Share Analysis. Not Data.

PRANA-DATA

DA.1 The State of the Art in Privacy-Preserving Data Analysis

Project	PRANA-DATA
Project leader	Wessel Kraaij, TNO
Work package	White paper
Deliverable number	DA.1
Authors	Sietse Ringers, Radboud University, Nijmegen, the Netherlands
Reviewers	Bart Jacobs, Radboud University
Date	31-03-2017
Version	1
Access Rights	Public
Status	Final

PRANA-DATA Partners:

Portavita, TNO, Radboud Universiteit Nijmegen, Maastricht UMC+, UMCG

COMMIT/ COMMIT is a public-private research community solving grand challenges in information and communication science shaping tomorrow's society

COMMIT/ A public- private research community

This publication was supported by the Dutch national program COMMIT.

Contents

Abstract

This document describes recent developments in privacy-preserving data analysis, with particular focus on secure multiparty computation and machine learning.

As the availability and accessibility of measurement and storage devices increases, so too does the body of medical data that is potentially available for statistical research. This puts us in a difficult position: how can we use this data to further advance our understanding of important medical and biological processes, while respecting the privacy-sensitive nature of the data, as well as the interests of the relevant stakeholders? Although such research generally has no need for personal attributes such as for example names or home addresses, it could still be necessary to take sensitive information such as the age of the data subject into account. Thus we cannot always avoid the usage of privacy-sensitive data, so that we have to find other ways to ensure that such data is handled responsibly. Although there are many moral, legal and technological aspects to this issue, in this paper we will focus on the technological side: we examine the state in the art in areas such as secure multiparty computation, machine learning and homomorphic encryption to perform privacy-respecting data analysis on (possibly distributed) data repositories.

1 Introduction

1.1 Secure multiparty computation

Secure multiparty computations allow multiple parties to jointly compute some function in such a way that each party keeps its input data private to itself. For example, a number of hospitals may wish to jointly mine their medical data, or a patient may want to evaluate his personal medical data against a remotely executed algorithm without divulging his data. An immediate question that arises, which is out of the scope of this paper, is whether the output of such computations is itself sufficiently privacy-preserving, and thus safe for publication. Instead we will assume that the result of the computation is either safe or deemed essential. Apart from the privacy requirement (no party should learn anything more than what can be derived from its prescribed output), correctness is also important (each party is guaranteed that the output that it receives is correct), as well as guaranteed output delivery (dishonest parties should not be able to prevent honest parties from receiving their output) and fairness (dishonest parties should receive their outputs if and only if the honest parties also receive their outputs). In the 2-party case a generic protocol was created in 1986 by Yao [28] that allows two parties to evaluate any function of their inputs, which was later extended [17] to be secure against active adversaries (see Section 1.3), and to more than two parties [15]. In realistic scenarios, however, these generic protocols are often insufficiently efficient, so that specialized protocols are necessary for specific problems to achieve acceptable running times. All schemes that we discuss in this paper fall in this category. For an extensive introduction and overview of secure multiparty computation, a more elaborate description of Yao's generic protocol, and a number of other protocols, we refer to [19].

1.2 Machine learning

Machine learning is a branch of artificial intelligence that gives computers the ability to process and classify data and to detect patterns from large, noisy or complex data sets, using a variety of statistical, probabilistic and optimization tools. For example, data from X-ray images or biomedical samples may be used to estimate the chance of someone getting cancer [10]. There are a number of distinct types of machine learning algorithms, of which supervised and unsupervised learning algorithms are the most important. In the former, the algorithm is presented a labeled set of training data or examples in a training phase. After analyzing the training data a function is inferred, which can then be applied to new data to for example label or group it. Important supervised algorithms include Support Vector Machines (SVM) [9]; Decision Trees [24] and Naive Bayes classifiers. By contrast, in unsupervised algorithms there is no separate training phase; instead a set of examples (without labels) are given, and it is up to the learner to find the pattern or discover the groups. An important example is K-means clustering [20]. We will discuss all of these techniques in more detail in Section 2.

