SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.910

SPECIAL ISSUE PAPER

Enhancing information lookup privacy through homomorphic encryption

N. Fotiou[1], D. Trossen[2], G. F. Marias[1], A. Kostopoulos[1] and G. C. Polyzos[1,3]

[1] Mobile Multimedia Laboratory, Athens University of Economics and Business, 11362 Athens, Greece
[2] TecVis LP, Colchester, U.K.
[3] Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA, U.S.A.

ABSTRACT

Revealing one's interests in communication has been recognized as a growing problem in the Internet. We postulate that it is desirable for future information retrieval systems to provide privacy in both what information is requested and what information is received, without raising obstacles to the deployment of accounting and access control mechanisms. This paper outlines a solution that fulfills this requirement in the context of broker-based systems, that is, systems in which brokers facilitate the communication between a consumer and a provider (of information). Broker-assisted communication is a common paradigm used in many settings, including contemporary information-centric networking approaches. We present the design and the evaluation of a solution that conceals consumers' interests, without hiding consumer identity or location. The developed solution is applied over a system of hierarchically organized brokers; similar systems are used in many information lookup services. Because in these systems, information is distributed in various locations, traditional private information retrieval (PIR) protocols exhibit significant communication overhead. Our solution achieves up to 97% less communication overhead compared with a PIR protocol, without additional computational overhead. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS

adjustable privacy; broker-based communication; information-centric networking

***Correspondence**

N. Fotiou, Mobile Multimedia Laboratory, Athens University of Economics and Business, 11362 Athens, Greece.
E-mail: fotiou@aueb.gr

1. INTRODUCTION

Profiling users has become commonplace in today's Internet to an increasing dismay of end users and policy makers alike [1]. Initiatives in standardization and legislation currently introduce measures to outlaw the tracking of end users' selections and preferences during a communication session [2]. These initiatives highlight the privacy threats that the trails of a communication session may entail and leverage the role of the service provider to a privacy guard for the user, by using a "code of honor" according to which the provider will not track users that have explicitly expressed their desire not to be tracked. However, an obvious question arises: is it possible to hide user preferences and still be able to provide the same services? In this paper, we answer this question in the context of broker-assisted information lookup: we build solutions for hierarchical brokering systems that allow information lookup while hiding user queries (and preferences)–but not the user identity, or location–from the brokering system.

In our work, we assume the following system setup. First, a *consumer* is interested in a specific information item. Second, a *provider* holds this item (possibly among many others). Third, a brokering system that matches a *query* for an information item to a pointer to the provider that holds this information item. We aim at devising a solution that provides consumer *unobservability* [3], that is, a solution in which the brokering system matches a consumer query to a provider, without being able (deterministically or with high probability) to determine the provider that holds the item that satisfies the query of the consumer or the information item requested. Moreover, no provider, other than the one that holds the desired information item, should learn anything about the consumer's preferences. We refer to this property of the brokering system as *information lookup privacy* in the following.

Our main contribution, is that our solution is applied in a *hierarchically* organized information space, distributed among many brokers. Hierarchically organized information spaces are a common approach for efficiently organiz-

ing information. This approach is even used in the most common information lookup service: the DNS system.

We recognize that a brute force solution to hiding a particular query (and its result) might be that of returning the entire information space stored in the brokering system to the consumer. We argue that two main reasons could stand against this simplistic approach. First, it is often not an option because of the size of the space. Second, the brokering system business model might rely on a per-item-lookup charge. Sending the entire information space would negate this business model. Our solution provides information lookup privacy without the restrictions of the brute force approach. The proposed solution is based on a private search mechanism, developed by Bethencourt *et al.* [4], which utilizes homomorphic encryption in order to enable users to privately search a stream stored in a single location. Our work extends [4] in order to support private search over a brokering system. We anticipate that the proposed solution can be applied in many diverse broker-based systems and network types. One area that we see directly benefiting from our work is that of *information-centric networking* (ICN) [5–7]. This paradigm shifts internetworking from communication between location endpoints to communication over information identifiers; information items are directly requested by name (or by identity) through a publish–subscribe service model. For this, the matching between provider and consumer of information is performed by brokers, potentially by for-profit third parties and, therefore, privacy concerns are significantly increased. Our solution can provide an important piece in addressing a major concern for solutions in this space. Another area where we see potential for realizing our architecture is gateway-based sensor systems, such as those presented in [8], enabling anonymity for the sensor queries being issued by interested applications. The proposed architecture can also be mapped onto many topic-based publish–subscribe systems. For instance, publish–subscribe event (overlay) systems–such as SIENA [9]–can be extended to support privacy through our scheme.

The remainder of the paper is organized as follows. We first discuss related work in this area in Section 2. We continue with the basic design of our solution in Sections 3 and 4. Section 5 presents our evaluation regarding security and performance but also addressing certain socioeconomic struggles. Finally, we conclude our paper with a discussion and conclusions in Section 6.

2. RELATED WORK

Onion routing and mix-based mechanisms–such as Tor [10]–are common approaches for protecting user privacy. These mechanisms are based on an overlay network or on intermediate proxy servers, which can be used to route consumer queries anonymously through circuits. However, such mechanisms do not provide full privacy as queries are eventually revealed to the provider: if an end-to-end authentication mechanism is used (e.g., only logged on consumers are allowed to make queries), then both consumer identity and query are exposed to the provider. But even if consumers remain anonymous, the content of their queries can reveal their identity, as it happened with the anonymized query database released by AOL in 2006 [11]. Moreover, circuits introduce latency and hide the consumer's real location, making it hard to deploy multicast and mobility solutions, affecting the user's quality of experience.

Mechanisms developed for privacy preserving data analysis [12] are not suitable for our goal. These mechanisms protect query responders' privacy by adding permanent fuzziness to the responses as well as by hiding their identity using proxies. In our approach, the fuzziness introduced in responses is not permanent: a consumer can recover the exact information requested. Moreover, our approach deliberately does not hide the end-points identities, which allows for deploying dedicated access control mechanisms.

