Enhanced Functionality and Confidentiality for Database Search and Publish/Subscribe Protocols

Giovanni Di Crescenzo(✉), Euthimios Panagos, and Brian Coan

Applied Communication Sciences, Basking Ridge, NJ, USA
{gdicrescenzo,epanagos,bcoan}@appcomsci.com

Abstract. We show a privacy-preserving and performance-preserving approach to provably transform any database search protocol into a (pull-mode or batch-mode) publish-subscribe protocol, and viceversa. This enhances functionality of both protocol types, notably implying practically efficient publish-subscribe solutions for a large class of subscriptions (e.g., index, keyword, range and conjunction). Previous work either missed practicality or focused on customized solutions for specific subscription types. We also show simple padding techniques that enhance the confidentiality of database search and publish-subscribe protocols against communication eavesdroppers. Specifically, these techniques provide optimal hiding of the number of matching database records or publications, while restricted to keeping the communication increase below a specified limit.

1 Introduction

Private information retrieval, in its early results (i.e., [1,8]), showed the surprising possibility of accessing data while provably not leaking undesired information about a database or a query, although at significant applicability restrictions [1] or performance costs [8]. After several advances, more recent literature on provably privacy-preserving database retrieval (DR) protocols contains constructions with practical efficiency (i.e., only a constant factor slower than an analogue non-private solution to the problem) for specific query types, and in a 3-party model. There, a help server facilitates a querying client and a data owner achieve their goal, where the only leakage is to the help server and can be provably characterized as 'access-pattern' over encrypted data. Intriguing questions related to this area include: what other types of protocols are possible with similar (or better) privacy and efficiency guarantees?

In this paper, we answer this question for a large class of publish/subscribe (PS) protocols. We show a general paradigm to transform a class of DR protocols into a related class of (pull-mode or batch-mode) PS protocols, while preserving privacy and practical efficiency. The resulting PS protocols provably protect the privacy of publications and subscriptions, and have efficiency only a constant

S. Ranise and V. Swarup (Eds.): DBSec 2016, LNCS 9766, pp. 52–60, 2016.
DOI: 10.1007/978-3-319-41483-6_4

factor slower than an analogue non-private solution to the problem. Moreover, they can benefit from practically efficient 3-party database retrieval protocols, without inheriting their drawbacks ('access pattern' leakage to the matching server). To the best of our knowledge, this is the first example in the area of an application where this combination of properties is achievable. We also show a converse transformation of a class of (pull-mode) publish-subscribe protocols into a related class of database retrieval protocols that have practical latency and provably protect privacy of database and queries. Finally, we show how simple padding approaches can further enhance confidentiality against eavesdroppers for both DR and PS protocols. To capture tradeoffs between privacy and communication, we formulate a restricted padding problem, and define a simple padding algorithm that provably increases the eavesdropper's uncertainty about the number of matched database records or publications. Using entropy, we can then quantify the improved confidentiality, and show that the proposed simple padding algorithm is optimal within the considered restriction model.

Related work. We note that designing PS or DR protocols using general solutions from the area of secure function evaluation protocols in the 2-party [10] or 3-party [4,6] model, would not result in practically efficient solutions. Practically efficient 3-party DR protocols with provable privacy include [3,7,9] for the case of index, keyword, range and conjunction queries. Practically efficient PS protocols with provable privacy in the 3-party model include [2] for the case of subscriptions based on boolean circuits. (See references therein for more related work on DR and PS protocols.)

2 Models and Definitions for DR and PS Protocols

DR protocols. A *database* is an n-row, $(m + 1)$-column matrix $Db = (A_1, \ldots, A_{m+1})$, where each row is associated with a data *record*, denoted as rec_i, for $i = 1, \ldots, n$, each column is associated with an *attribute*, denoted as A_j, for $j = 1, \ldots, m+1$, and each *entry* is denoted as $A_j(i)$. The first m columns are *value attributes*, where entries $A_j(i)$ are values in a *domain* $Dom_j = \{0, 1\}^\ell$ allowing suitable operations, and the last column A_{m+1}, is a *payload attribute*, where entries are from a domain $Dom_{m+1} = \{0, 1\}^r$, for integers $\ell, r > 0$. The database *schema*, including all parameters and domain descriptions, is known to all parties. A *query* is a sequence $q = (qv_1, \ldots, qv_s, mc)$, where $s \geq 1$, mc is a boolean *(matching) circuit* and, for $h = 1, \ldots, s$, each *query value* $qv_h \in Dom_j$, for some $j \in \{1, \ldots, m\}$. An *equality query gate* $(A(i), q, j, h)$, for some $j \in \{1, \ldots, m\}$ and $h \in \{1, \ldots, s\}$, is a function that takes as input $A_j(i)$ and qv_h, and outputs 1 if $A_j(i) = qv_h$ and 0 otherwise. An *equality-based query* is a query where $mc = mc'(x_1, \ldots, x_t)$, where mc is a boolean circuit and, for $h = 1, \ldots, t$, each x_h is the output of the h-th equality query gate.

