

# Privacy-preserving statistical data analysis on federated databases

Dan Bogdanov<sup>1</sup>, Liina Kamm<sup>1,2</sup>, Sven Laur<sup>2</sup>, Pille Pruulmann-Vengerfeldt<sup>3</sup>,  
Riivo Talviste<sup>1,2</sup>, and Jan Willemson<sup>1,4</sup>

<sup>1</sup> Cybernetica, Mäealuse 2/1, 12618 Tallinn, Estonia  
{dan,liina,riivo,janwil}@cyber.ee

<sup>2</sup> University of Tartu, Institute of Computer Science, Liivi 2, 50409 Tartu, Estonia  
swen@ut.ee

<sup>3</sup> University of Tartu, Institute of Journalism, Communication and Information  
Studies, Lossi 36, 51003 Tartu, Estonia  
pille.vengerfeldt@ut.ee

<sup>4</sup> ELIKO Competence Centre in Electronics-, Info- and Communication  
Technologies, Mäealuse 2/1, Tallinn, Estonia

**Abstract.** The quality of empirical statistical studies is tightly related to the quality and amount of source data available. However, it is often hard to collect data from several sources due to privacy requirements or a lack of trust. In this paper, we propose a novel way to combine secure multi-party computation technology with federated database systems to preserve privacy in statistical studies that combine and analyse data from multiple databases. We describe an implementation on two real-world platforms—the SHAREMIND secure multi-party computation and the X-Road database federation platform. Our solution enables the privacy-preserving linking and analysis of databases belonging to different institutions. Indeed, a preliminary analysis from the Estonian Data Protection Inspectorate suggests that the correct implementation of our solution ensures that no personally identifiable information is processed in such studies. Therefore, our proposed solution can potentially reduce the costs of conducting statistical studies on shared data.

**Keywords:** secure multi-party computation, federated database infrastructures, linking sensitive data, privacy-preserving statistical analysis

## 1 Introduction

During the last decade we have witnessed a rapid growth of e-government technology adoption, e.g. online services for citizens based on federated state databases. The Estonian X-Road system is one of the more successful platforms. It is operational since 2001 and mediates the vast majority of the governmental data exchange requests of today. State agencies maintain their individual databases and queries are made over the X-Road as needed. The distribution of data between several owners also prevents the creation of “superdatabases”, that contain an extensive amount of information on a single person or company. However, there

are analytical tasks that require the linking of one or more databases. For example, the government needs to combine tax records with educational records to analyse the efficiency of educational investments.

However, every such database combination is a privacy risk for citizens. Even if the databases are pseudonymised or anonymised, records become larger in the joined databases. This, in turn, gives the attackers the ability to use additional information available from public sources like social networks to restore the identities. Potential attackers vary from malicious hackers to misbehaving officials. The latter class of attackers is the most complicated one to handle, as the official may have a completely legal access to the raw datasets, and just use them to gain more information than is necessary for completion of legitimate tasks.

Our goal is to perform all the required computations without revealing any microdata to the computing entities and mitigate the privacy risk. This paper proposes a novel extension of federated database systems that adds the feature of privacy-preserving data analysis using secure multi-party computation.

Secure multi-party computation (SMC) has been researched and developed for several decades. For years, SMC was rightfully considered too inefficient for practical use. However, in recent years, several fast implementations have been developed [9,17]. Still, SMC has not yet become popular in practice, as people have managed without such a technology for a long time and have replaced it with social solutions like non-disclosure agreements and hope that their partners protect the data being shared.

In this paper, we describe a joint architecture combining federated database environments with SMC. We describe a concrete solution based on the Estonian X-Road federated database platform and the SHAREMIND [7] secure multi-party computation system. However, our idea is general enough to be applied to other database environments and secure computation technologies.

The paper is organised as follows. First, Section 2 illustrates the benefits of combining data in a federated database setting, describes the X-Road platform and discusses candidate microdata protection mechanisms. Section 3 explains secure multi-party computation, gives a privacy definition for SMC applications and explains how to satisfy it in data analysis algorithms. Section 4 discusses how to combine a federated database environment like the X-Road with SMC to provide a secure data analysis environment. In Section 5, we validate the need of the proposed solution through a set of interviews with potential end users.

We then focus on the practical implementation of the solution. Section 6 describes our first pilot project where we apply the proposed technology. Section 7 describes our implementation of secure multi-party statistical analysis algorithms and Section 8 analyses their performance with a set of benchmarks.

**Related work.** There have been efforts to implement statistical analysis using SMC. Cryptographic primitives for evaluating statistical functions like mean, variance, frequency analysis and regression were proposed in [14,18,19]. Early implementations of filtered sums and scalar products are described in [47,52]. Solutions based on secret sharing include a protocol for mean value [34,33]. A

protocol for calculating weighted sums over homomorphically encrypted sensitive data is given in [27].

In 2004, Feigenbaum *et al.* proposed to use SMC for analysing faculty incomes in the annual Taulbee Survey [21]. The protocols designed for this study can be found in [1]. In 2011, Bogdanov *et al.* deployed SMC for financial data analysis for the Estonian Association of Information Technology and Telecommunications [10]. Kamm *et al.* have shown how to conduct secure genome-wide association studies using secure multi-party computation [31].

**Our contribution.** We provide a novel privacy definition for data analysis using SMC and give guidance on how to achieve it. We propose a privacy-preserving data sharing solution in federated database environments based on SMC. We validate the proposed solution by analysing interviews conducted with potential end-users of the technology. We analyse the responses of our interviewees and identify their expectations towards an SMC-based solution. We illustrate the use of the solution by describing a first pilot project following its design.

We then describe the most complete secure multi-party statistics implementation made to date, supporting the calculation of mean, variance, standard deviation, frequency tables and quantiles. We show how to clean the data and apply custom filters. We give descriptions of privacy-preserving hypothesis testing using standard and paired t-tests, Wilcoxon tests and the  $\chi^2$ -test. We report on our experimental validation of the proposed algorithms on the SHAREMIND SMC platform and provide performance results proving their feasibility.

## 2 Data sharing in federated state databases

### 2.1 Benefits of openness in state databases

There exist several initiatives that promote access to and usage of open data to provide enhanced services and greater public transparency to the citizens; the Open Data Foundation<sup>5</sup>, ePSI platform<sup>6</sup> and Open Access to scholarly research results [46] just to name a few.

Lane, Heus and Mulcahy [35] discuss the role of publicly accessible sources in research, and identify four essential arguments to support data openness.

1. **Data utility:** data are useful only when they are being used.
2. **Replicability:** original data sources for a scientific result need to be published so that independent scholars could verify the work.
3. **Communication:** research results are always subject to interpretation, and results relying on closed sources are more prone to be misinterpreted.
4. **Efficiency:** data collection is a time-consuming and costly process, hence it makes sense to open it to bring down the social cost of research.

---

<sup>5</sup> <http://www.opendatafoundation.org>

<sup>6</sup> <http://www.epsiplatform.eu>

They also acknowledge the need for data protection and discuss four levels of it – technological, statistical, operational and legal protection. In 2008 when the paper [35] was published, the authors saw VPN and Citrix-like thin-client approach as the main technical protection mechanisms. Whereas these solution mitigate the risks caused by the need to have a copy of the dataset available for the research, they still assume direct access to the data. Our paper can be seen as extension of [35], enriching the pool of available data processing tools with secure multi-party computations tools.

Privacy-preserving technologies like secure multi-party computation support the cause of the open data movement by providing a platform for linking and aggregating confidential data sources into less sensitive, publishable streams. For example, if we securely link two personalised databases and aggregate the individual records into demographic groups, we can publish the resulting groups. This is especially useful when the source databases are confidential and should not be openly linked as is the case for many government databases.

## 2.2 The X-Road secure data exchange infrastructure

By early 2000s, the level of computerisation in the Estonian state databases had reached both the level of sufficient technical maturity and a certain critical volume so that the need for a unified secure access mechanism was clear. The development activities on the modernisation of national databases started in the beginning of 2001 [30,28]. The first version of the developed X-Road infrastructure was launched on December 17th 2001. The number of queries and replies mediated through the infrastructure per year exceeded 240 million in 2011 [29].

