Implementation and evaluation of an efficient secure computation system using 'R' for healthcare statistics

Koji Chida,[1] Gembu Morohashi,[1] Hitoshi Fuji,[1] Fumihiko Magata,[1] Akiko Fujimura,[1] Koki Hamada,[1] Dai Ikarashi,[1] Ryuichi Yamamoto[2]

[1]Secure Platform Laboratories, NTT Corporation, Tokyo, Japan
[2]Department of Health Management and Policy, Graduate School of Medicine, University of Tokyo, Tokyo, Japan

Correspondence to
Dr Koji Chida, Secure Platform Laboratories, NTT Corporation, 3-9-11 Midori-cho, Musashino, Tokyo 180-8585, Japan; chida.koji@lab.ntt.co.jp

Received 6 January 2014
Revised 3 April 2014
Accepted 7 April 2014
Published Online First
24 April 2014

To cite: Chida K, Morohashi G, Fuji H, *et al*. *J Am Med Inform Assoc* 2014;**21**:e326–e331.

ABSTRACT

Background and objective While the secondary use of medical data has gained attention, its adoption has been constrained due to protection of patient privacy. Making medical data secure by de-identification can be problematic, especially when the data concerns rare diseases. We require rigorous security management measures.

Materials and methods Using *secure computation*, an approach from cryptography, our system can compute various statistics over encrypted medical records without decrypting them. An issue of secure computation is that the amount of processing time required is immense. We implemented a system that securely computes healthcare statistics from the statistical computing software 'R' by effectively combining secret-sharing-based secure computation with original computation.

Results Testing confirmed that our system could correctly complete computation of average and unbiased variance of approximately 50 000 records of dummy insurance claim data in a little over a second. Computation including conditional expressions and/or comparison of values, for example, t test and median, could also be correctly completed in several tens of seconds to a few minutes.

Discussion If medical records are simply encrypted, the risk of leaks exists because decryption is usually required during statistical analysis. Our system possesses high-level security because medical records remain in encrypted state even during statistical analysis. Also, our system can securely compute some basic statistics with conditional expressions using 'R' that works interactively while secure computation protocols generally require a significant amount of processing time.

Conclusions We propose a secure statistical analysis system using 'R' for medical data that effectively integrates secret-sharing-based secure computation and original computation.

BACKGROUND AND SIGNIFICANCE

As digitization of medical records advances and data processing platforms evolve—for example, with the migration to the cloud—the secondary use of medical records is gaining momentum. But because medical records involve the privacy of patients, we require some rigorous security management. For example, let us consider a case where medical records pertaining to rare diseases are collected from multiple medical institutes and placed in a cloud database. Medical researchers access the database to perform statistical analysis. In such a case, the medical records of individual patients stored in the database may be accessed inappropriately. If a data leak occurs, there is the risk of harm to the patients' privacy.

Security control measures for externally accessing databases include access controls and database encryption. Information identifying individual patients is not necessary for the purpose of statistical analysis. Therefore an effective method for protecting patient privacy is to store de-identified medical records in databases. However, risks and threats still remain, including inappropriate internal control, operational errors by administrators, as well as external attacks. Also, because in general encrypted data must be decrypted for use, the aforementioned risks and threats are still not completely eradicated. Furthermore, when it comes to de-identified data, the more unique the medical data, the easier it is to surmise the identity of the patient to whom the medical record belongs. Actually, it has been reported that individuals can be uniquely identified from de-identified medical data by comparing it with other data.[1–4]

In this research, we focus on *secure computation*,[5–7] which is capable of processing data in encrypted state, as the technology for making secondary use of medical databases. We then implement our system in software to evaluate its performance. Because the secure computation protects the data during use, it can provide high-level data privacy. However, this feature gives rise to the issue of tremendous processing times required compared with original computation. Thus, we focus on a *secret-sharing-based* secure computation to implement a versatile and fast system capable of computing various statistics. Secret sharing[8] is a technique of encrypting the original data into multiple pieces of data so that the original data cannot be reconstructed unless at least a certain number of pieces of data are obtained. In a well-known secret sharing scheme known as the (k,n)-threshold scheme,[8] where k and n are integers with value of 2 or greater satisfying $n \geq k$, n pieces (fragments) of encrypted data are created from the original data. If arbitrary k of n fragments are obtained, the original data can be reconstructed. However, the original data cannot be reconstructed even partially with less than k fragments. Therefore, in general the greater the value of k, the more difficult it is to inappropriately collect the fragments and reconstruct the data.