1.3 Adversary models

Many cryptographic schemes are argued or proven secure with respect to a certain adversary model: that is, a set of capabilities that an adversarial entity can use without compromising the security goals of the scheme. Which party is the adversary, what capabilities the adversary has, and what it means for a scheme to be secure, differs depending on the scheme and on the situation. Ideally a scheme stays secure if any party deviates in any way from the protocol (i.e., the active or malicious adversary model), but this is often very difficult to achieve at realistic speeds. For this reason many schemes (including the schemes that we discuss in this paper) use the semi-honest or passive adversary model, in which the adversary must always follow the established protocol, but is allowed to retain and study all data that it receives during an execution of the protocol. It is clear that this model offers less security, and depending on the situation it may not suffice. To fill the gap between these two models, Aumann and Lindell [4] proposed a new covert adversary model, in which the adversary has a (non-negligible) probability to be caught when it deviates from the protocol, deterring it from doing so, at relatively little extra cost compared to the passive adversary model. They include a protocol for oblivious transfer (in which the sender has two inputs x0, x1; the receiver has a bit b; and the protocol ensures that the receiver receives xb but not x¯ b, without the sender learning b), that when = 1/2 is 4 times more expensive than contemporary protocols in the passive adversary model such as, e.g., [8]. Protocols for computing dot products and set intersections in the covert adversary model were introduced in [21].

2 The State of the Art

2.1 Training machine learning algorithm

As explained in Section 1.2, supervised machine learning algorithms require a training phase before usage, taking into account that there may legal obstructions from the raw training data to be shared and copied away from their origin – for example, for many hospitals there are stringent privacy regulations forcing medical data to be kept on the premises. In these cases one might take the approach of the Personal Health Train [2], and instead bring the algorithms to the data. In the Personal Health Train metaphor, each station is a tightly regulated data location that a "train" can use. The data at each station is "FAIR" (findable, accessible, interoperable and reusable) [27], and can be granularly regulated by the station owner via house rules, specifying what trains have access to what data. The "trains" are algorithms executing specific research questions or use cases that need FAIR data, under control of the researcher. The "track" sets maintain and check the rules of interaction and is the interface between trains and stations. The PHT initiative supports both horizontally partitioned data (where the data subjects are partitioned across multiple databases containing the same attributes for each patient) and vertically partitioned data (where the attributes of data subjects is spread across multiple databases) [16], depending on the use case. Only certified research trains and certified FAIR data stations are allowed on the track and their use is fully controlled and auditable.

Closely related is a recent method by Damiani et al. [11] to train SVMs across multiple (horizontally partitioned) datasets simultaneously, without needing to merge all patient data in advance at a central site. Based on the ADMM method by Boyd et al. [6], which is also used by the PHT initiative mentioned above, each dataset runs a software agent called a slave node that trains on the local data and exchanges part of the result with a master node. This master node collects all results from the slave nodes, calculates new coefficients, and sends these back to the slave nodes. This process is performed until the slave nodes converge to the same result, and it is such that during the process no information about any patient can be derived from the messages passing between the master and slave nodes. When run on data with 2 classes (namely alive or deceased), 2 to 10 features per patient, 50 to 500 patients per site, and 2 to 10 sites, the systems takes some 2000 iterations to converge. The paper does not mention how long this takes, but since this concerns the training phase of the SVM instead of the usage phase (for which e.g., the earlier mentioned method by Bost et al. might be used), the running time is arguably of lesser importance. The authors have published an (unlicensed) open source implementation of their work.

2.2 Using machine learning algorithms

An important (unsupervised) learning technique is K-means clustering [20], in which a group of n real vectors are partitioned into K sets such that each vector belongs to the cluster with the nearest mean. This can be done by randomly selecting K locations as initial cluster centers, assigning each vector to the closest center, recomputing the centers of the resulting clusters, and repeating this process until the cluster centers converge. In such cases it may be necessary to hide the vectors as well as the resulting clusters from the party that does the clustering. Since this was introduced in 2000 [3,18] a number of schemes have been proposed, e.g., [7,13,22], of which [13] by Erkin et al. seems particularly promising. In their scheme, the computation is facilitated by a central entity that learns essentially nothing about the input vectors or the intermediate and final clustering. The scheme uses the passive adversary model, although the scheme can be adapted to work in the active adversary model, at a cost. Alternatively, the trust can be distributed by having a number of helper users assist in the computations, with the additional advantage of substantially increasing the efficiency of the scheme. The paper does not contain a formal security proof, but is nevertheless convincing. A preliminary implementation shows that the scheme scales well: clustering 100,000 vectors of dimension 12 into 10 clusters, using 64 helper users, takes 24 minutes. The authors do not seem to have published their implementation, although many of its ingredients are provided in the SeComLib library [12].