One of the more significant benefits of X-Road is the reduction of data duplication and the ability to combine data from many national databases. Furthermore, all communication is bilaterally authenticated and encrypted, meaning that parties on the X-Road can always prove the source of a request or a data item. Today, already the fifth generation of the X-Road is in operation and the sixth generation is under development, adding new features as high-performance qualified digital signatures [3] and increased availability under cyber threats [4]. Detailed technical descriptions of the whole system can be found in [2,51].

## 2.3 Privacy Protection Mechanisms for Data Sharing

**The Use of Pseudonymisation in X-Road and its Shortcomings.** Since the beginning of 2011, X-Road provides a solution for joining different databases without revealing the identities of the persons included [50]. In order to protect these identities, but still facilitate connecting the data items corresponding to the same individuals, *pseudonymisation* is used. On the technical level, the pseudonyms are computed by encrypting the ID codes of the individuals by a common symmetric AES key distributed via offline means.

Even though this solution offers some protection against curious data analysts, it is not sufficient to resist more determined and targeted attacks. As the data fields of the records are not encrypted, it is possible to breach the privacy

by comparing these fields to other datasets, e.g. publicly available data on the person’s gender, age, education, home town. Thus it is practically impossible to give any kind of security guarantee to a pseudonymisation-based solution.

**Other Candidates for Microdata Protection Mechanisms.** Several microdata protection mechanisms have been proposed such as  $k$ -anonymity [44,48] and  $\ell$ -diversity [42]. The main idea on  $k$ -anonymity is to ensure that each record in a dataset is indistinguishable from at least  $k - 1$  other records with respect to so-called quasi-identifiers, i.e. certain sets of attributes that can be used to identify at least one person uniquely. Machanavajjhala *et al.* [42] showed that  $k$ -anonymity approach has several practical weaknesses. For example, the  $k$ -anonymity approach does not take into account a possible background knowledge of attackers. The  $\ell$ -diversity approach [42] was designed to overcome the weaknesses of  $k$ -anonymity, but it has also been shown to have limitations [40].

### 3 Secure multi-party computation as a privacy-enhancing technology

Secret sharing [45] is a concept of hiding a secret value by splitting it into random parts and distributing these parts, called shares, to different parties so that each party has only one share. Depending on the secret sharing scheme used, all or a known threshold of shares are needed to reconstruct the original secret value.

Secure multi-party computation allows to compute functions of secret shared values so that each party learns only the corresponding function output value and no inputs of other parties. For example, given a secret value  $x$  that is split into  $n$  shares so that party  $P_i$  has  $x_i$ , all parties can collaboratively compute

$$y_1, \dots, y_n = f(x_1, \dots, x_n)$$

so that party  $P_i$  learns only the output value  $y_i$ .

There are a lot of SMC implementations with various features. Most of the them are academic research implementations that solve a problem in a very specific setting and are thus not easily usable together with other solutions. More mature implementations with programmable protocol suites include VIFF [16], SEPIA [13], FairplayMP [6] and SHAREMIND [7].

The SHAREMIND application server is a practical implementation of secure multi-party computation technology that allows privacy-preserving computation on secret shared data. It is an SMC implementation powering several real-world applications [10,15]. Its applications are developed using the SecreC programming language [8]. At the time of writing this paper, SHAREMIND is a secure multi-party computation platform with the largest selection of practical features. Therefore, we have chosen it as the platform of choice for this paper.

#### 3.1 Modelling SMC deployments

We define three fundamental roles in an SMC system—the input party  $\mathcal{I}$ , the computation party  $\mathcal{C}$  and the result party  $\mathcal{R}$ . Input parties collect and send data

to the SMC system. The SMC system itself is hosted by computation parties who carry out the SMC protocols on the inputs and send results to result parties in response to queries.

We use the following notation for modelling SMC applications. Let  $\mathcal{I}^k = (\mathcal{I}_1, \dots, \mathcal{I}_k)$  be the list of input parties,  $\mathcal{C}^m = (\mathcal{C}_1, \dots, \mathcal{C}_m)$  be the list of computing parties and  $\mathcal{R}^n = (\mathcal{R}_1, \dots, \mathcal{R}_n)$  be the list of result parties. Let  $\Pi$  be an SMC protocol for performing a specific task.

In the following,  $\mathcal{ICR}$  refers to a party that fills all three roles, similarly,  $\mathcal{IC}$  refers to a party with roles  $\mathcal{I}$  and  $\mathcal{C}$ . We use superscripts  $(k, m, n \geq 1)$  to denote that there are several parties with the same role combination in the system.

Real world parties can have more than one of these roles assigned to them. The set  $\{\mathcal{I}, \mathcal{C}, \mathcal{R}\}$  has 7 non-empty subsets and there are  $2^7$  possibilities to combine them. However, we want to look only at cases where all three roles are present. This leaves us with  $128 - 16 = 112$  possible combinations. Not all of these make sense in a real-world setting, but we claim that all deployments of SMC can be expressed using these 112 combinations.

### 3.2 Privacy expectations and definitions

We are mainly interested in the privacy-preserving properties of SMC. Therefore, we want the private inputs of the input parties remain hidden from the computing parties and the result parties.

While it is tempting to define privacy so that the computing parties and result parties learn nothing about the values of the input parties, such a definition would be rather impractical. First, we would need to hide the sizes of all inputs from the computing parties. There are several techniques for hiding the input size (e.g. [23,41]), but no generic solution exists and practical protocols often leak the upper bound of the size.

Second, we would need to hide all branching decisions based on the private inputs. While this can be done by always executing both branches and obviously choosing the right result, we can significantly save resources when we perform some branching decisions based on published values. However, such behaviour can partially or fully leak the inputs to the computing parties (and also to the result parties, should they measure the running time of  $\Pi$ ).

This directs us to a relaxed privacy definition, that allows the computing parties to learn the sizes of inputs and make limited branching decisions based on published values that do not directly leak private inputs. Finally, to support practical statistical analysis tasks, we also allow the result parties to learn certain aggregate values based on the inputs (e.g. percentiles). In a real-world setting, we prevent the abuse of such queries using query auditing techniques, that reject queries or query combinations that are extracting many private inputs.

**Definition 1 (Relaxed privacy of a multi-party computation procedure).** *A multi-party computation procedure  $\Pi$  evaluated by parties  $\mathcal{I}^k, \mathcal{C}^m, \mathcal{R}^n$  preserves the privacy of the input parties if the following conditions hold:*

**Source privacy** *During the evaluation of  $\Pi$ , computing parties cannot associate a particular computation result with the input of a certain input party.*

**Cryptographic privacy** *During the evaluation of  $\Pi$ , computing parties learn nothing about the intermediate values used to compute results, including the individual values in the inputs of input parties, unless any of these values are among the allowed output values of  $\Pi$ . As an additional exception, if a computing party is also an input party, it may learn the individual values in the input of only that one input party.*

**Restricted outputs** *During the evaluation of  $\Pi$ , the result parties learn nothing about the intermediate values used to compute results, including the individual values in the inputs of input parties, unless any of these values are among the allowed outputs of  $\Pi$ . Additionally, if a result party is also an input party, it may learn the input of only that one input party.*

**Output privacy** *The outputs of  $\Pi$  do not leak significant parts of the private inputs.*

### 3.3 Adapting private data analysis procedures for SMC

We now describe general guidelines for designing privacy-preserving algorithms that satisfy Definition 1. For source privacy, we require that computing parties cannot associate an intermediate value with an individual input party that contributed to this value. For instance, we may learn the smallest value among the private inputs, but we will not know which input party provided it. This can be achieved by starting the protocol by *obviously shuffling* the data [38].

Cryptographic privacy is achieved by using SMC protocols that collect and store inputs in a protected (e.g. encrypted, secret-shared) form. This prevents the computing parties from recovering private inputs on their own. Furthermore, the protection mechanism must be maintained for private values throughout the algorithm execution. The computing parties must not remove the protection mechanism to perform computations. Examples of suitable techniques include homomorphic secret sharing, homomorphic encryption and garbled circuits.

