to group the different records with the same unique identifier together. The year of observation became obsolete during this aggregation and was left out of the resulting dataset.

The date of admission was renamed as date of first admission to account for students who had started in the same curriculum several times. The minima of the values were selected as the date of first admission. The date of graduation/termination was similarly taken as the maximum of the available values to obtain the latest of these dates. The status of studies was the trickiest of the attributes. There were four options—in progress, quit, graduated, in progress at the end of 2012—and the logic was the following.

1. The resulting status can never be "in progress".
2. The resulting status can only be "quit", if the person never graduated from the curriculum in question not even after several tries.
3. The resulting status is "graduated", if the person graduated from the curriculum in question. A person is not allowed to reapply for a curriculum they have graduated from.
4. The resulting status is "in progress 2012" if the person's studies were still ongoing by the end of 2012.

If we give these options codes 1 for in progress, 2 for quit, 3 for graduated, and 4 for in progress 2012, we can take the maximum of these values as the result.

The original dataset did not have the fourth option. In fact we only needed this for one special case in which a person had quit their studies in a curriculum, then re-enrolled, and was still studying at the end of the period under analysis (i.e., at the end of 2012). In such a case, the status after aggregation must be "in progress" instead of "quit", which the maximum operation would have returned without the added fourth option. We added this option with an MPC operation that changed the values that are "in progress" in 2012 to "in progress 2012". Now taking the maximal element in a group would always return the right code.

Note that, during data import, codes are given to classifier values automatically based on their order of appearance. Hence, an imported dataset might not immediately have the classifier options we require. For this purpose, we added

a feature to RMIND to allow the users to obliviously recode the classifier values by providing the codes they wish the options to have. Thus, regardless of the codes that the classifier denoting the status of studies originally had, they can be reordered so that the maximal element is chosen during aggregation.

All other values were the same within a group based on a person's ID and the curriculum ID. In these cases, a random element was taken from the values of each of these attributes. Having done this, we had obtained our desired table format.

4.5 Transforming tax data

The Tax and Customs Board input data with the following attributes: person ID, year, month, payment for which social security has been charged, dividend income, income from self-employment, whether the employer was from the ICT field, whether the employer was a member of ITL.

Similarly to the education dataset, the tax dataset was too detailed for our needs. Specifically, there was a record for each source of income per month per person. This means that if a person got a salary from two companies for a year, there were 24 records in the table for that person that year. For our study, we only needed information about the average salary per year and the number of months that a person worked during a year. In fact, our goal was to receive a table where each row corresponded to one person's salary information for all ten years.

As the first step, we added some new attributes to the dataset based on the existing data. Namely, we added attributes for whether the person received income from self-employment and whether the person received dividend income. These attributes generalised some of the more detailed attributes in the original dataset.

Second, we wanted to combine a person's salaries during one month for the cases where a person was holding multiple jobs (Aggregation 2 in Figure 2). For this, we grouped the data by person ID, year and month, and calculated the sum of the payment attributes within a group. During this operation, we left out the detailed attributes (dividend income, income from self-employment) and took the maximum of the corresponding generalised attributes.

Next, we averaged the monthly income into average income per year (Aggregation 3 in Figure 2). We did this per month of employment, meaning that if a person worked for one month, their average yearly income would be that month's salary. For this, we grouped the dataset by person ID and year. We computed the average income and counted the records in each group to get the number of months that the person had worked during that year. In addition, we took the maximal element for the Boolean attributes (is ICT curriculum, works in a company belonging to the ITL, was self-employed, received dividend income) again.

Next, we expanded the table adding an attribute for each year for all the attributes: income, number of months, is an ICT company, is not an ICT com-

pany, works in an ITL company, received income from self-employment, received dividend income. Our table had $7 \cdot 10 = 70$ new columns.

Let us look at this process using an example record for person X for year 2006. This record has all the attributes for that person during that year. After expansion, the record has 70 new columns. To fill these columns we do the following.

1. We used MPC to build a mask vector for each record based on its year $a \in \{2004, \ldots, 2013\}$. The mask vector is an element-wise secret-shared binary vector, where 1 denotes the position(s) where the predicate holds. In our example, the mask vector is built by comparing 2006 to all possible years, so the resulting mask vector is $(0, 0, 1, 0, \ldots, 0)$.
2. For each of the attributes we expanded, we multiplied the corresponding attribute with the mask vector and saved the result as the corresponding expanded attribute. In our example this means that the original average salary s for person X is saved as $(0, 0, s, 0, \ldots, 0)$.

The result was a fairly sparse table containing one record per person per year. The final step was to group by person ID (Aggregation 4 in Figure 2) to receive an expanded table that had one record per person which included the data for all the years in question. Hence, we used sum as the operation for all of the grouped attributes. This worked for the Boolean attributes as well, as within a group, each column had exactly one value.

4.6 Privacy-preserving record shifting

After separate processing, we merged the education and salary information into one data table using left join, so that people with no salary information were also retained in the resulting database.

We required the salary information to be relative to the person's studies, meaning that we wanted the salaries to correspond to the study years i. Hence, we converted the data so that, for each student, there would be an attribute for work and salary information for each year i of their studies beginning with admission. The resulting table was relatively sparse. As an example, consider a person who started her studies in 2007. For this student, the salary information in the attribute $\mathbf{salary}_1$ is information from 2007. For another student, starting in 2010, the same attribute $\mathbf{salary}_1$ contains salary information from 2010.

