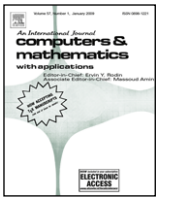
Computers and Mathematics with Applications 59 (2010) 1568–1577

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa



Optimal configurations for peer-to-peer user-private information retrieval

Klara Stokes *, Maria Bras-Amorós

Universitat Rovira i Virgili, Catalonia, Spain

ARTICLE INFO

Article history:
Received 8 May 2009
Received in revised form 5 January 2010
Accepted 5 January 2010

Keywords:
Combinatoric configurations
Private information retrieval
Projective plane
Ramanujan graphs
Expander graphs

ABSTRACT

User-private information retrieval systems should protect the user's anonymity when performing queries against a database, or they should limit the servers capacity of profiling users. Peer-to-peer user-private information retrieval (P2P UPIR) supplies a practical solution: the users in a group help each other in doing their queries, thereby preserving their privacy without any need of the database to cooperate. One way to implement the P2P UPIR uses combinatoric configurations to administrate the keys needed for the private communication between the peers.

This article is devoted to the choice of the configuration in this system. First of all we characterize the optimal configurations for the P2P UPIR and see the relationship with the projective planes as described in finite geometry. Then we give a very efficient construction of such optimal configurations, i.e. finite projective planes. We finally check that the involved graphs are Ramanujan graphs, giving an additional justification of the optimality of the constructed configurations.

1. Introduction

Public access databases are an indispensable source of information, especially when the information needs to be constantly updated. But the databases also imply a high risk for the privacy of the users, since a curious administrator of a database can track the queries of a user and deduce her interests and necessities. There are several disciplines studying different aspects of privacy in communications. We will describe some of these shortly with the purpose to contrast them with the protocol treated in this article (P2P UPIR, first described in [1]).

1.1. PIR

The discipline studying how a user should retrieve an element from a database or a search engine, without the system or the server being able to deduce which element is the object of the user's interest, is called PIR — Private Information Retrieval. The name and the discipline were introduced in the works of Chor, Goldreich, Kushilevitz and Sudan [2,3].

A first result, found in these works, says that the only way to guarantee complete privacy, when using one single database, is by making the user access all the information in the database. Because of this, the first PIR protocols were initially designed for situations where there exist several copies of the same database, without these copies being intercommunicated. In this case, privacy refers to each of the servers individually [2,3].

Later, computational PIR (cPIR) was introduced, dealing with privacy against one single database [4,5]. In this case, there is one unique server with limited computational capacity and the privacy is relaxed to computational privacy. This means

* Corresponding address: Universitat Rovira i Virgili, 43007 Tarragona, Catalunya, Spain. Tel.: +34 697690447.
E-mail address: klara.stokes@urv.cat (K. Stokes).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.01.003

that the computations the server has to perform in order to gather enough information on the searches of a user to vulnerate her privacy, exceeds the capacity of the server. To distinguish the original PIR from the computational PIR, the former is called information theoretic PIR.

A major issue with cPIR schemes is that they are computationally expensive. The database needs to process all its entries for every query sent by the users, since otherwise it would be able to deduce in what entries the user is not interested. New protocols have been presented lately, based on noise over lattices instead of on number theoretical problems. The computational cost is then lowered, but communication performance obtained is worse [6]. One interesting example of this is [7] where a higher efficiency is obtained accepting a minor probability of error in the answer of the query.

Apart from cPIR, there are PIR protocols based on the assumption that a trusted hardware is installed in the database, so-called trusted-hardware based PIR (thPIR). This is a rather prosperous assumption, and several ideas for thPIR protocols have been presented. However, the assumption of the existence of a trusted hardware, restricts the applications of these protocols to particular situations.

The drawbacks we observe with the existing PIR protocols are the following:

- PIR protocols model the database as a vector in which the user knows the physical address of the item she is interested in. This is a very unreal assumption, e.g. think of a user querying a search engine;
- Theoretical PIR protocols have complexity that is linear in the size of the database. To avoid giving the server any clues of the interests of the user, the protocol must be such that the server process all entries in the database for every query;
- It is assumed that the database server cooperates in the PIR protocol. But it is the user who is interested in her own privacy, whereas the motivation for the database server is dubious; actually, PIR is likely to be unattractive to most companies running queryable databases, as it limits their profiling ability.

1.2. Mixing

Digital mixes were invented by David Chaum in 1981, in order to provide anonymous email. A mix works at the network layer, and tries to hide the meta-data associated to a communication, i.e. avoid traffic analysis. One common approach for the construction of mixes is to use a chain of multiple untrusted relays, e.g. MorphMix [8] and Tarzan [9], but it is also possible to use a single trusted relay, as is the case with the interesting example of PIR applied to mixing that can be found in [10].

A well-known software, providing traffic analysis resistance for interactive communication, is Tor [11]. It is an example of mixing using repeated public key cryptography through a chain of untrusted relays, called *onion-routing*. However, these are not intended to offer private information retrieval. They protect the transport of data, but give no end-to-end protection (at the application level). A server may link the successive queries submitted by the same user (e.g by using cookies), and in that way be able to profile and re-identify the user.