A *secure database retrieval (DR)* protocol in the *3-party model* is an interactive protocol between 3 types of efficient parties: a *querier* Q, having as input a query; a *data owner* D, having as input database $Db = (A_1, \ldots, A_{m+1})$; and a *help server* HS, helping Q and D to more efficiently reach their goals. To align with many results in the area, we consider DR protocols with the following 4 subprotocols, as detailed in Fig. 1:

1. (*Key Setup*) D and Q share a key k that is unknown to HS;
2. (*Db Setup*) D sends an encrypted (using k) version of its database to HS;
3. (*Query*) Q sends an encrypted (using k) version of its query value(s) to HS;
4. (*Answer*) HS computes an answer over the received encrypted data, possibly interacting with Q and without involving D, and resulting in Q returning an output.

We define correctness and privacy requirements for DR protocols.
Correctness. The protocol's outcome should be Q's retrieval of the payload(s) $A_{m+1}(i)$ such that C's query is 'matched' by attribute values $A_1(i), \ldots, A_m(i)$. Important examples of queries and matching conditions are as follows:

1. index query: an index $ind \in \{1, \ldots, n\}$; matching condition: $i = ind$;
2. keyword query: a keyword $v \in Dom_j$; matching condition: $A_j(i) = v$;
3. conjunction query: multiple keywords $v_1 \in Dom_1, \ldots, v_t \in Dom_t$; matching condition: $(A_1(i) = v_1) \wedge \ldots \wedge (A_t(i) = v_t)$ for a specific column j;
4. range query: a range $[v_1, v_2] \subseteq Dom_j$; matching condition: $v_1 \leq A_j(i) \leq v_2$.

Privacy. The protocol communication should not reveal to any efficient adversary Adv corrupting any one among C, S or HS, any information other than system parameters $\sigma, m, s, r, \ell, \kappa, n$, or the following: (a) when Adv corrupts C, the query and the matching payloads which C is entitled to retrieve in the correctness requirement; (b) when Adv corrupts HS, 'access pattern' information relative to when HS accesses encrypted data provided by D. Given this intended leakage, a formal privacy definition can be derived using known approaches frequently used in the cryptography literature [5]. Note that such protocols, even when all communication is encrypted, can leak information to an eavesdropper, such as an upper bound on the number of matching records [3]. We study how to limit this leakage in Sect. 4.

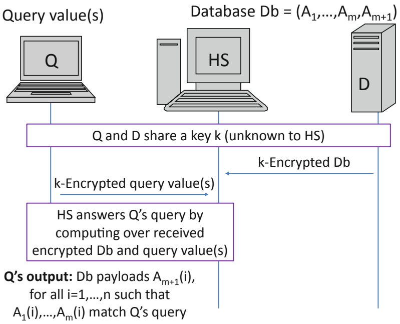
Query value(s)
Database Db = (A1,…,Am,Am+1)
Q
HS
D
Q and D share a key k (unknown to HS)
k-Encrypted Db
k-Encrypted query value(s)
HS answers Q's query by computing over received encrypted Db and query value(s)
Q's output: Db payloads Am+1(i), for all i=1,…,n such that A1(i),…,Am(i) match Q's query

Fig. 1. Structure of 3-party DR protocols

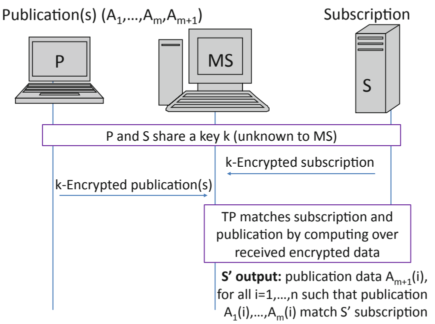
Publication(s) (A1,…,Am,Am+1)
Subscription
P
MS
S
P and S share a key k (unknown to MS)
k-Encrypted subscription
k-Encrypted publication(s)
TP matches subscription and publication by computing over received encrypted data
S' output: publication data Am+1(i), for all i=1,…,n such that publication A1(i),…,Am(i) match S' subscription