To fix this, we needed the possibility to shift vector elements. The shifting function is a typical transformation from physical time into semantic time used in statistical studies. Oblivious vector shifting means that elements in the vector are shifted left by a number of spaces k, where k is a private value that none of the computation parties know. The k-th element and all those to the left of it are copied to the end of the vector and are marked as not available in the corresponding availability vector.

Using this function, we added attributes for all the years i since admission for the following information:

– Whether the person was working during year i;
– Whether the person was working and studying during year i; whether they were working for at least 3, 6, and 9 months during year i;
– Exactly how many months the person was working during year i;
– Salary during studies during year i; salary if they were working for at least 3, 6, and 9 months during studies in year i;
– Whether the person was working in an ICT company during the nominal period during year i;
– Whether the person was working in a non-ICT company during the nominal period during year i;
– Whether the person was working in an ITL company during the nominal period during year i.

We also created another shift to reflect working during years j after graduation with the following attributes:

– Whether the person was working during year j after graduation, whether they were working for at least 3, 6, and 9 months during year j;
– Salary during year j after graduation; salary if they were working for at least 3, 6, and 9 months during year j after graduation.

Unfortunately, oblivious shifting required us to align the dataset with the person whose studies had been the longest, i.e., if the earliest admission date was from 1994, we would have to make columns for study years 1 through 20 for everyone. This made the data matrix extremely sparse because most of the studies will be within 2 to 6 years. In addition, we did not have salary information before 2004, so the extreme cases could only be used to analyse graduation in expected time for different fields.

We created this sparse data matrix because later it would be easier to formulate the necessary analysis queries. Recall, that we had the salary data as a vector of ten values for years 2004-2013. We needed to obliviously shift this salary vector according to each person's individual year of admission. As there were ten elements in the salary vector, but we might have more that ten years of study, we also added a padding of zeros before the shifting process. The shift essentially selected the first element from the salary info of the year of admission and added all the previous salary data to the end of the vector, changing the corresponding last elements to not available.

To obtain the necessary attributes, the shift based on years since admission ranged from one to $2013 - \mathbf{min}(\text{year of admission}) + 1$. We needed a shorter period of time for the shift reflecting working after graduation, as we knew that the earliest graduation information we had was from 2006. Hence, we shortened the salary vector to include only years 2007 to 2013 and shifted this, instead, based on the year of the end of studies plus one. The index for this shift ranged from one to seven.

We generalised some of the attributes that were too detailed in the joined data table. Namely, based on the year of graduation, we made attributes for dividend income before and after graduation, and did the same for income from

self-employment. We also found out whether the person was working in an ICT company or an ITL member company during their studies.

4.7 Final analysis and result presentation

After the ETL process was completed, the statisticians performed a number of queries to answer the questions motivating this work. Thanks to a well-designed ETL process, these were relatively easy to perform, even with MPC. We list the queries in Appendix A.

Visualisation through plots allows people to perceive analysis results more easily. Thus, the goal was to have the final results visualised by Rmind. Rmind supports multiple types of plots—histograms, boxplots, heatmaps and a generic two-dimensional plot of lines or points. Plotting is done using the published results after Sharemind has securely performed a query. The Rmind client application uses the gnuplot[2] tool to visualise the results. While gnuplot has a lot of features, its programming interface turned out to be not suited for this kind of application.

Tools like gnuplot expect the user to input data that the tool aggregates itself (e.g., computes the quartiles for a boxplot). In the private setting the data are aggregated in Sharemind, hence, we cannot give gnuplot the source data. Thus, we had to work around this limitation.

The Rmind plotting procedures have a similar interface to tools in traditional statistical analysis software. For example, the command to plot a histogram of the vector `x` is `hist(x)`. The `hist` procedure will return an object representing the plot which can be displayed or saved as an image file. The main problem with this implementation choice is that when parameters of the plot are changed (e.g., dimensions, axis labels), the procedure has to be executed again, including the private aggregation which may be costly.

Also, when the study has been finished, the data will be deleted, but the analyst may need to change graphical parameters of the plot to suit different outputs of the study (articles, websites). In our study, we modified the Rmind client application to output generated gnuplot scripts so that they could be changed later. However, this was inconvenient for the analyst and a better solution will be required for future studies.

5 Deploying MPC in practice

5.1 Achieving legal compliance

Estonia, being a member of the European Union, has implemented the EU Data Protection Directive 95/46/EC with the Personal Data Protection Act [1]. All personal data processing has to be conducted in compliance with this law. Data owners expect the organisation planning a study to demonstrate the legal justification according to which they can process personal data and show compliance

[2] gnuplot homepage. http://www.gnuplot.info/

with it. If sensitive personal data are involved, an explicit permission needs to be acquired from the Data Protection Inspectorate. In Estonian state information systems, tax data of natural persons are considered sensitive.

The second legal barrier was the Taxation Act [2]. It explicitly names the few organisations to whom personal tax data can be forwarded for use in governmental statistical or financial analysis. The Tax and Customs Board required us to prepare an application to the Data Protection Inspectorate and explain how the study complies to the law. If data extraction is too big a burden for the data owner, or if they see risk to the data, data owners reserve the right to deny access to their data even if the legal compliance and permission from the Data Protection Inspectorate has been acquired.

We prepared the application to the Data Protection Inspectorate as a combination of standard and non-standard elements to describe MPC as a new privacy technology. The standard elements included the formal application itself with the detailed description of data on the attribute level, and details about which organisations would be processing the data using MPC.