This last observation is generally true for all systems working on the network layer, hence for all mixers. Although anonymity on network level is achieved, so that the user's ip is maintained in secret, the collection of network traffic originating from her (secret) ip will reveal her by its content, e.g. through user names, query contents, etc.

1.3. Other systems

1.3.1. Goopir

In [12] a system named Goopir is proposed in which a user masks her target query by ORing it with $k-1$ fake queries and then submits the resulting masked query to a search engine or large database which does not need to cooperate (in fact, it does not even need to know that the user is trying to protect her privacy). Strictly speaking, Goopir does not achieve PIR as defined above; rather, it provides $h(k)$-private information retrieval, in that it cloaks the target query within a set of k queries of entropy at least $h(k)$. This system works fine but it assumes that the frequencies of keywords and phrases that can appear in a query are known and available: for maximum privacy, the frequencies of the target and the fake queries should be similar, so that the uncertainty $h(k)$ of the search engine about the real target query is maximum.

1.3.2. TrackMeNot

TrackMeNot [13] is a software available as a plugin for Firefox. It periodically issues randomized search queries to popular search engines, e.g., AOL, Yahoo!, Google, and MSN. In this way it hides the users actual search trails in a cloud of 'ghost' queries, significantly increasing the difficulty of aggregating such data into accurate or identifying user profiles. While practical at a small scale, if the use of TrackMeNot became generalized, the overhead introduced by ghost queries would significantly degrade the performance of search engines and communications networks. Also, the way the automatic ghost queries are submitted may be distinguishable from the way real queries are submitted, which could provide clues on how to identify the latter type of queries.

1.4. UPIR

We define UPIR to be the discipline studying how a user should retrieve an element from a database or a search engine without the system or the server being able to deduce who the retrieving user is. Since UPIR does not hide the content of

the query for the database, but instead obstructs the possibilities for the database of profiling users, formally a UPIR protocol does not have to be a PIR protocol.

UPIR and mixers both deal with anonymity, but the concepts are different. As we commented before, mixers provide anonymity on the network layer, but the user can still be profiled through e.g. cookies. UPIR deals with anonymity on the application layer.

The protocol treated in this article is Peer-to-Peer Private Information Retrieval [1,14]. The idea is that users submit queries on behalf of other users. The way in which users share communications spaces (memory sectors and cryptographic keys) is defined using combinatorial configurations. Regarding this protocol we consider that the following should be stressed:

- P2P UPIR has none of the disadvantages of cPIR listed in Section 1.1, e.g. it does not need the cooperation of the server, it has sublinear complexity, and the database does not have to be modeled as a vector. Of course it is not a fair comparison, since the P2P UPIR protocol does something quite different from what is described in Section 1.1;
- Unlike mixers, P2P UPIR hides the profile of the user in front of the database/server. The users send queries on behalf of others, i.e. it is something like a mixer on application level;
- Unlike Goopir, no knowledge of the frequencies of all possible keywords and phrases that can be queried is required;
- Unlike TrackMeNot, the overhead of ghost query submission is avoided.

In addition to preserving the privacy of a user's query profile in front of the database and external intruders, P2P UPIR offers privacy versus peer users, because peers are anonymous to each other using mixers.

As mentioned before, one way to implement P2P UPIR uses combinatoric configurations to administrate the keys needed for the private communication between the peers [1,14]. The main problem when dealing with configurations is that they are very easy to define but not so easy to find. This work is devoted to combinatorial configurations in the context of P2P UPIR.

The rest of this article is organized as follows. In Section 2 we describe the P2P UPIR with combinatorial configurations. In Section 3 we characterize the optimal configurations for the P2P UPIR and see the relationship with the projective planes as described in finite geometry. In Section 4 we give a very efficient construction of such optimal configurations. We finally check in Section 5 that the involved graphs are Ramanujan graphs, giving an additional justification of the optimality of the constructed configurations.

2. Peer-to-peer private information retrieval

In the articles [1,14] a peer-to-peer user private information retrieval system was introduced. This is an alternative system capable of providing user privacy against one single database.

Suppose that a community of users share a communication space formed by one memory sector and one cryptographic key. The community users use the communication space to write their query requests, to read the query requests of other users and then commit these, and finally to write the answers to the queries. In that way all users collaborate for the good of the group and the database can not know who is asking what, nor elaborate any profiles, at least not more specific profiles than one describing the entire community. This system is really good if we consider the privacy against the database, but it is not so good when it comes to privacy between users. Actually, although it is not known who made a particular query request, all requests from a certain user pass through the shared communication space.

