Private Information Retrieval

An overview and current trends

Dmitri Asonov[*]
Humboldt-Universität zu Berlin
asonov@dbis.informatik.hu-berlin.de

Abstract

In e-commerce, the protection of user privacy from a server was not considered feasible until the private information retrieval (PIR) problem was stated recently.

A PIR protocol allows a user to retrieve a record from a database while hiding the identity of the record from a database server.

We explain a motivation for PIR by demonstrating e-commerce examples, where only PIR techniques help. All-out overview of what is done by the community is given without details of concrete algorithms.

We conclude with future work, which is mostly aimed at making PIR practical.

1 Introduction

Formally, private information retrieval (PIR) is a general problem of *private* retrieving the i-th bit out of an N-bit string stored at the server. “Private” means that the server does not know about i, that is, the server does not learn which bit the client is interested in.

Where did the need for PIR come from? In our days, knowledge about user preferences is an information with a well-recognized importance and value. This information may often play a bad role if used against the user.

Until recently, the user preferences were treated as a secret for everybody except the server. The assumption, that the server will not employ user preferences against the user, had been taken for granted for a long time. However, there is no reason for such an assumption. One of the biggest on-line media traders stated that his database containing millions user profiles and shopping preferences is one of the company’s assets. Therefore, this database can be a subject of a commercial deal, i.e., the database can basically be sold to another company without the users’ permission [52, 21].

Even worse, the server may be ”honest but stupid”. That means, the server may have a flaw in its security level, thus allowing an intruder to access user preferences. Up to half of the top on-line servers are reported to compromise the user privacy in such a way [54, 46].

Finally, the company may be forced to *sell* the user preference database due to bankruptcy [7, 56, 24].

In current systems, the user preferences depend on the good face of the company owning the server, and the quality of the server’s security tier, and the financial situation of the server’s company. There are too many assumptions to be true simultaneously and forever.

The solutions for the PIR problem would make it possible for a user to keep his preferences private from everybody including the server.

Structure. At the next section we provide a motivation for PIR. Section 3 brings a classified list of what has been done in PIR, from that open problems stated in Section 4 are derived. Section 5 is a conclusion.

2 Motivation

In this section we give application examples for PIR protocols; we also explain why some naive approaches do not work well enough to be treated as PIR solutions.

2.1 Application Examples

In the following we describe concrete as well as hypothetical examples, where PIR protocols might be useful.

Patent Databases. If the patent server knows which patent the user is interested in, this could cause a lot of problems for the user, if the user is a researcher, an inventor, or an investor. Imagine, that a scientist discovers a great idea, for example, that ”2+2=4”. Naturally, he wants to patent

[*] This research was supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316.)

it. But first, he checks at an international patent database whether a such or similar patent already exists. The administrator of that server has access to the scientist's query "Are there patents like 2+2=4", and this automatically gives him the following information:

- The "2+2=4" may possibly be an invention. Why not to try to patent it first?
- The area, the scientist (or the research laboratory) is working at, is also notable.

Both observations are highly critical and should not be revealed. PIR solves this problem: the user may openly pay with his credit card for downloading a single patent; and the server will not know which patent the user just downloaded.

Pharmaceutical Databases. Usually, pharmaceutical companies are specialized either in inventing of drugs or in gathering information about the basic components and their properties (pharmaceutical databases). The process of synthesizing a new drug requires information on several basic components from this databases. To hide the plans of the company, drug designers buy the entire pharmaceutical database. These huge expenses could be avoided if the designers use a PIR protocol to buy only the information about a few basic components needed [62].

Media Databases. These are commercial archives of electronic publications, music (mp3) files, photos, video, etc. As it was shown above, it is too risky to put customer data in server's trust. In this context, the user may be interested in hiding his preferences from the server while buying one of the digital products on-line. That means, the user may be interested in a PIR protocol.

Academic Examples. Special Operations department of the defense ministry plans an operation in a region R. To get a high-resolution map of R, there should be made an appropriate request to the IT department's map database. Thus, the staff of the IT department can figure out, that there will be a special operation in the region R soon. Is that possible to keep the secret inside the Special Operations department and still process the query at the external database? It is generally possible, if PIR is used [58].

