MS Thesis Proposal

Achieving Symmetric Private Information Retrieval via Oblivious Transfer

Laura Beth Lincoln
Rochester Institute of Technology
Department of Computer Science
lbl6598@cs.rit.edu

October 9, 2004

Stanisław Radziszowski - *Committee Chair*

??? - *Reader*

Abstract

There exist many situations where a user may wish to access a database without any record of what was viewed, or found, being recorded. This is solved by Private Information Retrieval. An example being Google's newly released e-mail program GMail, the server scans each e-mail for matching textual advertisements, the scanning process can be viewed as a user query. Some users may refrain from using this new e-mail client because someone with access to the server, or the website of a resulting ad, could easily infer the content of private e-mails based on the matching textual ads.

There also exist situations in which the database should also have a level of privacy, and this is solved with Symmetric Private Information Retrieval. Two or more medical institutions may wish to mine for trends amongst the union of each institution's data. Since all medical records are confidential it is important to be able to query the databases without the results from a specific institution per query being known and to also not allow the data mining process to retrieve more information than that received from the queries posed and from information that could be inferred from the results of the queries.

Oblivious Transfer is a cryptographic primitive that can retrieve data from a sender without the sender revealing anymore data other than the data requested and without the sender knowing which piece of data was revealed to the receiver. This primitive can achieve Symmetric Private Information Retrieval. The forthcoming thesis should review past works in the area and then should expand upon these works through implementation and experimentation.

Overview and Definitions

There exist many databases, both public and private, and there exist several users that may be hesitant to obtain information from these databases for various reasons involving the perseverance of the user's privacy. Knowledge of the data retrieved by a party could cause such hazards as unwanted profiling. Given a scenario with a public database, a Private Information Retrieval (PIR) protocol will provide the user a means to retrieve data without revealing his query.

Private Information Retrieval (PIR) [GIKM98] Allows a user to retrieve information from a database while maintaining the privacy of his query.

An example, Google's GMail, where a user's e-mail is scanned, and textual ads are retrieved from Google's database based on the text of the e-mail. The e-mail text can easily be viewed by a database administrator, and if the user pursues the resulting ads, the owners of these sites could potentially infer the text of a user's e-mails. Considering the e-mail text as the private query, the textual ads as the result of the private query, and Google's database of textual ads as public, this scenario becomes a PIR problem.

When the privacy of the database must also be maintained, a Symmetric Private Information Retrieval (SPIR) protocol can achieve this functionality.

Symmetric Private Information Retrieval (SPIR) Allows a user to retrieve information from a database in a manner that preserves the privacy of the users query and the privacy of the database.

The privacy of the data is dependent upon the functionality of the particular database. A private database may restrict the user to a specific number of results based on clearance or subscription level or to a variable number of results dependent upon the query. This functionality is first discussed by Gertner, et. al [GIKM98].

In 1981, Rabin introduced, the cryptographic primitive, Oblivious Transfer [Rab81], which has also been presented by Yao as Oblivious Circuit Evaluation [Yao86]. SPIR, by definition, can be achieved by the use of Oblivious Transfer.

Oblivious Transfer (OT) In the traditional two-party scenario, a sender A has n records of data and a receiver B would like to receive k of those records provided the following hold:

- A can be guaranteed that B does not receive any information about the n records A has beyond the k records B is allowed to receive.
- B can be guaranteed that A does not know which of the k records of the n possible records B received.

The simplest implementation of an OT protocol would be in the ideal model.

Ideal Model The ideal model, in the two-party scenario, consists of the two parties that wish to communicate and a trusted third party. The two parties will send their inputs to the trusted third party. The trusted third party will then perform the computation and send the appropriate output to each party. In the event that the trusted third party detects cheating, an error will result.

While the ideal model is simple, it requires the presence of a trusted third party. Two parties that wish to share information obliviously may not trust any third party. The ideal model is not really an option to explore. The purpose of the ideal model is to provide a basis for what protocols in the real scenario must achieve.