We can think of another system where each user shares a different communication space with every other user. In that way she can spread her query requests between them. The privacy against the database is maintained. Every user reads only a portion of the query requests of a user with whom he collaborates, but on the other hand he can be certain of who of the users requested the query. In the mentioned work an intermediate solution is proposed, using combinatoric configurations. That is, we have a set of n_c communication spaces, all of them consisting of a memory sector and a cryptographic key and a set of n_u users, all of them having access to a subset of d_u communication spaces so that every communication space is shared by d_c users and every pair of users share at most one communication space. The combinatoric configurations are the combinatoric object defining the distribution of the users among the mentioned communication spaces. A good reference for combinatorial configurations is [15] and [16] is a more general reference for combinatorial designs. In [17] there are some results on the existence of combinatorial configurations. Configurations are also called *partial linear spaces* or *semi-linear spaces*. As in the previous systems the users in the community submit queries to the database on behalf of other users, using the different communication spaces to request queries from the others, to read query results and to commit queries for the others.

Actually, both situations explained before, the one with a single communication space shared by the totality of the users in the community, as well as the one with a different communication space for each pair of users, define configurations. In the first case we have $n_c = 1$, $d_u = 1$, $d_c = n_u$, and in the second case $n_c = \frac{n_u(n_u-1)}{2}$, $d_u = n_u - 1$, $d_c = 2$. It is already justified that these configurations have some deficiencies. We are now interested in verifying what kind of configuration is the most appropriate for the scenario of peer-to-peer user-private information retrieval.

3. Optimal configurations for peer-to-peer private information retrieval

Suppose that we use a mathematical (n_c, n_u, d_c, d_u)-configuration for the realization of the P2P UPIR protocol. We next analyze its performance by means of the parameters n_c, n_u, d_c and d_u. There is a well-known property of configurations

saying that necessarily

$$n_u d_u = n_c d_c. \tag{1}$$

We have that:

- The number of required keys and memory sectors is $n_c = n_u d_u / d_c$ (by equality (1)). Therefore, as d_c grows there is a reduction in the number of required keys and memory sectors (storage efficiency);
- One can check that in a (n_c, n_u, d_c, d_u)-configuration the number of blocks intersecting any specific block is $d_u(d_c - 1)$. Now since the number of blocks intersecting a specific block cannot be greater than $n_u - 1$ we get that

$$d_u(d_c - 1) \leq n_u - 1. \tag{2}$$

  The overall number of keys stored by the users therefore satisfies

$$\{\text{overall number of keys}\} = n_u d_u \leq \frac{n_u(n_u - 1)}{d_c - 1}, \tag{3}$$

  and therefore, as d_c grows there is a reduction in the overall number of keys stored by the users (storage efficiency);
- In addition to storage, another performance metric is how long it takes for a user to get her query submitted and answered. Clearly, the greater the number d_c with whom the user shares a selected key, the shorter the expected waiting time (time efficiency);
- The risk that a user can profile and thereby re-identify another user decreases as d_u increases, since the user then distributes her queries to a wider subset of communication spaces (privacy in front of other users);
- The query profile of a particular user is diffused among the $d_u(d_c - 1)$ users with whom the user shares a key and confused among the other queries submitted by those users (privacy in front of database).

We therefore deduce that the privacy of the users against the database is an increasing function of $d_u(d_c - 1)$.

From (2) and from the fact that the privacy of the users facing the database is an increasing function of $d_u(d_c - 1)$, we deduce that the parameters for an optimal configuration for the P2P UPIR, considering the privacy against the database, should satisfy the following relation:

$$d_u(d_c - 1) = n_u - 1. \tag{4}$$

These configurations exist, as is shown in the following example.

Example. Let $n_u = 9$, $n_c = 12$, $d_u = 4$ and $d_c = 3$. Consider the following adjacency list of users and communication spaces:

u_1 : c_1 c_2 c_3 c_4
u_2 : c_1 c_5 c_6 c_7
u_3 : c_1 c_8 c_9 c_{10}
u_4 : c_2 c_5 c_8 c_{11}
u_5 : c_2 c_6 c_9 c_{12}
u_6 : c_3 c_5 c_{10} c_{12}
u_7 : c_3 c_7 c_9 c_{11}
u_8 : c_4 c_6 c_{10} c_{11}
u_9 : c_4 c_7 c_8 c_{12}

c_1 : u_1 u_2 u_3
c_2 : u_1 u_4 u_5
c_3 : u_1 u_6 u_7
c_4 : u_1 u_8 u_9
c_5 : u_2 u_4 u_6
c_6 : u_2 u_5 u_8
c_7 : u_2 u_7 u_9
c_8 : u_3 u_4 u_9
c_9 : u_3 u_5 u_7
c_{10} : u_3 u_6 u_8
c_{11} : u_4 u_7 u_8
c_{12} : u_5 u_6 u_9

In the following lemma we see what relation d_u and d_c should keep for these configurations.