The other standard component was the description of physical, organisational and technological security control deployed to protect personal data during the whole study—from acquisition to deletion. The non-standard element we added to the application was an explanatory memo describing Sharemind and its underlying cryptographic methods, secret sharing and MPC. The explanatory memo used a risk-based assessment of the whole study and detailed all the participants and their roles in each step of the process.

We also used a process map in Business Process Modeling Notation (BPMN) to help the inspectorate better understand the organisational setup of the study. In addition to the supplied application, the description of used information security controls and the explanatory memo, the Inspectorate also required a meeting where they wanted to discuss the privacy guarantees of data analysis using MPC.

When selecting the justification for legal processing of personal data, getting consent from all persons participating in a study is one option. In the case of hundreds of thousands of participants, as in a study leveraging existing databases, this is not feasible. We selected the approach with novel technological security measures, where data remains encrypted during processing. However, the difficulty of explaining the novel technology to all involved technological and legal experts was a significant challenge and took several months.

After reviewing our application to process personal data during the study in encrypted form the Data Protection Inspectorate indicated that we did not require a permission to process personal data. In fact, their reply stated that, in the described form, we were not actually processing personal data and, thus, would not need permission for an MPC-based analysis with only statistical summary output [12].

We note that, since the study, we have asked a research team of privacy legislation at the University of Göttingen to evaluate the technology and deployment scenario within the FP7 project PRACTICE. The resulting legal analysis confirmed that the way the study was designed, no personal data processing took

place in Sharemind [11]. However, the analysis concluded that, under current law, no general guidance can be given, as each study plan would have to be checked to make sure that the MPC engine is not made to publish personally identifiable information—only summary statistics.

Having received the assessment from the Data Protection Inspectorate, we proceeded with setting up the prerequisites to receiving data for the study. The IT department of the Ministry of Finance; legal and oversight departments of the Tax and Customs Board; legal, IT and analysis departments of the Ministry of Education and Research, all needed both written and face-to-face explanations of MPC and the data analysis system we built using Sharemind. In addition, the IT department of the Ministry of Finance (RMIT) wanted to conduct a code review of the deployed software before giving their permission to proceed. As RMIT is the legal processor of tax data, they were representing its legal controller, the Tax and Customs Board in the technical risk assessment. They also deployed the study software internally so their analyst could test what could and what could not be learned about the collected data through queries.

We delivered the full source code of the Sharemind Application Server, Rmind statistical analysis application and the data import tool to RMIT. During the review, we answered their questions about the coding conventions, the structure of the source code tree, and the build system including internal and external dependencies, as well as our testing practices and the coverage of automated regression tests.

The verdict from RMIT was that the software suite has been engineered with good quality and according to conventions. Due to a lack of cryptographic expertise, they were unable to assess the security of the cryptographic protocols, but accepted successful peer review in the research community as sufficient for now. After having reviewed the Sharemind source code, RMIT gave us the permission to proceed with deployment, acceptance testing and preparation of contracts. Only after all contracts were prepared and acceptance testing with generated data passed, was the internal control department of the Tax and Customs Board prepared to review the process, the contract chains and the risk analysis. They gave their permission for the Tax and Customs Board to contract the analyst and release actual tax data of the involved students to be secret shared and processed with Sharemind.

5.2 Secure multi-party contracting

The data owners' need to legally cover the processing of the personal data of citizens was the main driving force for the contracting. As the Ministry of Finance IT centre is the legal processor of tax data, their legal basis for processing those data was already covered in their statutes. Taking part in the study process as a computing party gave them a strong technical control over everything that was computed. Thus, they could participate under a single agreement.

The Ministry of Education and Research, however, did not take part in the computation and, thus, they had to contract the set of organisations participating in the secure multi-party computation to have the required organisational

security measures in place. As we learned from the legal risk analysis [11], data processing with MPC in Sharemind on secret-shared data does not constitute personal data processing in the sense of the Personal Data Protection Act (that is compliant with the current EU Data Protection Directive). However, when the Ministry of Education and Research sent student IDs to the Tax and Customs Board for a pre-filter of students, this was done without MPC. This clearly had to comply with the Personal Data Protection Act and, therefore, we arranged for a separate contract between them to cover this exchange.

The trilateral analyst contract between the result party (CentAR) and the input parties (Ministry of Education and Research, and the Tax and Customs Board) regulated the usage of personal data to be exchanged between the Ministry and the Tax and Customs Board, and personal data to be secret shared into Sharemind. The quadrilateral contract between the input parties and computation parties was signed between the Ministry of Education and Research as the input party and the Estonian Information System Authority, RMIT and Cybernetica as computing parties. Thus, these organisations took the obligation to jointly compute the allowed functions and to refrain from collusion and declassifying personal data, fulfilling the security assumption of the MPC protocols used in the study. They also agreed with the data owner on the deletion date of study data.

The computing parties considered it important that they did not have to deal with legal risks arising from the processing of personal data. It would also have been possible to completely eliminate the movement of personal data between input parties, if the selection of persons to the study happened using MPC. However, this was not done as the implementation of the study was already challenging enough.

5.3 Secure software delivery

After the code review, RMIT compiled all of the necessary components (the Sharemind Application Server, the Rmind statistical analysis application and the data importing tool) from the source code provided by Cybernetica. RMIT set up its own Sharemind Application Server installation and shared the binaries with RIA. Cybernetica compiled and deployed the binaries on their own. After the initial delivery by RMIT, it was decided that Cybernetica will deliver application binaries to all involved parties during the remainder of the project.