The system implemented in this research runs the statistical computing software 'R'[9] as a front-end application. The secret sharing scheme used is the (k,n)-threshold scheme, where the parameters k and n are arbitrary integers satisfying $(n+1)/2 \geq k \geq 2$. To measure performance, various statistics of approximately 50 000 records of dummy insurance claim data were computed using 'R' for $(k,n)=(2,3)$ and $(3,5)$. We confirmed that the statistics could be processed within a few minutes.

RELATED WORKS

Secure computation is an approach being researched in cryptography. The concept was introduced by Yao in 1986.[5] Assuming that two parties hold secret values x and y, respectively, Yao's protocol makes it possible to compute an arbitrary function f(x,y) without each party revealing its secret value to the other party. Ben-Or *et al*[6] and Chaum *et al*[7] then independently proposed secret-sharing-based secure computation schemes that allow function values to be computed when secret values are held by three or more parties without revealing the secret held by any party. To compute a function, both schemes require the parties' computers to communicate with one another. Thus this type of secure computation is also called *multi-party computation*.

From the beginning, an issue of multi-party computation is the tremendous processing time required compared with original processing. Research on improved processing time is still being vigorously pursued today. For example, Bogdanov *et al*[10] proposed a fast multi-party computation system using a simple secret sharing scheme that generates three fragments called *Sharemind*. Kamm *et al*[11] applied this system and demonstrated the genome-wide analysis of approximately 300 000 single-nucleotide polymorphisms belonging to roughly 1000 donors. Many development projects for implementing practical multi-party computation system such as SEPIA,[12] TASTY,[13] and VIFF[14] have been ongoing.

The multi-party computation has received considerable attention recently in the research field of *privacy-preserving data mining* (PPDM).[15–18] PPDM aims to obtain the mining (or statistical) result without violating privacy of persons who agreed their own personal data to be used. PPDM also focuses on distributed databases which allow computation of some aggregate statistics over the entire dataset without disclosing any dataset. The entire dataset can be seen either horizontally partitioned or vertically partitioned. In horizontally partitioned datasets, each dataset has the same attributes. Lindell and Pinkas constructed a decision tree induction method called ID3 without disclosing horizontally partitioned datasets. For horizontally partitioned datasets, many other approaches related to PPDM are proposed to realize various statistical analyses and data mining operations.[19–23] In vertically partitioned datasets, the same individuals are included in each dataset. Vaidya and Clifton proposed a privacy-preserving association rule mining in vertically partitioned datasets.[24] Related studies include Vaidya *et al*, Vaidya and Clifton, and Sanil *et al*.[25–27]

Meanwhile, secure computation research based on approaches different from multi-party computation is also being advanced. In particular, schemes based on *fully homomorphic* encryption, proposed by Gentry in 2009,[28] are receiving attention. The distinction of Gentry's scheme is its ability to allow an arbitrary third party to compute by itself a function with the encrypted input data remaining in encrypted state and return the true function value only to the owner of the decryption key. A software implementation of the scheme has been released[29]; however, currently the processing time is a major issue. To realize performance at a practical level for large-scale statistical analysis and complex processing, we would need further research breakthroughs. Also, MONOMI[30] and similar systems, which can process various requests (SQL commands) over an encrypted database, are at a near-practical stage. However, the operations that can be processed are limited to addition, determining equality, and comparing values. Each database in these systems also cannot be accessed from unspecified multiple people because the database is encrypted by a symmetric key, so they are not suitable for purposes such as integrated analysis.

Recently, El Emam *et al*[31] proposed a *secure linking protocol*, which allows for the exact matching of records among registries and the computation of statistics on the linked data, using a homomorphic encryption while keeping patient confidentiality and privacy. Statistics include OR and the χ^2 test. The protocol has acceptable computation time, for example a χ^2 test based on a four-cell cross tabulation for 50 000 patients can be evaluated within 1000 to 2000 s.