Another hypothetical application is suggested in [14] by Isabelle Duchesnay. A spy disposes of a corpus of various state secrets. In his catalogue, each secret is advertised with a tantalizing title, such as "where is Abu Nidal". He would not accept to give away two secrets for the price of one, or even partial information on more than one secret. You (the potential buyer) are reluctant to let him know which secret you wish to acquire, because his knowledge of your specific interests could be a valuable secret for him to sell to someone else (under the title: "who is looking for terrorists"). You can *privately retrieve* a secret of your choice using PIR; and both parties remain happy.

2.2 Naive Approaches And Why They Do Not Work

There are at least two straight-forward approaches to the PIR problem. Both fail to solve the real-world problem but they point us to the properties, that the practical PIR solution must have.

Entire Database Download. Theoretically speaking, the entire database transfer (from the server to the client) solves the PIR problem: The client can process queries on the local copy of the database. Thus, the server is unaware of the user queries' content, and consequently, the server is unaware of the user preferences.

This approach cannot be applied for real, because of the great cost the user has to pay for all records of the database. Additional cost is a communication, which is equal to the size of the database. But this cost is usually negligible in comparison with the cost of the entire database content.

Anonymization techniques. Using a traffic anonymization technique (like [26]), a user can anonymously send queries to a server and anonymously receive the answers. In addition, using an anonymous payment system (like [32]) the user can anonymously pay for executing a query.

One might think this is a PIR solution. It is not, since the server can still gather some general statistics on the user preferences. For example, the server can trace which record has been accessed more than others. Or, the server can count how much a specific record has been accessed at a given time interval. Data mining on such statistics and some additional efforts may break the user privacy.

Another drawback of this approach is that most network anonymization techniques as well as payment anonymization techniques are either:

1. dependent on a third trusted party [5, 48, 32] (And we are back to the beginning: The client has to trust the third party now instead of trusting the server.)
2. or insecure under the "all-against-one" attack, when all participants cooperate against one user [16, 51, 53].

3 PIR Approaches

Over 20 scientific papers have been published on the PIR subject since the PIR problem was first formulated in [19]. We classify the results accordingly to the assumptions, that authors rely on in these papers. Not even one algorithm is explained due to the space limitations. Instead, basic ideas of some of the algorithms are given.

3.1 Theoretical Private Information Retrieval

”Theoretical” stands for the fact, that the user privacy is assumed to be unbreakable independently from the computational power of a cheater. Chor et al. prove, that any Theoretical PIR solution has a communication with a lower bound equal to the database size [19]. Thus, downloading the entire database is an optimal solution with respect to the communication amount. Such a solution is called trivial. Consequently, a non-trivial PIR solution is one, that has communication amount less than the database size.

With the idea of getting a non-trivial Theoretical PIR solution in mind, Chor et al. relax the problem setting. They assume that there are several (instead of one) non-communicating to each other database servers with the same data. This assumption makes the non-trivial Theoretical PIR feasible. (The very basic idea in [19] is to send several queries to several databases. The queries are constructed in such a way, that they give no information to the servers about the record that the user is interested in. But, using the answers from the queries, the user can construct the desired record.) There is also a case considered, when up to t of the servers are allowed to cooperate against the user.

Ambainis [4] improves results of Chor et al., while leads to the following non-trivial Theoretical PIR solutions:

1. A k database scheme (i.e., a scheme with k identical databases non-communicating to each other), for any constant $k \geq 2$, with communication complexity $O(N^{1/(2k-1)})$.
2. A $\Theta(\log N)$ database scheme with communication complexity $O(\log^2 N * \log\log N)$.

Further research on Theoretical PIR appears in [33, 34, 42, 50, 13, 63, 10, 35].

PIR of Blocks is an extension of a PIR problem in the sense, that database records are assumed to be blocks of several (instead of one) bits. Theoretical PIR of blocks is introduced in [19] and further investigated in [18]. Techniques for PIR of blocks are important for making PIR practical. The cases for blocks were also partially considered in the papers, mentioned in the next sections.

