Exploring Complexity in Health: An Interdisciplinary Systems Approach
A. Hoerbst et al. (Eds.)

doi:10.3233/978-1-61499-678-1-152

Evaluation of Secure Computation in a Distributed Healthcare Setting

Eizen KIMURA[a,1], Koki HAMADA[b], Ryo KIKUCHI[b], Koji CHIDA[b], Kazuya OKAMOTO[c], Shirou MANABE[d], Tomohiko KURODA[c], Yasushi MATSUMURA[d], Toshihiro TAKEDA[d] and Naoki MIHARA[d]
[a]*Department of Medical Informatics, Ehime University Department of Integrated Medicine*
[b]*NTT Secure Platform Labs., NTT Corporation*
[c] *Division of Medical Information Technology and Administrative Planning, Kyoto University Hospital*
[d]*Medical Informatics, Osaka University*

Abstract. Issues related to ensuring patient privacy and data ownership in clinical repositories prevent the growth of translational research. Previous studies have used an aggregator agent to obscure clinical repositories from the data user, and to ensure the privacy of output using statistical disclosure control. However, there remain several issues that must be considered. One such issue is that a data breach may occur when multiple nodes conspire. Another is that the agent may eavesdrop on or leak a user's queries and their results. We have implemented a secure computing method so that the data used by each party can be kept confidential even if all of the other parties conspire to crack the data. We deployed our implementation at three geographically distributed nodes connected to a high-speed layer two network. The performance of our method, with respect to processing times, suggests suitability for practical use.

Keywords. Secure computation, secondary usage, trusted party, personal privacy

1. Background

Integrating heterogeneous data, such as the "-omics" biological and clinical data, has led to a new era of translational bioinformatics. However, there is no widely held consensus on the disclosure of clinical data repositories to third parties, which has slowed the progress of biomedical research [1]. To protect sensitive - including personal data, we must preserve "input privacy" and "output privacy". Input privacy ensures that even the administrator of a data repository cannot extract personal information from the repository. Output privacy serves to limit the available information so that analysts cannot make inferences regarding the identity of specific individuals using the data obtained from their queries. To enable studies to transition from the laboratory bench to the bedside, the biology community has already implemented essential infrastructure called "i2b2" with "SHRINE" [2]. In the architecture of SHRINE, the trusted third-party (aggregator) exists in addition to data repositories and users. When a user issues a query to the aggregator, the aggregator gathers the result of the query on each repository, conducts Statistical

[1] Corresponding Author: kandalva@gmail.com

Disclosure Control (SDC) that assures output privacy, and sends the result to the user. Although SHRINE is a novel architecture for aggregation and assuring output privacy, it is not concerned about input and output privacy for the aggregator. Therefore, there is potential risk that the data and privacy breach if the aggregator is corrupted.

To exclude aforementioned risk in the medical informatics field, several works [3,4] employed secure computation that is a method of proceeding statistical analysis while the data remains encrypted. They assume the existence of several servers in addition to data repositories and users and proceeds as follows. Clinical data of each repository is encrypted and sent to the servers in a way that each server cannot obtain information about the clinical data. When a user issues a query, the servers compute and interact with each other, obtains the encrypted result of the query while the clinical data has never decrypted, and finally, the user decrypted it and obtained the result. One of the main drawbacks of their results is that they assume the servers do not collude. If the servers collude, the clinical data is to be disclosed completely. Furthermore, a popular method of SDC is excluding the value of a small number of persons from the result, and this method is difficult in secure computation since the data is encrypted.

To realize the federated clinical repositories that are open to researchers with both input and output privacy, we should develop a security framework that can resist collusion and SDC capability. The framework should be able to perform secret computation in sufficient performance even when the parties are distributed geographically. In this study, we developed a secure computation framework that ensures secure computing among geographically dispersed nodes. In this framework, each node directly conducts secure computation in response to a user's query and additional participants are not required. This framework is secure against collusion, i.e., clinical data of a repository will be not disclosure even if the other repositories collude, and capable of SDC.

2. Method

We implemented a secure computation engine using secret sharing technique [5][6] and its enhancing method [7]. Secretly-shared random numbers used in the engine are generated and stored in advance by a trusted third party. This setting can be substituted to offline phase used in the method developed by Damgård et al.[7]. The engine supports arithmetic functions, including shuffle [8], sort [9], and comparison [10]. These operations are done in a finite field with order $p=2^{61}-1$ (Mersenne prime number) for a technical reason for efficiency. We also implemented floating point arithmetic (input, output, addition, subtraction, multiplication, and division). By combining these arithmetic functions, we developed secure calculations using SDC based, on statistical analysis methods: one-way analysis of variance (ANOVA), the Kruskal–Wallis test, Pearson’s correlation coefficient and Spearman's rank correlation coefficient and summary statistics with SDC.