MATERIALS AND METHODS

Like Bogdanov *et al*'s [10] system, the secure computation system implemented in our research is made up of the following: an unspecified number of *Data Provider Clients*, which encrypt the original data into n fragments through a secret sharing scheme and send each fragment separately to a server; n *Secure Computation Servers*, which take the fragment as input and perform secure computation; and *Analysis Clients*, which issue requests for the desired operations to the n Secure Computation Servers to obtain the computation results. Unlike Bogdanov's system, however, our system allows for more than three Secure Computation Servers; data security can be increased by increasing the number of servers.

The original data, which serves as input to the Data Provider Clients, is expressed in table (CSV) format. Each row in the table represents an individual record. Each column contains the values of an attribute. The data is encrypted into n fragments by applying a secret sharing scheme on a cell data basis. The Analysis Clients receive computation result fragments from k Secure Computation Servers, and restore the computation result from k fragments. The basic operations that can be processed through secure computation are summation, sum of squares, logical operations (NOT, AND, OR), determining equality, comparing values, random shuffle of specified columns, and sorting of specified columns. The sample size can be obtained by counting the number of fragments in a column. The results of these basic operations are passed to the statistical computing software 'R', which is widely used for the analysis of medical statistics. By processing on 'R', the Analysis Clients can obtain the results of a variety of statistical methods. Details are described later.

The secret sharing scheme implemented by our proposed system is Shamir's (k,n)-threshold scheme,[8] a well-known method. Fragments obtained using Shamir's scheme are known to have the property of additive homomorphism. In other words, using x,y to represent fragments of the original data and E to denote the encryption function, $E(x)+E(y)=E(x+y)$. Thus the result of adding x+y with the original data in encrypted state can be determined by computing $E(y+y)$. In this way, secure computation of summation can be easily realized. Because subtraction and constant multiplication can be realized in a similar manner, if the original data is represented as 1-bit (true: 1, false: 0), then the logical operation of negation $x \rightarrow 1-x$ can be performed. The method of computing the multiplication of fragments x and y as they remain encrypted follows Gennaro *et al*'s [32] protocol. By combining this method with addition, sum of squares can also be securely computed. If the original data is 1-bit, logical operations AND: $x, y \rightarrow xy$ and OR: $x, y \rightarrow x+y-xy$ are also possible. For determining equality, we follow Cramer and Damgård's[33] protocol. With E(x) and E(y) as the input, this method enables the output of $E(b)$, where $b=1$ if $x=y$ and $b=0$ if not. Similarly, we follow Damgård *et al*'s [34] protocol to compare values. For random shuffle, we follow Laur *et al*'s[35] protocol. For sorting, we follow Hamada *et al*'s[36] protocol.

In addition, a major feature of our proposed system is the ability of the Analysis Clients themselves to find the results of desired statistical methods by combining (on 'R') the results of basic statistics and conditional expressions, without the need for the desired statistical results to be consistently processed through secure computation. This allows statistical techniques to be flexibly programmed on the Analysis Clients. Furthermore, because the Secure Computation Servers execute the minimum secure computation of operations, we can expect processing time to be reduced. For example, in our system an average value is computed as follows. First, the Secure Computation Servers compute the fragments of the summation of specified attribute values *sum* and the number of the values *num* and return them. Next, an Analysis Client obtains the average value by computing *sum*/*num* after reconstructing *sum*. Due to the help of Analysis Clients, the Secure Computation Servers can avoid the secure computation of division. Note that attribute values of specific individuals are disclosed neither to the Analysis Clients nor to the Secure Computation Servers even during statistical analysis. Also, since the Analysis Clients can get the average value *sum*/*num* only if it gets *sum* and *num* based on the (k,n)-threshold setting, the average value also satisfies (k,n)-threshold.

In summary, our proposed secure computation system has the following features: (1) A multi-party secure computation system using (k,n)-threshold sharing scheme is implemented. (2) The statistical programming language 'R', which is widely used for medical statistical analysis, can be used as the front-end application. (3) Minimum operations such as basic statistics and conditional expressions through secure computation without revealing the attribute values of specific individuals are executed. This reduces overall processing time compared with performing all operations through secure computation. (4) Users can flexibly make programs to perform statistical analysis using 'R' by combining the results of operations carried out by secure computation.