Private information retrieval (PIR) schemes (see [13] for a survey on these systems) are similar to the technique used in our approach. These schemes are used in order to retrieve a record from a database, without revealing the record or the query. In their basic form, PIR schemes model the database as a large string, or an array, from which bits are retrieved. Variations of PIR schemes use multiple replicas of the same database (e.g., [14]) or split a single information item in many sub-databases (e.g., [15]). Our approach considers a different organization of information: in terms of PIR, we consider many individual databases, hierarchically organized, in which the index of a record denotes the path to the database in which the record is stored. This organization is very similar to the information space organization of many lookup services.

Information lookup privacy can be regarded as the reverse of searching over encrypted data systems (e.g., [16]). In such systems, data is encrypted and queries are revealed to the service provider. Nevertheless, even if the provider does not have access to the data, it can infer certain information about consumers by simply examining their queries. For example, in a system in which consumers query for stock prices, the provider will not learn the stock prices but he will learn the stock names in which a consumer is interested. Therefore, these systems do not satisfy our goal of query anonymity. Schemes that support searching over encrypted data using encrypted queries–such as [17]–overcome this shortcoming, but they limit the number of consumers that can perform queries over a set of (encrypted) data items. In these systems, queries and data cannot be encrypted using independent keys; therefore, the consumers that perform queries over the same data items have to share a secret. Our scheme does not impose any relationship among the consumers that lookup the same information items.

Broker-based privacy-preserving schemes presented in [18] and in [19] have similar goals with our work. However, in both solutions, and in terms of our system model,

queries and information identifiers are encrypted. Moreover, in [19] any *subject* may learn some of the keywords included in the *issuer*'s query. In our approach, information advertisements are not encrypted, in order to facilitate the information space management, and no entity, apart from the provider that holds the desired information item, learns any information about the consumer preferences.

We believe ICN can benefit from our work. To the best of our knowledge, privacy preservation in the context of ICN, has not been widely studied. DiBenedetto *et al.* [20] have proposed a Tor-like system which however suffers from the same limitations as the traditional Tor system. Arianfar *et al.* [21] introduced a solution that is based on the creation of many algorithmically related identifiers for the same item. These identifiers have the properties that they can be easily generated, and that they do not give any information about the item they identify. This solution however, adds significant network overhead as not only identifiers have to be advertised to the broker, but they have also to be changed periodically. Moreover, the provider has to perform many computations in order to achieve a significant level of privacy.

3. PRELIMINARIES

In this section, we give a high level description of our architecture, and we provide the necessary encryption background that is utilized in our work.

3.1. High-level architecture

As illustrated in Figure 1, we base our high-level architecture on three main components, namely the information provider, the information consumer and a brokering system that mediates demand and supply for information items. The brokering system can be regarded as a directed acyclic graph with every node of the graph being a broker and each edge of the graph being a pointer to another broker. In such broker-aided (rendezvous-based) communication, it is assumed that the information identity space is structured, and each broker is responsible for a portion of this space, that is, each broker is responsible for managing certain information identifiers. In the example of Figure 1, the topmost broker manages the information subsubspace *B1*, the second broker manages the information sub-space *B1*/*B2* and so forth. The leaf brokers maintain tuples of the form $<id, pointer_1, pointer_2, \ldots, pointer_n>$ where *id* corresponds to an information item identifier, and $pointer_{\{1..n\}}$ are pointers to providers. The pair $<$*information subspace identifier*, *id*$>$ identifies uniquely an information item. In the example of Figure 1, the globally unique identifier of the first information item managed by the leftmost broker is *B*1/*B*2/*B*5/*A*. In the considered architecture, the provider advertises information item identifiers to the brokering system while the consumer queries the brokering system for provider locations. In our example, the provider P1 advertises an information item identified by *B*1/*B*4/*B*6/*B* (step 1). This advertisement results in the creation of a record in the rightmost broker (step 2). When a consumer queries the brokering system for the same identifier (step 3), the brokering system responds with the location of the provider P1 (step 4).

3.1.1. Goals.

With this high-level architecture in mind, we aim at the following extension to the matching process. We define a *private information lookup* architecture for hierarchically organized brokering systems in which the following properties hold:

- The consumer is able to query for any information item without needing to reveal his choice to the brokering system.
- The brokering system is able to perform a demand/supply matching operation without being able to determine the exact item that matched.
- No third-party, including the providers that have not advertised the desired item, learns any information about consumer's preferences.
- Consumers' identities are not hidden from the brokering system.
- Information item identifiers are well known, and they are not hidden from the brokering system.

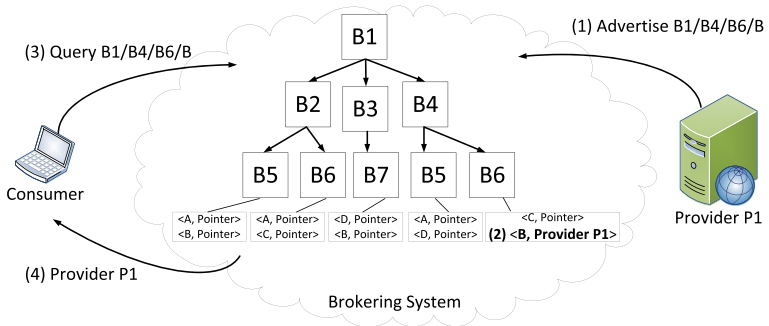

Figure 1. High-level architecture.