More recently Bost et al. [5] proposed a promising set of protocols for a wide array of classifiers, including SVMs, naive Bayes, and decision trees. Their scheme keeps both the training model and the user input private to the server and the user, respectively. They achieve this by introducing protocols for a number of core operations, namely comparison based on [26] (for equality or which of two inputs is the largest), argmax, and dot products, that can be composed without losing security (in the passive adversary model) to the mentioned classifiers. These protocols are modular in the sense that they could also be of use in other models or situations. The paper includes formal security proofs, and the authors provide a freely available open source implementation of their scheme that achieves impressive runtimes (e.g., 3.8 seconds for a naive Bayes classifier with 24 classes and 70 features). The paper is well-written and broadly applicable.

2.3 Handling sensitive data responsibly

Especially for privacy-sensitive data it is important that only authorized agents can access and modify the data. For example, a doctor may need to access and modify data of one patient, while a researcher may need (read-only) access to the medical data of all patients, although he will generally not be interested in the identities of the patients. At the Radboud University in Nijmegen, The Netherlands, a framework called PEP [25] is being developed that can cryptographically ensure that the data can only be accessed by authorized parties, encrypting it in transit and during storage. PEP, standing for Polymorphic encryption and Pseudonymisation, uses the flexibility of the ElGamal partially homomorphic encryption scheme [14] to cryptographically force that all involved parties are aware of the minimum amount of information that they need to perform their role. In this context, it is important to distinguish three levels of identification (c.f. [23]):

– Some data can can completely identify the data subject (e.g., a name);

– the data can be psuedonymized: two data records can be linked together as coming from the same data subject, but the identity of the data subject remains unknown (for example, two data records may belong to patient 7486613);

– the data can be anonymized: multiple data records cannot be linked as coming from the same data subject

For each data record the PEP system internally keeps track of from which data subject it originated, but depending on the need of the data user or handler, it can ensure all three levels of identification. As long as the data is in the PEP system it is encrypted at all times. The data is at a storage facility, from whose perspective the data is psuedonymized. Later on it can be decided who can decrypt the data (for example, a doctor or a researcher). This decision is made on the basis of a policy, in which the data subject can play a key role. In order to ensure that only the intended recipient can indeed decrypt the data, the ciphertext is modified by a trusted party, called the transcryptor, to “fit” the ElGamal key of the recipient. This is done after the data was encrypted and stored at the storage facility, so that when the data enters the system it need not yet be known who will need to access it. The transcryptor only handles encrypted data so it is unaware of its contents, and from its perspective, all data is anonymized. Thus, although the transcryptor plays a critical role in the system by handling all data, it is completely unaware of both the contents and the owners of the data. When the data is decrypted, the identity of the data subject may or may not be revealed to the receiver, depending on the relevant policies. In addition, for accountability the system ensures that all transactions are logged. The PEP system will also allow researchers to securely export the data that they can access to the DRE system [1], for Digital Research Environment, allowing them to make use of their own analytical online workspace to process the data in a secure environment and safely share it with other researchers.

3 References

1. DRE: Digital research environment, http://nieuws.radboudumc.nl/ radboudumc-biedt-onderzoekers-digital-research-environment, in dutch, see also http://www.nlhealthresearch.nl/2016/11/25/ research-service-officially-launched-health-ri-conference-2016/. Accessed: 2017-03-24

2. The Personal Health Train, http://www.dtls.nl/fair-data/ personal-health-train/, accessed: 2017-03-22

3. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) Proceedings of the 2000 ACM SIGMOD International Conference. pp. 439–450. ACM (2000)

4. Aumann, Y., Lindell, Y.: Security against covert adversaries: Efficient protocols for realistic adversaries. J. Cryptology 23(2), 281–343 (2010), https://eprint.iacr. org/2007/060

5. Bost, R., Popa, R.A., Tu, S., Goldwasser, S.: Machine learning classification over encrypted data. In: 22nd Annual Network and Distributed System Security Symposium – NDSS 2015. The Internet Society (2015), https://eprint.iacr.org/ 2014/331