Restricting outputs is quite straightforward. First, the computing parties must publish to other parties only the result values that  $\Pi$  allows to publish. Everything else must remain protected. Trivially, it follows that the computing parties must run only the procedures to which the computational parties have agreed. Furthermore, the computing parties must reject all queries from the result parties that the computing parties have not agreed to among themselves. In practice, all computation nodes audit their copy of the code and if they do not agree with the operations it wants to perform on the data, they can reject the code. This effectively halts the computation process, as the code needs to be executed in parallel by all of the involved parties for the computation to work.

Output privacy is the most complex privacy goal, requiring a more creative approach. The most complex part in algorithm design is to control the leakage of input value bits through published outputs. There are many measures for this leakage, including input entropy estimation and differential privacy [20]. Regardless of the approach, the algorithm designer must analyse the potential

impact of publishing the results of certain computations. In some cases, such an analysis is straightforward. For example, publishing the results of aggregations like sum and mean is a negligible leak unless there are only a few values.

Typically, directly publishing a value from the private inputs should not be allowed. However, there are exceptions to this rule. For example, descriptive values, such as the minimal value in a private input, are used by statisticians to evaluate data quality. The main concern of data analysts in our interviews was that if we take away their access to individual data values, we need to give them a way to get an overview of the data in return. That is the reason why our privacy model allows the publishing of descriptive statistics.

## 4 Using SMC for secure data sharing and analysis

### 4.1 Solution architecture

We propose a way how secure multi-party computation can be used among federated databases to process sensitive data. We will model the solution after SHAREMIND and the Estonian X-Road as two practical implementations. First, we need to deploy secure multi-party computation nodes within the X-Road infrastructure. These nodes must be coupled with X-Road security servers or they can be standalone computation servers with high network bandwidth. The only restriction for the SHAREMIND nodes is that they must be operated by independent parties to avoid collusion. If the SHAREMIND computation nodes are operated by the same institutions as X-Road security servers belonging to separate government institutions, then secure multi-party computation protocols working in the honest-but-curious model are sufficient as government institutions have no incentive to collude. Furthermore, collusion requires some level of co-operation between the institutions and it is enough for a host to reject suggestions for collusion to defeat the institutional attacker. In the case of an outside attacker, the theft of a single SHAREMIND node does not leak the sensitive data. Should an attack occur, the privacy-preserving *resharing* procedure will ensure that stealing other parts of the secret-shared database will not compromise the sensitive information.

Upon receiving a query that spans databases of multiple service providers, the service consumer contacts the security servers of involved X-Road service providers as usual. These security servers request the needed database from their internal network, perform secret sharing on its contents and distribute the shares among previously chosen SHAREMIND computation nodes. This data sharing step may also be completed before the actual computation request.

The service consumer issuing the request then asks the involved SHAREMIND computation nodes to perform the actual computation on the secret-shared data. Once the secure multi-party computation is finished, the computation nodes send output shares back to the X-Road service consumer who can then reconstruct the final answer for the request. This workflow is shown on Figure 1.

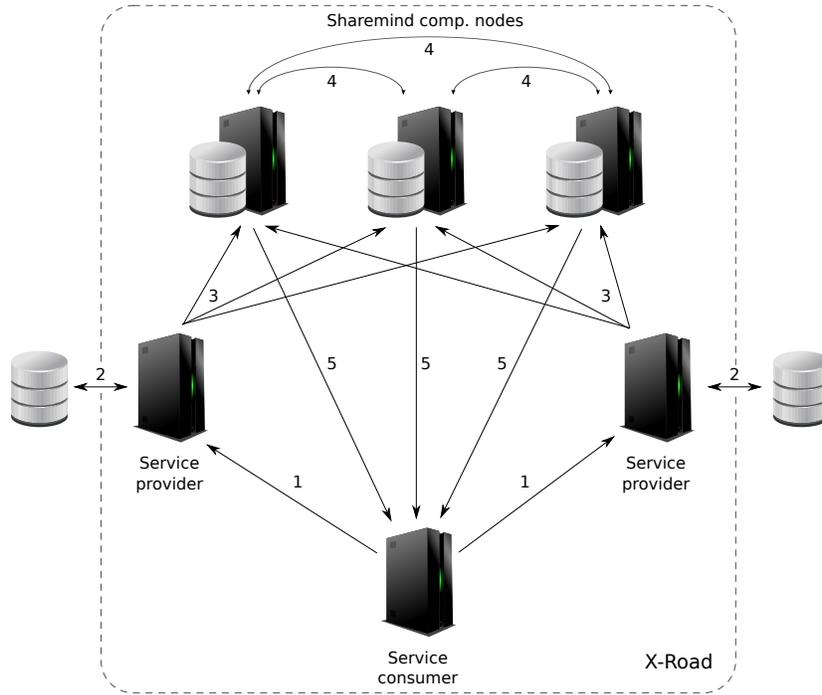


Fig. 1: A step-by-step workflow of using SHAREMIND together with X-Road.

## 4.2 Choosing computing nodes

For each computation request, the input data has to be shared and distributed between the same computation nodes. Hence, choosing these nodes has to be coordinated, and this turns out to be a non-trivial task. Consider a situation where X-Road service providers have a preference list of computation nodes to whom they are willing to entrust their secret-shared data. Such a list may be based on a reputation system as described in [5]. Then, upon receiving the computation request from a service consumer, the service providers involved in that request must communicate to agree on a common set of trusted computation nodes to use in the following secure computation. It may happen that such an intersection with a sufficient size does not exist.

Alternatively, we can state that all the deployed SHAREMIND computation nodes are equally trustworthy. Then the task becomes easier as the service consumer issuing the request can itself dictate which nodes to use and just informs the security servers of the involved service providers about its decision.

If a participating X-Road service provider requires more control over the computation process, it can host a SHAREMIND computation node. This gives the service provider an opportunity to halt the secure multi-party data processing if it suspects that anything is wrong. When choosing the computation nodes, this possible limitation has to be taken into account.

## 5 Validating the solution with potential end users

One of the goals of our work on SMC was to validate the real-world need for secure multi-party computation solutions. For this, we performed and analysed a number of interviews with stakeholders from a variety of fields to find out whether data holders see a need for this technology.

As previous research has indicated, a serious obstacle in user-driven innovation and involving users in the early stages of development work is the problem of explaining such a complex technology to the end-user who is rarely an expert [25,39]. Therefore, we first devised a way to explain the emerging technology to potential end users.

### 5.1 Using visualization to explain SMC to non-specialists

As our goal was to validate the need for privacy-preserving data sharing, we discussed SMC with people from different areas and asked them if they had had problems with sharing data in their field. We assumed that the interviewees did not have a background in computer science so approaching them with the usual SMC descriptions was out of the question.

We planned to visualise typical SMC applications to make the idea understandable. Fortunately, our role-based model translates easily into illustrative diagrams. See Table 1 for examples of deployment models inspired by published research on SMC applications.

We prepared for the interviews by designing 12 deployment models, some of which were based on existing SMC applications and some were hypothetical. We designed large colourful and easily readable figures to help us describe SMC to stakeholders during the interviews. On these figures we did not use the *ICR* syntax, but rather real-world roles that the interviewee could relate to. The description of each model included the security and trust guarantees that SMC provides for the parties. We could not include the figures here due to size constraints, but they can be found in [43].