Fig. 2. Structure of 3-party PS protocols

PS protocols. We formally define a data model for PS protocols so to exactly mirror the one for DR protocols; that is, publications are defined like database records, subscriptions like queries, and equality-based subscriptions like equality-based queries. A *secure publish/subscribe retrieval* protocol is an interactive protocol between 3 types of efficient parties: a subscriber S, having as input a subscription; a publisher P, having as input a publication; and a *matching server*, denoted as MS, maintaining a repository rp and helping subscribers and publishers to store their subscriptions, publications and carry out their desired functions, including matching publications with subscriptions based on matching circuit mc. To align with some results in the area, we consider publish-subscribe protocols as made of the following 4 subprotocols, with a specific structure, as detailed in Fig. 2:

1. (*Init*): P and S share a key k that is unknown to MS;
2. (*Subscribe*): S sends an encrypted (using k) version of its subscription to MS;
3. (*Publish*): P sends an encrypted (using k) version of its publication to MS;
4. (*Pull-based Match*): Upon S' request, MS determines if there is a match between the subscription and the publication, based on the received encrypted data and matching predicate mc, resulting in S returning an output.

We define correctness and privacy requirements for pull-mode PS protocols.
Correctness. At the end of the protocol S should receive a publication item $data_i$ for all publications issued by S and matching with C's current subscription.
Privacy. The communication transmitted during the protocol should not reveal to any efficient adversary Adv that corrupts any one among S, P or HS, any information other than system parameters $\sigma, m, s, r, \ell, \kappa, n$, and the following: (a) when Adv corrupts S, the matching publication data items which S is entitled to retrieve in the correctness requirement; (b) when Adv corrupts MS, the number of matching publication data items in each execution of subprotocol *PbMatch*. Given this intended leakage, a formal privacy definition can be derived using known approaches from the cryptography literature [5]. Note that such protocols, even when all communication is encrypted, can leak information to an eavesdropper, such as the number of matching publications, as it is intended to be leaked to S by the correctness requirement. We study how to limit this leakage in Sect. 4.

3 Enhanced Functionality for PS and DR Protocols

In this section we describe our privacy-preserving transformations of any DR protocol into a PS protocol. Our first result is the following

Theorem 1. *Assuming the existence of pseudo-random functions, and of a secure 3-party DR protocol π_{dr} for equality-based queries, there exists (constructively) a secure 3-party pull-mode PS protocol π_{ps} for equality-based subscriptions, satisfying:*

1. publication correctness;
2. privacy against any polynomial-time adversary *Adv* corrupting any one among S, P or MS (that is, other than intentionally revealed data, π_{ps} only leaks to MS the number of matching publication data items in any execution of *PbMatch*)
3. latency and round complexity of π_{dr} is the same as those of π_{ps};
4. if π_{dr} has communication complexity linear in number of matching records, then π_{ps} has communication complexity linear in the number of matching publications.

Among mentioned examples of equality-based query and subscription types, Theorem 1 is applicable to index subscriptions, keyword, range and conjunction subscriptions. Remarkably, even if π_{dr} leaks information like 'access pattern' to encrypted data to the help server HS, its application in constructing π_{ps} does not result in any leakage of this same type. This is due to the following: in PS protocols, encrypted publications are only processed once and are deleted afterwards, while in DR protocols, encrypted data records remain stored with HS until they are explicitly deleted.