3.2 Computational Private Information Retrieval

In oder to get better communication complexity, another assumption was weakened by Chor and Gilboa [17]. ”Computational” means that database servers are presumed to be computationally bounded. I.e., under an appropriate intractability assumption, the databases cannot gain information about i. For every $\varepsilon > 0$, [17] presents a two database Computational PIR scheme with communication complexity $O(N^{\varepsilon})$.

In [47] Ostrovsky and Shoup construct PIR protocols with the option to write i-th record at the database in a way, that the database servers do not know about i. There are protocols both for the Theoretical PIR and Computational PIR with two or more servers. For example, for Theoretical PIR with three servers, they offer a protocol with communication complexity $O(N^{1/3}\log^3 N)$. The Computational PIR protocol with poly-logarithmic communication complexity requires $O(\log N)$ rounds in comparison to one round for the most PIR schemes in this review.

Computational PIR with Single Database. Recall that in the first paper on PIR it was proven, that the Theoretical PIR problem has no non-trivial solutions for the case of single database. Surprisingly, the substitution of an information-theoretic security with an intractability assumption allows to achieve a non-trivial PIR protocol for a single database schema [38]. Its communication complexity is $O(N^{\varepsilon})$ for any $\varepsilon > 0$. They use an intractability assumption, described in [30]. (The basic approach is to encrypt a query in such a way, that the server still can process it using special algorithms. However, the server recognizes neither the clear-text query nor the result. The result can be decrypted only by the client.)

This was also a first single-database protocol, where designers consider and provide database privacy (see Section3.3).

Using another intractability assumption [15], Cachin et al. demonstrated a single database Computational PIR protocol, that has polylogarithmic communication. This is an improvement in compare to polynomial communication complexity in [38]. This result looks particular effective, because the user has to send minimum $\log N$ bits just to address the i-th bit (the bit he wants to receive) in the database, independently from whether the protocol preserves privacy or not. A scheme with better results appears in [37].

3.3 Symmetrical Private Information Retrieval

Symmetrical PIR is a PIR problem, where the privacy of the database is considered. I.e., a Symmetrical PIR protocol must prevent user from learning more than one record of the database during a session. Clearly, symmetrical privacy (database privacy) is a very important property for practical applications, since an efficient billing is only then possible. Symmetrical PIR protocol for single server was first considered in [38]; and for several servers it was considered in [28]. Other symmetrical PIR were later proposed in

[42, 43, 44].

3.4 Hardware-based Private Information Retrieval

Smith and Safford [60, 61] considered the single database PIR problem under the assumption, that a special tamper-proof device is used. To understand the basic idea, imagine a secure coprocessor (SC) [64, 59, 31] installed on the server side. The user encrypts a query ”give me the i-th record”, and sends it to the SC to process. The SC decrypts the query, processes it, and then encrypts the answer and send it to the user.

The server has no evidence of what the query is, because

1. The main property of a SC is that the server, where the SC is installed, cannot access the build-in RAM of the SC. Thus, the server cannot catch sight of how the (decrypted) user queries look like.

2. To process a query, the SC reads all the records from the database not to reveal the record, the user is interested in.

3.5 Further Extensions of the Problem Setting

It can be seen from previous sections, that most of the initial work on PIR has focused on the goal of optimizing communication, because communication was considered the most expensive resource. Despite considerable success in realizing this goal, the real-life applicability of the proposed solutions remains questionable [12]. This is because in most solutions the *computation* required by the servers is at least linear in database size[1]; and typical scenario for using PIR protocols is when the database is big.

To solve this problem, Gertner et al. propose a scheme, where most computation is moved from the database server to special purpose servers [27]. While their protocols reduce computation for the database server to $O(1)$, the computation of the special-purpose servers is still linear for every query.

Di-Crescenzo et al. present another PIR scheme [22], that utilize special-purpose servers. In this model, the most computation and communication is moved off-line (i.e., it is performed only once, independently from the number of further queries). Both in [22] and in [27] the user privacy is not protected in case all servers cooperate against the user.