3.2. The Paillier cryptosystem

The Paillier cryptosystem [22] is a public key-based approach. The public key K_{pub} is a pair of numbers (n, g), where n is the product of two large primes (p, q), and g is a random number in $Z^*_{n^2}$ †. The private key K_{prv} is the least common multiple of $(p-1)$, $(q-1)$. The encryption function $E(m)$ of a message $m \in Z_n$ uses as input of the public key K_{pub}, and a randomly selected number $r \in Z^*_n$. The resulting ciphertext c belongs to $Z^*_{n^2}$, that is, the ciphertext is twice as large as the plaintext. For the decryption $D(c)$ of the ciphertext only the private key, K_{prv}, is required, that is, the random number r is not used during the decryption.

The most interesting property of the Paillier cryptosystem is its homomorphism, that, let $a, b \in Z_n$, then‡:

$$E(a) \cdot E(b) = E(a + b)$$

Moreover, it is possible to multiply an encrypted number a with a known number b, without revealing a or the result; given $E(a)$ and b, $E(a \cdot b)$ can be calculated as follows:

$$E(a \cdot b) = E(a)^b$$

4. DESIGN

In this section, we provide details about our scheme. Throughout this section, it is assumed that a consumer can learn, using an out-of-band mechanism, an ordered list of the identifiers that a broker manages (or in case this information is confidential or the size of the identities is big, a list of their hashes). Moreover it is assumed that all pointers are of equal size. Finally it is considered that all identifiers belong to Z_n.

Initially we consider a simple model in which the information space is treated as being flat, and a PIR protocol is applied. Then we take advantage of the hierarchical organization of the information space and we construct two new types of queries. In all cases the brokering system of Figure 1 is considered. The entire information space depicted in this figure consists of 10 items.

4.1. A PIR model

In this section, initially we treat the entire information space as being stored in a single broker, and we define a PIR model based on [4] that uses the following functions:

CreateQuery(ID, S)

The *CreateQuery* function is executed by a consumer, in order to construct a query. This function takes as input, an ordered list $ID = id_1, id_2, \ldots, id_m$ of the identifiers managed by the brokering system, and a corresponding set of numbers $S = s_1, s_2, \ldots, s_m$. From all items in ID, the consumer is interested in one of them. The values of S are chosen in such a way that the result of the linear expression $s_1 \cdot id_1 + .. + s_m \cdot id_m$ is the identifier of the item in which the consumer is interested. As an example, suppose that a consumer is interested in the identifier $B1/B2/B6/A$, which is the third element of the ID set; then, S can be of the form $s_3 = 1$, and $s_{i\{i=1..10, i\neq 3\}} = 0$. The function generates a public/private key pair, denoted as K_{pub}/K_{prv} and outputs a query $Q = \{K_{pub}, A\}$§, where $A = E(S)$ is a new set, which is constructed by encrypting all elements of S using K_{pub}, that is, $a_1 = E(s_1)$, $a_2 = E(s_2)$,.., $a_m = E(s_m)$. Note that because of the probabilistic property of the Paillier cryptosystem if $s_x = s_m$ then $a_x \neq a_m$. Figure 2 illustrates how the set A is constructed.

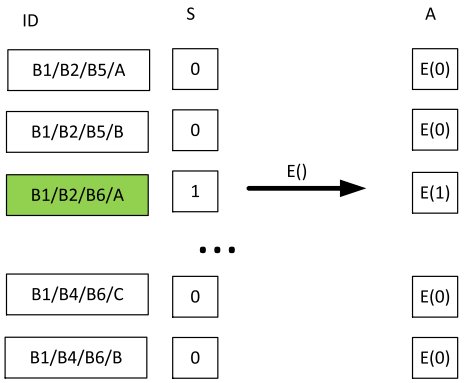

Figure 2. Query creation. The consumer is interested in the third element in the ID set, therefore the third element of the set A is E(1).

CreateResponse(Q, P)

The CreateResponse function is executed by a broker when a query $Q = \{K_{pub}, A\}$ is received. It takes as input the query Q and the set ID used for the query construction. The function outputs a response R by exponentiating each element a_i of the set A to the pointer that corresponds to the i^{th} identifier of the ID set–denoted as $pointer_i$–and by multiplying all the exponents. Assuming that the size of a pointer is less than n, R is calculated as follows:

Algorithm 1 CreateResponse(Q, ID)

$R = 1$
for each id in ID **do**
/* a_i is the i^{th} element of set A included in Q*/
$R = R^* \left(a_i^{pointer_i}\right) \bmod n^2$
end for

If the size of a pointer is bigger than n then, each pointer is divided in blocks and the above algorithm is accordingly

† For the homomorphic operations of our scheme only n is required [23].

‡ All operations are *mod* n^2.

§ K_{pub} includes only the n part of the key.

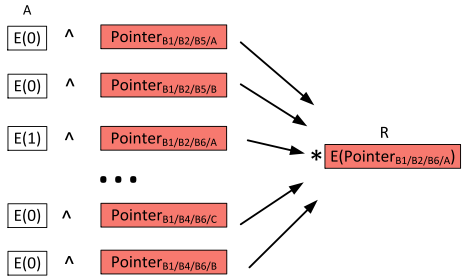

Figure 3. Response creation. The symbol ∧ denotes exponentiation and the symbol ∗ denotes multiplication.

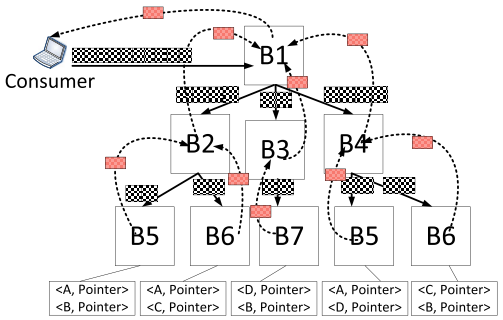

Figure 4. PIR queries over a hierarchical brokering system.

adapted. Due to the Paillier cryptosystem's homomorphism properties, this algorithm outputs the result of the following expression:

$$(a_1)^{pointer_{id_1}} \cdot \ldots \cdot (a_m)^{pointer_{id_m}} = E\left(pointer_{id_1} \cdot s_1\right) + \ldots + E\left(pointer_{id_m} \cdot s_m\right)$$

Although R is generated by combining all pointers, its size the size of single encrypted pointer. Figure 3 illustrates how R is constructed based on our example.