RESULTS

Using an electronic medical billing system's master lists of diseases, medical procedures, and drugs,[37] we created 50 001 dummy insurance claim data records. We then computed various statistical methods over the data and measured the performance. The number of records in the dummy insurance claim dataset was decided based on the assumption that it approximates the number of rare diseases-related data records in a month. For the sake of convenience, only a few values were used for the year and month of the medical procedures. The dummy insurance claim data has 38 attributes, as shown in table 1.

Execution of a command from the Data Provider Client causes the dummy insurance claim data to be secret shared on a cell data basis (total 50 001×38=1 900 038 cells). Each fragment is sent to a Secure Computation Server. The secret sharing process is executed by a call to a C++ library installed in the Data Provider Client. Secure computation operations that can be executed by commands from 'R' are average, unbiased variance, minimum value, maximum value, median, tabulation, cross tabulation, conditional expressions (=, >, ≥, <, ≤, !=) and t test. As described before, these operations are carried out by using 'R' to combine the results of securely computed basic operations. Each basic operation is executed by a call from 'R' to a C++ library installed in the Analysis Client. Conditional expressions can combine logical AND and OR. Secure computation commands and basic operations that can be executed from 'R' are shown in table 2. The overall scheme of the implemented system and a snapshot of commands executed from 'R' are shown in figures 1 and 2, respectively.

Table 1 Attributes of dummy insurance claim data

Name of attribute	Attribute type	Scope of value/number of categories
ID number	Number	[100000001, 100050001]
Sex	Category	2
Year of birth	Number	[1950, 1990]
Month of birth	Number	[1, 12]
Year of medical procedure	Number	[2008, 2011]
Month of medical procedure	Number	[1, 12]
Inpatient/outpatient	Category	2
Primary disease code	Category	41
Primary disease name	Category	41
Disease2 code	Category	18 363
Disease2 name	Category	18 363
Disease3 code	Category	20
Disease3 name	Category	20
Medical procedure1 code	Category	467
Medical procedure1	Category	467
Medical procedure2 code	Category	418
Medical procedure 2	Category	418
Medical procedure3 code	Category	417
Medical procedure3	Category	417
Medical procedure4 code	Category	417
Medical procedure4	Category	417
Medical procedure5 code	Category	1676
Medical procedure5	Category	1676
Drug1 code	Category	4175
Drug1	Category	4175
Dosage1	Number	[1, 84]
Drug2 code	Category	4176
Drug2	Category	4176
Dosage 2	Number	[1, 84]
Drug3 code	Category	4176
Drug3	Category	4176
Dosage 3	Number	[1, 84]
Drug4 code	Category	4176
Drug4	Category	4176
Dosage 4	Number	[1, 84]
Drug 5 code	Category	4036
Drug5	Category	4036
Dosage 5	Number	[1, 84]

The processing times of each command for the (k, n)-threshold sharing scheme shown in table 3. We evaluated the implemented system only when (k,n)=(2,3) and (3,5) due to the restriction of our current execution environment. The execution environment is shown in table 4. Note that the operations of minimum value, maximum value, and median are all executed in a similar manner: (1) securely computing ascending sort; and (2) restoring only the necessary column (column 1 for minimum value, column 50 001 for maximum value, and column 25 001 for median). Thus only the results of computing the median as the representative operation are shown.

Observations based on the results of table 3 are as follows.

Table 2 Secure computation commands and basic operations that can be executed from 'R'

Command	Mathematical operation	Basic operations	Data type applied to
sec.mean	Average	Summation, sample size	Number
sec.var	Unbiased variance	Summation, sum of square, sample size	Number
sec.min	Min value	Sort	Number
sec.max	Max value	Sort	Number
sec.median	Median	Sort	Number
sec.xtabs	Tabulation	Random shuffle	Category
sec.xtabs	Cross tabulation	Random shuffle	Category
sec.subset.eq	conditional expression (=)	Determine equality	Number, category
sec.subset.gt	Conditional expression (>)	Comparison of numbers	Number
sec.subset.ge	Conditional expression (≥)	Comparison of numbers, determine equality, OR	Number
sec.subset.lt	Conditional expression (<)	Comparison of numbers	Number
sec.subset.le	Conditional expression (≤)	Comparison of numbers, determine equality, OR	Number
sec.subset.ne	Conditional expression (!=)	Determine equality, NOT	Number, category
sec.ttest	t test	Summation, sum of square, sample size (average, unbiased variance, sample size)	Number