In comparison to [27], where most computation was moved to a more reasonable place (special-purpose servers), in [12] most computation is shifted to a more reasonable time (off-line). It is demonstrated that, while without any preprocessing linear computation is unavoidable, with preprocessing and some extra storage, computation can be reduced. Namely, Beimel et al. have the following results for the Theoretical PIR and any $k \geq 2$ and $\epsilon > 0$:

1. A k-server protocol with $O(N^{1/(2k-1)})$ communication, $O(N/\epsilon log^{2k-2} N)$ work, and $O(N^{1+\epsilon})$ extra storage bits.

2. A k-server protocol with $O(N^{1/k+\epsilon})$ communication and work, and $O(N^{1+\epsilon})$ extra storage bits.

The targeted web advertising without revealing user preferences (a problem similar to PIR) is investigated in [36].

Comparative Security Analysis of PIR Relationships between different security primitives and the PIR problem are lighted in [23, 41, 39, 11, 20].

3.6 Related Work

We briefly mention in this section the work, which does not directly solve the PIR problem, but some ideas from this related work may be used or are already used to construct a PIR protocol.

Protocols for Theoretical PIR in [19, 4] have used ideas from instance hiding problem [1, 8, 9], and multiparty communication complexity problem respectively.

An oblivious transfer problem is similar to the single database PIR problem, but its research history is 15 years older (see, for example, [49, 14, 45]). The similarities and differences between oblivious transfer and PIR are discussed in [23].

The PIR problem can also be seen as a simple case of secure multiparty computations in general, and as a computing with encrypted function problem in particular. For example, the single database PIR protocol in [38] has the same base idea as used in the scheme of computing with encrypted function introduced in [55]. And a hardware-based PIR solution [60] is a particular case of the secure multiparty computations based on secure coprocessors [64].

To make the picture complete, we mention, that the earliest (to our best knowledge) record of a problem, that is similar to PIR, takes place in 17-18 century[2]; the author is unknown.

4 Open Problems

Open problems can be easily collected after an analysis of the existing results.

[1] The server has to read the entire database to answer one query. If the server-side protocol leaves one of the records unread, then the server can conclude that this record is not preferred by the user. This breaks the user privacy.

[2] We refer the reader to the story ”Go there, I won’t tell you where; Bring me that, I won’t tell you what”. The story can be found in a ”Russian Fairy Tails” collection [2].

- It looks profitable for practical applications to further optimize the on-line computation and communication, and gain a full use of such real-world assumptions, as preprocessing and off-line communication. Note, that the work done in this direction [27, 22, 12] relies on the assumption, that all the servers will never cooperate against the user. Similar single database PIR protocols with off-line communication and $O(1)$ on-line communication and computation complexity were independently developed in [6, 57].

- It looks also important for real-world applications to extend the query definition from the ”give me the i-th block” to some more general and real-database-like form. Some steps are made in [18, 29, 25]. One of the application examples is from biological databases [25]. A client has a DNA sequence, and he wants to perform a similarity search on some external DNA database to find out whether similar sequences exist or not. To perform comparisons privately, normal PIR setting is of no use. New algorithms are needed to operate with queries other than just ”give me the i-th block”.

- To have well-optimized algorithms and extended database query possibilities is still not enough for practical PIR applications. One needs an e-commerce platform to apply PIR practically. One of the parts of this platform is, for example, payment algorithms for PIR. A solution is proposed recently of how to perform PIR in case the prices of digital goods are different [3]. Our results are in preparation on how PIR and digital rights management might coexist.

- Finally, in previous work, the application-specific PIR protocols were not considered. For example, PIR protocols for digital libraries may differ in underlying assumptions from PIR protocols for conventional databases.

Our first two observations about the possible future work are very similar to those given in a Ph.D. thesis of Tal Malkin [40].

5 Conclusion

Private Information Retrieval protocols allow a user to protect his privacy by hiding the identity of database items being retrieved by the user.

We gave a comprehensive introduction to the PIR problem, focusing the potential applications, existing results, and on future work. While theoretical variations of the problem appear to be well investigated, the most future work has to be aimed at design PIR protocols, that can be applied in practice for real-world e-commerce scenarios.

References

[1] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. Journal of Computer and System Sciences, 39(1):21–50, 1989.

[2] A. Afanas’ev. Russian Fairy Tales. Random House, Oct. 1976.

