CYBERNETICA
Institute of Information Security

Constructing Privacy-Preserving Information Systems Using Secure Multiparty Computation

Dan Bogdanov, Liina Kamm

T-4-13 / 2011

Dan Bogdanov[1,3], Liina Kamm[2,3].
[1] Cybernetica, Institute of Information Security,
[2] Software Technologies and Applications Competence Center,
[3] University of Tartu, Institute of Computer Science

The research reported here was supported by:

1. Estonian Science foundation, grant(s) No. 8124,
2. the target funded theme SF0012708s06 "Theoretical and Practical Security of Heterogenous Information Systems",
3. the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS, and the Software Technology and Applications Competence Centre, STACC

Cybernetica research reports are available online at
`http://research.cyber.ee/`

Mailing address:
AS Cybernetica
Akadeemia tee 21
12618 Tallinn
Estonia

Constructing Privacy-Preserving Information Systems Using Secure Multiparty Computation

Dan Bogdanov, Liina Kamm

June 8, 2011

Abstract

The physical deployment of an information system is an important factor in its security. When an information system is hosted in an untrusted environment, its owner may lack controls to prevent unauthorized access to the data in the system. For example, if the system is hosted by a cloud provider, there is often no way for controlling which country the system is geographically located in. If the data owners do not trust the information system, then they may be unwilling to use it for processing their data. In response to these threats, the data security research community has developed secure multiparty computation techniques that reduce the trust requirements for the hosts of information systems. In this paper, we describe how to construct information systems that allow the use of secure multiparty computation techniques for achieving improved security and privacy guarantees.

1 Introduction

Organizations in both the public and private sector need to gather and process private data to achieve their goals. For example, government agencies need to be aware of changes in the economical and medical well-being of the population. Similarly, medical institutions have to collect patient data to improve treatments and perform research. Furthermore, private companies are collecting customer information to make business decisions. Since private information is processed during these activities, they are regulated by data protection laws that require the organizations to preserve the privacy of the involved individuals.

The regulations directly influence the design of information systems. Although the privacy requirements have existed for years, systems with strong privacy guarantees are still rare. Also, numerous incidents [2, 12] involving the loss of large databases on portable storage devices such as USB sticks or misplaced backups

show that the technologies and procedures of privacy preservation are not yet mature.

Research in the area of information security and cryptography has provided information system developers several techniques for improving the security of privacy-preserving data processing. One of the most promising results with regard to practical security guarantees are currently provided by secure multiparty computation. In secure multiparty computation, multiple servers exchange information in order to perform privacy-preserving computations.

Secure multiparty computation has been applied for solving practical problems [5,6]. However, the technology is not yet commonly used, as it sets restrictions on the deployment of applications and requires developers to rethink application architectures. In this article we discuss these restrictions and security guarantees that secure multiparty computation can provide for information systems that process private data.

In this paper we use privacy terminology from the ISO/IEC 29100 privacy framework standard [1]. *Personally identifiable information* (*PII*) is any information that:

1. can be used to identify the person to whom such information pertains,
2. from which such information can be derived, or
3. that is or might be directly or indirectly linked to a natural person.

The information system will have a set of *privacy safeguarding requirements* and privacy is *breached* when PII is not processed according to these requirements. The privacy safeguarding requirements restrict the ways in which the system is allowed to process PII. Therefore, techniques that help enforce such restrictions, also called *privacy enhancing technologies* or PETs reduce the risk of privacy breaches. For example, with secure multiparty computation, the parties processing the data do not have to see the values provided by the data owner in order to compute statistical results. This reduces the risk of PII breaches during processing through insider attacks or negligence.

The main contributions of this article are an analysis of how secure multiparty computation affects information system design, two generic information system architectures suitable for deployment in a secure multiparty computation setting and three examples that implement these architectures for improved privacy guarantees. In Section 2 we introduce secure multiparty computation as a technique for secure computations. Section 3 describes how secure multiparty computation and privacy requirements affect information system design. In Section 4 we propose two generic architectures for private information systems for which we give usage examples in Section 5.