Lemma 1. *Given a configuration with $d_u(d_c - 1) = n_u - 1$, we always have $d_c \leq d_u$.*

Proof. Suppose as before that the user u_1 is assigned the communication spaces $c_1, \ldots, c_{d_u}$. From the condition $d_u(d_c - 1) = n_u - 1$ we get that $u_2, \ldots, u_{n_u}$ are assigned to one and only one of the communication spaces $c_1, \ldots, c_{d_u}$. We also suppose, without loss of generality, that u_2 is assigned the communication spaces c_1 and c_j with $j > d_u$. Then each of the other $d_c - 1$ users assigned to the space c_j, should be assigned to another space between c_2 and c_{d_u}. Therefore $d_c - 1 \leq d_u - 1$ and the result follows. □

By the arguments at the beginning of the section, larger d_c's give better performance as for storage, and shorter expected waiting time per query. On the other hand, larger d_u's give more privacy against peers. If we only focus on privacy against the search engine then we are interested in configurations with the largest possible d_c. By Lemma 1 this means that

$$d_u = d_c. \tag{5}$$

Now by (1) we also have that $n_u = n_c$ and therefore we are dealing with symmetric configurations. Let $n := n_u = n_c$ and $d := d_u = d_c$. From the condition (4) we deduce that $n = d^2 - d + 1$ and we also have that every pair of users share one and only one communication space while every pair of communication spaces is assigned simultaneously to one and only one user. In the area of finite geometry these configurations are called projective planes [18] (see Section 4 for a formal definition). The order q of the projective plane corresponds to the value of $d - 1$. Hence the number of users (and memory sectors) in the configuration, i.e. the number of points (and lines) in the projective plane is $n = d^2 - d + 1 = q^2 + q + 1$.

We conclude that the optimal configurations for the peer-to-peer user-private information retrieval are, indeed, the projective planes. It is known that projective planes of order q exist whenever q is a power of a prime number, but when q is an integer in general the existence is not guaranteed. Actually there is not a single known example of a projective plane where q is not a power of a prime. In [18] it is specified that the existence of projective planes of arbitrary orders is one of the most difficult questions within finite geometry.

In our discussion we did not take into account the privacy against other peers. Alternative solutions for avoiding collusions of peers are analyzed in [19].

4. Construction of the optimal configurations

A standard construction of a finite projective plane of order q uses homogeneous coordinates over the finite field of order q. This method gives us one plane of every order q: $\mathbb{P}(\mathbb{F}_q)$. Many more projective planes can be constructed using algebraic structures with less postulates than fields. M. Hall defined the concept of *ternary ring* (see below) as the algebraic structure that exactly corresponds to the structure needed in the construction of projective planes [20,21].

Another very simple construction is given by the existence in some projective planes of the so-called Singer cycles. The projective planes that have Singer cycles can be constructed by defining the lines of the plane as the successive translations of a difference set. Apart from being a very simple construction (once given a difference set), it gives a compact way of representing the projective plane, since defining one of the lines is enough to deduce the entire plane. However, not all planes can be constructed in this way and, more important, constructing a difference set is equivalent to constructing a projective plane.

When q is prime there is a rather straightforward algorithm for constructing $\mathbb{P}(\mathbb{F}_q)$. We will here give one variant of this straightforward algorithm an efficient and explicit expression, at the same time generalizing it in order to be able to efficiently construct any projective plane (whenever it exists). In particular we construct projective planes of order a power of a prime number, using an algorithm that is efficient and of easy implementation. The generalization uses the mathematical structure ternary ring, which was first introduced by Hall as a tool for his construction and classification of projective planes [20]. Any ternary ring will give us a projective plane and any projective plane defines a ternary ring [22]. Ternary rings make possible a general formulation of the algorithm, and simplify the proof. The reader who only needs an efficient algorithm to calculate a projective plane, and does not care about fancy mathematical concepts, can take only a quick glance at Proposition 4, skip the proof and go straight to the examples.

Observe that the concept of ternary ring has little to do with the concept of a ring. A ternary ring has one ternary operation, while a ring has two binary operations.

Definition 2 (*Ternary Ring*). A ternary ring is a set R with two distinguished elements 0,1 and a ternary operation $T : R^3 \to R$ satisfying the following conditions:

- (T1) $T(1, a, 0) = T(a, 1, 0) = a$ for all $a \in R$;
- (T2) $T(a, 0, c) = T(0, a, c) = c$ for all $a, c \in R$;
- (T3) If $a, b, c \in R$, the equation $T(a, b, y) = c$ has a unique solution y;
- (T4) If $a, a', b, b' \in R$ and $a \neq a'$, the equations $T(x, a, b) = T(x, a', b')$ have a unique solution x in R;
- (T5) If $a, a', b, b' \in R$ and $a \neq a'$, the equations $T(a, x, y) = b$, $T(a', x, y) = b'$ have a unique solution x, y in R.

Since we are only interested in finite projective planes, and therefore only in finite ternary rings, we can forget about (T5). When R is finite, the condition (T5) is redundant.

Also it can be useful to have the definition of a projective plane fresh in memory while following the proof.

Definition 3. A projective plane is a set of elements called points, together with a family of subsets called lines, satisfying the following axioms:

- (P1) Any two distinct points belong to exactly one line;
- (P2) Any two distinct lines meet in exactly one point;
- (P3) There exists a quadrilateral; a set of four points, no three on any line.

We will represent a finite projective plane using an adjacency list, i.e. a list of the subsets defining the lines of the plane, using the set of integers $\{1, \ldots, n\}$ to represent the points.

The following defines a general and efficient algorithm for the construction of finite projective planes.

