MEDINFO 2007
HISA Ltd Australia
© 2007 The authors. All rights reserved.

Agent-Based Secure Aggregation of Distributed Health Data

Zim Chan[a,b], Peter Croll[a], Anthony Maeder[b], David Hansen[b]

[a] *Faculty of Information Technology, Queensland University of Technology, Australia*
[b] *CSIRO E-Health Research Centre, Australia*

Abstract

A common problem in the health sector is in balancing the need for patient health information against the risks posed by releasing such private information. The idea of Secure Multi-party Computation (SMC) provides potential ways in which this can be mitigated, by allowing for analysis of data distributed amongst multiple parties to take place without the need to disclose it. This paper presents a two-party SMC protocol which can be used to calculate some basic aggregate functions. In addition, we discuss a rationale for using an agent-based framework to utilize SMC techniques in order to support privacy-preserving access to health information.

Keywords:

data linkage, patient data privacy, data security, secure multi-party computation

Introduction

Consider two separate databases, Diagnoses and Prescription, containing pre-linked patient information about illnesses and drug prescriptions respectively. Suppose that as part of a study into drug adverse effects, a researcher wishes to find a count of people who were diagnosed with cancer less than 3 years after being prescribed with drug X.

Normally, this would require the release of potentially sensitive information – for instance, it would be necessary to release the cancer diagnosis date and drug prescription date in order to be able to perform some meaningful comparison.

This paper presents a two-party SMC protocol for computing a scalar product on *privately selected* elements of private vectors, while ensuring that:

- no party discovers which elements in the vector are used for the aggregation
- no party discovers the values of any individual data, or the partial aggregates at each database

Such a protocol can be used to calculate basic aggregates such as counts and weighted/un-weighted sums. In addition, we discuss how this protocol could be extended towards the calculation of other statistics such as average, standard deviation and covariance in a true (i.e., >2 participants) multi-party scenario.

Finally, we discuss the rationale for an agent-oriented framework to support SMC operations for providing aggregate processing of distributed health information.

Method

The protocol, shown in Algorithm 1, is based on using a series of SMC primitives, whereby the output of each step is fed into the input of the next phase. Standard cryptographic notions of secret-sharing and encryption ensure that while these intermediate values can be used in their successive stages, they do not leak information.

Step 1 of the protocol makes use of general SMC theory, which states that *any* function which can be represented as a combinatorial circuit can be evaluated securely [1]. What prevents the execution of the entire protocol in this manner is that the circuit size is polynomial with respect to input size and complexity. In the case of Step 1, the function is expected to be a relatively simple computation. For the example, it is necessary to reduce the following predicate to circuit form:

(has_cancer == TRUE & uses_drug_x == TRUE & diagnosis_date > prescription_date & (diagnosis_date + 365*3) < prescription_date).

While SMC remains mainly in the theoretical domain, implementations such as Fairplay [2] exist that allow such high level expressions to be compiled and executed.
In Step 2, Alice and Bob engage in a 1-2 Oblivious Transfer protocol using their secret shares obtained in Step 1 for the input. An Oblivious Transfer ensures that Alice cannot determine which message Bob has received. As it is encrypted, Bob is similarly unable to determine which message is received.
The messages are aggregated in Step 3 – By exploiting the properties of a homomorphic public-key cryptosystem, Bob performs a blindfold evaluation of the scalar product in the same manner as used in [3]. Note that, if the predicate P evaluated to 0 (or false), then the value received by Bob, $c_{i,b_i} = E(0,r)$, and Bob is essentially multiplying a zero payload.

Private Input: Databases DbA, DbB owned by Alice and Bob respectively, where DbA_i, DbB_i is the data corresponding to patient i, $|DbA| = |DbB| = N$, and $DbA_i.attr$ refers to the attribute *attr* in DbA.