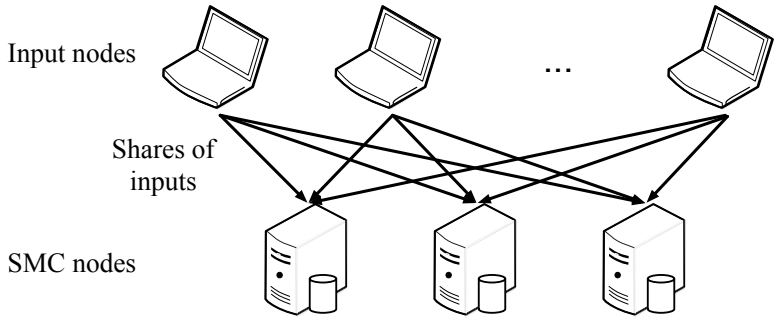

Figure 1: The input nodes connect to all SMC nodes and store their data using secret sharing.

2 Secure Multiparty Computation

Before discussing the use of secure multiparty computation in the context of information systems and privacy we give a generic overview of the technology.

Secure multiparty computation (SMC) with n parties is defined as the computation of a function $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ so that the result is correct and the input values of all the parties are kept private. Every party P_i will provide an input value x_i and learn only the result value y_i. For some functions f and some input values it might be possible to deduce other parties' inputs from the result. Still, in the case of data aggregation algorithms it is not possible to learn the inputs of other parties from the result.

In order use secure multiparty computation for implementing larger algorithms, it makes sense to use *universally composable* secure multiparty computation protocols [7]. Such protocols can be executed sequentially and in parallel without loss of privacy.

Most secure multiparty computation methods are based on encryption or secret sharing. However, due to the computational complexity of encryption-based solutions, methods based on secret sharing achieve a better performance. For this reason in this paper we concentrate on methods based on secret sharing—also called *share computing* techniques.

Share computing uses *secret sharing* for the storage of data. Let s be the secret value. An algorithm $\mathbf{S}$ defines a *k-out-of-n threshold secret sharing scheme*, if it computes $\mathbf{S}(s) = [s_1, \ldots, s_n]$ and the following conditions hold [13]:

1. **Correctness:** s is uniquely determined by any k shares from $\{s_1, \ldots, s_n\}$ and there exists an algorithm $\mathbf{S}'$ that efficiently computes s from these k shares.

2. **Privacy:** having access to any $k-1$ shares from $\{s_1, \ldots, s_n\}$ gives no infor-

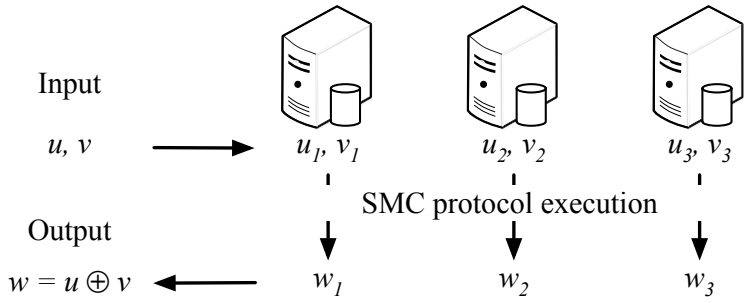

Figure 2: The SMC nodes can run SMC protocols to compute useful results from secret-shared data.

mation about the value of s, i.e., the probability distribution of $k-1$ shares is independent of s.

An input party who provides data for secure computations, distributes it data into shares using secret sharing. The input parties will then send one share of each value to a single SMC party. The data storage process with three SMC nodes is illustrated on Figure 1. This separation of nodes into input nodes and SMC nodes is useful, since it does not force every party in the information system to run SMC protocols. This reduces the complexity of the system.