[3] B. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In Proc. of Eurocrypt’01, May 2001.

[4] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In Proc. of 24th ICALP, 1997.

[5] The Anonymizer. `http://www.anonymizer.com`.

[6] F. Bao, R. H. Deng, and P. Feng. An efficient and practical scheme for privacy protection in the e-commerce of digital goods. In Proc. of 3rd ICISC, Dec. 2000.

[7] C. Beaumont. What price privacy when dotcoms go down? NEW ZEALAND HERALD, Sept. 2000.

[8] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In Proc. of the 7th STACS, LNCS Vol. 415, Springer Verlag, 1990.

[9] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Security with low communication overhead. In Proc. of CRYPTO’90, Springer-Verlag, pages 62–76, 1991.

[10] A. Beimel and Y. Ishai. Information-theoretic private information retrieval: A unified construction. ECCC Report TR01-015, Feb. 2001.

[11] A. Beimel, Y. Ishai, E. Kushilevitz, and T. Malkin. One-way functions are essential for single-server private information retrieval. In Proc. of 31st STOC, 1999.

[12] A. Beimel, Y. Ishai, and T. Malkin. Reducing the servers computation in private information retrieval: PIR with preprocessing. In Proc. of CRYPTO’00, 2000.

[13] C. Blundo, P. D’Arco, and A. D. Santis. A t-private k-database information retrieval scheme. International J. of Information Security, July 2000. `http://dx.doi.org/10.1007/s102070100005`.

[14] G. Brassard, C. Crpeau, and J. Robert. All-or-nothing disclosure of secrets. In Proc. of Crypto’86, 1986.

[15] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In Proc. of EUROCRYPT’99, 1999.

[16] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2):84–88, Feb. 1981.

[17] B. Chor and N. Gilboa. Computationally private information retrieval. In Proc. of 29th STOC, 1997.

[18] B. Chor, N. Gilboa, and M. Naor. Private information retrieval by keywords. Technical report, Technion: Israel Institute of Technology, 1997.

[19] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In Proc. of 36th FOCS, 1995.

[20] H. C. Chun-Yun. Private information retrieval does not imply one-way permutations. Master’s thesis, National Taiwan University, 2001.

[21] CNN. Amazon client checks out. CNN Financial Network, `http://cnnfn.cnn.com/2000/09/13/technology/privacy/index.htm`, Sept. 2000.

[22] G. D. Crescenzo, Y. Ishai, and R. Ostrovsky. Universal service-providers for database private information retrieval. In Proc. of 17th PODC, 1998.

[23] G. D. Crescenzo, T. Malkin, and R. Ostrovsky. Single database private information retrieval implies oblivious transfer. In Proc. of EUROCRYPT'00, 2000.

[24] J. Disabatino. Disney offers to buy toysmart.com customer list. CNN News Online, www.cnn.com/2000/TECH/computing/07/14/disney.toysmart.list.idg/, June 2000.

[25] W. Du and M. J. Atallah. Protocols for secure remote database access with approximate matching. In Proc. of the First Workshop on Security and Privacy in E-Commerce, Nov. 2000.

[26] E. Gabber, P. B. Gibbons, D. M. Kristol, Y. Matias, and A. Mayer. Consistent, yet anonymous, web access with LPWA. CACM Journal, 42(2):42–47, Feb. 1999.

[27] Y. Gertner, S. Goldwasser, and T. Malkin. A random server model for private information retrieval. In Proc. of 2nd RANDOM, 1998.

[28] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In Proc. of 30th STOC, 1998.

[29] N. Gilboa. Topics in Private Information Retrieval. PhD thesis, Technion - Israel Institute of Technology, 2000.

[30] S. Goldwasser and S. Micali. Probabilistic encryption. Journal of Computer and System Sciences, 1984.

[31] P. Gutmann. An open-source cryptographic coprocessor. In Proc. of the 9th Usenix Security Symposium, Aug. 2000.

[32] iPrivacy. http://www.iprivacy.com.

[33] Y. Ishai and E. Kushilevitz. Improved upper bounds on information-theoretic private information retrieval. In Proc. of 31st STOC, pages 79–88, 1999.