Each computation node host deployed the binaries themselves. Cybernetica and RMIT had a virtual machine with 2 CPU cores at 2.4 Ghz, while RIA had a physical machine with 12 CPU cores at 2.0 GHz. Computation nodes provided by RMIT and RIA had 8 GB of memory as requested, Cybernetica's node had 32 GB of RAM to handle extensive profiling data collected from the computations.

All Sharemind components use mutually authenticated and encrypted TLS tunnels (with AES-GCM) for communication. Hence, each party generated a 2432-bit RSA key pair and distributed the public key among other parties. As public keys were distributed via insecure e-mail, a representative of each party

(except for the Ministry of Education and Research) digitally signed their public key with the Estonian ID-card to ensure its authenticity.

Once all three computation nodes were set up, RMIT tested the whole analysis process from importing the data to running the ETL queries, and creating figures on a small generated test dataset.

5.4 Importing the data

We developed a special data importing tool that loads the data in comma-separated value (CSV) format and the data model description in XML. The importer then checks that the data in the CSV file corresponds to the expected format. The importer then secret shares each value in the table to the correct Sharemind secure data type and uploads the shares to the computing parties. If there is a mismatch between formats, the tool warns its operator and stops the import.

The input parties generated the CSV files by exporting the data from their own databases. It was critical that secret sharing is performed by the data owners so that private information never leaves the organisation.

6 Running the study

This section describes the timeline and the issues we faced during the running of the study beginning with the licence agreement and ending with data deletion. We also discuss the main setbacks and lessons that are useful for conducting future studies. In addition, we talk about the differences between the study that we performed on data in secret-shared format, and the study that CentAR performed on anonymised data.

6.1 Timeline

The timeline of the study was the following.

- Nov. 10, 2014: Software licensing agreement to RMIT signed, source code delivered to RMIT.
- Dec. 10, 2014: Cybernetica, RMIT and RIA sign the secure multi-party computation agreement. The three Sharemind servers of the production environment are successfully interconnected for the first time.
- Dec. 17, 2014: RMIT imports a small test dataset (test set A) into the production environment.
- Dec. 30, 2014: RMIT successfully runs all ETL steps on the test set in the production environment.
- March 2, 2015: HTM imports the education dataset into the production environment.
- March 5, 2015: RMIT imports the tax data into the production environment.

ETL script	Test set B (Laboratory)	Real data (Production)
(1) Aggregation of education data	25 min	2 h
(2) Aggregation of tax data (monthly income)	18 h 10 min	221 h 55 min
(3) Aggregation of tax data (average yearly income)	1 h 55 min	15 h 14 min
(4) Joining the two datasets	32 min	4 h 15 min
(5) Compiling the analysis table (shifting)	39 h 3 min	141 h 11 min
Total ETL time	60 h 5 min	384 h 35 min

Table 1. Running times of the privacy-preserving ETL scripts on test set B in a laboratory environment and the final imported data in the production environment.

- March 30, 2015: Cybernetica moves their computation server into a data centre near to (but not co-located with) the others to decrease the network latency with other computation servers.
- June 17, 2015: All ETL steps are finished.
- July 1, 2015: End of the analysis step, data shares are deleted by the computing parties.

6.2 Performance

We generated two sets of test databases: a smaller set for correctness testing that contained 354 education records and 8,201 tax records (test set A); and a larger set that was comparable in size to the expected real dataset (test set B) with 831,424 education records and 16,205,641 tax records. We used the larger dataset to test performance on a Sharemind installation in a local area network. The final real-world data imported by the data owners contained 623,361 education records and 10,495,760 tax records.

However, when the result party started the ETL process in the production environment, we soon discovered that the running time did not scale linearly with respect to the network bandwidth and latency. Applying aggregation functions on many small datasets is bound by latency, while operations on large datasets (e.g., the whole tax database) are bound by bandwidth. For reference, Table 1 details the ETL running time on test set B on our local cluster and on the real data in the production environment.

Through analysis of the performance results, we saw that the current implementation of aggregation did not fully use the resources available to the Sharemind deployment. Most notably, during scripts 2 and 5, the network bandwidth use was significantly lower than it should have been. By further profiling the implementation in a laboratory setting, we found that the large number of small groups in aggregations caused the extensive running time. We believe that by

further parallelisation of the aggregation procedure, we could reduce the running time of scripts 2 and 5 in a production setting by up to an order of magnitude.

6.3 The cost of human errors in MPC

As shown in Table 1, the ETL process was divided into steps that run from a couple of minutes to a couple of days. Each such step loaded one or more data tables saved by previous steps, performed privacy-preserving operations on them and saved the result as a new table. Making a human error (e.g., using a wrong variable name or constant) usually means that the relevant step or its part has to be performed again. Although we had tested the ETL scripts on generated test sets, a couple of errors were discovered during the real analysis.

The first such error was related to how missing values are handled by Rmind. When aggregating tax data, a person's income was computed by adding together two different income origins of which one or both could be empty (missing). The expected outcome was that empty values would be replaced with zeros before the addition. However, Rmind handles missing values similarly to R: the output of an operation containing at least one missing value yields a missing value. Hence, most of the students were counted as not having an income at all.

The mistake was fixed by replacing missing values with zeros and all of the ETL steps starting from the fixed script were run again, losing about 6 days worth of computation time. However, checking the results later it was discovered that the analyst using Rmind had mistakenly still used the old version of the ETL script that did not replace the missing values. Hence, most of the ETL steps had to be run once more, this time losing more than 8 days of computation time.