ReadResponse(K_{prv}, R)

The ReadResponse function is executed by a consumer upon receiving a query response R. This function decrypts R using the private key K_{prv}, generated by the CreateQuery function, and returns the pointer to the provider of the item in which the consumer is interested.

4.1.1. PIR queries over pseudo flat information spaces.

We now expand our PIR model in order to be applicable in a hierarchical brokering system.

The consumer still treats the entire information space as being flat and constructs a query over the (flat) information space. The query is sent to the root broker of the brokering system. The root broker splits the query and forwards to each of its descendant brokers the corresponding part. This procedure is recursively repeated until all leaf brokers receive the part of the query that corresponds to them. Then each leaf broker calculates a response and forwards it to its parent broker. Each broker multiplies the responses it receives from its children and forwards the product to its parent; this procedure is repeated until the top most broker receives the responses from all of its children. Finally, the topmost broker forwards the response to the consumer. Figure 4 illustrates this approach. Because the entire information space of our example consists of 10 items, the consumer creates a query which contains a set A with 10 elements. The query is split and forwarded until all leaf brokers receive their part (solid arrows). The reverse approach is followed for forwarding the response to the root broker of the system (dotted arrows).

4.2. Queries over hierarchically organized information spaces

In this section, we take advantages of the hierarchical organization of the information space, and we construct two new type of queries.

A first approach for implementing queries over a hierarchically organized space is based on the fact that given an ordered list of the identifiers managed by the brokering system, it is possible to reconstruct the brokers hierarchy. The query is now constructed using the following procedure: the consumer considers that each level of the brokering system hierarchy contains only the broker that manages the part of the information space of interest, then for each level of the hierarchy constructs a sub-query that filters the pointers maintained by the assumed unique broker; if not all brokers of the same level have the same number of pointers, then the consumer considers that the assumed unique broker has as many pointers as the broker that has the maximum number of pointers in this level. When done, all sub-queries are concatenated. Consider for example the brokering system of Figure 5, supposedly a consumer is interested in the information item identified by $B1/B3/B7/B$. Initially, the consumer constructs a sub-query that filters all but the second pointer of the broker that manages $B1$; this sub-query contains a set A of the form E(0), E(1) and E(0). Similarly, the consumer treats the second level of the brokering system as being composed only by the broker that manages $B3$. Moreover, the consumer con-

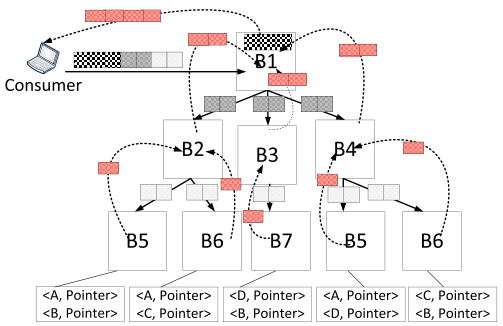

Figure 5. Queries over a hierarchical identity space.

siders that this broker has two pointers to other brokers and creates a query that filters all but the first, that is, the query contains a set A of the form E(1) and E(0). Finally, the consumer treats the third level of the brokering system as being composed only by the broker that manages $B7$ and creates a sub-query that filters all but the second pointer of this broker, that is, this sub-query contains a set A of the form E(0) and E(1). The consumer concatenates the sub-queries and sends the result (i.e., a query with an A set of the form E(0), E(1), E(0), E(0), E(1), E(0), E(1)) to the root broker. The root broker keeps the part of the query that corresponds to it and forwards the rest of the query to its child brokers. These brokers in return keep the part of the query that corresponds to them and forward the rest of the query to their children. This procedure is repeated until the query reaches the leaf brokers. In our example, this procedure will result in the broker that manages $B1$ having a query with an A set of the form E(0), E(1), and E(0), the brokers that manage $B1/B2$, $B1/B3$, and $B1/B4$, a query with an A set of the form E(0) and E(1) and the brokers that manage $B1/B3/B5,\ldots,B1/B4/B6$, a query with an A set of the form E(0) and E(1). The leaf brokers apply the query they hold to the pointers they maintain and forward the result to their parents. Then each broker applies the query it holds to the results it receives from its children and forwards the outcome to its parent, and so forth. The output of the root broker is the desired pointer encrypted as many times as the levels of the hierarchy. It should be noted here that each intermediate broker, before applying the query it holds to the results it receives from its children, has to split the result in blocks of maximum size of n. Then it should use the same element of A for all blocks of the same result. This is a procedure similar to the one followed in [24]. As we discuss in Section 5, a consumer needs to perform several decryptions in order to obtain the plaintext.

A second approach that can be used in order to implement queries over a hierarchically organized space is a variant of the PIR approach. In this case, the leaf brokers, instead of sending their responses to their parent broker, send them directly to the consumer. In this case, because the query is only applied to the leaf brokers, its A set has to be as big as the number of records of the leaf broker with the maximum number of records. This procedure is depicted in Figure 6. With this approach, the consumer will receive some redundant responses, which can be disposed.

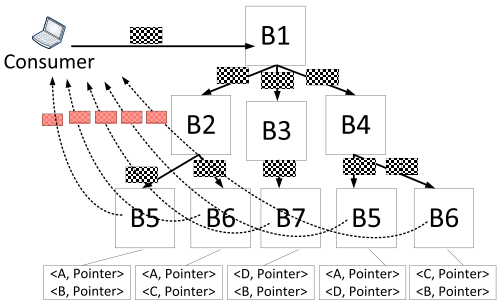

Figure 6. Queries where leaf brokers reply directly to the consumer.

