Proceedings of Statistics Canada Symposium 2016
Growth in Statistical Information: Challenges and Benefits.

Practical Applications of Secure Computation for Disclosure Control

Luk Arbuckle, Khaled El Emam[1]

Abstract

Microdata dissemination normally requires data reduction and modification methods be applied, and the degree to which these methods are applied depend on the control methods that will be required to access and use the data. An approach that is in some circumstances more suitable for accessing data for statistical purposes is secure computation, which involves computing analytic functions on encrypted data without the need to decrypt the underlying source data to run a statistical analysis. This approach also allows multiple sites to contribute data while providing strong privacy guarantees. This way the data can be pooled and contributors can compute analytic functions without either party knowing their inputs. We explain how secure computation can be applied in practical contexts, with some theoretical results and real healthcare examples.

Key Words: Disclosure control; secure computation; remote analysis.

1. Introduction

Considering a risk-based approach to disclosure control, secure computation can be formulated as “protected” pseudonymous data. The encryption ensures the "protection" in that no human is working directly with the data: instead they are only seeing the statistical results on key variables. Although the source data is protected by encryption for secure computation, there would still be a need to address some of the concerns that exist with remote analysis systems in general. Secure computation is well suited to scenarios in which there is a continuous, systematic collection and analysis of data, because the computations can be defined and optimized and then continuously applied.

1.1 Risk-Based Disclosure Control

There is long history of statistical disclosure control methods that consider the relationship (marginal distribution) from key variables or quasi identifiers to protect against identity disclosure (Duncan et al. 2011). Regulations and guidance documents suggest a framework for de-identification that is risk based—i.e., that incorporates the context of the data release into the risk assessment framework—to have strong assurances that the risk is “reasonable”.

We will assume that masking unique or direct identifiers is well understood and instead focus on the key variables. Consider the probability of re-identification given that an attacker makes an attempt as $\Pr(\text{reid} \mid \text{attempt})$ (Marsh et al. 1991). We can therefore formulate the problem as the probability of re-identification and an attack using

$$\Pr(\text{reid, attempt}) = \Pr(\text{reid} \mid \text{attempt}) \times \Pr(\text{attempt}).$$

The probability of an attacker attempting a re-identification is given by the context of the data release, using a subjective assessment of risk (Morgan et al. 1992; Vose 2008) based on expert opinion and precedent (e.g., (Centers for Disease Control and Prevention 2004; Statistics Canada 2007; Subcommittee on Disclosure Limitation Methodology 2005). The factors that affect an attempt include the security and privacy practices of the data requestor and contractual obligations. Furthermore, a defensible risk threshold can be determined based on precedent by evaluating the potential invasion of privacy.

[1] Luk Arbuckle, Children’s Hospital of Eastern Ontario Research Institute, 401 Smyth Road, Ottawa, Canada, K1H 8L1; Khaled El Emam, Children’s Hospital of Eastern Ontario Research Institute, 401 Smyth Road, Ottawa, Canada, K1H 8L1, and Faculty of Medicine, University of Ottawa, 451 Smyth Road, Ottawa, Canada, K1H 8M5.

All of the above factors can be used to formulate a repeatable risk assessment framework (El Emam 2013). In fact many standards and guidelines have incorporated such a risk-based framework in their recommendations for sharing health data (Committee on Strategies for Responsible Sharing of Clinical Trial Data; Board on Health Sciences Policy; Institute of Medicine. 2015; Health Information Trust Alliance 2015; Information Commissioner's Office 2012; Office for Civil Rights 2012; PhUSE De-Identification Working Group 2015; The Expert Panel on Timely Access to Health and Social Data for Health Research and Health System Innovation 2015).

1.2 Secure Computation

Assume that multiple parties want to pool data and compute a function without either party knowing their inputs. The basic idea of secure computation is to compute a function on encrypted data, without ever decrypting the data to achieve the desired output. Cryptographic primitives, or building blocks, to create secure computation protocols can come from homomorphic encryption, garbled circuits, secret sharing, or others, each with their own advantages and disadvantages.