Another error was related to the way in which the data importer tool handles classifiers. Generally, the CSV files to be imported contain two types of data: numerical values and discrete textual values (classifiers such as gender and university name). The latter have to be first encoded as a numerical values. The data importer tool automatically generated and showed a classifier for each data field that contained discrete textual values, (e.g., *0—male, 1—female*). Input parties revealed these classifiers to the analysts so they could use the correct values in the Rmind scripts.

Different data sets can yield different classifier values for the same data field (e.g., if the first record belongs to a female, the same classifier becomes *0—female, 1—male* instead). Therefore, we introduced a `recode` operation in Rmind. The `recode` operation allowed to securely swap the values in a classifier to avoid changing the numerical values in all of the analysis scripts independently.

In this study, the analysts used the `recode` operation for the study status classifier. Unfortunately, having many test versions of the same Rmind script, the analyst used a wrong mapping in the `recode` operation and hence the students' study status was wrong throughout the analysis for a while. Moreover, as one of the status values was used in further computations, it was not possible to correct the classifier afterwards. As the graphs drawn with the wrong classifier initially looked more or less plausible, the mistake was discovered very late during the

study. While the statisticians were able to correct some outputs of the study, not all scripts could be run again in time. Consequently, some of the plots using the studying status attribute were incorrect and not usable in the final report. As the computing parties followed the agreements and took SHAREMIND offline, the statisticians were unable to fix the missing plots.

6.4 Comparison to the anonymised study

In order to be sure that the results computed on encrypted data correspond to results that would be received when using traditional methods, two study processes were designed. The control study used k-anonymity measures [17] with groups of three, based on education data. The process designs revealed immediately in the planning and risk analysis phase that, in the anonymised study, the statistical analyst was a privacy risk to data. This is because the merged database was created and retained by the analyst during the whole study, giving them access to individual values.

In the privacy-preserving study using MPC, both the location of the linked data and their form changed. Three separate organisations, of which RMIT was the legal processor of tax data, were responsible for the secure multi-party computation. MPC gave the data owner real technical control over their data during analysis and implemented dynamic consent in practice. When the data owner wanted to stop the processing, it was as easy as stopping their SHAREMIND server. Other computing parties could not restart it without agreeing with RMIT.

The privacy-preserving study introduced additional steps to the study process, namely reviewing the SHAREMIND and RMIND source code, and determining beforehand what would be queried on the data. This improved the data owners' involvement in how their data would be used, especially in comparison with the anonymised study that was limited to reviewing the risk assessment and setting punitive measures into the study contract. The punitive measures are reactive, whereas technically enforced MPC guarantees are proactive and prevent leakage instead of measuring damages when it happens.

In addition to stronger security features and eliminating the biggest security threat to data, we also wanted to know the impact on the sample and final results. In the anonymised study, the statisticians measured 10% to 30% sample loss resulting from k-anonymity enforcement. Though the remaining sample sizes are still large in comparison with survey-based research, there was a systematic bias problem resulting from the removal of unique observations that did not have two other observations with similar educational parameters.

Initially, the data owners wanted to use k-anonymity measures on tax data as well. However, this resulted in sample losses from 84% to 97%, making the approach completely infeasible. The statisticians proved the resulting bias by demonstrating increasing sample loss with the growing number of observation years and severely skewed estimates of the number of studying, graduated and interrupted students as to the original distributions. Thus, k-anonymity was enforced only in the education dataset.

The impact of the 10% to 30% sample loss was evident when the MPC-based study was completed. The statisticians reported that the numbers and distributions of students in the MPC-based study was equal to the real-world ones. However, in the final reports, the sample bias introduced by k-anonymity had a negative influence on the accuracy of estimates of working during studies. The difference of the biased results to precise ones was from 4% to 13% where the results from the MPC study represented the unbiased actual result. In typical social studies, analysts do not have the exact results with which to compare the bias in the estimates. Even stratification that could be used to compensate for the bias is difficult to apply, as the biases do not solely depend on measurable input dimensions.

This suggests that a social study on existing databases that uses MPC to enforce privacy can give more accurate results than the same study run using k-anonymity measures.

6.5 Lessons learned

From a deployment perspective, the main concern was the instability of the system when deployed over the wide area network. At the time of conducting the study, it was necessary that the client application Rmind was connected to the Sharemind computation nodes throughout the runtime of a script as the commands were sent to computation nodes one by one.

The ETL steps running up to several days constituted a problem because the client connection was often interrupted. Most probably, this was due to the fact that the result party used a broadband connection meant for private individuals, whereas computing parties had enterprise-level service-level agreements (SLAs) with their network service providers. A client application disconnecting in the middle of a computation left the servers in a state where they did not accept any new incoming connections. For this reason, the Sharemind servers had to be restarted about 25 times during the whole study. As each server was hosted by an independent organisation, each such restart could take up to a whole business day. Consequently, we are redesigning the Sharemind system to allow the client to disconnect and reconnect during long queries.

To avoid human errors described in Section 6.3, data-dependent sanity checks, e.g., publishing some non-sensitive aggregates, have to be added to the analysis scripts. These add to the overall computational cost of the analysis but, in our experience, may decrease the number of reruns. Another option would be to make currently implicit behaviours explicit, i.e., the user would have to state how to handle missing values in arithmetic operations and not doing so would be considered an error. The user would be forced to think what the desired behaviour is beforehand instead of debugging the program when unexpected output occurs. The disadvantage is that this would make Rmind less similar to R which was one of the design goals of Rmind.