Goldreich, in [Gol04], presented an OT protocol using enhanced one-way trapdoor permutations. Brassard and Crépeau developed Zig-Zag functions in [BCS96]. Zig-Zag functions were based on self-intersecting codes, and since self-intersecting codes are understood relatively well, Zig-Zag function based OT protocols should be reasonable. Later, Brassard and Crépeau, in [BC97] considered Privacy Amplification as a basis for an OT protocol. The Privacy Amplification based protocol provided a more computationally efficient OT protocol compared to previous implementations at the cost of an exponentially small probability of privacy loss. Pinkas has produced a significant amount of literature pertaining to OT and privacy preserving data mining [NP01, Pin02]. His contributions also include Oblivious Polynomial Evaluation (OPE) [NP99]. An OPE protocol is claimed to be useful in solving the database intersection problem. Other extensions include Conditional Oblivious Transfer [COR99] and Committed Oblivious Transfer[CvdGT95].

The majority of past constructed Oblivious Transfer protocols are secure with in the context of the semi-honest model.

Semi-Honest Model [Gol04] The semi-honest model consists of semi-honest parties. Loosely speaking, a semi-honest party is one who follows the protocol properly with the exception that it keeps a record all its intermediate computations.

The semi-honest model requires the parties to trust that neither deviates from the protocol. It is most likely sufficient to assume that neither party will deviate from the protocol, if the software through which the user and database communicate is closed source and distributed by a trusted third party, but the protocol should not rely on the existence of a trusted third party; therefore, the protocol should force the parties to not deviate from the protocol.

Malicious Model [Gol04] The malicious model consists of one or more malicious parties which may only deviate from the protocol in the following three ways:

- A party may refuse to participate in the protocol when the protocol is first invoked.

- A party may substitute his local input by entering the protocol with an input other than the one provided to him.
- A party may abort the protocol prematurely

The protocol should yield it impossible for the malicious parties to deviate from the protocol in any way not mentioned above.

Literature which presented a semi-honest model OT protocol, usually failed to produce a malicious model construction without oracle-aided sub-protocols.

Functional Specification

Oblivious Transfer (OT) has been constructed for the oblivious transfer of data ranging from bits to strings to records. Since it is possible to transfer large data objects via several invocations of smaller data object transfers (ie. sending a string bit by bit), previously constructed OT protocols, concerning any size of data transfer, should be reviewed. The results of such review should lead to the analysis and comparison of previously constructed OT protocols with respect to construction, communication, and computational efficiency versus level of privacy preserved.

A Symmetric Private Information Retrieval (SPIR) protocol can be seen as an OT protocol extended by the introduction of querying ability. There are at least two approaches to extending an OT protocol with querying ability, either a private query protocol can be performed to produce the static address (index) of the desired result followed by using the resulting address in a traditional OT protocol, or the private query process can be interleaved within the OT protocol. Dmitri Asonov's Doctoral Thesis [Aso04] should serve as a starting point and foundation for construction of protocols that maintain the privacy of queries.

Goldreich [Gol04] notes three functionalities that must exist to force semi-honest behavior in the malicious model: input commitment, augmented coin tossing, and authenticated computation. Since semi-honest behavior cannot be guaranteed, these functionalities should be briefly reviewed for previous use in practical application. Partial functionality should improve security.

After the above review of OT, PIR, SPIR, Private Query protocols, and functionalities to force semi-honest behavior, constructed PIR and SPIR protocols should implementable. Protocols in the forthcoming thesis should not be reliant on oracle-aided protocols [Gol04] or a secure coprocessor [**?**]. There is a possiblity of the protocols being constructed to use multiple general pupose processors though. Additionally, both PIR and SPIR protocols should be constructed using OT protocols that transfer various amounts of data in a single invocation. The implemented protocols should also be implemented with varying degrees of privacy leakage within the malicious model. All constructed protocols should be secure in the semi-honest model. The constructed protocols should be assessed for expected construction, communication, and computational efficiency versus level of privacy maintained within the malicious model before implementation

for comparison with experimental results. Implementations will be used to privately submit a query and retreive the appropriate information from a chosen database, yet to be determined. With consideration to the constructed SPIR protocols, the privacy of the database will be accounted for in addition to the privacy of the user.