After the data has been stored, the SMC nodes can perform computations on the shared data. Notice that none of the SMC nodes can reconstruct the input values thanks to the properties of secret sharing. We now need to preserve this state during computations. This is achieved by using secure multiparty computation protocols that specify which messages the SMC nodes should exchange in order to compute new shares of a value that corresponds to the result of an operation with the input data.

Figure 2 shows the overall structure of secure multiparty computations. Assume, that the SMC nodes have shares of input values u and v and want to compute $w = u \oplus v$ for some operation $\oplus$. They run the SMC protocol for operation $\oplus$ which gives each SMC node one share of the result value w. Note that the protocols do not leak information about the input values u and v . For details and examples on how this can be achieved, see the classical works on SMC [3, 7, 8, 15].

After computations have been finished, the results are published to the client of the computation. The SMC nodes send the shares of the result values to the client node that reconstructs the real result from the shares. See Figure 3 for an example. Note, that it is important not to publish results when they could still leak information about the inputs. The algorithms running in the SMC nodes must be audited to ensure that they do not publish private inputs.

SMC protocol suites provide different levels of privacy depending on the adopted

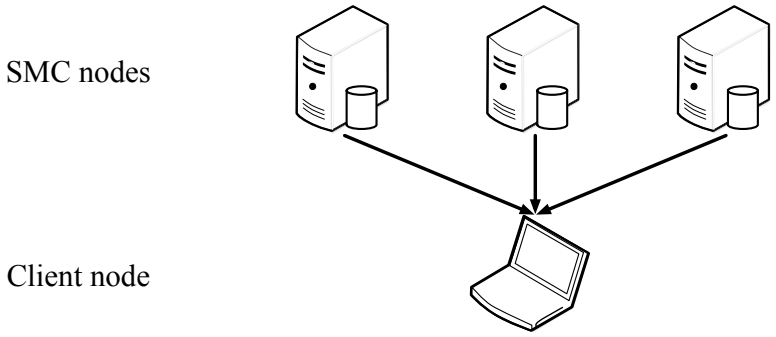

Figure 3: Results are published after the computation is complete.

security model. In the semi-honest model it is assumed that SMC parties follow protocol, but they may also perform more computations to try and figure out the input values. In the malicious model, SMC nodes can deviate from the protocol, but the deviation can be detected and the necessary corrective actions can be taken.

A significant parameter of SMC systems is the number of misbehaving parties in a system. Classical results [3,8] show that there is a correct, private and robust share computing protocol evaluating a function f with n parties in the malicious model if and only if no more than $\frac{n}{3}$ players are corrupted. In the semi-honest model no more than $\frac{n}{2}$ parties can be corrupt. These bounds are optimal and therefore we assume that there is an honest majority among parties in the system. Protocols without honest majorities exist, but they are weaker to cryptographic attacks.

We note that protocols that are proven secure in the semi-honest model provide reasonable privacy preservation. Intuitively, if the secret sharing scheme is secure and the SMC protocols are secure in the semi-honest model, then even the system administrator of a SMC node cannot reconstruct the individual values provided by the data input nodes. For example, in a three-node SMC system that is secure in the semi-honest model, two SMC nodes will have to collude to reconstruct input values. In the real world, such collusion is prevented by choosing SMC parties in a way that they are motivated to preserve privacy. This can be achieved by contractual means. It also helps to choose the SMC nodes from the data providers so that they are also preserving their own privacy.

Several secure multiparty computation frameworks have implementations [9,11, 14]. While the majority of implementations are research-oriented, it is expected that more practical systems will be created as the technology matures.

3 Secure Multiparty Computation in Information Systems

3.1 Prerequisites for Using SMC

The use of secure multiparty computation and, more precisely, share computing, adds the following requirements to an information system.

1. **Multiple server nodes**. The processing of data requires a minimum of three SMC nodes.

2. **Focus on aggregation.** SMC improves security guarantees the most in statistics and data mining systems.

3. **Limited amount of data.** The current SMC technologies are less efficient than standard computing. This limitation must be considered during application planning.