5. EVALUATION

We realized our broker design as a networked prototype. As a networking environment, we chose an architecture that follows the information-centric networking paradigm [25]. This paradigm is at the heart of a growing number of efforts (see [5,6,9] for a few). Common to all these efforts is the identification of information objects through dedicated identifiers, which in turn are used for disseminating the information throughout the network. Given the information-centrism of this paradigm, the question of preserving the anonymity of queries for information within any such system seems a natural one, albeit unanswered so far. Although we see our efforts as an important contribution to this particular research area, this particular implementation choice does not, however, restrict the general applicability of our work.

Information identifiers format within the system in [25] is similar to the chosen identification scheme presented in Section 3.1, that is, each information item is identified by a stack of labels that indicates the broker (named the *rendezvous point*) that is responsible for handling queries and advertisements for that item. Each such information item is advertised toward the rendezvous point by the publisher (provider) with the subscriber (consumer) subscribing (querying) for it. Hence, the system in [25] is well aligned with our high-level architecture.

The communication API of our system is implemented using a prototype of [25]–code-named Blackadder–whereas the Paillier cryptosystem is implemented, using the advanced crypto software collection [26].

5.1. Security analysis

In this section, we analyze the security properties of the proposed system. Initially, we define a threat model, and then we evaluate our system based on this model.

5.1.1. Threat model.

For our security analysis, we assume a threat model in which an adversary wants to learn information about a consumer's preferences. We assume that a security attack is successful when an adversary learns, without being detected, some information about (i) the item in which a consumer is interested; or (ii) the items in which a consumer *is not* interested. In our threat model, we consider the following type of adversaries:

- Passive third party adversary (eavesdropper).
- Active third party adversary. This is a third party adversary that may modify transmitted packets.
- Honest-but-curious broker. This is a broker interested in learning consumers' preferences, without deviating from the specified protocol.

DOI: 10.1002/sec

5.1.2. Security evaluation.

Initially, we examine the case of an eavesdropper that learns the response of the brokering system. All responses are Paillier ciphertexts, and because the Paillier cryptosystem is semantically secure [22], an adversary can not deduce any information. All responses are encrypted with a public key generated by the consumer, therefore given two responses for the same identifier, but targeting different consumers, an adversary is not able to tell if they concern the same identifier or not. Moreover, because the Paillier cryptosystem is probabilistic, two responses that concern the same identifier, targeting the same consumer, will differ from each other. An active third party may be able to tell if two different responses targeting the same consumer, concern the same information item, simply by discarding the second response and by repeating the first one: if the consumer proceeds as no error has occurred, then these two responses concern the same item. A solution that can be used by a consumer in order to mitigate this attack is to record all the responses (or their hashes) that has received and discard all duplicates. An honest-but-curious broker has the same abilities as an eavesdropper.

We now examine the case in which an eavesdropper intercepts a query. The queries of the PIR model, as well as the second type of queries over a hierarchical information space are adaptations of [4] in a hierarchical system of brokers, therefore their security against eavesdroppers can be trivially proved. Similarly, the first type of queries over a hierarchical information space can be proved to be secured against eavesdroppers using the work in [24]. An active third party may be able to learn some information about the items in which a consumer is not interested by interchanging two elements of the set A of the query; if the consumer proceeds as no error has occurred, then it means that these two elements are encryptions of zero, therefore the consumer is not interested in the corresponding information items. In order to prevent this type of attack, the query should be digitally signed by the consumer (or it should be encrypted using an authentication encryption mechanism and a key shared between the consumer and the brokering system), and the brokering system should be honest in order to inform the consumer if the query has been tampered. An honest-but-curious broker has the same abilities as an eavesdropper.

5.1.3. The case of malicious brokers.

In this sub-section, we discuss a case which is not considered in our threat model: the case of malicious brokers that deviate from the specified protocol. As we will see when a malicious broker deviates from the specified protocol, it is possible to learn some information about the consumer's non-preferences. All the following attacks are based on the fact that it is hard to tell if a broker has used all the available data in order to create a response, and a solution for these attacks has been left as future work.

The first attack concerns deviation from the response creation procedure. In the PIR query type, as well as, in the first type of queries over a hierarchical information space, it can be observed that if the intermediate response of a broker, that *does not* manage part of the identifier of the item in request, is discarded then (i) the final response will still be valid; and (ii) the consumer will not be able to tell that the response has been manipulated. The same applies if a leaf broker does not include in the response calculation, a pointer that is not of interest of the consumer. Therefore by discarding an intermediate response or a pointer, and by observing if the consumer proceeds as no error has occurred, then a malicious broker is able to tell some of the identifiers in which the consumer is not interested. When the second type of queries over a hierarchical information space is used, this attack can only by launched by leaf brokers that do not consider some of their pointers during the response generation.

Our next attack is achieved when a malicious broker manipulates a consumer's query. If the PIR model or the first type of queries over a hierarchical information space are used, a malicious broker can omit to forward to a subsequent broker the corresponding part of the query, excluding this broker–and its children–from the response creation procedure. If the consumer proceeds as no error has occurred, then the malicious broker is able to tell some of the identifiers in which the consumer is not interested. When the second type of queries over a hierarchical information space is used and if the consumer knows the expected number of responses, that is, knows the number of leaf brokers, then this attack is not applicable. However, in this case a malicious broker may alter the query it forwards to its children (instead of omitting it). Again, if the consumer proceeds as no error has occurred, then the malicious broker learns some of the identifiers in which the consumer is not interested. Nevertheless, this attack can be mitigated using digital signatures.

Variations of these attacks are applicable to many PIR systems, including [24] and [4], and to our knowledge no solution has been provided¶.

Table I. Notation.