As stated above, the database to be used in experiments is still undetermined. The content of the data is irrelevant to results of the experiments, but the size of the data and the database could have significant effect on experiment results, so time permitting multiple databases of various sizes could be obtained for further experiments. The languages used in implementations will be held as consistent as possible amongst all implemented protocols. At this time, the languages to be used are also undetermined. Possible candidates are Java for its easy to use and readly available classes for basic cryptography, networking and database manipulation, SQL for database manipulation, C/C++ as a logical alternative to Java and preferred imperative, Perl for its flexibility, or Haskell for its brevity. The language chosen will be directly inherent and useable with the constructed protocols.

Schedule

Below are the expected completion dates for intermediate steps in the thesis process. The earlier portion of the schedule has been stretched to permit sufficient time for courses in progress this Fall. Possible changes to this schedule and progress will be posted to the website `http://www.cs.rit.edu/~lbl6598`.

Septmeber 20, 2004 Review of bit OT_1^k for two parties in all models of [Gol04]

October 8, 2004 Thesis Proposal Approved

November 5, 2004 Review of Private Database Query, [Aso04] and others

December 3, 2004 Review of OT protocols for data larger than a bit (ie. strings and records)

December 31, 2004 Brief Review of Input Commitment Schemes, Augmented Coin Tossing, and Authenticated Computation

January 14, 2005 Construct and implement PIR and SPIR protocols with Private Querying in Semi-Honest Model

February 4, 2005 Construct and implement PIR and SPIR protocols in Malicious Model

March 11, 2005 Complete Experimentation and Analysis of Experiments

April 1, 2005 Complete and submit thesis document

April 15, 2005 Thesis Defense

Deliverables

Upon completion of this thesis a formal document will be bound and submitted. The contents of this document should be roughly organized as follows:

1. Overview
   (a) Oblivious Transfer of Variable Lengths of Data
   (b) Querying Databases Privately
   (c) Achieving a Malicious Model: Input Commitment, Augmented Coin Tossing, and Authenticated Computation
   (d) Formal Definitions
2. Construction of a Private Query Protocol
3. Construction of an OT Protocol
   (a) Semi-Honest Model
   (b) Malicious Model
4. Construction of a PIR Protocol
   (a) Semi-Honest Model
   (b) Malicious Model
5. Construction of a SPIR Protocol
   (a) Semi-Honest Model
   (b) Malicious Model
6. Results and Discussion
   (a) Comparison of Security
   (b) Comparison of Computation
   (c) Comparison of Communication
   (d) Applications and Practicalities
   (e) Further Work
7. Appendices
   (a) Source Code
   (b) Scripts

Although scource code and scripts will be printed in the formal document, an electronic copy will be submitted to be posted along with the formal document on the RIT Department of Computer Science Masters Theses and Projects webpage to aid in the further study of the proposed thesis area.

Bibliography

[Aso04] D. Asonov. *Querying Databases Privately: A New Approach To Private Information Retrieval.* SpringerVerlag, 2004.

[BC97] G. Brassard and C. Crépeau. Oblivious transfers and privacy amplification. *Lecture Notes in Computer Science*, 1233:334–??, 1997.

[BCS96] G. Brassard, C. Crépeau, and M. Sántha. Oblivious transfers and intersecting codes, 1996.

[COR99] G. Di Crescenzo, R. Ostrovsky, and S. Rajagopalan. Conditional oblivious transfer and timed-release encryption. *Lecture Notes in Computer Science*, 1592:74–??, 1999.

[CvdGT95] C. Cr'epeau, J. van de Graaf, and A. Tapp. Committed oblivious transfer and private multi-party computation. In *CRYPTO*, number Theory, pages 110–123, 1995.

[GIKM98] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 151–160. ACM Press, 1998.

[Gol04] O. Goldreich. *Foundations of Cryptography*, volume 2. Cambridge University Press, May 2004.

[Kil88] J. Kilian. Founding crytpography on oblivious transfer. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 20–31. ACM Press, 1988.

[NP99] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254. ACM Press, 1999.

[NP01] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 448–457. Society of Industrial and Applied Mathematics, 2001.

[Pin02] B. Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.*, 4(2):12–19, 2002.

[Rab81] M. Rabin. How to exchange secrets by oblivious transfer. Technical report, Technical Report TR 81, Aiken Computation Lab, Harvard University, 1981.

[Yao86] A. C. Yao. How to generate and exchange secrets. In *Proceedings 27th Symposium on Foundations of Computer Science (FOCS)*, volume 14, pages 162–167. IEEE, 1986.