Recently, significant advances have been achieved in alternative secure computation techniques. The best known result from recent years is the construction of a fully homomorphic encryption scheme [10]. The use of fully homomorphic encryption could remove the requirement for multiple server nodes. However, the technique is several orders of magnitude slower than share computing and requires significantly more storage space. Therefore, we do not consider it in this paper for efficiency reasons.

In the following we discuss the effect that the use of secure multiparty computation has on the design of the information system. We concentrate our analysis on atomic database operations and generic information system processes.

3.2 Working with Data

We start by discussing the effect of secret sharing on the level of atomic database operations. We categorize the operations into creating, reading, updating and deleting data. For each operation we discuss both risks and security guarantees associated with the use of this technology.

Storing new PII. Input nodes insert PII into the database in a secret-shared form. As a result, the SMC nodes are unable to learn the actual input values. This operation has low privacy risks, since the SMC nodes learn only the shares of the values.

Reading PII. We have to distinguish between reading operations initiated by the SMC nodes and by all other nodes. The first kind is secure since moving shares around inside the SMC node does not compromise the PII. However, if shares are transferred outside the SMC node, we have to consider it as publishing.

It is easy to see why individual values in the database must not be published as it could directly compromise the PII provided by the respective input node. It follows directly, that it is impossible to completely preserve privacy in information systems that need to process and present PII on the level of individual records.

Therefore, our focus is on systems where the main task is PII aggregation. The respective category of applications would be statistical analysis tools and data warehouses. Indeed, publishing aggregate results computed from collected PII is less probable to leak information. Of course, this depends, among other things, on the amount of PII records, the aggregate function and the distribution of the PII values.

Running aggregation functions on a subset of the PII specified by filters is a complex issue. It is clear that the SMC nodes may not learn information about the exact identity of records that belong to the specified subset. We note, however that oblivious selection techniques such as the ones described in [4] can be used to perform filtered aggregations without disclosing the records that corresponded to the filtering condition.

To conclude, the SMC nodes are allowed to read shares from the database and run computations. They should also do their best to not publish non-aggregated PII. This requires the SMC nodes to be aware of data processing algorithms and possibly have them built into the system. This way the client nodes can request the results of the whole computation that can be performed without having to publish the input data or intermediate values.

Updating PII. We need to distinguish between two cases of updating PII values in the database. Global updates that change all the records in the database without filtering are acceptable. If the changes replace old values or update them with results of secure computations then the privacy of stored PII is preserved.

However, updating individual records based on some criteria is more complex, as the matching records have to be identified. It is again possible to use the oblivious selection techniques to prevent the SMC nodes from knowing which rows were updated. However, it must be noted that this technique requires the whole database to be modified and this can be inefficient. This again stresses the suitability of privacy preserving techniques in creating data warehouses, since the latter have less update operations.

Deleting data. Deleting data is a low-risk operation when performed on a whole

table or set of PII. However, deletions based on a criteria cannot be done as the SMC nodes would have to separate the respective PII records. The oblivious selection technique does not helps us, because there is not yet a good way to perform oblivious deletions. A feasible alternative is to use a private attribute to determine, which records are "inactive" and use this attribute to separate the records in further computations. Since private values can be updated private in a way mentioned above, this private inactivity flag can be also be set based on securely evaluated conditions.

If we introduce public attributes in the database in addition to private ones, we gain the ability to distinguish between values in the database for some cases. The main requirement is that the public value may in no way compromise the secret-shared values that it will be associated with. Artificial identifiers are suitable for such use. Given such public values, we can separate, update and delete private values if the conditions are evaluated based on public values and not PII.

3.3 Processes

We will now discuss a subset of the basic processes of information systems and how they are affected by the use of secure multiparty computation.

Gathering data. Data should be secret-shared as close to the source as possible. In other words, PII should be accessible by as few nodes as possible. Preferably, every input node performs secret sharing and transmits only shares to the SMC nodes.

PII can originate from human users with a desktop or web-based interface, or from an existing database or a sensor network. The choice of input source has little effect on the technical privacy guarantees.