Moreover, classifiers for discrete textual values generated by the data import tool should be made automatically available for use in client applications. Classifier information can be saved together with the secret-shared values on the

server so Rmind could automatically replace the textual value with the correct classifier value in its scripts on runtime and show the classifier of an attribute to its users as well.

To improve plotting, data aggregation and visualisation of each plotting procedure should be separated from the analysis. The aggregated data can be saved in a form that Rmind could read and visualise. This would make experimenting with different sets of graphical parameters faster as the aggregation results could be reused instead of recomputing them every time. In addition, the appearance of the plots could be easily changed after the sensitive input data have been deleted.

We also noticed that the growing performance of standard computing hardware does not require statisticians to write optimised analysis scripts anymore. In the MPC setting, efficiency is still an important concern. At the time of writing this paper, the Rmind client application is a simple interpreter that directly interprets the abstract syntax tree of the program and performs no optimisations. Possible improvements would be to make the interpreter restructure the query for efficient MPC execution, caching intermediate results, or improving documentation to help users optimise their code.

6.6 Study results

The statisticians at CentAR compiled their findings into a 26-page study report [3], with over 20 plots generated using our study tools. The main findings were as follows:

1. The graduation rate of ICT students in the observed period is low, around 20%. Female students graduate with higher probability than male students and this applies to both timely and non-timely graduation.
2. During bachelor's studies, ICT students are not working more than non-ICT students. However, ICT students in master's studies work more than their non-ICT colleagues.
3. Most bachelor students in ICT do not work in the ICT sector. Among final year students in bachelor's and applied higher education studies, 30% of ICT students work in ICT companies. On the master's level, this increases to 50%.

For an excerpt of the study results, and example plots see Appendix B.

7 Conclusion

We have successfully solved a real-world privacy problem. Previously, it was impossible to conduct a secure statistical study on confidential tax information without losing in precision and privacy. With our implementation, we resolved legal concerns and were able to process the full population instead of losing parts of the sample to the anonymisation procedure. A legal precedent has currently

been established only in Estonia, but a second legal analysis based on European law suggests that it could be extendable.

This technology can significantly impact governance. Any government planning a new policy can first analyse the economic trends caused by current policies. Using this baseline and new planned policy, one could then simulate the effect of the new policy, if it had come into effect, for example, five years ago. In addition, such analytics can help detect fraud in welfare and other governmental grant programs.

While our study was conducted on government data, the technology is not limited to the public sector use or social studies. In fact, the capability that we have created allows any organisation to make better decisions by basing them on the best possible data. For example, analytics companies providing consulting services to different industries can collect more data to improve their predictions. Similarly, financial institutions can pull together more data for better fraud detection or analytics.

The vigilance we encountered during the validation of a new privacy technology is natural. Previous attempts at bringing cryptographic secure computing into practice (e.g., when Feigenbaum et al. attempted to apply MPC in the Computing Research Association Taulbee Survey in 2004) have also met similar resistance. Even though we were proposing a technology that would significantly improve the protection of personal data, we were challenging the established state of the art and had to convince all involved parties that our solution actually reduced risks. One could compare this situation with how new drugs are validated by proving their greater efficacy through an expensive trial. We approached the technical, organisational and regulatory hurdles of this study in a similar way and achieved success. However, further trials are needed to ensure the breakthrough of privacy-enhancing technologies such as secure multi-party computation.

Acknowledgments

This work was supported by the European Regional Development Fund from the sub-measure "Supporting the development of the R&D of information and communication technology" through the Archimedes Foundation (project Private Statistics PRIST). It has also received funding from the Estonian Research Council under Institutional Research Grant IUT27-1 and from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 284731.

The authors acknowledge the outstanding patience and effort from the parties involved in making this study a reality: the Estonian Data Protection Inspectorate, the Information System Authority, the Ministry of Education and Research, the Information Technology Centre of the Ministry of Finance, the State Information System Agency and the Tax and Customs Board.

Special thanks go to the team in the Estonian Center for Applied Research for designing the social study, using SHAREMIND-based statistical analysis tools and guiding us towards making MPC friendlier for real-world users.