Data registration: The time required for secret sharing of data consisting of approximately 1.9 million cells was less than 10 s for (k,n)=(2,3) and a little over 20 s for (k,n)=(3,5). For (k,n)=(2,3), the total size of the fragments was approximately three times the size of the original data; for (k,n)=(3,5), it was approximately five times. However, because the original data size is about 15 MB, the length of time involved in secret sharing, including sending time, is considered to be in real-time. Theoretically, the total size of fragments and the computational cost required to generate the fragments increases in proportion to n.

Basic statistics (average, unbiased variance, median): Average and unbiased variance could be computed in 1–2 s, resulting in near-instantaneous results. Computing the median required time for sorting through secure computation, is a basic operation. For (k,n)=(2,3), slightly more than 1.5 min was required, whereas for (k,n)=(3,5) it took slightly more than 6 min, just less than a fourfold difference. Even with the more secure case of (k,n)=(3,5), the system can securely compute some basic statistics with conditional expressions using 'R' that works interactively while secure computation protocols generally require a significant amount of processing time. Theoretically, the total computation and communication costs required to compute a summation, which is the basic operation of average value, increase in proportion to k. For computing a square (or multiplication), which is a basic operation of unbiased variance, the total computation and communication costs increase roughly in proportion to n′(n′−1), where n′=2k−1. For computing a sort, which is a basic operation of median, the total communication cost increasing roughly in proportion to $(2^{n'}/\sqrt{n'})n'^2$ becomes bottleneck.

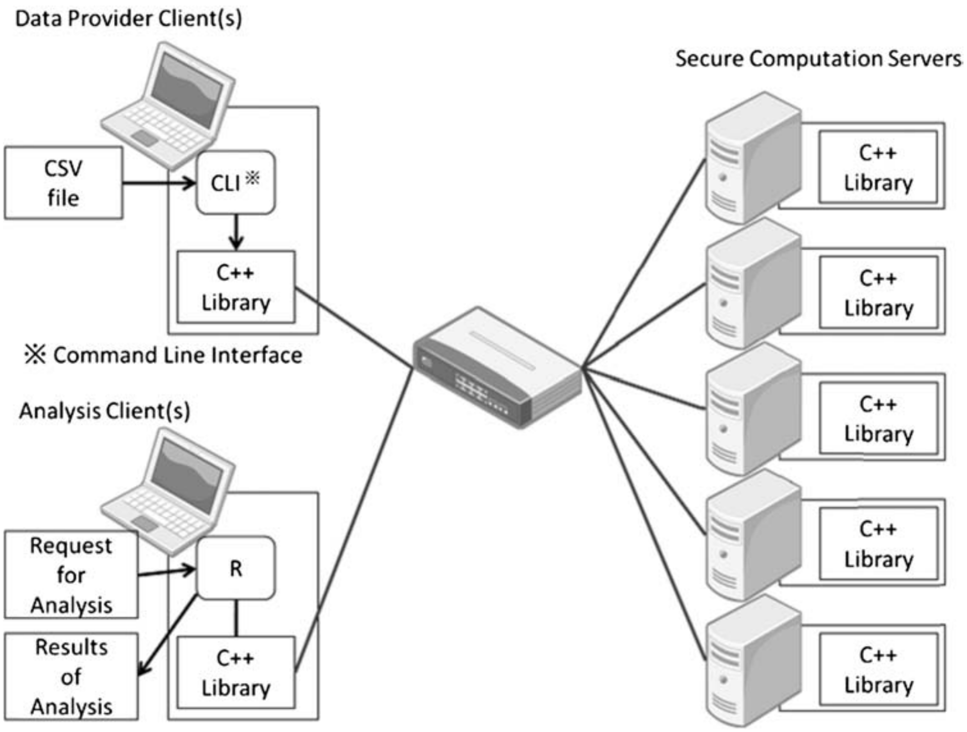
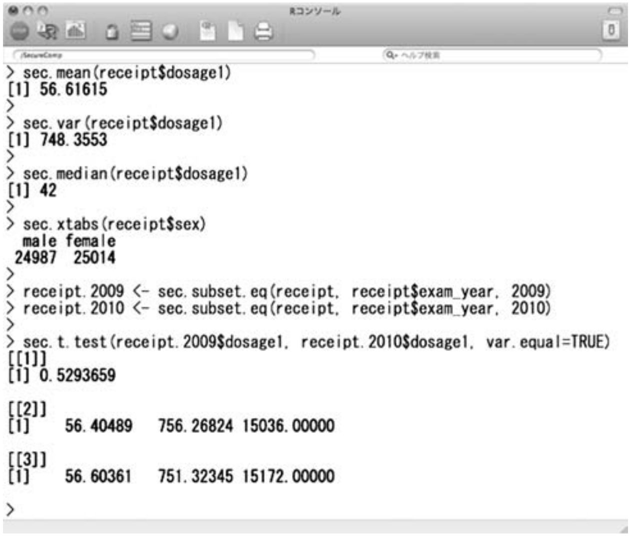