Proposition 4. *Let R be a ternary ring with q elements $R_0 = \{0, \ldots, R_{q-1}\}$ and ternary operation T. Let $\iota(R_i) = i$. Consider the matrix $A = (a_{i,j})$ defined by*

$$a_{i,j} = 2 + iq + j,$$

with $i \in \{0, \ldots, q\}$ and $j \in \{0, \ldots, q-1\}$, and the q matrices $B^0 = (b^0_{i,j})$, $B^1 = (b^1_{i,j})$, $\ldots$, $B^{q-1} = (b^{q-1}_{i,j})$ defined by

$$b^k_{i,j} = 2 + (j+1)q + \iota(T(R_j, R_k, R_i)),$$

with $i \in \{0, \ldots, q-1\}$ and $j \in \{0, \ldots, q-1\}$. The following matrix gives us an adjacency list defining a projective plane of order q:

$$C_q = \left(\begin{array}{c|c} 1 & \\ 1 & \\ \vdots & A \\ 1 & \\ \hline 2 & \\ 2 & \\ \vdots & B^0 \\ 2 & \\ \hline 3 & \\ 3 & \\ \vdots & B^1 \\ 3 & \\ \hline \vdots & \vdots \\ \vdots & \vdots \\ \hline q+1 & \\ q+1 & \\ \vdots & B^{q-1} \\ q+1 & \end{array}\right)$$

Proof. First we observe that in the $(q+1) \times q$ matrix A, all numbers in $\{2, \ldots, q^2+q+1\}$ appear once.

Now let A' be the matrix A, dropping the first row, i.e. the row containing the numbers $\{2, \ldots, p+1\}$. We observe that by property T2 the matrix B^0 is A' transposed, and therefore no two numbers appearing in the same row in A' (and hence in A) can appear in the same row in B^0.

Consider now the construction of the matrices B^k. In every column b_j of B^k, the elements

$$b^k_{i,j} = 2 + (j+1)q + \iota(T(R_j, R_k, R_i))$$

by property T3 take all the values in

$$\{2 + (j+1)q, \ldots, 1 + (j+2)q\}, \tag{6}$$

as i goes through $\{0, \ldots, q-1\}$. We will call (6) the element sequence of column b_j.

Each of these sequences corresponds to the elements in a row of A', so if two numbers appear in the same row in A they can not appear in the same row in B^k for any k.

On the other hand, all the sequences are disjoint, so all numbers in $\{q+2, \ldots, q^2+q+1\}$ appear exactly once in each B^k. Consequently, the only case in which two elements may still coincide in more than two rows is when the two rows belong to two different minors

$$\left(\begin{array}{c|c} \vdots & \\ k+2 & B^k \\ \vdots & \end{array}\right)$$

and

$$\left(\begin{array}{c|c} \vdots & \\ k'+2 & B^{k'} \\ \vdots & \end{array}\right)$$

with $k \neq k'$.

Suppose that these two rows are the ith row in $(k+2|B^k)$ and the i'th row in $(k'+2|B^{k'})$. Let the two repeated elements be b^k_{ij} and $b^k_{ij'}$, with $j \neq j'$. Since the set of elements of the jth column in B^k is the same as the set of elements of the jth column in $B^{k'}$, and the analogous case is true for the j'th column, we must have $b^k_{ij} = b^{k'}_{i'j}$ and $b^k_{ij'} = b^{k'}_{i'j'}$, i.e. $T(R_j, R_k, R_i) = T(R_j, R'_k, R'_i)$. By condition T4 it must be $j = j'$, a contradiction. □

Proposition 5. *The computational cost of the algorithm given by Proposition 4 is $O(q^3)$, provided that q is prime and that the ternary ring we use corresponds to the arithmetic of $\mathbb{F}_q$. In particular the number of operations used in each case is*

- *Additions: $q + q^2 + 2q^3$;*
- *Multiplications: $q + q^2 + 2q^3$;*
- *Modulo operations: q^3.*

Proof. The matrix A needs $q(q+1)$ multiplications and $2q(q+1)$ additions. If we do not count addition of the constant 2 we get $q(q+1)$ additions. The matrix B^k needs $2q^2$ multiplications, q^2 modulo operations and $4q^2$ additions. If we continue not counting addition of the constants 1 and 2 we get $2q^2$ additions. Considering that $k \in \{0, \ldots, q-1\}$ we get $q(q+1)+q(2q^2) = q+q^2+2q^3$ multiplications, $q(q+1)+q(2q^2) = q+q^2+2q^3$ additions and q^3 modulo operations. □

In order to clarify how to implement the algorithm, we will now see some examples. In the following, if nothing else is said, the operations $+$ and $\cdot$ stand for ordinary integer sum and product.

Example. $(\mathbb{Z}/p\mathbb{Z}, +, \cdot)$, the integers modulo a prime number p, give rise to a ternary ring with ternary operation $T(x, y, z) = xy + z \pmod p$. With the notation from Proposition 4 we calculate $A = (a_{i,j})$ using

$$a_{i,j} = 2 + ip + j,$$

and for $k \in \{0, \ldots, p-1\}$ we calculate $B^k = (b^k_{i,j})$ using

$$b^k_{i,j} = 2 + (j+1)p + [i + jk \pmod p],$$

with $i \in \{0, \ldots, p-1\}$ and $j \in \{0, \ldots, p-1\}$.