References

1. Isikuandmete kaitse seadus (Personal Data Protection Act of Estonia). Passed 15.02.2007 - RT I 2007, 24, 127; RT I, 12.07.2014, 51. English translation available at https://www.riigiteataja.ee/en/eli/509072014018/consolide.
2. Maksukorralduse seadus (Taxation Act of Estonia). Passed 20.02.2002 - RT I 2002, 26, 150; RT I, 11.07.2014, 11. Taxation Act, English translation available at https://www.riigiteataja.ee/en/eli/501092014002/consolide.
3. Sten Anspal, Dan Bogdanov, Liina Kamm, Baldur Kubo, Ville Sokk, and Riivo Talviste. The working habits of ICT students. Overview of study results (in Estonian). http://www.centar.ee/case-studies/ikt-erialade-tudengite-tootamine/, 2015.
4. Dan Bogdanov. *Sharemind: programmable secure computations with practical applications*. PhD thesis, University of Tartu, 2013.
5. Dan Bogdanov, Liina Kamm, Sven Laur, and Ville Sokk. Rmind: a tool for cryptographically secure statistical analysis. Cryptology ePrint Archive, Report 2014/512, 2014. http://eprint.iacr.org/.
6. Dan Bogdanov, Peeter Laud, and Jaak Randmets. Domain-polymorphic programming of privacy-preserving applications. In *Proceedings of the Ninth Workshop on Programming Languages and Analysis for Security*, PLAS'14, pages 53–65. ACM, 2014.
7. Dan Bogdanov, Riivo Talviste, and Jan Willemson. Deploying secure multi-party computation for financial data analysis (short paper). In *Proceedings of FC 2012*, pages 57–64, 2012.
8. Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas P. Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael I. Schwartzbach, and Tomas Toft. Secure Multiparty Computation Goes Live. In *Proceedings of FC 2009*, pages 325–343, 2009.
9. Koji Chida, Gembu Morohashi, Hitoshi Fuji, Fumihiko Magata, Akiko Fujimura, Koki Hamada, Dai Ikarashi, and Ryuichi Yamamoto. Implementation and evaluation of an efficient secure computation system using 'R' for healthcare statistics. *Journal of the American Medical Informatics Association*, 04, 2014.
10. Ivan Damgård, Kasper Damgård, Kurt Nielsen, Peter Sebastian Nordholt, and Tomas Toft. Confidential Benchmarking based on Multiparty Computation. Cryptology ePrint Archive, Report 2015/1006, 2015.
11. Ernesto Damiani, Valerio Bellandi, Stelvio Cimato, Gabriele Gianini, Gerald Spindler, Matthis Grenzer, Christopher Schwanitz, David Koppe, Niklas Heitmüller, Sonja Hagenhoff, and Tim Kostka. D31.1 Risk assessment and current legal status on data protection. http://practice-project.eu/downloads/publications/D31.1-Risk-assessment-legal-status-PU-M12.pdf, 2014.
12. Data Protection Inspectorate of Estonia. Notification for the application to use delicate personal data in a study. January 27th, 2014. Document 2.2.-7/13/557r registered in the document management system of the DPI (in Estonian)., 2014. http://adr.rik.ee/aki/dokument/3679385/.

13. Khaled El Emam, Saeed Samet, Jun Hu, Liam Peyton, Craig Earle, Gayatri C. Jayaraman, Tom Wong, Murat Kantarcioglu, Fida Dankar, and Aleksander Essex. A Protocol for the Secure Linking of Registries for HPV Surveillance. *PLoS ONE*, 7(7):e39915, 07 2012.
14. Joan Feigenbaum, Benny Pinkas, Raphael Ryger, and Felipe Saint-Jean. Secure computation of surveys. In *EU Workshop on Secure Multiparty Protocols*, 2004.
15. Liina Kamm. *Privacy-preserving statistical analysis using secure multi-party computation*. PhD thesis, University of Tartu, 2015.
16. Sven Laur, Riivo Talviste, and Jan Willemson. From Oblivious AES to Efficient and Secure Database Join in the Multiparty Setting. In *Proceedings of ACNS'13*, volume 7954 of *LNCS*, pages 84–101. Springer, 2013.
17. Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

A Queries in the study plan

In Estonia, there are four major schools that teach ICT subjects: University of Tartu, Tallinn University of Technology, Tallinn University and Estonian Information Technology College. In the following, the statistics done across all schools refer to these four institutions.

First, descriptive statistics for the dataset were computed using the following queries on the education database:

- Number of students starting ICT studies across all schools in different levels of study during the years 2006-2012 (1 query);
- Number of students graduating from ICT studies across all schools in different levels of study during the years 2006-2012 (1 query);
- Number of students quitting their ICT studies across all schools in different levels of study during the years 2006-2012 (1 query);
- Percentage of students graduating their bachelor's studies in nominal time based on year of admission during the years 2006-2009 in *ICT and non-ICT fields*. The same for professional higher education studies and master's studies; the same queries for graduation of *ICT in nominal time* across different universities (6 queries).

To study general employment during studies, the following queries were performed on the analysis database:

- Percentage of working students based on year of admission during the three years of bachelor's studies in *ICT and non-ICT fields*; the same for professional higher education studies and master's studies; the same queries based on working during studies in *ICT across the schools* (6 queries);
- Number of months worked during a calendar year during nominal study time during the three years of bachelor's studies in ICT; the same for professional higher education studies and master's studies (3 queries).

To study employment in ICT companies and ITL member companies during studies, the following queries were performed on the analysis database:

- Percentage of students working in *ICT companies* based on year of admission during the years of bachelor's studies in ICT and non-ICT fields; the same query for professional education studies and master's studies; the same queries for *ITL member companies* (6 queries);
- Percentage of students working in *ICT companies* based on year of admission during the three years of bachelor's studies in ICT across three universities; the same for professional higher education studies and master's studies; the same queries for *ITL member companies* (6 queries).

To study employment after graduation or quitting studies, the following queries were performed on the analysis database:

– Rate of employment after graduation or quitting studies in *ICT and non-ICT fields* one to three years after graduating/quitting bachelor's studies; the same for professional higher education studies and master's studies; the same queries based on *ICT studies across the schools* (6 queries).

To study income during studies, the following queries were performed on the analysis database:

– Median monthly income of *ICT and non-ICT students* during bachelor's studies based on year of admission and the fact of graduation/quitting; the same for professional higher education studies and master's studies; the same queries for *ICT students across schools* (6 queries);
– Median monthly income of ICT students across schools during the *nominal time* bachelor's studies based on year of study and the fact of graduation/quitting; the same for professional higher education studies and master's studies (3 queries).