Description of protocol π_{ps}. We now describe the four subprotocols (*Init* for initialization, *Subscribe* for subscription, *Publish* for publication, and *PbMatch* for pull-mode matching) of our PS protocol. The overall main idea consists of the following ingredients: P and MS create an encrypted database from a batch of publications issued by P; S defines a subscription as a database query, and finally an execution of the *PbMatch* subprotocol can be defined as an execution of the *Answer* subprotocol. We note that our publication, subscription and pull-based match models well mirror the database record, query and answer models, respectively. Thus, this main idea almost defines the entire construction, by preserving efficiency of the original DR protocol. Only two more refinements are needed to satisfy correctness and privacy requirements. With respect to correctness, we note a potential issue: *Subscribe* may be run before *Publish*, while Query needs to run after *DbSetup*. We circumvent this issue as follows: during *Init*, a first batch of publications is collected and used by MS with Db-Setup to create a first publication database; later, the next batches of publications for the next publication databases are collected by MS between any two consecutive executions of *PbMatch*. With respect to privacy, note that repeated use of the same key k during Subscribe and Publish may result in subscription leakage to MS (e.g., repeated occurrences of the same subscription). We avoid this issue using a fresh session key k_i at the i-th execution of *Subscribe*, for $i \geq 1$. Each session key is derived from the originally agreed upon key k using standard key derivation techniques. A high-level pictorial description of the protocol can be found in Fig. 3.

Our next result and protocol are somewhat dual and simpler than our first ones. Formally, we obtain the following

Theorem 2. Assuming the existence of pseudo-random functions, and of a secure 3-party pull-mode PS protocol π_{ps} for equality-based queries, there exists

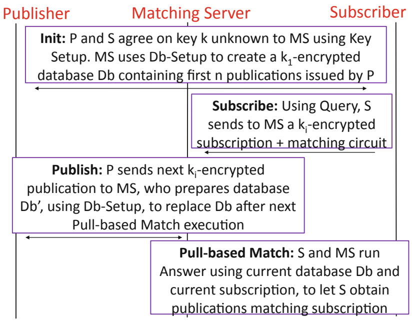
Publisher
Matching Server
Subscriber
Init: P and S agree on key k unknown to MS using Key Setup. MS uses Db-Setup to create a k_1-encrypted database Db containing first n publications issued by P
Subscribe: Using Query, S sends to MS a k_i-encrypted subscription + matching circuit
Publish: P sends next k_i-encrypted publication to MS, who prepares database Db', using Db-Setup, to replace Db after next Pull-based Match execution
Pull-based Match: S and MS run Answer using current database Db and current subscription, to let S obtain publications matching subscription

Fig. 3. Our construction of PS protocols

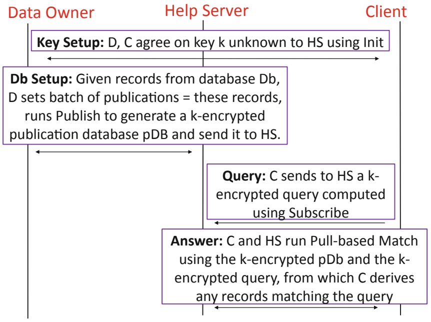
Data Owner
Help Server
Client
Key Setup: D, C agree on key k unknown to HS using Init
Db Setup: Given records from database Db, D sets batch of publications = these records, runs Publish to generate a k-encrypted publication database pDB and send it to HS.
Query: C sends to HS a k-encrypted query computed using Subscribe
Answer: C and HS run Pull-based Match using the k-encrypted pDb and the k-encrypted query, from which C derives any records matching the query

Fig. 4. Our construction of DR protocols

(constructively) a secure 3-party DR protocol π_{dr} for equality-based subscriptions, satisfying:

1. publication correctness;
2. privacy against any polynomial-time adversary Adv corrupting any one among C, D or HS (that is, other than intentionally revealed data, π_{dr} only leaks the number of matching records to HS for each execution of subprotocol $Answer$);
3. latency and round complexity of π_{ps} is the same as in π_{dr};
4. if π_{ps} has communication complexity linear in number of matching publications, then π_{dr} has communication complexity linear in the number of matching records.

Among the mentioned examples of equality-based query and subscription types, Theorem 2 is directly applicable to index, keyword, conjunction and range queries. A pictorial description of the protocol can be found in Fig. 4.

4 Enhanced Confidentiality for both Types of Protocols

In this section we describe our main results on enhanced confidentiality against eavesdroppers of DR and PS protocols. First, we define the problem of confidentiality against eavesdroppers in both protocol types. Then, we define a padding algorithm that reduces confidentiality loss while limiting communication increase. Using entropy, this loss can be shown to be optimal within the considered class of padding algorithms.