We now present the results from this construction for $p = 2$, $p = 3$.

$$C_2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 4 & 5 \\ 1 & 6 & 7 \\ 2 & 4 & 6 \\ 2 & 5 & 7 \\ 3 & 4 & 7 \\ 3 & 5 & 6 \end{pmatrix} \qquad C_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 \\ 1 & 8 & 9 & 10 \\ 1 & 11 & 12 & 13 \\ 2 & 5 & 8 & 11 \\ 2 & 6 & 9 & 12 \\ 2 & 7 & 10 & 13 \\ 3 & 5 & 9 & 13 \\ 3 & 6 & 10 & 11 \\ 3 & 7 & 8 & 12 \\ 4 & 5 & 10 & 12 \\ 4 & 6 & 8 & 13 \\ 4 & 7 & 9 & 11 \end{pmatrix}.$$

Example. The integers modulo p prime is a finite field of order prime. For every $q = p^n$ power of a prime, there is a finite field $\mathbb{F}_q$ generalizing the nice behavior of $\mathbb{Z}/p\mathbb{Z}$. A finite field $\mathbb{F}_q$ defines a ternary ring with ternary operation $T(x, y, z) = xy + z$, where the sum and the product follows the arithmetic rules of $\mathbb{F}_q$. With the notation from Proposition 4 we calculate $A = (a_{i,j})$ using

$$a_{i,j} = 2 + iq + j.$$

We represent the elements of $\mathbb{F}_q$ with the integers in the array $F = (0, \ldots, q-1)$. With the zero of the field represented as 0 and the unit represented as 1, the elements of the matrices B^k, for $k \in \{0, \ldots, q-1\}$, can now be calculated as

$$b^k_{i,j} = 2 + (j+1)q + [F_i + F_{k-1}F_j],$$

with $i \in \{0, \ldots, q-1\}$ and $j \in \{0, \ldots, q-1\}$. The arithmetic of the elements of the array F must follow the arithmetic rules of $\mathbb{F}_q$.

Observe that the previous example is a special case of this example.

We now present the result from this construction for $q = 4$.

$$C_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 6 & 7 & 8 & 9 \\ 1 & 10 & 11 & 12 & 13 \\ 1 & 14 & 15 & 16 & 17 \\ 1 & 18 & 19 & 20 & 21 \\ 2 & 6 & 10 & 14 & 18 \\ 2 & 7 & 11 & 15 & 19 \\ 2 & 8 & 12 & 16 & 20 \\ 2 & 9 & 13 & 17 & 21 \\ 3 & 6 & 11 & 16 & 21 \\ 3 & 7 & 10 & 17 & 20 \\ 3 & 8 & 13 & 14 & 19 \\ 3 & 9 & 12 & 15 & 18 \\ 4 & 6 & 12 & 17 & 19 \\ 4 & 7 & 13 & 16 & 18 \\ 4 & 8 & 10 & 15 & 21 \\ 5 & 6 & 13 & 15 & 20 \\ 5 & 7 & 12 & 14 & 21 \\ 5 & 8 & 11 & 17 & 18 \\ 5 & 9 & 10 & 16 & 19 \end{pmatrix}.$$

In this article the focus is on constructing optimal configurations for P2P UPIR, and it is probably enough with one configuration for a given $d = q - 1$. It is conjectured that all projective planes have order a power of a prime number. Therefore it is highly probable that all projective planes constructed by our algorithm will have this property. The finite projective planes constructed using finite fields constitute a subset of all finite projective planes, with the particularity that they satisfy the theorem of Desargues. Although the existence of finite fields is restricted to q a power of a prime number, since there always exists one finite field for every q, we will always get at least one projective plane of order q, using the previous example. It is therefore of little interest to continue the examples further.

Observe though that some projective planes constructed using less 'regular' (i.e. satisfying less axioms) ternary rings could be interesting when some properties associated to the theorem of Desargues are to be avoided.

5. The optimal configurations are Ramanujan graphs

A network is efficient if the information held by a particular node can be transmitted in a fast way to all the other nodes. To formulate this idea mathematically, the network is represented by a graph and every node by a vertex in the graph. Two vertices are adjacent if the corresponding nodes have direct communication.

Of course, the information is transmitted with the highest efficiency when the network is a complete graph, that is, when every pair of nodes is connected. But from a practical point of view, the complete connection is not viable. Rather, it is to prefer if the graph has few edges. The objective then is to maximize the border of every set of vertices, where the border of the set of vertices V is the set of vertices not in V being adjacent to some vertex in V. Graphs with a good relation between the border and the number of vertices for every subset V are called good expander graphs. There are plenty of applications for good expanders graphs. Some of these are construction of error correcting codes, construction of intense access networks, generation of pseudo-random numbers, randomized and derandomized algorithms, computational complexity theory and parallel architectures [23].