Figure 1 An overview of the proposed system.



```
> sec.mean(receipt$dosage1)
[1] 56.61615
>
> sec.var(receipt$dosage1)
[1] 748.3553
>
> sec.median(receipt$dosage1)
[1] 42
>
> sec.xtabs(receipt$sex)
  male female
 24987  25014
>
> receipt.2009 <- sec.subset.eq(receipt, receipt$exam_year, 2009)
> receipt.2010 <- sec.subset.eq(receipt, receipt$exam_year, 2010)
>
> sec.t.test(receipt.2009$dosage1, receipt.2010$dosage1, var.equal=TRUE)
[[1]]
[1] 0.5293659

[[2]]
[1]     56.40489    756.26824 15036.00000

[[3]]
[1]     56.60361    751.32345 15172.00000

>
```

Figure 2 A snapshot of command executions on 'R' interface.

Conditional expressions: Determining equality for (k,n)=(2,3) took slightly more than 10 s; for (k,n)=(3,5) it took roughly double the time. For comparing values, (k,n)=(2,3) took slightly more than 45 s; for (k,n)=(3,5) the processing time was roughly double. However, it was less than the time for computing median, so the performance can be considered to be sufficient for 'R' in an interactive mode. The main basic operation for equality and comparison checks is multiplication. Therefore, the total computation and communication costs required to check equality and comparison can be estimated to increase roughly in proportion to $n'(n'-1)$.

Tabulation: For tabulation and cross tabulation, secure computation of random shuffle is executed, and its results are tabulated by 'R'. Results of both operations could be obtained in a few seconds. The total computation and communication costs required to execute a random shuffle can be estimated to increase roughly in proportion to $n'(n'-1)$.

Table 3 Processing time for each type of operation

	Processing time (unit: seconds)	
Type of operation	**(k,n)=(2,3)**	**(k,n)=(3,5)**
Secret sharing	9.70	22.51
Average [dosage1]	1.19	1.43
Unbiased variance [dosage1]	1.28	1.54
Median [dosage1]	92.39	345.64
Condition [dosage1=10]	12.57	24.00
Condition [year of medical procedure=2009 OR year of medical procedure=2010]	21.38	43.74
Condition [dosage1<10]	45.82	98.15
Condition [dosage≥10 AND<25]	85.99	190.26
Tabulation [primary disease name]	1.91	2.88
Cross tabulation [primary disease name×sex]	3.12	5.10
t test		
Group A←conditional expression [year of medical procedure=2009]	11.98	23.61
Group B ← conditional expression [year of medical procedure=2010]	11.87	23.52
Test [group A's dosage, group B's dosage]	1.64	2.20
Total	25.49	49.32

Table 4 Execution environment

Data provider client, analysis client	
CPU	Intel core i5 1.7 GHz×2 cores
OS	Mac OS X Snow Leopard
RAM	4 GB
'R'	V.3.0.0
Secure computation server 1, 2, 3	
CPU	Intel core i7-2640M 2.8 GHz×4 cores
OS	Ubuntu 12.04
RAM	8 GB
Secure computation server 4, 5	
CPU	Intel core i5-2540M 2.6 GHz×4 cores
OS	Ubuntu 11.10
RAM	8 GB
NW	1 Gbps NW switch
C++ compiler	g++ V.2.4.1

Tests: A sample 'R' script for t test was created by combining basic statistics and conditional expressions. Overall, the t test for (k,n)=(2,3) took slightly more than 25 s; it was less than 50 s for (k,n)=(3,5), a less than twofold difference. In either case, the results could be obtained in less than 1 min. As stated above, because cross tabulation and median can be computed within real-time limits, tests using the results of cross tabulation, such as the χ^2 test and the Mann–Whitney U test, would also be able to be carried out by 'R' in an interactive mode. In particular, the χ^2 test can be computed using the result of cross tabulation.