Homomorphic encryption became practical with the introduction of the Paillier cryptosystem because computation time is reasonable, but it has a limited set of operations (Paillier 1999). Our practical examples in the next sections will focus on this technology. Yao's garbled circuits (Yao 1986) have been considered impractical due to computation time and memory requirements, although new methods may change this (Gueron et al. 2015; Songhori et al. 2015). A secret sharing scheme such as Shamir's (Shamir 1979) would be computationally efficient in a homomorphic scheme (Benaloh 1986). Some have raised concerns that parties must reveal their shares at the time of secret reconstruction (i.e., to get the final output), in which case other techniques could be employed (Desmedt and Frankel 1989), or implementations can randomize new shares representing the same secret value without disclosing the result itself.

The field of secure computation is advancing rapidly with more efficient and scalable methods (Rohloff and Cousins 2014), as well as specialized hardware to accelerate computations (Cousins et al. 2015). Furthermore, general purpose software for statistical analysis using cryptosystems is also being developed (Dan Bogdanov et al. 2014).

Secure computation is consistent with guidance provided by regulators as a means to protect personal health information while sharing encrypted data for the purposes of collaborative analysis. With secure computation the highest levels of security controls are implied, and there would be no processing of personal health information by humans. With appropriate contractual obligations and measures to ensure there are no leakages of personal information from the results themselves (O'Keefe and Chipperfield 2013), secure computation can be thought of in a risk-based framework as *protected* pseudonymous data with a very low risk of re-identification.

2. Practical Applications

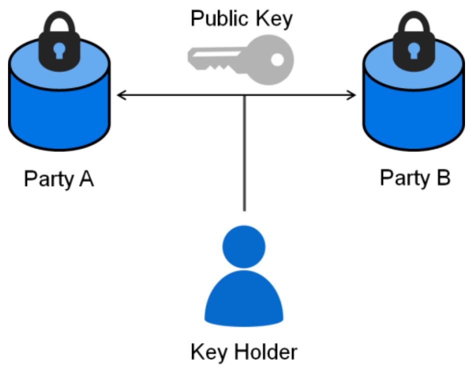
2.1 Secure Linking

Linking can be used to perform record lookup, or database matching for deduplication. Often, however, the best fields for linking are ones that cannot be disclosed (e.g., social security number, first and last name). The goal of secure linking is to link without sharing sensitive or personal information, and this is not limited to just the fields used for linking. Revealing to another party which data subjects are in a database may itself be a disclosure if membership to the database is sensitive.

In our secure linking protocol (El Emam and Arbuckle 2013 chap. Secure Linking) we use secure computation with a semi-trusted third party (sTTP)—trusted to run the protocol, but unable to obtain sensitive or personal information about data subjects even if it wanted to. Because the data and the computations on the data are protected by encryption, the parties do not need to trust one another or the sTTP. None of the parties involved can "peak" into the data or the computations. Furthermore, a breach at any one site would not reveal the identity or personal information of data subjects.

Referring to Figure 2.1-1, the first step in our secure linking protocol is for the key holder to generate private and public keys, and to distribute the public key to the data custodians who will use it to encrypt the link variables. The encrypted data will then be shared with a central aggregator, or one data custodian sends encrypted data to the other. A homomorphic equality test is then run on the pooled encrypted data. The encrypted match results are then sent to the key holder, who uses the private key to decrypt the results.

Figure 2.1-1
Distributing the public key to data custodians



This secure protocol is used by the Institute for Clinical Evaluative Sciences (ICES) for linking de-identified data (matching on insurance number, name, and date of birth), and was proposed to determine Chlamydia screening and testing rates with a public health agency (matching electronic medical records from family doctors to lab testing). It was also proposed for a human papillomavirus (HPV) vaccine initiative impact assessment, where more details about the protocol can be found (El Emam et al. 2012a).