Given a graph with m vertices, its adjacency matrix is the square and symmetric matrix $A = (a_{i,j})$ of dimension $m \times m$ such that $a_{i,j} = 1$ if there is an edge between the ith and the jth vertex and $a_{i,j} = 0$ if there is not. The expansion degree of a graph has to do with the eigenvalues of its adjacency matrix. If a graph is d-regular (every vertex is connected to exactly d other vertices) and connected, then the eigenvalues of its adjacency matrix $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{m-1}$ satisfy $d = \lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{m-1} \geq -d$. Ramanujan graphs are a very important example of good expanders.

Definition 6. Given a connected and d-regular graph G with eigenvalues of its adjacency matrix $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{m-1}$ we define $\lambda(G) = \max_{|\lambda_i|<d} |\lambda_i|$. We say that the graph is Ramanujan if $\lambda(G)$ is defined and $\lambda(G) \leq 2\sqrt{d-1}$.

Lubotzky, Phillips and Sarnak [24] described a method to construct infinite families of Ramanujan graphs using quotients of quaternions and defining the Cayley graph with vertices in $\mathrm{PGL}_2(\mathbb{F}_q)$ for q prime. One extension of this construction was elaborated by Morgenstern [25] for q power of a prime. In general, the constructions in the existing literature use quite complex mathematical concepts, not suitable for an engineering context.

The configurations defined by the projective planes (hence optimal configurations for P2P UPIR) are a particular case of the graphs treated in [26] and, as is proved there, they therefore are Ramanujan graphs. However we will with the following give a direct and simple proof of this property.

Consider a projective plane with $n = q^2 + q + 1$ points and the same number of lines, so that there are $d = q + 1$ points on every line and $q + 1$ lines passing through every point, with no two (different) points on more than one line.

Consider a bipartite graph with $n + n$ vertices corresponding to the n points together with the n lines, so that the vertex that corresponds to a point is incident exactly with the vertices corresponding to the lines containing the point.

Lemma 7. *Let A be the adjacency matrix of the bipartite graph corresponding to a projective plane of order q. We have*

$$A^2 = \begin{pmatrix} q+1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 \\ 1 & q+1 & \dots & 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & q+1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & q+1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 & 1 & q+1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & q+1 \end{pmatrix}.$$

Proof. This is because the square of the adjacency matrix of a graph, in position i, j has the number of paths of length two between the ith and the jth vertex. □

Lemma 8. *The eigenvalues of the matrix A^2 in the previous lemma are $(q + 1)^2$ and q.*

Proof. The vectors

$$(\underbrace{0, \dots, 0, 1^{(j)}, 0, \dots, 0, -1}_{q^2+q+1}, \underbrace{0, \dots, 0}_{q^2+q+1}),$$

with $j \in 1, \dots, q^2 + q$ and

$$(\underbrace{0, \dots, 0}_{q^2+q+1}, \underbrace{0, \dots, 0, 1^{(j)}, 0, \dots, 0, -1}_{q^2+q+1}),$$

with $j \in 1, \dots, q^2 + q$, are all linearly independent and they are eigenvectors for the eigenvalue q. On the other hand the vectors

$$(\underbrace{1, \dots, 1}_{q^2+q+1}, \underbrace{0, \dots, 0}_{q^2+q+1})$$
$$(\underbrace{0, \dots, 0}_{q^2+q+1}, \underbrace{1, \dots, 1}_{q^2+q+1})$$

are linearly independent and also linearly independent to the previous eigenvectors, being eigenvectors for the eigenvalue $(q + 1)^2$. As these vectors together form a base, there are no more eigenvalues. □

Theorem 9. *The bipartite graph corresponding to a projective plane is a Ramanujan graph.*

Proof. Let A be the adjacency matrix of the bipartite graph corresponding to a projective plane. The squares of the eigenvalues of A must be eigenvalues of A^2. From Lemma 8 we get that the only possible eigenvalues of A are $\pm(q + 1)$ and $\pm\sqrt{q}$. □

Observe that since A is an adjacency matrix of a bipartite graph, it has a symmetric spectrum with respect to the origin. Hence the eigenvalues of A are exactly

- $q + 1$ (with multiplicity 1),
- $-(q + 1)$ (with multiplicity 1),
- $\sqrt{q}$ (with multiplicity $q^2 + q$) and finally
- $-\sqrt{q}$ (with multiplicity $q^2 + q$).

Since we already proved the optimality of the projective planes as configurations for the P2P UPIR, the fact that they are also Ramanujan graphs may seem to be superfluous knowledge (in the context of P2P UPIR). However, it is useful knowledge considering that it helps to understand the optimality from another point of view. Also, knowing that the projective planes have the Ramanujan property is valuable per se.

6. Conclusion

In this article we have proved that the optimal configurations for the P2P UPIR protocol presented in [1,14] are the finite projective planes. We have also presented an efficient and explicit algorithm for the construction of finite projective planes. Finally we have given another aspect of the optimality of finite projective planes; giving a short proof of the fact that they are Ramanujan graphs.