The eavesdropper confidentiality problem. As proved in [3], in any DR protocol in our model, including both those from the literature and the one obtained from Theorem 2, an eavesdropper can infer information about the number of matching database records. Note that this happens even when the communication is encrypted, since encryption, as is well known, does not hide the length of the plaintext. Padding is an often mentioned approach to reduce such leakage. We study a constrained version of the problem where we use an additive constraint on the amount of affordable padding, and ask the following questions: (1) what is the reduction in leakage to the adversary under any such padding strategies, and (2) is there an optimal padding strategy, where optimality is in the sense of minimizing leakage about m to an eavesdropper. (Note that although we study the problem for DR protocols, a similar study can be done for PS protocols, where an eavesdropper can infer information about the number of publications matching a given subscription.)

Let $X(i)$ denote the random variable that is $= 1$ (resp., 0) if the i-th database record matches (resp., does not match) the client's query, for $i = 1, \ldots, n$. We assume that all $X(i)$ are independently and uniformly distributed on $\{0, 1\}$. Also, let hwX denote the random variable that is equal to the Hamming weight (i.e., the number of 1's) in the vector $X = (X(1), \ldots, X(n))$.

Let pA be an efficient (possibly probabilistic) *padding algorithm* that takes as input $m \in \{1, \ldots, n\}$ and always returns a non-decreasing output $m' = pA(m)$; that is, for any $m \in \{1, \ldots, n\}$, with probability 1, it holds that $m' \geq m$. We say that pA is a *c-restricted* padding algorithm if for any $m \in \{1, \ldots, n\}$, with probability 1, it holds that $m' = pA(m) \leq c \cdot m$. Let $phwX$ denote the random variable returning the output of algorithm pA on input a value drawn from random variable hwX.

To analyze the information leaked about m, we use the well-known notion of entropy of a random variable, denoting as H the entropy function which maps a random variable to a real number ≥ 0. In what follows, we study the conditional entropy $H(X|phwX = m')$, modeling the uncertainty that an (even infinitely powerful) adversary has on matching bits $X(1), \ldots, X(n)$, after eavesdropping a communication consistent with m' matching records, for some m' returned by a c-restricted padding algorithm.

Entropy-based confidentiality analysis. First of all, we analyze the uncertainty on X from a value for the number of matching records hwX. Then, we define a c-restricted padding algorithm pA and show the implied uncertainty on X from the resulting value for $phwX_{pA}$. Finally, we show that algorithm pA is optimal, in that it maximizes the uncertainty among all c-restricted padding algorithms.

Uncertainty on X from a value for hwX. Let m be an integer in $\{1, \ldots, n\}$. We observe that $\text{Prob}[\,X = x|hwX = m\,]$ is 0 when the Hamming weight of n-bit vector x, denoted as $hw(x)$, is $\neq m$. Otherwise, when $hw(x) = m$, we have that

$$\begin{aligned}\text{Prob}[\,X = x|hwX = m\,] &= \frac{\text{Prob}[\,X = x\,] \cdot \text{Prob}[\,hwX = m|X = x\,]}{\sum_x \text{Prob}[\,X = x\,] \cdot \text{Prob}[\,hwX = m|X = x\,]} \\ &= \frac{2^{-n} \cdot 1}{\sum_{x:hw(x)=m} 2^{-n} \cdot 1} = \frac{2^{-n}}{\binom{n}{m} \cdot 2^{-n}} = \frac{1}{\binom{n}{m}},\end{aligned}$$

where the first equality follows from Bayes' rule, and the second on the assumption of X's distribution. Denoting $p_{x,m} = \text{Prob}[\,X = x|hwX = m\,]$, we obtain that

$$H(X|hwX = m) = -\sum_x p_{x,m}\log(p_{x,m}) = -\sum_{x:hw(x)=m} p_{x,m}\log(p_{x,m}) = \log\binom{n}{m}.$$

Defining an algorithm pA. Let m be an integer in $\{1,\dots,n\}$. We define the c-restricted padding algorithm pA as the algorithm that maps m to the next larger integer m' that is an integer multiple of c. Formally, $m' = (q+1)c$, where (q,r) is the only pair of non-negative integers such that $m = qc + r$. Note that pA is a deterministic algorithm.
Uncertainty on X from a value for $phwX_{pA}$. We observe that $\mathrm{Prob}[\,X = x|phwX = m\,]$ is 0 when $hw(x) \notin \{m-c+1,\dots,m\}$ or m is not an integer multiple of c. Otherwise, when $hw(x) \in \{m-c+1,\dots,m\}$ and $m = kc$, for some positive integer k, we have that