Backup and restoring. Every SMC node can back up its part of the secret-shared database. It can be restored as usual. Notice, that should a database or a backup of a single SMC node be stolen, the SMC nodes can protect the PII by initiating the *resharing* procedure. During resharing, all values in the database will be recomputed from the previous ones by computing new shares that represent the same values. The process does not leak information about the shares and the stolen shares become effectively useless. However, if the resharing process takes place, all SMC nodes must destroy previous backups as they pose a security risk together with the compromised shares. Additionally, they must and make new backups immediately, as the loss of one SMC node's database will render all shares useless.

Performing computations. The client nodes may request that the SMC run an

aggregation algorithm on collected PII that is stored in the database. The SMC nodes will synchronously run the necessary protocols and compute shares of the results. These results can also be stored in the SMC nodes' local databases as input for future computations.

Note, that the security guarantees for input data can also be extended to the query parameters provided by client nodes. This means that it is possible to query the secure database so that the SMC nodes do not learn the true values of the query parameters. This may be important if the client nodes have something to lose by disclosing their query parameters. This, however, does not mean that the client nodes can do whatever they want with the PII, as the SMC nodes restrict the list of allowed analysis algorithms beforehand.

The algorithms used in the computations have to be agreed upon before computation. Every SMC node must have the same version of the algorithm. If a single SMC node does not support an algorithm due to technical or privacy reasons, the computation cannot proceed. Since the SMC nodes are ultimately responsible for controlling the publishing of results, they must independently check that the algorithms do not leak information. Hence, even though the SMC nodes cannot learn the query parameters, they will validate the data analysis operations needed to execute the query. This can be a part of the audit process applied to secure systems.

Access control and auditing. Access control in a privacy-aware information system is not different from a standard information system. Users have access levels and they are enforced as usual.

However, an important issue in privacy-preserving information systems is the auditing process. Since the organizations hosting the SMC nodes have no way to compromise PII without colluding with other SMC nodes, the need to prevent technical personnel from accessing data is not as pressing as in normal information systems. Audit logging is still an important feature, but extra care must be taken to make sure that shares of the database do not become public through system logs.

A limitation of privacy-preserving systems is the lack of data auditing measures. Since in data aggregation we are not disclosing PII to the client nodes, we cannot allow monitoring applications that let the user inspect individual records for correctness of data. It is, however, possible to check the range of user inputs by using cryptographic techniques. Also, one can analyze the consistency of the collected PII by statistical methods such as value distribution and aggregate statistics.

4 Deployment Architectures

The deployment of nodes is an important consideration in the development of a privacy preserving information system based on secure multiparty computation. Choosing input nodes and client nodes is simple—everyone who provides data is an input node and everyone who needs the analysis results is a client node.

There are more factors to consider when deciding who should host the SMC nodes. Firstly, one should choose organizations that are sufficiently trusted by possible individuals or organizations at the input nodes. Second, the SMC node hosts should have no incentive for colluding with each other. This can be achieved by selecting competing organizations as hosts. If the selection of SMC nodes fails because of these restrictions, then any trusted third party trusted by input nodes and already selected SMC nodes, can fulfill the role of the final SMC node.

Recall, that for share computing, the number of SMC nodes should be at least three, as two-party secure computation systems are more inefficient than ones with three or more parties. On the other hand, the number should be as low as possible, as every additional node decreases the efficiency of computation by increasing communication complexity.

Taking that into account, the most efficient scenario is the one with three SMC nodes. They should be chosen in such a way that the rest of the parties can trust them to not collude. This is called *trust transfer*—the input nodes trust the group of SMC nodes to handle their PII without abusing the trust.

We now look at two example deployment scenarios for selecting organizations to host the SMC nodes.

A single service provider. If an information system has one hosting organization, but wishes to apply secure multiparty computation, then it needs to find at least two more organizations to reach the required three. These organizations should be trusted by the input nodes and they should not be motivated to compromise the PII being collected. Good candidates are competing organizations and secure hosting providers. Keep in mind, that the users do not have to trust the organizations unconditionally, just the fact that they will not collude. The reduced trust requirements can lead to greater participation in such studies.