Acknowledgements

This work was partly supported by the Spanish Government through projects TIN2009-11689 "RIPUP" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", and by the Government of Catalonia under grant 2009 SGR 1135. The authors would like to thank Josep Domingo-Ferrer for many helpful discussions.

References

[1] J. Domingo-Ferrer, M. Bras-Amorós, Peer-to-peer private information retrieval, in: J. Domingo-Ferrer, Y. Saygin (Eds.), Privacy in Statistical Databases, in: Lecture Notes in Computer Science, vol. 5262, Springer, 2008, pp. 315–323.
[2] B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, Private information retrieval, in: IEEE Symposium on Foundations of Computer Science, FOCS, 1995, pp. 41–50.
[3] B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, Private information retrieval, Journal of the ACM 45 (1998) 965–981.
[4] B. Chor, N. Gilboa, Computationally private information retrieval, in: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, STOC'97, El Paso, Texas, May 4–6, 1997, ACM Press, New York, 1997, pp. 304–313.
[5] E. Kushilevitz, R. Ostrovsky, Replication is not needed: Single database, computationally-private information retrieval, in: Proc. of the 38th Annual IEEE Symposium on Foundations of Computer Science, 1997, pp. 364–373.
[6] C. Aguilar-Melchor, P. Gaborit, Single-database private information retrieval protocols: Overview, usability and trends, Research Report, 2007.
[7] W. Gasarch, A. Yerukhimovich, Computational inexpensive PIR, 2006. http://www.cs.umd.edu/arkady/papers/pirlattice.pdf.
[8] M. Rennhard, B. Plattner, Practical anonymity for the masses with MorphMix, Financial Cryptography (2004) 233–250.
[9] M.J. Freedman, R. Morris, Tarzan: A peer-to-peer anonymizing network layer, in: ACM Conference on Computer and Communications Security, CCS02, ACM Press, Washington, DC, USA, 2002, pp. 193–206.
[10] C. Aguilar-Melchor, Y. Deswarte, Trustable relays for anonymous communication, Transactions on Data Privacy 2 (2) (2009) 101–130.
[11] The Tor project, inc. Tor: Overview. http://torproject.org/overview.html.en.
[12] J. Domingo-Ferrer, A. Solanas, J. Castellà-Roca, $h(k)$-private information retrieval from privacy-uncooperative queryable databases, Online Information Review 33 (4) (2009) 720–744.
[13] D.C. Howe, H. Nissenbaum, Trackmenot: Resisting surveillance in web search, in: I. Kerr, C. Lucock, V. Steeves (Eds.), Lessons from the Identity Trail: Privacy, Anonymity and Identity in a Networked Society, Oxford University Press, Oxford, UK, 2009. Software downloadable from: http://www.mrl.nyu.edu/dhowe/trackmenot/.
[14] J. Domingo-Ferrer, M. Bras-Amorós, Q. Wu, J. Manjón, User-private information retrieval based on a peer-to-peer community, Data & Knowledge Engineering 68 (11) (2009) 1237–1252.
[15] H. Gropp, Configurations, in: Charles J. Colbourn, Jeffrey H. Dinitz (Eds.), in: Handbook of Combinatorial Designs, Chapman and Hall/CRC, Kenneth H. Rosen, 2007, pp. 353–355.
[16] C.J. Colbourn, J.H. Dinitz, Handbook of Combinatorial Designs, Chapman and Hall/CRC, Kenneth H. Rosen, 2007.
[17] M. Bras-Amorós, K. Stokes, On the existence of combinatorial configurations, arXiv:0907.4230v2.
[18] L. Storme, Finite geometry, in: Charles J. Colbourn, Jeffrey H. Dinitz (Eds.), Handbook of Combinatorial Designs, Chapman and Hall/CRC, Kenneth H. Rosen, 2007, pp. 702–729.
[19] M. Bras-Amorós, K. Stokes, M. Greferath, Using (0,1)-geometries for collusion-free P2P user private information retrieval, in: 19th International Symposium on Mathematical Theory of Networks and Systems, Budapest, 2010.
[20] M. Hall, Projective planes, Transactions of the American Mathematical Society 54 (2) (1943) 229–277.
[21] M. Hall, The Theory of Groups, The Macmillan Co., New York, NY, 1959.
[22] C. Weibel, Survey of non-desarguesian planes, Notices of the AMS 54 (10) (2007).
[23] Fan R.K. Chung, Spectral Graph Theory, American Mathematical Society Press, Providence, RI, 1997.
[24] A. Lubotzky, R. Phillips, P. Sarnak, Ramanujan graphs, Combinatorica 8 (3) (1988) 261–277.
[25] M. Morgenstern, Existence and explicit constructions of q+1 regular Ramanujan graphs for every prime power q, Journal of Combinatorial Theory. Series B 62 (1) (1994) 44–62.
[26] T. Høholdt, H. Janwa, Optimal bipartite Ramanujan graphs from balanced incomplete block designs: Their characterizations and applications to expander/ LDPC codes, in: M. Bras-Amorós, T. Høholdt (Eds.), Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, AAECC-18, in: Lecture Notes in Computer Science, vol. 5527, Springer, 2009, pp. 53–64.