DISCUSSION

Benefits and remaining limitations: If medical records are simply de-identified and encrypted in a normal fashion, the risk of leaks exists because decryption is usually required during statistical analysis. If the de-identified but unencrypted medical records are leaked out, the risk of re-identification occurs.[1–4] Our system possesses high-level security because medical records remain in encrypted state even during statistical analysis. Also, our system can securely compute some basic statistics with conditional expressions using 'R' that works interactively while secure computation protocols generally require a significant amount of processing time. For example, a cross tabulation, which is a basic statistical operation to obtain the result of a χ^2 test, can be computed within a few seconds while the existing systems require much more time (a χ^2 test for 1080 donors takes several tens of seconds in the genome-wide analysis system in Kamm *et al*[11] and that for 50 000 patients takes 1000 to 2000 s in El Emam *et al*'s[31] secure linking system).

On the other hand, in our system remains at risk of 'collusion attack' where fragments of the de-identified data are inappropriately gathered from k or more corrupted secure computation servers to reconstruct the de-identified data. If the incident happens, we are back to the basic solution by simple de-identification and access control.

We are working to further improve processing time by fine-tuning secure computation algorithms and system implementation. We also plan to extend the 'R' language for statistical analysis. Remaining issues to be addressed going forward include creating access control methods to prevent attacks that aggregate the results of secure computation to surmise particular individuals, and devising appropriate methods for operating the secure computation server to prevent attacks where fragments of data are inappropriately gathered to reconstruct the data of individuals.

CONCLUSION

We proposed a secure and efficient statistical analysis system for medical data. Our system is capable of computing various statistical methods from the statistical computing software 'R' through the secure computation based on a (k,n)-threshold secret sharing scheme. We then evaluated the system's performance using approximately 50 000 records of dummy insurance claim data. We confirmed that the system could process statistical analysis within practical time limits so that 'R' works interactively. The special feature of this system is its ability to execute the most basic statistical operations through secure computation without revealing the data of specific individuals, and perform statistical analysis by using 'R' scripts. This allows users to flexibly define statistical analysis functions, and at the same time reduce secure computation's processing load. Our system demonstrated the completion of statistical methods within a few minutes.

Contributors KC, GM, HF, KH, and DI developed the proposed secure statistical analysis system using 'R'. KC, GM, FM, AF, and RY evaluated the proposed system from a practical viewpoint of healthcare statistics.

Funding This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Competing interests None.

Provenance and peer review Not commissioned; externally peer reviewed.