A secure computation server (SCS) that hosts the secure computation process, as well as an analysis client, are implemented as a combination of a C++ library and a command line interface-based application (Fig. 1). The SCS runs a server daemon on every node. The analyst launches the analysis client and enters a query; the analysis client then issues the query to all nodes. When every SCS accepts the query, each SCS loads comma separated value (CSV) files stored locally in the node, processes secure computations, and determines the shared secret value. All of the shared secret values are

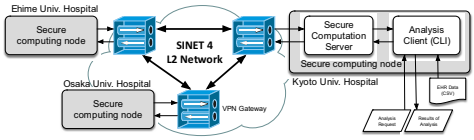

Figure 1. Overview of the secure computing system

Table 1. Network performance between pairs of nodes

Destination \ Source	Ehime	Kyoto	Osaka
Ehime	NA	13.67 (0.20) 576.0 (85.26)	9.50 (0.12) 529.20 (92.15)
Kyoto	13.69 (0.13) 573.20 (71.93)	NA	6.48 (0.140) 611.20 (59.15)
Osaka	9.46 (0.15) 598.00 (60.14)	6.48 (0.13) 580.40 (21.37)	NA

* Upper side: RTT(average/sd) msec Lower side: Bandwidth (average/sd)Mbps

Table 2. Attributes of the electronic medical record (EMR) data

Attribute	Type	Scope /Categories
Age	Number	[20,99]
Sex	Category	2
Hb1Ac	Number	[2.3,17.4]
Urinary Protein	Category	5
Serum Cr	Number	[0.1-19]
eGFR	Number	[2.3,1713.6]

Table 3. Processing time for each type of operation

Statistical method	Class	Time (Sec)
ANOVA		
	UP	30.67
	CKD	30.42
	eGFR	30.34
Kruskal-Wallis		
	UP	582.01
	CKD	618.77
	eGFR	607.38
Pearson		
	HbA1c~eGFR	19.58
Spearman		
	HbA1c~UP	941.93

Statistical method	Class	Time, sec (sd)
Count	all	1.16 (0.13)
Sum	all	1.89 (0.13)
Mean	all	10.58 (1.82)
Var	all	15.64 (2.77)
Min	all	2.00 (0.23)
Max	all	2.00 (0.23)
Median	all	19.97 (27.32)

returned to the analysis client, which finally decrypts the results of the query from these shared secret values. We established three SCS nodes at the Ehime, Osaka, and Kyoto University Hospitals. The nodes were interconnected with the layer two network of Science Information NETwork 4 (SINET 4). Virtual private network (VPN) routers (FITELnet F2000; Furukawa Co. Ltd., Tokyo, Japan) were set up as a gateway for every node build IPSec-based full mesh VPN network (Figure 1).

All of the nodes were installed on a personal computer [CPU, Intel Core i502540M 2.60 GHz (2 cores; Intel Co., Santa Clara, CA, USA); RAM, 8 GB; SSD, 128 GB (THNSNC12; Toshiba, Tokyo, Japan); operating system, Ubuntu (ver. 12.04; Canonical Ltd., London, UK); compiler, g++ 4.6.3; and Ethernet, 1 Gbps]. We benchmarked the network performance using iperf and ping.

We set $t = 10$ to be the SDC standard. This is the most severe level according to published SDC standards and is the standard used at The Centres for Medicare & Medicaid Services (CMS) of the US government. This limitation will apply to the sum, frequency, mean, variance, minimum, maximum and median. Every n–1 node receives only those results that meet the following condition: (the sum of the frequencies returned from each node) $\geq t$. We also applied the dominance rule ($m = 1$, $k = 80\%$) to suppress any risky cells that dominate over $k\%$ of the sum at m nodes; the rule suppresses every sum cell that does not meet the condition at every value of S (n–1 nodes) and X (m nodes, $X \subset S$) . Sum cells must also satisfy the following equation: (the sum of X)/(the sum of S) $\leq k/100$. We prepared a verification scenario that conducted a statistical analysis of the hemoglobin A1c (HbA1c), serum creatinine (Cr), and urinary protein (UP) levels, as well

as the estimated glomerular filtration rate (eGFR), and severity of chronic kidney disease (CKD), to estimate the relationships between the laboratory results and the severity of CKD. We selected patients who were more than 20 years of age, had underwent physiological testing between 2012/04/01 and 2014/03/31, and for whom all laboratory results were acquired within 30 days. The data on age, sex, Hb1Ac, UP, and Cr levels were extracted from the electronic medical records (EMRs) at the three university hospitals and stored as a CSV file. The patients were classified into nine groups: combination of all generations/elderly (more than 65 years of age) men/women, non-elderly (20-65 years) men/women and men/women of all ages. eGFR was stratified into six categories: <15, [15,30], [30,45], [45,60], [60,90], and >90%. HbA1c was stratified into five categories: <6.2, [6.2,6.9], [6.9,7.4], [7.4,8.4], and 8.4 >. To assess the performance of the secure computing, we built a matrix in which the horizontal axis was the eGFR interval and the vertical axis was the HbA1c interval. The basic statistics (sum, frequency, mean, variance, minimum, maximum, and median) of Cr were calculated for every cell in the matrix. Using the HbA1c intervals, ANOVA and Kruskal–Wallis analysis were conducted with respect to UP, severity of CKD, and eGFR. Pearson's correlation, of HbA1c and eGFR, and Spearman's rank correlation coefficient, of HbA1c and UP, were also calculated. To verify the accuracy of the results of the secure computing, we performed the same statistical analysis with R and SAS software, and compared the results with those of the secure computing. The primary goal was to confirm that the secure computing gave results equivalent to those obtained using the standard statistical software, and to confirm that the processing time was practical.