### 5.2 Interview process and results

Our sample of 25 people was designed with the aim to get as much diversity as possible. The recruitment was based on the fields which, according to previous literature, were considered potentially interested in using SMC applications and also the snowballing technique for furthering the sample. The interviewees were always given a possibility to propose additional fields outside of their own where this kind of technology could be beneficial. Not all of our interviewees could be considered potential users, some could rather be described as stakeholders with knowledge of potential social barriers. For instance, among others, we interviewed a lawyer and an ethics specialist in order to understand the larger societal implications. As we aimed for the maximum diversity, our interviewees originated from six different countries, came from academia, from both public and private sector organisations, from small and medium sized enterprises to

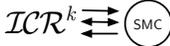
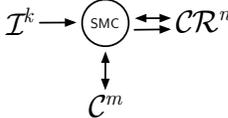
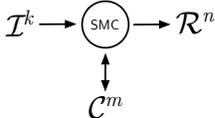
Basic deployment model	Example applications
	<p><b>The classic millionaires' problem [53]</b>  <i>Parties:</i> Two—Alice and Bob (both <math>\mathcal{ICR}</math>)  <i>Overview:</i> Millionaires Alice and Bob use SMC to determine who is richer.</p> <p><b>Joint genome studies [31]</b>  <i>Parties:</i> Any number of biobanks (all <math>\mathcal{ICR}</math>)  <i>Overview:</i> The biobanks use SMC to create a joint genome database and study a larger population.</p>
	<p><b>Studies on linked databases (this paper)</b>  <i>Parties:</i> Ministry of Education, Tax Board, Population Register (all <math>\mathcal{IC}</math>) and Statistics Bureau (<math>\mathcal{R}</math>).  <i>Overview:</i> Databases from several government agencies are linked to perform statistical analyses and tests.</p>
	<p><b>Outsourcing computation to the cloud [22]</b>  <i>Parties:</i> Cloud customer (<math>\mathcal{IR}</math>) and cloud service providers (all <math>\mathcal{C}</math>).  <i>Overview:</i> The customer deploys SMC on one or more cloud servers to process her/his data.</p>
	<p><b>Collaborative network anomaly detection [13]</b>  <i>Parties:</i> Network administrators (all <math>\mathcal{IR}</math>) a subset of whom is running computing servers (all <math>\mathcal{ICR}</math>).  <i>Overview:</i> A group of network administrators uses SMC to find anomalies in their traffic.</p>
	<p><b>The sugar beet auction [11]</b>  <i>Parties:</i> Sugar beet growers (all <math>\mathcal{I}</math>), Danisco and DKS (both <math>\mathcal{CR}</math>) and the SIMAP project (<math>\mathcal{C}</math>).  <i>Overview:</i> The association of sugar beet growers and their main customer use SMC to agree on a price for buying contracts.</p>
	<p><b>The Taulbee survey [21]</b>  <i>Parties:</i> Universities in CRA (all <math>\mathcal{I}</math>), universities with computing servers (all <math>\mathcal{IC}</math>) and the CRA (<math>\mathcal{R}</math>).  <i>Overview:</i> The CRA uses SMC to compute a report of faculty salaries among CRA members.</p> <p><b>Financial reporting in a consortium [10]</b>  <i>Parties:</i> Members of the ITL (all <math>\mathcal{I}</math>), Cybernetica, Microlink and Zone Media (all <math>\mathcal{IC}</math>) and the ITL board (<math>\mathcal{R}</math>).  <i>Overview:</i> The ITL consortium uses SMC to compute a financial health report of its members.</p>

Table 1: SMC deployment models and example applications

large multinational corporations, from local government to state level. The people we interviewed included representatives from the financial sector, agriculture, retail, security, mobile technologies, statistics companies and IT in general.

We sent the materials to the interviewees beforehand to let them prepare for the interview. We also used the figures during the interview process to trigger conversation and to assist in understanding the principles of the technology. During the interviews, we asked whether our interviewees recognised situations in their field of expertise where they need to share protected data with others.

Our interviewees outlined a number of different potential uses, some more realistic and closer to actual implementation, others brought examples of ideas where the concept of privacy-preserving computing might be useful or beneficial. While the interviewees struggled with identifying concrete applications, their conceptualisation of the technological framework identified several fields of inquiry. Of all the possible cases brought out in the deployment models, the cases concerning the use of databases from different data sources for performing statistical analysis were most discussed. It seems that the benefits of merging different databases for statistical analysis were easily comprehensible for the interviewees. On one hand, the interviewees had many concerns, such as SMC conflicting with the traditional ways of doing things and problems related to the existing legal and regulatory framework. At times, the interviewees could not distinguish between anonymisation and SMC, or understand the operational challenges of using this kind of solution in practice. On the other hand, the interviewees also saw many potential benefits of the possible applications of SMC.

Several interviewees brought out examples how SMC could be advantageous in their professional field, and pointed out that at the moment, the state collects information that is necessary for its purposes, but the public use and benefit of the same data suffers. With secret sharing, information given for general statistical purposes could support a wider range of goals. A researcher from the biomedicine field has an example of this kind of thinking:

*“For example, if I as a researcher get the data about the number of abortions but I also want to know how much all kind of associated complications cost, I need to get data from the national Health Insurance Fund. But I only get data from the Health Insurance Fund if I have the data from the abortion registry with names and national identification numbers and then I ask the medical cost records of those people. What I think is actually a really big security risk. If it would be possible to link them differently, so I would receive impersonalised data, that would be really good.” (Interview 11, Academic sector, Biomedicine)*

In addition to identifying data use possibilities directly relevant to their field of work, interviewees also pointed out how SMC could be used on a more general level. The idea of using different state databases for statistical analysis was seen as highly beneficial. The potential benefit was seen not only for the members of public getting data, but also with the potential of more efficient state systems or use of public funding. For instance, an official working in a state institution that coordinates the work of the national information system stated that making more data and information available for public use is a relevant problem.

*“After the presentation I thought that the state data should be made available for people this way: for researches, statisticians, universities. Publishing these data has always been a topic in the state, all the data have to be public, we should put them on the cloud or somewhere else. But do it in a secure way, I haven’t thought about it before, but it seemed to me that there were no good solutions.” (Interview 8, Public sector, IT security)*

People are concerned about the privacy and personal integrity of the individuals whose data could be shared in such way. The legal barriers protecting individuals have a limited understanding or awareness of SMC capabilities and use of such application is hence seen possible only in distant future. At the same time, several interviewees see that state information systems using such applications could set the standard and help to develop trust. Alternatively, an EU regulation can help to overcome some implementation barriers as then different state systems will have to implement these data use regulations.

Interestingly, interviewees whose work involves data processing remained somewhat critical, mostly because of the practical issues. Although an interviewee working in biomedicine saw the benefits of using different databases in scientific research, he also foresaw possible issues that could hinder their work. The main concern could be expressed as the necessity to “see” the data.

*“But in the context of genetics, the researcher who does the calculations, he has to see the data. He has to understand the data, because there the future work will be combined. You never take just means, but when you are already calculating genotypes and their frequencies, then you have to take into account some other factors all the time. Adjust them according to age, height, weight. And you need to see these data. Without understanding the data, you cannot analyse them.” (Interview 11, Academic sector, Biomedicine)*

This obviously raises the question as to what is actually meant by “seeing” and “understanding” the data. The visibility of the data seems to be crucial, but it does not necessarily mean that no alternative solutions or procedures are possible. The interviewees remarked that it would be possible to do scientific analysis without “seeing” the data but that it would make their work more complicated and therefore would be met with hesitation. Hence, it may be possible that the barrier here is the practiced and accepted way of doing things. Even now statistics offices often respond to data requests by disclosing sample databases that resemble the data so that researchers can script their queries.

However, the interviews also revealed that the visibility of data is necessary to guarantee their quality. This aspect was for instance stressed by an expert working in the Statistics Office. Similarly, the interviewee doing scientific research thought it possible that the quality of their work and data suffers if they do not have the full overview.

*“We cannot combine different statistical works if we don’t have the identifiers. To do statistics, to have good quality information, we need to have it /full overview of data/.” (Interview 13, Public sector, Statistics)*

This quote illustrates nicely the way new technologies are understood first and foremost in the context of existing practices and boundaries. Similarly, peo-

ple considering the importance of statistical analysis with SMC can imagine the activities they do in their current framework. Hence, statistical analysis comes down to finding means, comparing samples in valid ways, finding correlations and relationships within the data. And all this preferably with a user environment that is recognisable. While, for instance, the Statistics Office employees can write their own scripts for queries, for wider usability, future SMC systems will need to be similar to existing tools.

## **6 An example application: linking tax and education data**

### **6.1 Estimating economic trends in education**

Governments are often interested in knowing what the return of investment of various kinds of education is. The underlying question is, what kind of specialists are needed in order to ensure the future economic success of the country and what kind of specialist may need re-education. One possible way for estimating this is to look at the trends in the earnings of different individuals. In Estonia, this information is distributed among several government institutions—the Tax and Customs Board has information about incomes and the Education Information System has all the education records for recent years. In a common scenario, the state will contact its Statistics Bureau and ask it to conduct the analysis. The latter has to contact the Tax and Customs Board and the Education Information System and ask them for the relevant data. The compiled database has a very high risk factor due to containing both income and educational information.