$$\begin{aligned}\mathrm{Prob}[\,X = x\,|\,phwX_{pA} = m'\,] &= \frac{\mathrm{Prob}[\,X = x\,]\cdot\mathrm{Prob}[phwX_{pA} = m'|X = x\,]}{\sum_x \mathrm{Prob}[\,X = x\,]\cdot\mathrm{Prob}[phwX_{pA} = m'|X = x\,]}\\ &= \frac{2^{-n}\cdot 1}{\sum_{hw(x)\in[m'-c+1,m']} 2^{-n}\cdot 1} = \frac{2^{-n}}{\sum_{j=m'-c+1}^{m'}\binom{n}{j}\cdot 2^{-n}}\\ &= \frac{1}{\sum_{j=m'-c+1}^{m'}\binom{n}{j}},\end{aligned}$$

where the first equality follows from Bayes' rule. Denoting $p_{x,m'} = \mathrm{Prob}[\,X = x|phwX_{pA} = m'\,]$, we obtain that $H(X|phwX_{pA} = m')$ is equal to

$$-\sum_x p_{x,m'}\log(p_{x,m'}) = -\sum_{hw(x)\in[m'-c+1,m']} p_{x,m'}\log(p_{x,m'}) = \sum_{j=m'-c+1}^{m'}\binom{n}{j}.$$

Optimality of padding algorithm pA. Note that the described algorithm pA always increases the uncertainty on X since $H(X|phwX_{pA} = m')$ is strictly larger than $H(X|hwX = m)$. It turns out that pA is the best algorithm among all c-restricted padding algorithms. This can be proved into two parts, depending on whether we consider deterministic or probabilistic algorithms, and the proof is based on the above computed expressions and known properties of the entropy function.

Implications on DR and PS protocols. Consider a DR protocol, where a help server HS can augment its answer to C based on a c-restricted padding algorithm. Because of our analysis above, this increases the eavesdropper's uncertainty on the Hamming weight of vector X denoting how many database records were matched with C's query, and this increase is optimal among all c-restricted padding algorithms. Consider a PS protocol obtained from a DR protocol via Theorem 1, where additionally a matching server MS can augment its answer to S based on a c-restricted padding algorithm. Because of our analysis above, this increases the eavesdropper's uncertainty on the Hamming weight of vector X denoting the number of publications matching S' subscription, and this increase is optimal among all c-restricted padding algorithms.

Acknowledgements. Part of this work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC00520. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation hereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

1. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. J. ACM **45**(6), 965–981 (1998)
2. Di Crescenzo, G., Burns, J., Coan, B., Schultz, J., Stanton, J., Tsang, S., Wright, R.N.: Efficient and private three-party publish/subscribe. In: Lopez, J., Huang, X., Sandhu, R. (eds.) NSS 2013. LNCS, vol. 7873, pp. 278–292. Springer, Heidelberg (2013)
3. Di Crescenzo, G., Cook, D., McIntosh, A., Panagos, E.: Practical private information retrieval from a time-varying, multi-attribute, and multiple-occurrence database. In: Atluri, V., Pernul, G. (eds.) DBSec 2014. LNCS, vol. 8566, pp. 339–355. Springer, Heidelberg (2014)
4. Feige, U., Kilian, J., Naor. M.: A minimal model for secure computation (extended abstract). In: Proceedings of ACM STOC, pp. 554–563 (1994)
5. Goldreich, O.: General cryptographic protocols: the very basics. In: Secure Multi-Party Computation, pp. 1–27 (2013)
6. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Proceedings of ACM STOC, pp. 218–229 (1987)
7. Jarecki, S., Jutla, C.S., Krawczyk, H., Rosu, M., Steiner, M.: Outsourced symmetric private information retrieval. In: Proceedings of ACM CCS (2013)
8. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In: Proceedings of IEEE FOCS, pp. 364–373 (1997)
9. Pappas, V., Krell, F., Vo, B., Kolesnikov, V., Malkin, T., Choi, S.G., George, W., Keromytis, A.D., Bellovin, S.: Blind seer: a scalable private DBMS. In: Proceedings of IEEE SOSP (2014)
10. Yao, A.C.-C.: How to generate and exchange secrets (extended abstract). In: Proceedings of IEEE FOCS, pp. 162–167 (1986)