3. Results

Table 1 shows the bandwidth and delays among the three nodes. The average performance of disk input/output (I/O) at the nodes was 440.94 MB/sec. The EMRs at the three university hospitals contained data on 33,552 patients (Table 2). The combination of patient groups, eGFR interval and HbA1c resulted in $6 \times 6 \times 5 \times 11 = 1{,}980$ ways. Table 3 shows the results of the analysis of all patients, as a representative result. The processing time of the Pearson analysis was 19 s, the total communication among nodes was 14Mb. The respective values were 30 s, 42 MB for the ANOVA; 942 secs, 39 GB for the Spearman's analysis; and 619 s, 23 GB for the Kruskal–Wallis analysis. The SDC, with the participation of $n = 3$ nodes, was properly performed for every pair of nodes. We confirmed that the threshold rule (t=10) was applied for the cases with eGFR > 90%, and HbA1c $\geq$ 8.4. The results generated using our system were consistent with those obtained using the R and SAS programs, after rounding the result to the output digits of R and SAS, respectively, and we thus confirmed that our program is valid for the use in a statistical processing environment.

4. Discussion

Statistical analysis of the results of non-Spearman's correlation and Kruskal–Wallis tests requires frequent but small amount of communications among the nodes, so the overall processing time largely corresponds to the communication delays. In comparison, node traffic with the Spearman's correlation and Kruskal–Wallis tests exceeds the numbers of

GB, indicating that a high-speed network is a critical requirement for an efficient and secure computing framework. In this study, we did not compare the performance archived using our framework with the theoretical performance of the secure computation. For computation of the Spearman's rank correlation coefficient and Kruskal–Wallis tests, the amount of communication among nodes was on the order of $\mathbf{O}(n \log n)$ with respect to the data number n. To improve the performance of secure the computing framework, we need to investigate the performance of the memory read/write, disk access, and local computation times to identify the optimal cut-off in the network-local computing trade off. Currently, every node holds data in a CSV file, and the data must be manually extracted to the CSV files form the EMRs. In the future, we will implement an interface for i2b2 using our framework, and add the ability to obtain a request for the statistical analysis and extract the data from the i2b2 repository simultaneously. In this study, we implemented a protocol for secure computing that was resilient against collusion for up to n-1 nodes, and as well as an SDC control feature. Our framework protects against the data breaches made by the node administrator and suppresses the identification of specific individuals by an analyst. Moreover, the framework can securely process statistical data analysis that require microdata and complete this process within a reasonable timeframe.

Acknowledgement

This study was supported by research funding from NTT and was designed by the academic researchers; the sponsor had no control over the management of this study. The corresponding authors had final responsibility for the decision to submit the manuscript for publication.

References

[1] Bellazzi R, Diomidous M, Sarkar IN, Takabayashi K, Ziegler A, McCray AT. Data analysis and data mining: current issues in biomedical informatics. Methods of information in medicine. 2011;50(6):536.
[2] Weber GM, Murphy SN, McMurry AJ, MacFadden D, Nigrin DJ, Churchill S, et al. The Shared Health Research Information Network (SHRINE): a prototype federated query tool for clinical data repositories. Journal of the American Medical Informatics Association. 2009;16(5):624-30.
[3] Kamm L, Bogdanov D, Laur S, et al. A new way to protect privacy in large-scale genome-wide association studies. Bioinformatics 2013;29:886–93.
[4] Chida K, Morohashi G, Fuji H, Magata F, Fujimura A, Hamada K, Ikarashi D, Yamamoto R. Implementation and evaluation of an efficient secure computation system using 'R' for healthcare statistics. J Am Med Inform Assoc. 2014;21:326-331.
[5] Ben-Or M, Goldwasser S, Wigderson A. Completeness theorems for non-cryptographic fault-tolerant distributed computation, STOC '88, ACM Press 1988;1-10.
[6] Chaum D, Crepeau C, Damgard I. Multiparty unconditionally secure protocols, STOC '88, ACM Press 1988;11-19.
[7] Damgård I, Keller M, Larraia E, Pastro V, Scholl P,et al.. Practical covertly secure MPC for dishonest majority–or: Breaking the SPDZ limits. Computer Security–ESORICS 2013: Springer; 2013. p. 1-18.
[8] Keller M, Scholl P. Efficient, oblivious data structures for MPC. Advances in Cryptology–ASIACRYPT 2014: Springer; 2014. p. 506-25.
[9] Hamada K, Kikuchi R, Ikarashi D, Chida K, Takahashi K. Practically efficient multi-party sorting protocols from comparison sort algorithms. ICISC 2012: Springer; 2012. p. 202-16.
[10] Nishide T, Ohta K. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. Public Key Cryptography–PKC 2007: Springer; 2007. p. 343-60.