The authors of this paper are participating in a project called “Privacy-preserving statistical studies on linked databases<sup>7</sup>” (PRIST) started in 2013. In this project, we are answering the above question without compiling a full database of people and their incomes. Instead, we are conducting the study using secure multi-party computation and, more specifically, SHAREMIND.

### **6.2 Deploying secure multi-party computation for the study**

As SMC technology is not yet integrated into X-Road, there are several extra steps that must be taken. First, we have to find partners who are willing to act as computing parties and who have the necessary technical knowledge. In PRIST, the role of a computation node is carried out by the IT department of the Ministry of Internal Affairs (the Tax and Customs Board is part of it), Estonian Information System’s Authority and Cybernetica AS. The latter also provides the secure multi-party computation technology. All of the partners have to audit the SMC platform before deploying it in their premises and have to engage in a pairwise public key exchange to establish secure channels during the computation.

---

<sup>7</sup> Funded 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.

The use of SMC technology requires that the secret sharing of input data is done at the data owner’s premises. Therefore, we will provide a special-purpose data import program to the Tax and Customs Board and the Ministry of Education and Science. This importer takes a database (e.g. a CSV file), applies secret sharing to each of its values and distributes the shares among the computing parties over a secure channel. Both institutions create this intermediate CSV database with a query to their internal databases.

A team of statisticians fulfils the role of the result party. Their job is to initiate the secure computation process, receive the statistical results and interpret them. They are also responsible for compiling the study plan. The plan is implemented on SHAREMIND using the SecreC statistical library (see Section 7). The resulting application is distributed among the computation nodes and each one of them audits the code separately. Each computation node tracks what kind of operations are requested on the data and which results are published. Only if they agree that the program does what is expected, they compile the code and deploy it into their SHAREMIND installation so it can be executed.

We also provided a description of the study process for analysis to the Estonian Data Protection Inspectorate. Their response states that if the study is conducted according to the description, then no permit for the processing of personally identifiable information needs to be requested, as no such processing takes place<sup>8</sup>. This is an important result, as it significantly simplifies such studies in the future without jeopardising the privacy of citizens.

### 6.3 Conducting the study

After the computation nodes have been set up, the analysis program code has been deployed and data have been imported, the actual secure multi-party computation process can be initiated by the statistical analysis team. In this project, the analysis starts by securely joining the database containing income information with the database containing education information. In SHAREMIND, the secure database join operation is similar to the one used in X-Road today (see Section 2.3) using AES block cipher to encrypt the key values and performing the actual join operation on ciphertexts [36]. However, in SHAREMIND the AES encryption works on secret shared plaintexts and with a secret shared key that no single party knows. This ensures that the sensitive inputs are not published to any person or computer during linking.

As the actual join operation also works on published (reconstructed) ciphertexts, this method leaks for each encrypted key the number of matching keys in the other database. However, the key values and their positions are not revealed. More formally, if we depict the key columns of the joinable databases as bipartite graph with an edge marking matching keys, the join operation leaks its edge structure with the precision of the graph isomorphism. This information leakage is usually acceptable. Nevertheless, a slightly less efficient version of the

---

<sup>8</sup> Official response in Estonian available at <http://adr.rik.ee/aki/dokument/2663016>.

oblivious database join operation that does not leak this information is described in [37].

After the join operation is done, the resulting table is ready for statistical analysis. In this study, we expect to perform a range of descriptive statistical analyses, statistical tests and regressions. The statistical analysis will be able to request previously agreed-to analyses and it will receive only the results of the analyses. The statistical tool receives one share of each result value from the SHAREMIND computation nodes and reconstructs the results. The statistical analysis library is prepared in a way that prevents query results that are aggregations of a single person's data.

#### **6.4 The benefits of integrating secure multi-party computation into X-Road**

The integration of SMC into X-Road will optimise the study process as follows. First, we avoid setting up the computation nodes separately for every such project. This does not include only installing the SMC software but also the key exchange process. The public keys of SHAREMIND computation nodes can be exchanged by using signed X-Road messages.

Second, there will be no need for a separate importer application. The X-Road security server can directly access the internal database, apply secret sharing to the relevant tables according to a given task description and distribute the shares among the computation nodes. It is important to notice here that this also eliminates the need to create a database view as a data file. Getting rid of this standalone data file also eliminates the risk of leaking this file. Based on this, we are planning to combine SHAREMIND with the next generation of the X-Road core developed in the SDSB project<sup>9</sup>.

## **7 Implementing statistical studies with secure multi-party computation**

### **7.1 Data import and filtering**

When collecting data from several input parties, a common data model has to be agreed upon and key values for linking data from different parties have to be identified. For efficiency, it is often useful to preprocess and clean data at the input parties before sending it to computing parties. This will not compromise data privacy as the data will be processed by the input party itself. We now look at how to filter and clean data once it has been sent to the computing parties.

In the following, let  $\llbracket x \rrbracket$  denote a private value  $x$ , let  $\llbracket \mathbf{a} \rrbracket$  denote a private value vector  $\mathbf{a}$ , and let binary operations between vectors be point-wise operations.

---

<sup>9</sup> Secure Distributed Service Bus—<http://www.eliko.ee/secure-service-bus>

**Encoding missing values.** Sometimes, single values are missing from the imported dataset. There are two options for dealing with this situation: we can use a special value in the data domain for missing values; or add an extra availability mask for each attribute to store this information. Let the availability mask  $\llbracket \text{available}(\mathbf{a}) \rrbracket$  of vector  $\llbracket \mathbf{a} \rrbracket$  contain 0 if the corresponding value in the attribute  $\llbracket \mathbf{a} \rrbracket$  is missing and 1 otherwise. The overall count of records in storage is public. If missing elements exist, that value does not reflect the number of available elements and it is not possible to know which elements are available by looking at the shares. The number of available elements can be computed as a sum of values in the availability mask.

**Evaluating filters and isolating filtered data.** To filter data based on a condition, we securely compare each element in the the private attribute vector  $\llbracket \mathbf{a} \rrbracket$  to the filter value and obtain a private vector of comparison results. This mask vector  $\llbracket \mathbf{m} \rrbracket$  contains 1 if the condition holds and 0 otherwise. If there are several conditions in a filter, the resulting mask vectors are multiplied to combine the filters. Such filters do not leak which records correspond to the conditions.

Most of our algorithms can use a provided filter automatically during calculations. However, in some cases, it is necessary to “cut” the vector—keep a subset vector containing only the filtered data. To cut the vector, we first obliviously shuffle the value and mask vectors, retaining the correspondence of the elements. Next, the mask vector is declassified and values, for which the mask vector shows a zero, are removed from the value vector. The obtained cut vector is then returned to the user.

This process leaks the number of values that correspond to the filters that the mask vector represents. This makes cutting trivially safe to use, when the number of records in the filter would be published anyway. Oblivious shuffling ensures that no other information about the private input vector and mask vector is leaked [38]. Therefore, algorithms using oblivious cut provide source privacy.

## 7.2 Linking multiple tables

After collecting input values and compiling filters for the outliers, we can link the input databases to form the final analysis database. There are various ways for linking databases in a privacy-preserving manner. As a minimum, we desire linking algorithms that do not publish private input values and only disclose the sizes of the input and output databases, as described in Section 6.3.