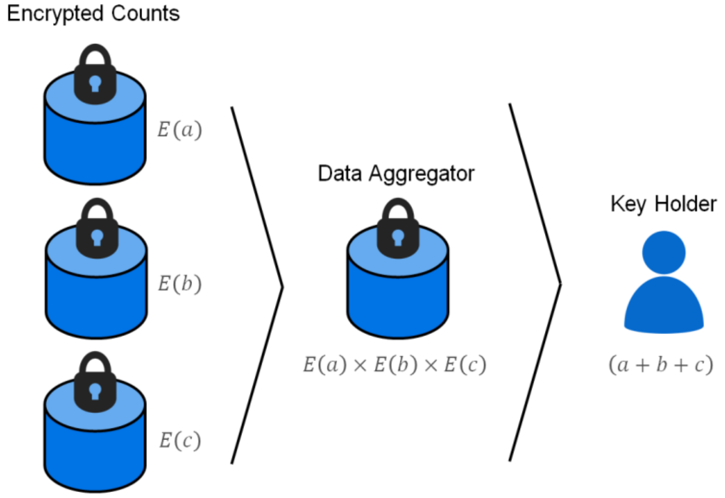
2.2 Prevalence of ARO's

Using a secure data collection system, providing strong privacy and confidentiality assurances, we were able to conduct a point prevalence study to assess rates of antimicrobial resistant organisms (ARO) in long term care homes in Ontario (El Emam et al. 2014). Although there is stigma attached to the identification of residents carrying ARO's in long term care homes, secure computation allowed the collection of colonization and infection data without revealing the rates for any of the participating homes. This addressed the need to collect data about the prevalence of ARO's in long term care homes for public health surveillance and intervention purposes.

The basic framework of the secure data collection system can be seen in Figure 2.2-2, in which $E(m)$ is used to denote encrypted cyphertext for plaintext m, and for the Pailier cryptosystem multiplication of cyphertext is equivalent to addition of plaintext, i.e., $E(a) \times E(b) = E(a + b)$. Long term care homes provided the counts, which were encrypted at the point of collection. These encrypted counts were then combined by a data aggregator using secure computation, which then passed the intermediate results to the key holder for decryption. Secure computation was used to determine the mean colonization or infection rates and the standard deviation, by region and facility size, and to run a two-sample randomization test (randomized t-test) for non-response bias.

All long term care homes in the province were asked to provide colonization or infection counts for methicillin resistant Staphylococcus aureus (MRSA), vancomycin-resistant enterococci (VRE), and extended-spectrum beta-lactamase (ESBL) as recorded in their electronic medical records, and the number of current residents. Data was collected online during the October-November 2011 period. Overall, 82% of the homes in the province responded, which is much higher than in previous attempts to collect data (without the use of secure computation). The microbiological findings and their distribution were consistent with available provincial laboratory data reporting test results for AROs in hospitals.

Figure 2.2-2
Computing point prevalence using a Paillier cryptosystem



2.3 Rare Adverse Drug Events

Logistic regression is commonly used in the analysis of adverse drug events. Data needs to be pooled, however, to detect rare events and ensure sufficient population heterogeneity to ensure the safety and effectiveness of a drug for subpopulations. We therefore developed a secure distributed logistic regression protocol using a single analysis center with multiple sites providing data (similar to the previous example) for post-marketing surveillance. We also extended the protocol to use generalized estimating equations (GEE) to account for correlated data, other generalized linear models (GLM), and survival models (El Emam et al. 2012b).

To estimate a generalized linear model (Agresti 2002), we can use the Newton-Raphson method and iteratively compute the parameter estimates b using

$$b(t+1) = b(t) - [I(t)]^{-1}u(t),$$

where $u(t)$ is the score vector and $I(t)$ is the information matrix for iteration t. With multiple sites contributing data (horizontally partitioned, so that they have the same covariates and coding formats), the score vector and information matrix are in fact computed individually at each site, and combined later. That is, for i sites,

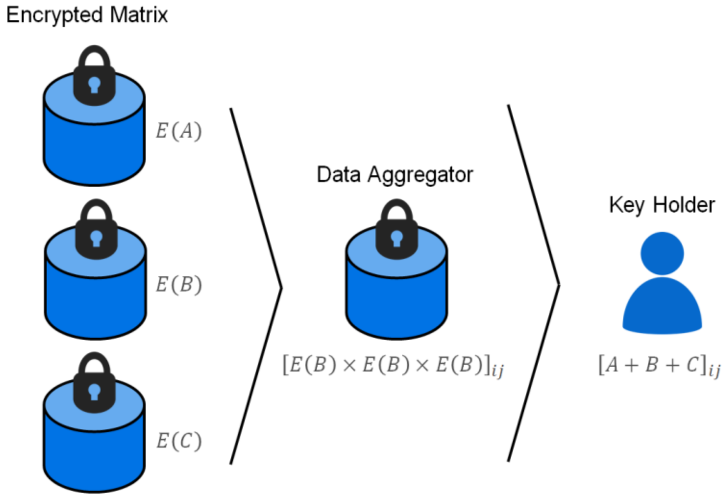
$$u(t) = \sum_i u_i(t) \text{ and } I(t) = \sum_i I_i(t).$$