Input nodes can be either individuals or other organizations. PII can be collected for the internal use of the service provider or, for example, marketing analysis results. Examples for such scenarios are medical organizations or research institutions conducting surveys and consulting companies performing data mining on data collected from companies.

Systems with two competing host organizations have to act similarly to the single service provider case and find a third partner.

Multiple service providers. Three or more competing organizations can create a common data warehouse without disclosing their own PII records to the partners. The input nodes may again be individuals and outside organizations, but they could also be the SMC node host organizations themselves. The organizations can collect the data for their own purposes or share the results with third parties.

Sharing business information in a data warehouse may be beneficial for making business decisions for a members of an industrial consortium. For example, Internet service providers could learn more about the global properties and shortcomings of the network by aggregating information about their individual deployments and monitoring results. This could also help detect distributed network attacks [6]. However, these organizations cannot disclose the information required for such an analysis as it contains PII and possibly business secrets and its publication may cause losses. In this setting, the prerequisites for SMC node selection are ideally achieved, since the organizations are motivated to process the data, but care enough for their own data to refrain from colluding with another SMC node.

With more than three candidates for SMC nodes, a decision could be made to elect three among them to collect and share the data. This would give the information system better performance. However, if this cannot be done due to trust issues, more parties could become SMC nodes. It is, in fact, possible to provide better security guarantees with more SMC nodes as was mentioned in Section 2. For example, with three SMC nodes none of the organizations can collude but with five SMC nodes up to two of them can collude without compromising PII. Note that in this scenario the SMC node host may have access to the values of its own PII records, but this does not give an advantage in determining the PII records of other input nodes.

5 Case Studies

5.1 A Survey System

The privacy guarantees of surveys will benefit greatly from secure multiparty computation. Given that the survey organizer finds partners to ensure privacy protection, the provided guarantees for people filling the survey can be better than with standard solutions. The identities of the individual input nodes cannot be determined neither directly nor indirectly from collections of attributes. Communicating these improved guarantees can help researchers arrange surveys with potentially embarrassing questions.

In SMC terms, every survey taker is an input party. It is important that the implementations of surveys that use secure multiparty computation provide user-friendly interfaces so that technological shortcomings will not stop the people from

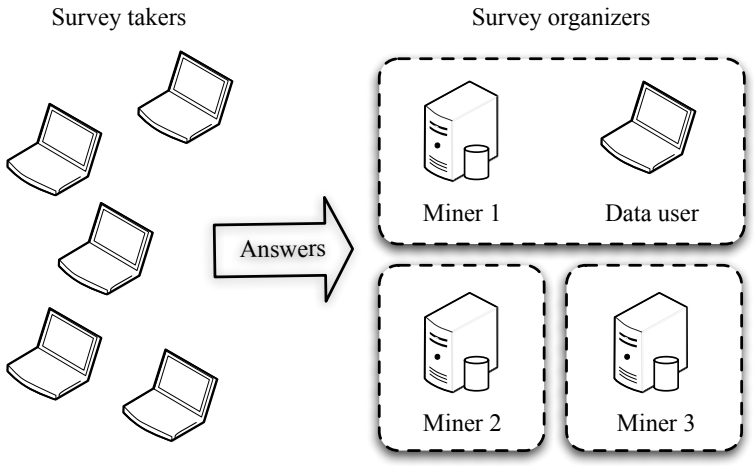

Figure 4: Example deployment of a survey system.

taking part in the survey. The data from the filled surveys will be collected and stored by the SMC nodes hosted by the survey organizer and the other enlisted parties.

During the following analysis the analysts can request basic statistics from continuous attributes, histograms over multiple-choice attributes and more complex statistical analyses such as correlations. The service provider may allow third parties to make queries on the collected data. However, this requires an agreement between all the SMC nodes since the responsibility for the PII is shared.