$\|ID\|$	The size of the identity space managed by the brokering system
$Size_p$	The size of a pointer
h	The height of the brokering system hierarchy
$max(p_i)$	The number of pointers of the broker with the maximum number of pointers at level i
$avg(p_i)$	The average number of pointers of a broker at level i
$\|leaf\|$	The number of leaf nodes in the brokering system hierarchy

5.2. Analysis of the introduced overhead

Throughout this section the notation of Table I is used.

¶ [19] provides a solution that assures that a *subject* includes in the response calculation a *property* that it really owns, however this solution does not tackle the case in which the *subject* does not include a (valid) *property* in the response calculation.

5.2.1. Communication overhead.

Assuming a public key of $Size_{Pub}$ bits[‖], the overhead introduced in the communication between a consumer and a brokering system is now calculated. In the PIR model the size of a query is $Size_{Pub} + |ID| \cdot 2 \cdot Size_{Pub}$ bits, where $2 \cdot Size_{Pub}$ is the size of the encryption of 0 or 1. The size of the response is $2 \cdot \lceil Size_p/Size_{Pub} \rceil \cdot Size_{Pub}$. In the first type of queries over a hierarchical information space the size of a query is $Size_{Pub} + \sum_{i=0}^{h-1} max(p_i) \cdot 2 \cdot Size_{Pub}$ bits. The size of the response is $2^h \cdot \lceil Size_p/Size_{Pub} \rceil \cdot Size_{Pub}$. In the second type of queries over a hierarchical information space the size of a query is $Size_{Pub} + max(p_{h-1}) \cdot 2 \cdot Size_{Pub}$ bits, where $max(p_{h-1})$ is the number of pointers of the leaf broker with the maximum number of pointers. In this type of querying the consumer receives as many responses as the leafs of the brokering system hierarchy, with each response being $2 \cdot \lceil Size_p/Size_{Pub} \rceil \cdot Size_{Pub}$. Therefore the brokering system to consumer communication overhead is $|leaf| \cdot 2 \cdot \lceil Size_p/Size_{Pub} \rceil \cdot Size_{Pub}$. The PIR model experiences the biggest communication overhead and this happens because it treats the information space as being flat. PIR optimizations are also applicable in our system, but they have to be applied per broker. Therefore if we consider the optimization proposed in [24], according to which the (per-broker) information space is organized in a 2-hypercube, then the query size of this optimized PIR model would be $|leaf|*log_2(avg(p_{h-1}))$ and the response size would be $4 \cdot \lceil Size_p/Size_{Pub} \rceil \cdot Size_{Pub}$.

We now evaluate the improvement of the communication overhead due to our approach. We consider a brokering system with varying height and branching factor. Moreover we consider that the inter-broker communication overhead is negligible.The leaf brokers of the brokering system maintain as many records required in order for $|ID|$ to be 1000. Moreover it is assumed that a single pointer fits to an encryption block and the size of the public key is 1024 bits. Figure 7 shows the communication overhead improvement when the branching factor is 4 and h varies from 2 to 5. Figure 8 shows the communication overhead improvement when h is 4 and the branching factor varies from 3 to 6. In both cases we consider the communication overhead between a single consumer and the brokering system, for a single query/response exchange. As it can be seen the only case in which our system experiences bigger communication overhead is when the information space is almost flat (i.e., h is 2). Moreover it can be seen than when the information space is highly distributed the communication overhead improvement is almost 97%.

5.2.2. Computational overhead.

The use of cryptography introduces computation overhead to both the consumers and the brokering system as these entities have to perform multiplications and exponentiations using large numbers. In an Ubuntu 12.04 based machine, using an Intel i5 processor at 2.8 GHz, 4 GB of RAM and the GMP library ver. 5.1.0, the modular multiplication of two 256-bytes numbers was performed in $2\mu s$ and the modular exponentiation of a 256-bytes number raised to a 128-bytes number was performed in 7.5 ms. The encryption of a number (i.e., the creation of an element of the set A of a query) was performed in 9.6 ms. Nevertheless this is a computation that can be made offline, i.e., a consumer can pre-calculate an arbitrary number of encryptions of 0 and 1[**]. The decryption of a 256-bytes ciphertext was performed in 7.5 ms.

Assuming that a pointer fits to an encryption block, in the PIR model all leaf brokers have to perform as many exponentiations as the number of pointers they maintain and all brokers have to perform as many multiplications as the number of pointers they maintain. In the first type of queries over a hierarchical information space all leaf brokers have to perform as many exponentiations and mul-

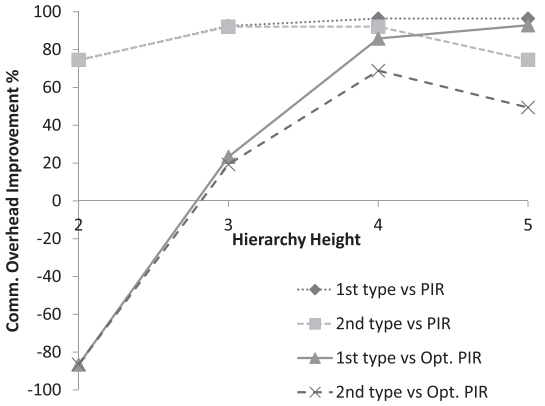

Figure 7. Communication overhead improvement with varying brokering system height.

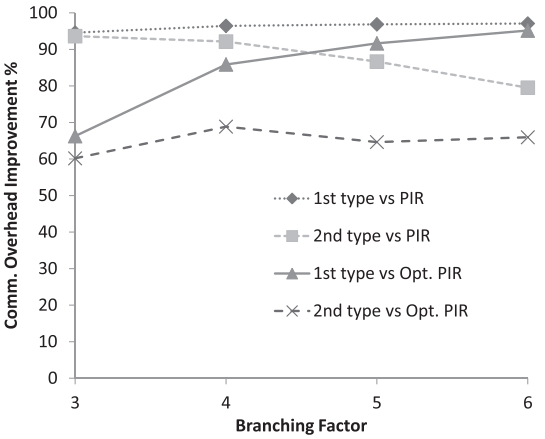

Figure 8. Communication overhead improvement with varying brokering system branching factor.