6. Boyd, S.P., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 3(1), 1–122 (2011)

7. Bunn, P., Ostrovsky, R.: Secure two-party k-means clustering. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007. pp. 486–497. ACM (2007), https://eprint.iacr.org/2007/231

8. Chou, T., Orlandi, C.: The simplest protocol for oblivious transfer. In: Lauter, K.E., Rodr´ıguez-Henr´ıquez, F. (eds.) Progress in Cryptology - LATINCRYPT 2015. Lecture Notes in Computer Science, vol. 9230, pp. 40–58. Springer (2015), https://eprint.iacr.org/2015/267

9. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)

10. Cruz, J.A., Wishart, D.S.: Applications of machine learning in cancer prediction and prognosis. Cancer informatics 2 (2006)

11. Damiani, A., Vallati, M., Gatta, R., Dinapoli, N., Jochems, A., Deist, T., Soest, J.V., Dekker, A., Valentini, V.: Distributed learning to protect privacy in multicentric clinical studies. In: Holmes, J.H., Bellazzi, R., Sacchi, L., Peek, N. (eds.) Artificial Intelligence in Medicine - AIME 2015. Lecture Notes in Computer Science, vol. 9105, pp. 65–75. Springer (2015)

12. Erkin, Z.: The Secure Computation Library (SeComLib), http://cybersecurity. tudelft.nl/content/secomlib, accessed: 2017-01-23

13. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Privacy-preserving distributed clustering. EURASIP J. Information Security 2013, 4 (2013)

14. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Information Theory 31(4), 469–472 (1985)

15. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A.V. (ed.) Proceedings of the 19th Annual ACM Symposium on Theory of Computing. pp. 218–229. ACM (1987)

16. Karr, A.F., Lin, X., Sanil, A.P., Reiter, J.P.: Secure statistical analysis of distributed databases. In: Statistical Methods in Counterterrorism, pp. 237–261. Springer (2006)

17. Lindell, Y.: Parallel coin-tossing and constant-round secure two-party computation. J. Cryptology 16(3), 143–184 (2003)

18. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) Advances in Cryptology - CRYPTO 2000. Lecture Notes in Computer Science, vol. 1880, pp. 36–54. Springer (2000), https://eprint.iacr.org/2008/197

19. Lindell, Y., Pinkas, B.: Secure multiparty computation for privacy-preserving data mining. IACR Cryptology ePrint Archive 2008, 197 (2008), https://eprint.iacr. org/2008/197

20. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: In 5-th Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297 (1967)

21. Miyaji, A., Rahman, M.S.: Privacy-preserving data mining in presence of covert adversaries. In: Cao, L., Feng, Y., Zhong, J. (eds.) Advanced Data Mining and Applications - 6th International Conference, ADMA 2010. Lecture Notes in Computer Science, vol. 6440, pp. 429–440. Springer (2010)

22. Oliveira, S.R.M., Za¨ıane, O.R.: Achieving privacy preservation when sharing data for clustering. In: Jonker, W., Petkovic, M. (eds.) Secure Data Management, VLDB 2004 Workshop, SDM 2004. Lecture Notes in Computer Science, vol. 3178, pp. 67– 82. Springer (2004)

23. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (aug 2010), http://dud.inf. tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, v0.34. Accessed: 201703-24

24. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986), http://dx.doi.org/10.1023/A:1022643204877

25. Verheul, E., Jacobs, B., Meijer, C., Hildebrandt, M., de Ruiter, J.: Polymorphic encryption and pseudonymisation for personalised healthcare. Cryptology ePrint Archive, Report 2016/411 (2016), https://eprint.iacr.org/2016/411. See also: http://pep.cs.ru.nl/, accessed: 2017-03-23

26. Veugen, T.: Comparing encrypted data, http://visionlab.tudelft.nl/sites/ default/files/Comparing%20encrypted%20data.pdf, accessed: 2017-03-22

27. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., et al.: The FAIR guiding principles for scientific

data management and stewardship. Scientific data 3 (2016), http://www.nature.com/articles/sdata201618

28. Yao, A.C.: How to generate and exchange secrets (extended abstract). In: 27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986. pp. 162–167. IEEE Computer Society (1986)