Public Input: Predicate function $P(DbA_i, DbB_i) \rightarrow \{0,1\}$

Private Outputs: Selective scalar product as secret shares

$$S_A + S_B = \sum_{i=0}^{N} DbA_i.x \cdot DbB_i.y \cdot P(DbA_i, DbB_i)$$

0. Setup. Alice generates a public/private keypair (E, D) for a homomorphic cryptosystem

1. Selection. For each $i \in \{1 \ldots N\}$ Alice and Bob evaluate $P(DataA_i, DataB_i)$ via circuit-based SMC evaluation. Alice obtains $a_i \in \{0,1\}$ and Bob obtains $b_i \in \{0,1\}$ such that $a_i \oplus b_i = P(DbA_i, DbB_i)$

2. Vector Transfer: For each $i \in \{1 \ldots N\}$:

a. Alice sets $c_{i,1-a_i} = E(DbA_i.x, r)$ and $c_{i,a_i} = E(0, r)$ for some random r

b. Alice and Bob execute a 1-2 Oblivious Transfer whereby Alice sends the pair ($c_{i,0}, c_{i,1}$) and Bob selects c_{i,b_i}.

3. Multiplication: Bob sets $w = \prod_{i=1}^{N} c_{i,b_i}^{DbB_i.y}$

Algorithm 1 – Private Selective Scalar Product

Finally, Bob blinds the final product with a random, before returning it to Alice, who can decrypt it. At this stage, both parties hold secret shares of the final output.

In its simplest form, Alice and Bob can set $x, y = 1$ to obtain secret shares of the count of elements.

Extensions

As the output of the protocol is a series of secret shares, it lends itself to a number of natural extensions. Principally, the idea is that the protocol can be run a number of times with x values, and the outputs of each run can then be combined via general SMC circuit evaluation. This has two major implications:

- It can easily be extended to a *true* multi-party scenario with greater than two participants, for instance, in the case where the data is further distributed horizontally (e.g. a scalar product over 3 vectors) or vertically (if a vector is partitioned amongst two parties), or a hybrid mix.
- Based on basic primitives, the protocol can be used to calculate other aggregates such as average (count, sum), standard deviation(sum of squares, average), and covariance (scalar product, average).

An agent-based framework

It is proposed to use software agents – autonomous, proactive, social software entities – towards realising a privacy-preserving framework utilizing SMC techniques on behalf of researchers and database custodians.

The use and execution of SMC protocols is a non-trivial task. They are interactive in nature, requiring databases to become active participants in the information gathering process. Agents are a natural paradigm for an environment involving distributed, complex interactions with multiple points of control [4].

Agents could provide the following functionality:

- Decomposition of researcher queries into SMC primitives
- Dynamic formation of coalitions for query answering, and role fulfillment
- Execution of SMC protocols
- Adapting to intermediate computation results.

Conclusion

Secure multi-party computation protocols, such as the one presented, offer potential ways in which private health data can be analysed without requiring its disclosure. An agent-based framework offers promising ways in which they can be effectively applied. Currently, such a framework is undergoing implementation in the Java Agent Development Framework (JADE) [5].

References

[1] Goldreich, Oded, Micali, Silvio and Wigderson, A., How to play ANY mental game, in: STOC '87: Proc. of the nineteenth annual ACM conference on Theory of computing, pages 218-229, ACM Press, New York, New York, United States, 1987.

[2] Malkhi, Dahlia, Nisan, Noam, Pinkas, Benny and Sella, Yaron, Fairplay - Secure Two-Party Computation System, in: Proc. USENIX Security Symposium, pages 287-302, San Diego, CA, USA 2004.

[3] Bart Goethals, Sven Laur, Helger Lipmaa and Taneli Mielikäinen. On Private Scalar Product Computation for Privacy-Preserving Data Mining, in ICISC 2004, vol. 3506 of Lecture Notes in Computer Science, pages 104-120, 2004.

[4] Jennings, Nicholas, Sycara, Katia and Wooldridge, Michael, A Roadmap of Agent Research and Development, in: Autonomous Agents and Multi-Agent Systems, vol. 1(1), pages 7-38, ISSN 1387-2532, 1998.

[5] Bellifemine, F., Rimassa, G., Poggi, A., JADE - A FIPA-compliant Agent Framework. In Proc. of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, pages 7-38, London, UK 1999.

Address for correspondence:

email: zim.chan@csiro.au
post: EHRC, PO Box 10842, Adelaide St. Brisbane Qld 4000