Unfortunately attempting to pool the intermediate statistics leaves the sites open to many potential disclosures. These disclosures can come from the information matrix, the covariance matrix, indicator variables, even the iterations themselves (for a summary of these potential disclosures see the appendix to El Emam et al. (2012b)). A core tenet of cryptography is to avoid any and all leakages of information—otherwise it could be used to find a way to extract the secrets that are meant to be protected.

Secure computation can be used to hide all of the intermediate computations. Our protocol, called Secure Pooled Analyis acRoss K-Sites (SPARK), uses the secure building blocks of addition, multiplication, dot product, matrix multiplication, matrix inverse, and two-norm distance and comparison, many of which we extended for the purposes of implementing SPARK. The protocol to implement secure distributed logistic regression was also evaluated to assess its computational performance on a variety of datasets, as performance is a common concern with the use of secure computation. Even on commodity hardware the time it took to fit a logistic regression model of one million records across five sites was only about five minutes (disregarding the communication time between sites).

The simplest example of a secure building block being used in the protocol is shown for matrix addition in Figure 2.3-3. In this case we simply need to multiply the individual cyphertext messages which are the matrix elements. Of course, the complete implementation of SPARK for a GEE or GLM is more complicated than this example would suggest. Nonetheless it shows that from the simple properties of the Paillier cryptosystem basic building blocks can be derived that allow for more complicated analysis to be performed.

Figure 2.3-3
Pooling score vector and information matrix using homomorphic addition



3. Conclusions

With appropriate contractual obligations and measures to ensure there are no leakages of personal information from the results themselves, secure computation can be thought of in a risk-based framework as *protected* pseudonymous data with a very low risk of re-identification. It is well suited to scenarios in which there is a continuous, systematic collection and analysis of data, such as public health surveillance, because the computations can be defined and optimized and then continuously applied.

Acknowledgements

This work was funded by the Canada Research Chairs program, and the Canadian Institutes of Health Research.

References

Agresti, A. (2002). *Categorical Data Analysis*. Wiley, Hoboken, New Jersey.

Benaloh, J. C. (1986). “Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret (Extended Abstract).” *Advances in Cryptology — CRYPTO’ 86*, Lecture Notes in Computer Science, A. M. Odlyzko, ed., Springer Berlin Heidelberg, 251–260.

Centers for Disease Control and Prevention. (2004). *Integrated Guidelines for Developing Epidemiologic Profiles: HIV Prevention and Ryan White CARE Act Community Planning*.

Committee on Strategies for Responsible Sharing of Clinical Trial Data; Board on Health Sciences Policy; Institute of Medicine. (2015). *Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk*. Washington (DC): National Academies Press (US);

Cousins, D., Rohloff, K., Peikert, C., and Sumorok, D. (2015). *SIPHER: Scalable Implementation of Primitives for Homomorphic EncRyption*. Final Technical Report, Raytheon BBN Technologies, Rome, NY, USA.

Dan Bogdanov, Liina Kamm, Sven Laur, and Ville Sokk. (2014). “Rmind: A Tool for Cryptographically Secure Statistical Analysis.” *IACR Cryptology ePrint Archive*, 512, 1–40.

Desmedt, Y., and Frankel, Y. (1989). “Threshold Cryptosystems.” *Advances in Cryptology — CRYPTO’ 89 Proceedings*, Lecture Notes in Computer Science, G. Brassard, ed., Springer New York, 307–315.

Duncan, G. T., Elliot, M., and Salazar-González, J.-J. (2011). *Statistical Confidentiality*. Springer New York, New York, NY.

El Emam, K. (2013). *Guide to the De-Identification of Personal Health Information*. CRC Press (Auerbach).