Figure 4 shows an example deployment of a survey system based on secure multiparty computation. On the left there are the survey takers who connect to all of the three SMC noes. On the right, there are the SMC nodes processing survey data. The organizational limits are shown by dashed lines. The data analysis results are published to the organization hosting the first SMC node.

Note, that surveys are a rather specific case for applications with a single service provider and actual use-cases may range from online auctions and voting systems to privacy-preserving billing systems.

5.2 Sales Data Analysis

Companies who run sales outlets like retail shops analyze the shopping habits of their customers. This is one reason for deploying customer loyalty programs. Customers have the possibility to request a customer card in exchange for some personal data. The company uses this data to analyze customer demographics. The customer then presents this card during each shopping trip at the cashier for a discount. The list of bought items is registered and shopping basket analysis can

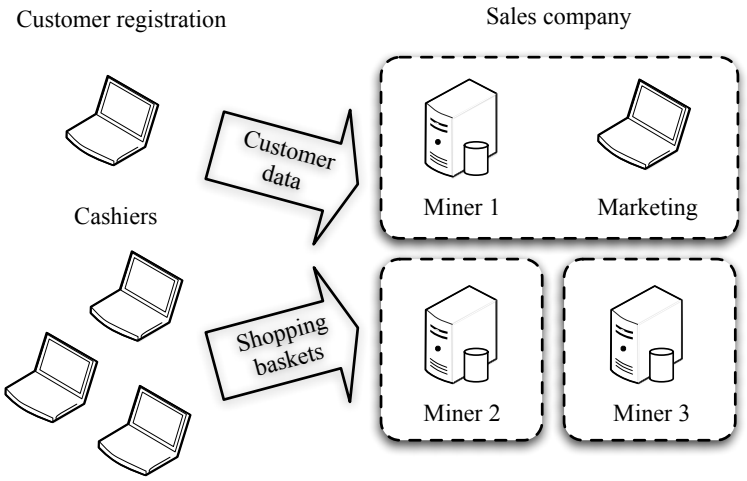

Figure 5: Example deployment of a sales data analysis system.

be performed.

To use secure multiparty computation, the company would find two partners host the secure data. Note, that if the other nodes are competitors who also provide similar data, this kind of an analysis can be performed on a much larger scale. Customer PII is collected at the customer service desks where cards are issued and at the cashier desks where shopping basket contents are stored. The shopping baskets can be associated with the customer through an artificial identifier assigned with the card.

The data will be analyzed by the marketing department of the companies. Useful analyses include a demographical profile of the customer base, frequent itemsets from shopping baskets and possibly correlations between customer and products. The received generic associations will help the company make business decisions for targeted advertising or shopping floor rearrangements. The privacy of the customer is again improved, as no shopping basket is associated to a certain customer.

Figure 5 shows the deployment of the described system.

5.3 Network Traffic Analysis

Finally, we look at a scenario where all the SMC nodes are stakeholders and provide PII for the information system. The example comes from the area of Internet service provision. Assume that ISPs are aggregating information about bottlenecks in their networks. Such actions require co-operation between service providers, but it is hard to achieve as traffic logs contain PII about the users'