[34] T. Itoh. Efficient private information retrieval. IEICE Transactions, E82-A(1):11–20, Jan. 1999.

[35] T. Itoh. On lower bounds for the communication complexity of private information retrieval. IEICE Transactions, E84-A(1), Jan. 2001.

[36] A. Juels. Targeted advertising... and privacy too. In Proc. of RSA, Apr. 2001.

[37] A. Kiayias and M. Yung. Secure games with polynomial expressions. In Proc. of 28th ICALP, 2001.

[38] E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: Single-database computationally private information retrieval. In Proc. of 38th FOCS, 1997.

[39] E. Kushilevitz and R. Ostrovsky. One-way trapdoor permutations are sufficient for single-database computationally-private information retrieval. In Proc. of EUROCRYPT'00, 2000.

[40] T. Malkin. A Study of Secure Database Access and General Two-Party computation. PhD thesis, Cryptography and Information Security Group, MIT, Feb. 2000.

[41] E. Mann. Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology, 1998.

[42] S. K. Mishra. On Symmetrically Private Information Retrieval. PhD thesis, Indian Statistical Institute, Calcutta, Aug. 2000.

[43] S. K. Mishra and P. Sarkar. Symmetrically private information retrieval (extended abstract). In Proc. of INDOCRYPT, LNCS 1977, Dec. 2000.

[44] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In Proc. of the 31th Annu. ACM Symp. on the Theory of Computing, 1999.

[45] M. Naor and B. Pinkas. Oblivious transfer with adaptive queries. In Proc. of CRYPTO'99, 1999.

[46] S. Olsen. Top web sites compromise consumer privacy. CNET News Archive, http://yahoo.cnet.com/news/0-1007-200-1500309.html, Dec. 1999.

[47] R. Ostrovsky and V. Shoup. Private information storage. In Proc. of 29th STOC, 1997.

[48] PrivateBuy. http://www.privatebuy.com.

[49] M. O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard, 1981.

[50] J.-F. Raymond. Private information retrieval: Improved upper bound, extension and applications. Master's thesis, McGill University, Montreal, Dec. 2000.

[51] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communication, 1998.

[52] K. Regan and C. Saliba. Privacy watchdogs blast amazon. E-Commerce Times, http://www.ecommercetimes.com/news/articles2000/000914-3.shtml, Sept. 2000.

[53] M. Reiter and A. Rubin. CROWDS: Anonymity for web transactions. Technical report, DIMACS, 1997.

[54] M. Rotenber. The online privacy protection act. Electronic Privacy Information Center, http://www.epic.org/privacy/internet/EPIC_testimony_799.pdf, July 1999.

[55] T. Sander and C. F. Tschudin. Towards mobile cryptography. Technical Report TR-97-049, International Computer Science Institute, Berkeley, Nov. 1997.

[56] G. Sandoval. Failed dot-coms may be selling your private information. CNET News Archive, http://yahoo.cnet.com/news/0-1007-200-2176430.html, June 2000.

[57] C. P. Schnorr and M. Jakobsson. Security of signed elgamal encryption. In Proc. of ASIACRYPT'00, LNCS 1976, Dec. 2000.

[58] S. W. Smith. Webalps: Using trusted co-servers to enhance privacy and security of web transactions. IBM Research Report RC-21851, IBM T.J. Watson Research Center, Oct. 2000.

[59] S. W. Smith, E. R. Palmer, and S. H. Weingart. Using a high-performance, programmable secure coprocessor. In 2nd Intl. Conf. on Financial Cryptography, 1998.

[60] S. W. Smith and D. Safford. Practical private information retrieval with secure coprocessors. Technical report, IBM T.J. Watson Research Center, July 2000.

[61] S. W. Smith and D. Safford. Practical server privacy with secure coprocessors. IBM Systems Journal, 40(3), 2001.

[62] G. Wiederhold. Private communication, June 2000.

[63] A. Yamamura. Private information retrieval scheme based on the subgroup membership problem. Symp. on Cryptography and Information Security, Jan. 2001.

[64] B. S. Yee. Using Secure Coprocessors. PhD thesis, CMU, 1994.