El Emam, K., and Arbuckle, L. (2013). *Anonymizing Health Data: Case Studies and Methods to Get You Started*. O’Reilly.

El Emam, K., Arbuckle, L., Essex, A., Samet, S., Eze, B., Middleton, G., Buckeridge, D., Jonker, E., Moher, E., and Earle, C. (2014). “Secure Surveillance of Antimicrobial Resistant Organism Colonization or Infection in Ontario Long Term Care Homes.” *PLoS ONE*, 9(4), e93285.

El Emam, K., Hu, J., Samet, S., Peyton, L., Earle, C., Jayaraman, G., Wong, T., Kantarcioglu, M., and Dankar, F. (2012a). “A Protocol for the Secure Linking of Registries for HPV Surveillance.” *PLoS ONE*, 7(7).

El Emam, K., Samet, S., Arbuckle, L., Tamblyn, R., Earle, C., and Kantarcioglu, M. (2012b). “A Secure Distributed Logistic Regression Protocol for the Detection of Rare Adverse Drug Events.” *Journal of the American Medical Informatics Association*.

Gueron, S., Lindel, Y., Nof, A., and Pinkas, B. (2015). “Fast Garbling of Circuits Under Standard Assumptions.” *22nd ACM Conference on Computer and Communications Security*, Denver, CO, 1–43.

Health Information Trust Alliance. (2015). *HITRUST De-Identification Framework*. HITRUST Alliance.

Information Commissioner’s Office. (2012). *Anonymisation: Managing Data Protection Risk Code of Practice*. Information Commissioner’s Office.

Marsh, C., Skinner, C., Arber, S., Penhale, B., Openshaw, S., Hobcraft, J., Lievesley, D., and Walford, N. (1991). “The Case for Samples of Anonymized Records From the 1991 Census.” *Journal of the Royal Statistical Society, Series A (Statistics in Society)*, 154(2), 305–340.

Morgan, M. G., Henrion, M., and Small, M. (1992). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge University Press, Cambridge; New York.

Office for Civil Rights. (2012). *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule*. Department of Health and Human Services, Washington, DC.

O’Keefe, C., and Chipperfield, J. (2013). “A Summary of Attack Methods and Confidentiality Protection Measures for Fully Automated Remote Analysis Systems.” 81(3), 426–455.

Paillier, P. (1999). “Public-key cryptosystems based on composite degree residuosity classes.” *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, EUROCRYPT’99, Springer-Verlag, Berlin, Heidelberg, 223–238.

PhUSE De-Identification Working Group. (2015). *De-Identification Standards for CDISC SDTM 3.2*.

Rohloff, K., and Cousins, D. B. (2014). “A Scalable Implementation of Fully Homomorphic Encryption Built on NTRU.” *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, R. Böhme, M. Brenner, T. Moore, and M. Smith, eds., Springer Berlin Heidelberg, 221–234.

Shamir, A. (1979). “How to Share a Secret.” *Commun. ACM*, 22(11), 612–613.

Songhori, E. M., Hussain, S. U., Ahmad-Reza, S., Schneider, T., and Koushanfar, F. (2015). “TinyGarble: Highly Compressed and Scalable Sequential Garbled Circuits.” *2015 IEEE Symposium on Security and Privacy*, San Jose, CA, 411–428.

Statistics Canada. (2007). “Therapeutic Abortion Survey.” <http://www23.statcan.gc.ca/imdb/p2SV.pl?Function=getSurvey&SDDS=3209>.

Subcommittee on Disclosure Limitation Methodology. (2005). “Statistical Policy Working Paper 22 - Report on Statistical Disclosure Limitation Methodology.” Federal Committee on Statistical Methodology.

The Expert Panel on Timely Access to Health and Social Data for Health Research and Health System Innovation. (2015). *Accessing Health And Health-Related Data in Canada*. Council of Canadian Academies.

Vose, D. (2008). *Risk Analysis: A Quantitative Guide*. Wiley, Chichester, England ; Hoboken, NJ.

Yao, A. C.-C. (1986). “How to Generate and Exchange Secrets.” *27th Annual Symposium on Foundations of Computer Science, 1986*, Toronto, ON, 162–167.