[‖] Considering only the n part of the key.

[**] Note however that the same encryption should not be used in two different queries in order to avoid pattern based attacks.

tiplications as the number of pointers they maintain. A broker at level i will receive $avg(p_i)$ responses of size $2^{h-i-1} * Size_{Pub}$. Each of these responses has to be split in 2^{h-i-1} blocks of size $Size_{Pub}$ and for each block an exponentiation and a multiplication has to be performed. A consumer in order to decrypt the final response, has to perform $\sum_{i=0}^{h-1} 2^i = 2^h - 1$ decryptions. Finally in the second type of queries over a hierarchical information space all leaf brokers perform as many exponentiations and multiplications as their records.

It should be noted that the time required to create a response is not proportional of the number of the operations, since all operations of the same level are done in parallel. Therefore, if the time of a multiplication is considered to be negligible, the total time required to create a response is the same whether the PIR model or the second type of queries over a hierarchical information space is used. Figures 9 and 10 show the estimated response calculation time based on the number of exponentiations

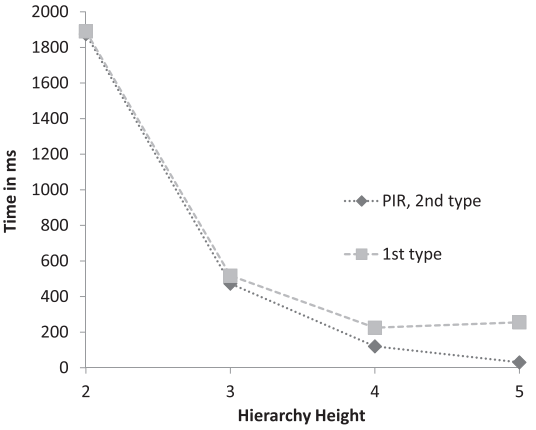

Figure 9. Response calculation time with varying brokering system height.

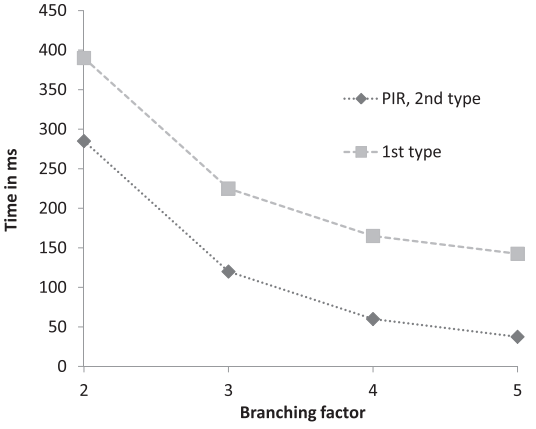

Figure 10. Response calculation time with varying brokering system branching factor.

and the aforementioned property, using the setup of the previous section. In Figure 9 the branching factor is 4, and in Figure 10 the height of the brokering system hierarchy is 4. As it can be observed the more distributed is the information space, the less time is required for the response calculation. The reason for that is, that the more distributed is the information space, the more brokers work on the response with each of them performing calculations on less data.

5.3. Socioeconomic struggles

Extending our security and performance analysis, it is worth investigating our proposed solution from the viewpoint of the socioeconomic incentives and tussles for adopting (or not) such solution. Based on the tussle taxonomy outlined in [27] we focus on security, trust, and information tussles, and how these tussles are impacted by our proposed mechanism.

Generally, applying policies for matching interests and availability, as well as the aspects of (avoiding) profiling usage and consumption, expose a tussle space for consumers and brokering services. Our solution clearly shapes this tussle space in favor of the consumer by allowing better privacy choices in the lookup service.

Another tussle concerns the control and protection of the information. For instance, the "right to be forgotten" [1] proposed by the European Commission, aims at addressing potential problems in data protection regulation. This underlines the difficulty that Internet users face concerning escaping their past, because they are not always able to apply policies about their published personal data. With our solution, we again clearly influence this tussle in favor of the consumer by allowing for disguising the query for an information item.

Given that our solution influences the playing field in favor of the consumers, profiling-based business propositions are clearly negatively impacted by our solution, as profiling becomes nearly impossible for the brokers. However, instead of relying on income from profiling end users, we see the emergence of brokering services that could charge consumers for the increased degree of privacy for each information lookup. It is shown in other research work [28] that placing a price on privacy is possible–but not trivial–for end users. Our overhead evaluation in Section 5.2 allows for a direct connection of overhead and achievable privacy in terms of disguising one's profile of lookup operations. We recognize, however, that more research is required to translate these results into tangible pricing models.

6. CONCLUSIONS AND FUTURE WORK

The various do-not-track initiatives have shown the need for solutions that counter the increasing profiling and therefore revelation of user information to third parties.

We address this need by designing a simple broker-based framework, which enhances privacy of information lookups, utilizing homomorphic encryption. This technique enhances privacy by hiding the nature of the query from the brokering system, while providing the desired information to the consumer. We outlined design choices that allow trading-off complexity of computation and communication overhead with adjustable degree of privacy.

We provided an evaluation of this trade-off by presenting the communication and computation overhead for the various design choices. We integrated our design into a working prototype selecting the emerging area of information-centric networking as our specific implementation area. We see this area particularly benefiting from our solution to a crucial problem of exposing consumer choices to third parties. This paper also provided an early insight into the possible socioeconomic struggles and opportunities that are created by our proposal. We recognize, however, that this angle to our solution will need significant extension in our future work.