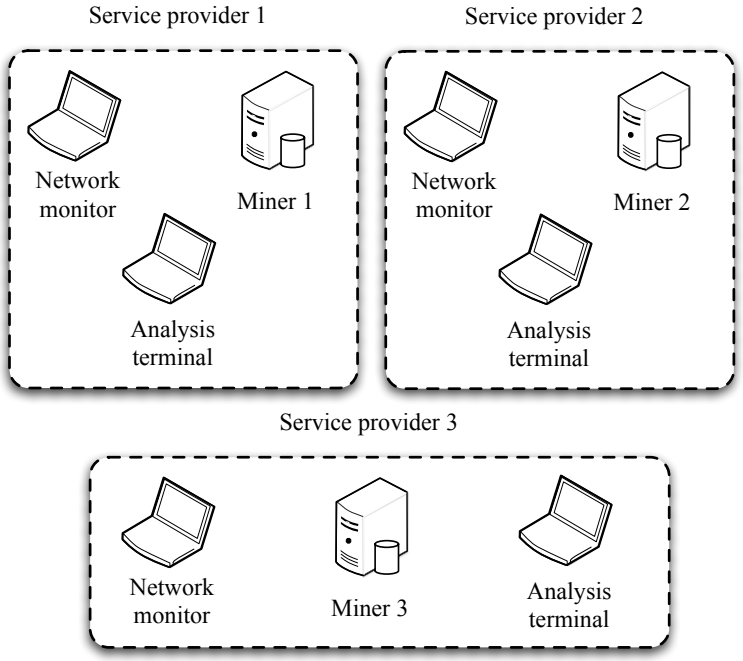

Figure 6: A privacy-preserving network monitoring system.

network using habits.

The ISPs can select among themselves a number of SMC nodes. Then, every service provider in the group acts as an input node and sends network usage data to the shared secure database. Subsequently, the service providers can act as client nodes and run queries on the jointly collected data.

This system is uniquely interesting because all involved organizations want to use the data and, at the same time, keep its privacy. Even, if the latter is caused by regulation, the trust relations are stronger than between weakly-connected organizations.

The deployment of secure multiparty computation parties can be seen on Figure 6. Every company is an input node, a SMC node and a client node at the same time. There may be additional organizations who do not act as SMC nodes and only provide data and request aggregated results.

6 Conclusion

Developments in secure multiparty computation techniques have made their application in real-world scenarios feasible. It is possible to create information systems with strong, provable security guarantees. However, the technology sets restric-

tions on the kinds of applications that can be built.

We have described how the use of secure multiparty computation affects information system design with regard to data management and generic processes. We found that while it may be impossible to preserve privacy in all information systems due to the invasive nature of business processes, the technology can successfully be applied to develop secure data warehouses.

Based on this we propose two architectural patterns for building information systems based on secure multiparty computation methods. Three example applications illustrate the use of these architectures.

References

[1] ISO/IEC FCD 29100. Information technology – Security techniques – Privacy framework, 2011.

[2] Michael Barbaro and Tom Zeller Jr. A face is exposed for AOL searcher no. 4417749. The New York Times, August 9th, 2006.

[3] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc of STOC '88*, pages 1–10, 1988.

[4] Dan Bogdanov, Sven Laur, and Jan Willemson. Sharemind: A framework for fast privacy-preserving computations. In *Proc. of ESORICS '08*, pages 192–206, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] Peter Bogetoft, Ivan Damgård, Thomas Jakobsen, Kurt Nielsen, Jakob Pagter, and Tomas Toft. A practical implementation of secure auctions based on multiparty integer computation. In *Proc. of Financial Cryptography*, LNCS 4107, 2006.

[6] Martin Burkhart, Mario Strasser, Dilip Many, and Xenofontas Dimitropoulos. SEPIA: privacy-preserving aggregation of multi-domain network events and statistics. In *Proc. of USENIX Security 2010*, USENIX Security'10, pages 15–15. USENIX Association, 2010.

[7] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[8] David Chaum, Claude Crèpeau, and Ivan Damgård. Multiparty unconditionally secure protocols. In *Proc. of STOC '88*, pages 11–19, 1988.

[9] SHAREMIND development team. The SHAREMIND framework. http://sharemind.cyber.ee, 2007-2011.

[10] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. of STOC 2009*, STOC '09, pages 169–178. ACM, 2009.

[11] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. Fairplay — a secure two-party computation system. In Proceedings of the 13th USENIX Security Symposium (2004), pp. 287-302., 2004.

[12] Open Security Foundation. DataLoss DB. http://datalossdb.org, 2009.

[13] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[14] VIFF development team. The virtual ideal functionality framework. http://viff.dk, 2007-2011.

[15] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *Proc. of the FOCS '82*, pages 160–164, 1982.