To study income after graduation or quitting studies, the following queries were performed on the analysis database:

– Median monthly income of *ICT and non-ICT students* after graduating or quitting bachelor's studies one to three years after graduating/quitting; the same for professional higher education studies; the same queries for *ICT students across schools* (4 queries).

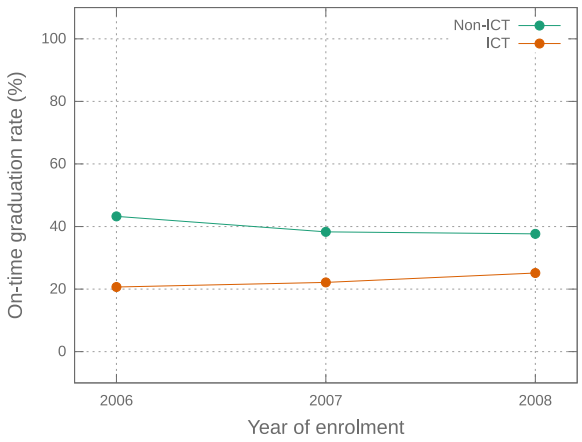

Fig. 3. Graduating in nominal time, ICT vs non-ICT students, bachelor's studies

B Excerpt from the study results

Graduating in nominal time is rare for all students, but things are especially bad for ICT students. Figure 3 shows nearly twice the difference in the graduation rates of ICT and non-ICT bachelor's students enrolled between 2006 and 2008. The graduation rate of ICT students is slowly increasing and future work will have to say whether the trend continues.

We see a clear increase in the employment rate in ICT companies, as ICT students progress in their studies. Figure 4 shows how, by year 3, 20%–25% of ICT students work in ICT companies. This does raise the question—where do ICT students work, if not in ICT companies? We also see that ICT companies respect the ICT diploma enough to not hire non-ICT students at any significant scale. As can be seen on Figure 5, in master's studies, the rate of employment is significantly higher. This is the largest risk to the sustainability of the academia, as it could reduce the number of good candidates to the doctorate programmes.

Furthermore, somewhat surprisingly, the working habits of ICT students and non-ICT students are pretty much the same (see Figure 6). Between 2006 and 2013, an average of 60% of all students worked. The sudden decline in working rates in 2008 shows the effect of the global financial crisis—a 10%–15% decline in the rate of employment among students. There is no quantitative explanation for the more sudden decline among ICT students. An idealist might attribute it to many students suddenly being enlightened and understanding that one should focus on studies and graduate to ensure a stable line of work.

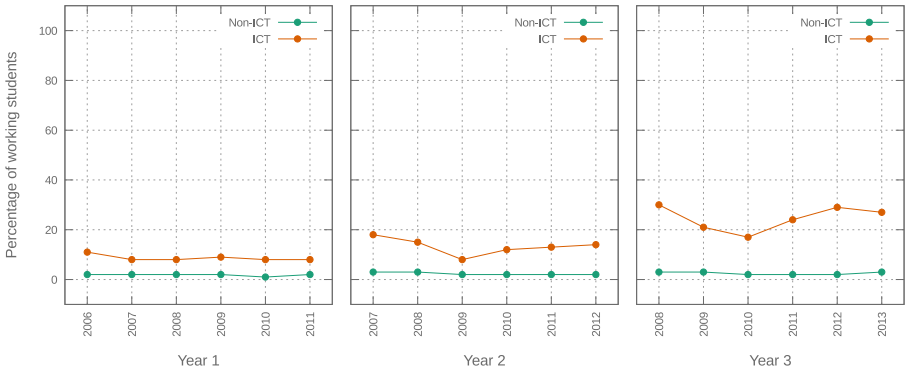

Fig. 4. Working in ICT companies, ICT vs non-ICT students, bachelor's studies

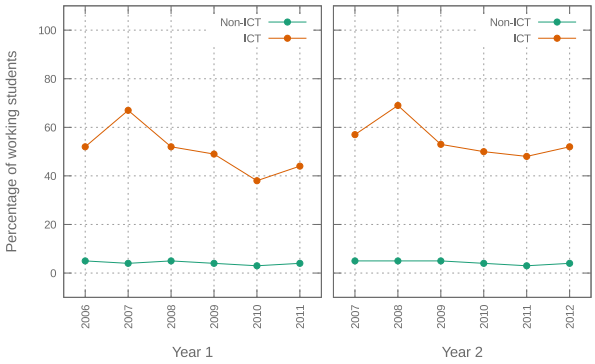

Fig. 5. Working in ICT companies, ICT vs non-ICT students, master's studies

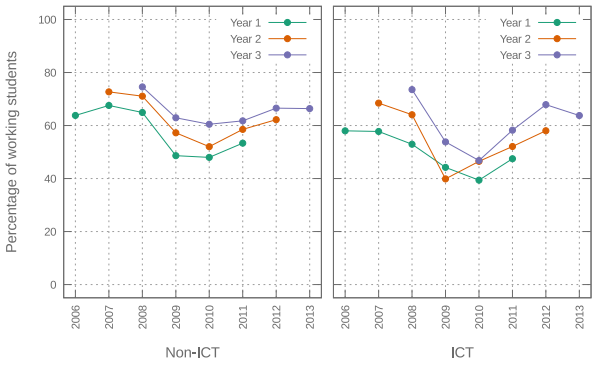

Fig. 6. Employment rates, ICT vs non-ICT students, bachelor's studies