## 7.3 Data quality assurance and visibility

**Quantiles and outlier detection.** Datasets often contain errors or extreme values that should be excluded from the analysis. Although there are many elaborate outlier detection algorithms like [12], outliers are often detected using quantiles. As no one method for computing quantiles has been widely agreed upon in the statistics community, we use algorithm  $\mathbf{Q}_7$  from [26], because it is

the default choice in our reference statistical analysis package GNU R. Let  $p$  be the percentile we want to find and let  $\llbracket \mathbf{a} \rrbracket$  be a vector of values sorted in ascending order. Then the quantile is computed using the following function:

$$\mathbf{Q}_\gamma(p, \llbracket \mathbf{a} \rrbracket) = (1 - \gamma) \cdot \llbracket \mathbf{a} \rrbracket[j] + \gamma \cdot \llbracket \mathbf{a} \rrbracket[j + 1] \text{ ,}$$

where  $j = \lfloor (n - 1)p \rfloor + 1$ ,  $n$  is the size of vector  $\llbracket \mathbf{a} \rrbracket$ , and  $\gamma = np - \lfloor (n - 1)p \rfloor - p$ . Once we have the index of the quantile value, we can use oblivious versions of vector lookup or sorting to learn the quantile value from the input vector.

We do not need to publish the quantile to use it for outlier filtering. Let  $q_0$  and  $q_1$  be the 5% and 95% quantiles of an attribute  $\llbracket \mathbf{a} \rrbracket$ . It is common to mark all values smaller than  $q_0$  and larger than  $q_1$  as outliers. The corresponding mask vector is computed by comparing all elements of  $\llbracket \mathbf{a} \rrbracket$  to  $\mathbf{Q}_\gamma(0.05, \llbracket \mathbf{a} \rrbracket)$  and  $\mathbf{Q}_\gamma(0.95, \llbracket \mathbf{a} \rrbracket)$ , and then multiplying the resulting index vectors. This way, data can be filtered to exclude the outlier data from further analysis. It is possible to combine the mask vector with the availability mask  $\llbracket \text{available}(a) \rrbracket$  and cache it as an updated availability mask to reduce the filtering load. Later, this mask can be used with the data attributes as they are passed to the statistical functions.

**Descriptive statistics.** As discussed in Section 5.2, one of the data analysts' main concerns was that they will lose the ability to see individual values. We claim that data quality can be ensured without compromising the privacy of individual data owners by providing access to aggregate values and enabling outlier filtering. While the aggregate value of an attribute leaks information about the inputs, the leakage is small and determined by the aggregate function.

A common aggregate is the five-number summary—a combination of the minimum, lower quartile, median, upper quartile and maximum of an attribute. We can compute the five-number summary using the quantile formula and use the published result to draw box-plots that give a visual overview of the data and effectively draw attention to outliers.

It is also important to analyse the distribution of a data attribute. For categorical attributes, this can be done by computing the frequency of the occurrences of different values. For numerical attributes, we must split the range into bins specified by breaks and compute the corresponding frequencies. The resulting frequency table can be visualised as a histogram. This publishes the number of bins and the number of values in each bin.

#### 7.4 Statistical testing

**The principles of statistical testing.** Many statistical analysis tasks conclude with the comparison of different populations. For instance, we might want to know whether the average income of graduates of a particular university is significantly higher than that of other universities. In such cases, we first extract two groups—the case and control populations. In our example, the case population corresponds to graduates of the particular university in question and the control group is formed of persons from other universities. Note that a simple

comparison of corresponding means is sufficient as the variability of income in the subpopulations might be much higher than the difference between means.

Statistical tests are specific algorithms, which formally quantify the significance of the difference between means. These test algorithms return the test statistic value that has to be combined with the sizes of the compared populations to determine the significance of the difference. While we could also implement a privacy-preserving lookup to determine this significant and prevent the publication of the statistic value, statisticians are used to including the statistic values and group sizes in their reports.

**The construction of case and control populations.** We first need to privately form case and control groups before starting the tests. One option is to select the subjects into one group and assume all the rest are in group two, e.g. students who go to city schools and everyone else. Alternatively, we can choose subjects into both groups, e.g. men who are older than 35 and went to a city school and men who are older than 35 who did not go to a city school. These selection categories yield either one or two mask vectors. In the former case, we compute the second mask vector by flipping all the bits in the existing mask vector. Hence, we can always consider the version where case and control groups are determined by two mask vectors.

In the following, let  $\llbracket \mathbf{a} \rrbracket$  be the value vector we are testing and let  $\llbracket \mathbf{m}_1 \rrbracket$  and  $\llbracket \mathbf{m}_2 \rrbracket$  be mask vectors for case and control groups, respectively. Then  $\llbracket n_i \rrbracket = \text{sum}(\llbracket \mathbf{m}_i \rrbracket)$  is the count of subjects in the corresponding population.

The tests need to compute the mean, standard deviation or variance of a population. We do this by evaluating the standard formulae using SMC. For improved precision, these metrics should be computed using real numbers.

**Student's t-tests.** The two-sample Student's t-test is the simplest statistical tests that allows us to determine whether the difference of group means is significant or not compared to variability in groups. There are two common flavours of this test [32] depending on whether the variability of the populations is equal.

In some cases, there is a direct one-to-one dependence between case and control group elements. For example, the data consists of measurements from the same subject (e.g. income before and after graduation), or from two different subjects that have been heuristically paired together (e.g. a parent and a child). In that case, a paired t-test [32] is more appropriate to detect whether a significant change has taken place.

The algorithm for computing both t-tests is a straightforward evaluation of the respective formulae using SMC with privacy-preserving real number operations. Both algorithms publish the statistic value and the population sizes.

**Wilcoxon rank sum test and signed rank test.** T-tests are formally applicable only if the distribution of attribute values in case and control groups follows the normal distribution. If this assumption does not hold, it is appropriate to use non-parametric Wilcoxon tests. The Wilcoxon rank sum test [24]

works on the assumption that the distribution of data in one group significantly differs from that in the other.

A privacy-preserving version of the rank sum test follows the standard algorithm, but we need to use several tricks to achieve output privacy. First, we need a more complex version of the cutting procedure to filter the database, the cases and controls using the same filter. Second, to rank the values, we sort the filtered values together with their associated masks by the value column.

Similarly to Student’s paired t-test, the Wilcoxon signed-rank test [49] is a paired difference test. Often, Pratt’s correction [24] is used for when the values are equal and their difference is 0. In a privacy-preserving version of this algorithm, we again need to cut several columns at once. We also need to obviously separate absolute values and signs from the signed inputs values and later sort these two vectors by the sign vector.

The computation of both tests is simplified by the fact that most operations are done on signed integers and secure real number operations are not required before computing the final z-score statistic. Both algorithms only publish the statistic value and the population sizes.

**The  $\chi^2$ -tests for consistency.** If the attribute values are discrete such as income categories then it is impossible to apply t-tests or their non-parametric counterparts and we have to analyse frequencies of certain values in the dataset. The corresponding statistical test is known as  $\chi^2$ -test.

The privacy-preserving version of the  $\chi^2$ -test is implemented simply by evaluating the algorithm using SMC operations. The algorithm can be optimised, if the number of classes is small, e.g. two. The algorithm publishes only the statistic value and the population sizes.

## 8 Practical implementation and benchmarks

We have implemented the described privacy-preserving statistics algorithms on the SHAREMIND SMC system. For an overview of the implementation, see Appendix A. Figure 2 shows our benchmarking scenario. Artificial data was used in the benchmarks, as the PRIST study is still in progress. We conducted the experiments on a SHAREMIND installation running on three computers with 3 GHz 6-core Intel CPUs with 8 GB RAM per core. While monitoring the experiments, we did not see memory usage above 500 MB per machine. The computers were connected using gigabit ethernet network interfaces.

Table 2 contains the operations, input sizes and running times for our experimental scenario. The output of the operations was checked against reference results from the R statistical toolkit and was found to be correct. We see that most operations in our experimental study take under a minute to complete. The most notable exceptions is the group median computation, as median computation has to be applied to the payments of 2000 subjects. This time can be reduced by vectorising the median invocations or conduct this aggregation before the data are converted into secret-shared form.

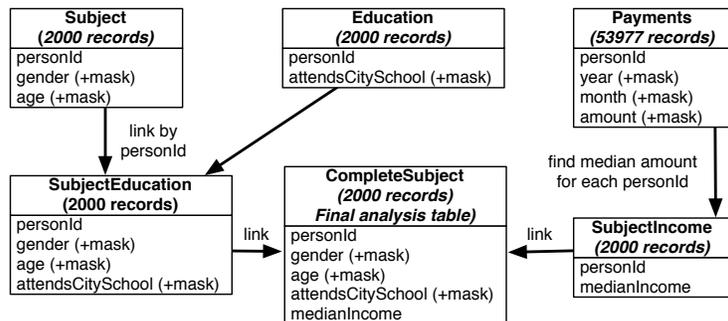


Fig. 2: The data model and table transformations in our experiment

To check scalability, we performed some tests on ten times larger data vectors. We found that increasing input data size 10 times increases running time about 5 times. Only histogram computation is actually slower, because it uses a more detailed frequency table for larger databases, actually increasing the work done.