REFERENCES

1 Sweeney L. Uniqueness of simple demographics in the U.S. population. LIDAP-WP4 Carnegie Mellon University, 2000.
2 Samarati P. Protecting respondents' identities in microdata release. *IEEE Trans on Knowledge and Data Engineering* 2001;13:1010–27.
3 Narayanan A, Shmatikov V. How to break anonymity of the Netflix prize dataset. arXiv:cs/0610105 22 November 2007. http://arxiv.org/abs/cs/0610105v1
4 Arrington M. AOL proudly releases massive amounts of private data. (Last updated 08/06/2006; cited 01/04/2014). http://techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/
5 Yao AC. How to generate and exchange secrets. *IEEE Symposium on Foundations of Computer Science*; 1986:162–7.
6 Ben-Or M, Goldwasser S, Wigderson A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. *ACM Symposium on Theory of Computing*; 1988:1–10.
7 Chaum D, Crepeau C, Damgård I. Multiparty unconditionally secure protocols. *ACM Symposium on Theory of Computing*; 1988:11–9.
8 Shamir A. How to share a secret. *Communications of the ACM* 1979;22:612–3.
9 The R Project for Statistical Computing. (Last updated 09/25/2013; cited 01/04/2014). http://www.r-project.org/
10 Bogdanov D, Laur S, Willemson J. Sharemind: A framework for fast privacy-preserving computations. *European Symposium on Research in Computer Security*; 2008:192–206.
11 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.
12 Burkhart M, Strasser M, Many D, *et al*. Sepia: Privacy-preserving aggregation of multi-domain network events and statistics. *USENIX Security Symposium*; 2010:223–40.
13 Henecka W, Kögl S, Sadeghi AR, *et al*. Tasty: Tool for automating secure two-party computations. *ACM Conference on Computer and Communications Security* 2010:451–62.
14 Geisler M. Cryptographic Protocols: Theory and Implementation. PhD thesis. University of Aarhus, 2010.
15 Lindell Y, Pinkas B. Privacy preserving data mining. *International Cryptology Conference CRYPTO* 2000:20–4.
16 Agrawal R, Srikant S. Privacy-preserving data mining. *ACM's Special Interest Group on Management of Data* 2000:439–50.
17 Aggarwal CC, Yu PS. *Privacy-preserving data mining: Models and algorithms*. Springer, 2008.
18 Vaidya J, Clifton CW, Zhu YM. *Privacy preserving data mining*. Springer, 2005.
19 Kantarcioglu M, Vaidya J. Privacy-preserving naïve Bayes classifier for horizontally partitioned data. *IEEE Workshop on Privacy Preserving Data Mining*; 2003.
20 Yu H, Jiang X, Vaidya J. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. *SAC Conference*; 2006:603–10.
21 Kantarcioglu M, Clifton C. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE TKDE J* 2004;16:1026–37.
22 Karr AF, Fulp WJ, Vera F, *et al*. Privacy-preserving analysis of distributed databases. *Technometrics* 2007;49:335–45.
23 Karr AF, Lin X, Sanil AP, *et al*. Secure regression on distributed databases. *J Comput Graph Stat* 2005;14:263–79.
24 Vaidya J, Clifton C. Privacy-preserving association rule mining in vertically partitioned databases. *ACM SIGKDD Conference*; 2002:639–44.
25 Vaidya J, Clifton C, Kantarcioglu M, *et al*. *Privacy-preserving decision trees over vertically partitioned data*. *ACM TKDD* 2008;2:1–27.
26 Vaidya J, Clifton C. Privacy-preserving naive Bayes classifier for vertically partitioned data. *The VLDB Journal* 2008;17:879–98.
27 Sanil AP, Karr AF, Reiter JP. Privacy preserving regression modelling via distributed computation. *ACM SIGKDD Conference*; 2004:677–82.
28 Gentry C. Fully homomorphic encryption using ideal lattices. *ACM Symposium on Theory of Computing* 2009:169–78.
29 shaih/HElib. (Last updated 01/03/2014; cited 01/04/2013). https://github.com/shaih/HElib
30 Tu S, Kaashoek M, Madden S, *et al*. Processing analytical queries over encrypted data. *Proceedings of the VLDB Endowment* 2013;Vol 6:289–300.
31 El Emam K, Samet S, Hu J, *et al*. A protocol for the secure linking of registries for HPV surveillance. *PLoS ONE* 2012;7:e39915–0.
32 Gennaro R, Rabin MO, Rabin T. Simplified VSS and fact-track multiparty computations with applications to threshold cryptography. *ACM Symposium on Principles of Distributed Computing* 1998:101–11.
33 Cramer R, Damgård I. Secure distributed linear algebra in a constant number of rounds. *International Cryptology Conference CRYPTO*; 2001:119–36.
34 Damgård I, Fitzi M, Kiltz E, *et al*. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. *Theory of Cryptography Conference*; 2006:285–304.
35 Laur S, Willemson J, Zhang B. Round-efficient oblivious database manipulation. *Information Security Conference*; 2011:262–77.
36 Hamada K, Ikarashi D, Chida K, *et al*. A linear time sorting algorithm on secure function evaluation. *Symposium on Cryptography and Information Security* 2011. (In Japanese)
37 Various information on Medical Fee. (Last updated 12/26/2013; cited 01/04/2014). http://www.iryohoken.go.jp/shinryohoshu/receMenu/doReceInfo (In Japanese)