ACKNOWLEDGEMENT

The work reported in this paper was supported by the FP7 ICT project PURSUIT, under contract ICT-2010-257217.

REFERENCES

1. Rosen J. The right to be forgotten. *Stanford Law Review Online* 2012; **64**: 88.
2. W3C. *The tracking protection working group*, 2012. http://www.w3.org/2011/tracking-protection/, last accessed: 5 September 2013.
3. Pfitzmann A, Khntopp M. Anonymity, unobservability, and pseudonymit a proposal for terminology. In *Designing Privacy Enhancing Technologies*, vol. 2009, Federrath H (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2001; 1–9, DOI: 10.1007/3-540-44702-4_1.
4. Bethencourt J, Song D, Waters B. New techniques for private stream searching. *ACM Transactions on Information and System Security* 2009; **12**(3): 16:1–16:32.
5. Koponen T, Chawla M, Chun BG, Ermolinskiy A, Kim KH, Shenker S, Stoica I. A data-oriented (and beyond) network architecture. *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, ACM: New York, NY, USA, 2007; 181–192.
6. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL. Networking named content. *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, ACM: New York, NY, USA, 2009; 1–12.
7. Fotiou N, Trossen D, Polyzos GC. Illustrating a publish-subscribe internet architecture. *Telecommunication Systems* 2012; **51**: 233–245.
8. Trossen D, Pavel D. Nors: an open source platform to facilitate participatory sensing with mobile phones. *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking Services, 2007. MobiQuitous 2007*, Philadelphia, PA,USA, 2007; 1–8.
9. Carzaniga A, Rosenblum DS, Wolf AL. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems* 2001; **19**(3): 332–383.
10. Dingledine R, Mathewson N, Syverson P. Tor: the second-generation onion router. *Proceedings of the 13th USENIX Security Symposium*, San Diego, CA, USA, 2004; 303–320.
11. Arrington M. *Aol proudly releases massive amounts of private data*, 2006.
12. Dwork C. Differential privacy: a survey of results. In *Theory and Applications of Models of Computation*, vol. 4978, Agrawal M, Du D, Duan Z, Li A (eds), Lecture Notes in Computer Science. Springer: Berlin / Heidelberg, 2008; 1–19.
13. Ostrovsky R, Skeith WE, III. A survey of single-database private information retrieval: techniques and applications. In *Public Key Cryptography PKC 2007*, vol. 4450, Okamoto T, Wang X (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2007; 393–411. DOI: 10.1007/978-3-540-71677-8_26.
14. Zhao F, Hori Y, Sakurai K. Two-servers pir based dns query scheme with privacy-preserving. *The 2007 International Conference on Intelligent Pervasive Computing, 2007. IPC*, Jeju Island, Korea, 2007; 299–302. DOI: 10.1109/IPC.2007.27.
15. Papadopoulos S, Bakiras S, Papadias D. pcloud: a distributed system for practical pir. *IEEE Transactions on Dependable and Secure Computing* 2012; **9**(1): 115–127. DOI: 10.1109/TDSC.2010.60.
16. Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G. Public key encryption with keyword search. In *Advances in Cryptology - EUROCRYPT 2004*, vol. 3027, Cachin C, Camenisch J (eds), Lecture Notes in Computer Science. Springer: Berlin / Heidelberg, 2004; 506–522.
17. Boneh D, Kushilevitz E, Ostrovsky R, Skeith WE, III. Public key encryption that allows pir queries. *Proceedings of the 27th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO'07, Springer-Verlag: Berlin, Heidelberg, 2007; 50–67.
18. Xiao Y, Lin C, Jiang Y, Chu X, Liu F. An efficient privacy-preserving publish-subscribe service

scheme for cloud computing. *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Miami, FL, USA, 2010; 1–5. DOI: 10.1109/GLOCOM.2010.5683310.

19. Shikfa A, Önen M, Molva R. Broker-based private matching. In *Privacy Enhancing Technologies*, vol. 6794, Fischer-Hner S, Hopper N (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2011; 264–284. DOI: 10.1007/978-3-642-22263-4_15.
20. DiBenedetto S, Gasti P, Tsudik G, Uzun E. Andana: anonymous named data networking application. *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, San Diego, CA, USA, 2012.
21. Arianfar S, Koponen T, Raghavan B, Shenker S. On preserving privacy in content-oriented networks. *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking*, ICN '11, ACM: New York, NY, USA, 2011; 19–24.
22. Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology EUROCRYPT 99*, vol. 1592, Stern J (ed), Lecture Notes in Computer Science. Springer: Berlin / Heidelberg, 1999; 223–238.
23. Damård I, Jurik M. A generalisation, a simplificcation and some applications of paillier's probabilistic public-key system. In *Public Key Cryptography*, vol. 1992, Kim K (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2001; 119–136. DOI: 10.1007/3-540-44586-2_9.
24. Chang YC. Single database private information retrieval with logarithmic communication. In *Information Security and Privacy*, vol. 3108, Wang H, Pieprzyk J, Varadharajan V (eds), Lecture Notes in Computer Science. Springer: Berlin / Heidelberg, 2004; 50–61.
25. Trossen D, Parisis G. Designing and realizing an information-centric internet. *IEEE Communications Magazine* 2012; **50**(7): 60–67.
26. *Advanced crypto software collection*, 2012. http://acsc.cs.utexas.edu/, last accessed: 5 September 2013.
27. Trossen D, Kostopoulos A. Techno-economic aspects of information-centric networking. *Journal of Information Policy* 2012; **2**: 26–50.
28. Riederer C, Erramilli V, Chaintreau A, Krishnamurthy B, Rodriguez P. For sale : your data: by : you. *Proceedings of the 10th ACM WORKSHOP on Hot Topics in Networks*, HotNets-X, ACM: New York, NY, USA, 2011; 13:1–13:6.