The improved efficiency per input data element is explained by the use of vectorised operations of the SHAREMIND framework. The operations in the SHAREMIND framework are more efficient when many are performed in parallel using the SIMD (single instruction, multiple data) model.

## 9 Conclusions

In this paper, we study the problem of performing privacy-preserving statistical studies on data collected from sources connected to a single federated database infrastructure. Our proposed solution is to use secure multi-party computation as a privacy-enhancing technology. The paper provides a description of the solution with an explanation of the privacy-guarantees and implementation guidelines. We have also validated the need for this technology with the end users.

We present practical designs for statistical analysis algorithms and their implementations on the SHAREMIND secure multi-party computation system. Experimental results show that our approach is sufficiently fast for practical use in non-real-time applications such as a statistical study. The technical strengths of our solution are generality, precision and practicality. First, we show that secure multi-party computation is flexible enough for implementing complex applications. Second, our use of secure floating point operations makes our implementation more precise. Third, we use the same algorithms as popular statistical toolkits like GNU R without simplifying the underlying mathematics.

We introduce a project that will validate the solution in practice—the linking of tax and education records in Estonia to study what kinds of specialists are needed in the ICT sector. A statement from the Estonian Data Protection Inspectorate indicates that our solution does not process personally identifiable information. This suggests that secure multi-party computation can provide a completely new level of privacy protection in the analysis of federated databases.

### Step 1: Data import

<i>Operation</i>	<i>Record count</i>	<i>Time</i>
Data import from offsite computer	2 000	3 s
	53 977	24 s

### Step 2: Descriptive statistics

<i>Operation</i>	<i>Record count</i>	<i>Time</i>
5-number summary (publish filter size)	2000	21 s
	20000	97 s
5-number summary (hide filter size)	2000	27 s
	20000	107 s
Frequency table	2000	16 s
	20000	222 s

### Step 3: Grouping and linking

<i>Operation</i>	<i>Record count</i>	<i>Time</i>
Median of incomes by subject	53 977	3 h 46 min
Linking two tables by a key column	2000×5 and 2000×3	28 s
Linking two tables by a key column	2000×7 and 2000×2	29 s

### Step 4: Statistical tests

<i>Operation</i>	<i>Record count</i>	<i>Time</i>
Student's t-test, equal variance	2000	167 s
	20000	765 s
Student's t-test, different variance	2000	157 s
paired t-test, known mean	2000 and 2000	98 s
paired t-test, unknown mean	2000 and 2000	102 s
$\chi^2$ -test, 2 classes	2000	9 s
	20000	10 s
$\chi^2$ -test, $n$ -class version, 2 classes	2000	20 s
$\chi^2$ -test, $n$ -class version, 5 classes	2000	23 s
Wilcoxon rank sum	2000	34 s
Wilcoxon signed-rank	2000 and 2000	38 s

Table 2: Running times of privacy-preserving statistics (in seconds)

## Acknowledgements

This research was supported by the European Regional Development Fund through Centre of Excellence in Computer Science (EXCS) and Competence Centre in Electronics-, Info- and Communication Technologies (ELIKO); the European Social Fund Doctoral Studies and Internationalisation programme DoRa; and by the Estonian Research Council under Institutional Research Grant IUT27-1. The end user validation and the development of privacy-preserving statistical tools is funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 284731.

The authors wish to thank the interviewees for their time and cooperation and the Estonian Center for Applied Research for their help in generating the artificial data used in the experiments of this paper.

## References

1. Gagan Aggarwal, Nina Mishra, and Benny Pinkas. Secure computation of the median (and other elements of specified ranks). *Journal of Cryptology*, 23(3):373–401, 2010.
2. Arne Ansper, Ahto Buldas, Margus Freudenthal, and Jan Willemson. Scalable and Efficient PKI for Inter-Organizational Communication. In *Proceedings of ACSAC 2003*, pages 308–318, 2003.
3. Arne Ansper, Ahto Buldas, Margus Freudenthal, and Jan Willemson. High-Performance Qualified Digital Signatures for X-Road. In *Proceedings of NordSec 2013*, number 8208 in LNCS, pages 123–138. Springer, 2013.
4. Arne Ansper, Ahto Buldas, Margus Freudenthal, and Jan Willemson. Protecting a Federated Database Infrastructure Against Denial-of-Service Attacks. In *Proceedings of CRITIS 2013*, number 8328 in LNCS, pages 26–37. Springer, 2013.
5. Gilad Asharov, Yehuda Lindell, and Hila Zarosim. Fair and Efficient Secure Multiparty Computation with Reputation Systems. In *Proceedings of ASIACRYPT 2013*, volume 8270 of *Lecture Notes in Computer Science*, pages 201–220. Springer Berlin Heidelberg, 2013.
6. Assaf Ben-David, Noam Nisan, and Benny Pinkas. FairplayMP: a system for secure multi-party computation. In *Proceedings of ACM CCS 2008*, pages 257–266, 2008.
7. Dan Bogdanov. *Sharemind: programmable secure computations with practical applications*. PhD thesis, University of Tartu, 2013.
8. Dan Bogdanov, Peeter Laud, and Jaak Randmets. Domain-Polymorphic Programming of Privacy-Preserving Applications. *Cryptology ePrint Archive*, Report 2013/371, 2013. <http://eprint.iacr.org/>.
9. Dan Bogdanov, Margus Niitsoo, Tomas Toft, and Jan Willemson. High-performance secure multi-party computation for data mining applications. *International Journal of Information Security*, 11(6):403–418, 2012.
10. 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.
11. 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.
12. Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *Proceedings of CM SIGMOD 2000*, pages 93–104, 2000.
13. Martin Burkhart, Mario Strasser, Dilip Many, and Xenofontas A. Dimitropoulos. SEPIA: Privacy-Preserving Aggregation of Multi-Domain Network Events and Statistics. In *Proceedings of USENIX 2010*, pages 223–240, 2010.
14. Ran Canetti, Yuval Ishai, Ravi Kumar, Michael K. Reiter, Ronitt Rubinfeld, and Rebecca N. Wright. Selective private function evaluation with applications to private statistics. In *Proceedings of PODC 2001*, pages 293–304. ACM, 2001.
15. Cybernetica. Income analysis of the Estonian Public Sector. Online service., 2013. <https://sharemind.cyber.ee/clouddemo/>, last accessed December 13<sup>th</sup>, 2013.
16. Ivan Damgård, Martin Geisler, Mikkel Krøigaard, and Jesper Buus Nielsen. Asynchronous multiparty computation: Theory and implementation. In *Proceedings of PKC 2009*, pages 160–179, 2009.

17. Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Proceedings of CRYPTO 2012*, volume 7417 of *LNCS*, pages 643–662. Springer, 2012.
18. Wenliang Du and Mikhail J. Atallah. Privacy-preserving cooperative statistical analysis. In *Proceedings of ACSAC 2001*, pages 102–110, 2001.
19. Wenliang Du, Shigang Chen, and Yung-Hsiang S. Han. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of SDM 2004*, pages 222–233, 2004.
20. Cynthia Dwork. Differential privacy. In *Proceedings of ICALP 2006*, volume 4052 of *LNCS*, pages 1–12. Springer, 2006.
21. Joan Feigenbaum, Benny Pinkas, Raphael Ryger, and Felipe Saint-Jean. Secure computation of surveys. In *EU Workshop on Secure Multiparty Protocols*, 2004.
22. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of STOC 2009*, pages 169–178. ACM, 2009.
23. Oded Goldreich and Rafail Ostrovsky. Software Protection and Simulation on Oblivious RAMs. *Journal of the ACM*, 43(3):431–473, 1996.
24. Myles Hollander and Douglas A Wolfe. *Nonparametric statistical methods*. John Wiley New York, 2nd ed. edition, 1999.
25. H.C.M. Hoonhout. Setting the stage for developing innovative product concepts: people and climate. *CoDesign*, 3(S1):19–34, 2007.
26. Rob J Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365, 1996.
27. Marek Jawurek and Florian Kerschbaum. Fault-tolerant privacy-preserving statistics. In *Privacy Enhancing Technologies*, volume 7384 of *LNCS*, pages 221–238. Springer, 2012.
28. Ahto Kalja. The X-Road Project. A Project to Modernize Estonia’s National Databases. *Baltic IT&T review*, 24:47–48, 2002.
29. Ahto Kalja. The first ten years of X-road. In *Estonian Information Society Yearbook 2011/2012*, pages 78–80. Department of State Information System, Estonia, 2012.
30. Ahto Kalja and Uno Vallner. Public e-Service Projects in Estonia. In *Proceedings of Baltic DB&IS 2002*, volume 2, pages 143–153, June 2002.
31. Liina Kamm, Dan Bogdanov, Sven Laur, and Jaak Vilo. A new way to protect privacy in large-scale genome-wide association studies. *Bioinformatics*, 29(7):886–893, 2013.
32. Gopal K Kanji. *100 statistical tests*. Sage, 2006.
33. Florian Kerschbaum. Practical privacy-preserving benchmarking. In *Proceedings of IFIP TC-11 SEC 2008*, volume 278, pages 17–31. Springer US, 2008.
34. Eike Kiltz, Gregor Leander, and John Malone-Lee. Secure computation of the mean and related statistics. In *Proceedings of TCC 2005*, volume 3378 of *LNCS*, pages 283–302. Springer, 2005.
35. Julia Lane, Pascal Heus, and Tim Mulcahy. Data Access in a Cyber World: Making Use of Cyberinfrastructure. *Transactions on Data Privacy*, 1(1):2–16, 2008.
36. Sven Laur, Riivo Talviste, and Jan Willemsen. From Oblivious AES to Efficient and Secure Database Join in the Multiparty Setting. In *Proceedings of ACNS 2013*, volume 7954 of *LNCS*, pages 84–101. Springer, 2013.
37. Sven Laur, Riivo Talviste, and Jan Willemsen. From Oblivious AES to Efficient and Secure Database Join in the Multiparty Setting (extended version). Cryptology ePrint Archive, Report 2013/203, 2013. <http://eprint.iacr.org/>.
38. Sven Laur, Jan Willemsen, and Bingsheng Zhang. Round-Efficient Oblivious Database Manipulation. In *Proceedings of ISC 2011*, pages 262–277, 2011.

39. Christopher Lettl. User involvement competence for radical innovation. *Journal of engineering and technology management*, 24(1):53–75, 2007.
40. Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and  $\ell$ -diversity. In *Proceedings of ICDE 2007*, 2007.
41. Yehuda Lindell, Kobbi Nissim, and Claudio Orlandi. Hiding the input-size in secure two-party computation. Cryptology ePrint Archive, Report 2012/679, 2012. <http://eprint.iacr.org/>.
42. Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkatasubramanian. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), March 2007.
43. Pille Pruulmann-Vengerfeldt, Liina Kamm, Riivo Talviste, Peeter Laud, and Dan Bogdanov. Deliverable D1.1—Capability model. <http://usable-security.eu/files/D1.1.pdf.pdf>, 2012.
44. Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13:1010–1027, 2001.
45. Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, November 1979.
46. Peter Suber. *Open Access*. MIT Press, 2012.
47. Hiranmayee Subramaniam, Rebecca N. Wright, and Zhiqiang Yang. Experimental analysis of privacy-preserving statistics computation. In *Proceedings of SDM 2004*, volume 3178 of *LNCS*, pages 55–66. Springer, 2004.
48. Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.
49. Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.
50. Jan Willemson. Pseudonymization Service for X-Road eGovernment Data Exchange Layer. In *Proceedings of EGOVIS 2011*, number 6866 in *LNCS*, pages 135–145. Springer, 2011.
51. Jan Willemson and Arne Ansper. A Secure and Scalable Infrastructure for Inter-Organizational Data Exchange and eGovernment Applications. In *Proceedings of ARES 2008*, pages 572–577. IEEE Computer Society, 2008.
52. Zhiqiang Yang, Rebecca N. Wright, and Hiranmayee Subramaniam. Experimental analysis of a privacy-preserving scalar product protocol. *Computer Systems Science & Engineering*, 21(1), 2006.
53. Andrew Chi-Chih Yao. Protocols for Secure Computations (Extended Abstract). In *Proceedings of FOCS 1982*, pages 160–164. IEEE, 1982.

## A Overview of implemented operations

Figure 3 showcases our privacy-preserving statistical functionality and its dependencies. The implementation is built on the arithmetical, comparison and oblivious vector operations provided by SHAREMIND. However, our algorithms can be ported to any SMC framework that provides the same set of features.

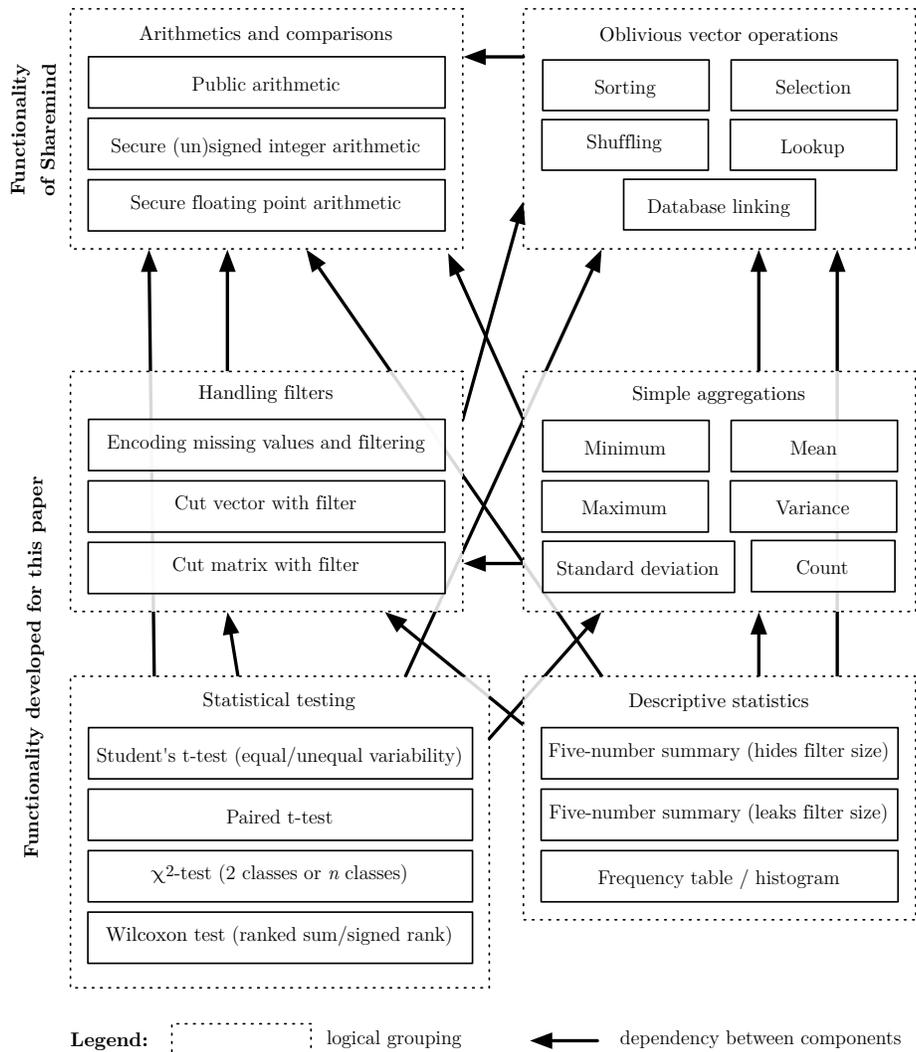


Fig. 3: Overview of operations implemented for our experiments