User-Private Information Retrieval

Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo

PST2011
Ninth Annual Conference on Privacy, Security and Trust
Wednesday July 20, 2011

Cette recherche a été effectuée en collaboration avec Colleen Swanson.

définitions

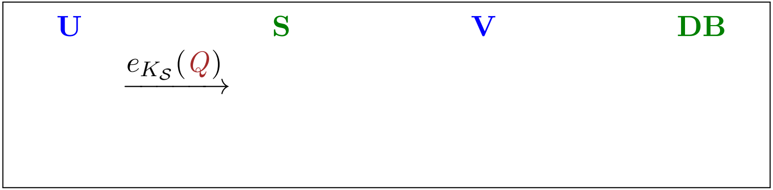
- Private information retrieval (PIR) involves hiding the content of queries from a database. The identity of the person making the query is not protected.
- User-private information retrieval (UPIR) involves hiding the identity of the person making the query. The content of the query is not protected.
- UPIR is a mechanism to provide anonymity (similar to Tor).
- The setting for UPIR is a co-operating community of users who act as proxies (mandataires) to submit each others' queries to the database.
- We investigate combinatorial techniques to enable UPIR, following the model introduced by Domingo-Ferrer and Bras-Amorós [1] and studied further in [2, 3, 4].

circulation de l'information

U **S** **V** **DB**

- **U** and **V** are two users (we allow $\mathbf{U} = \mathbf{V}$)
- **U** is the source of the query Q and **V** is the proxy.
- **S** is a memory space (e.g., a secure dropbox) to which **U** and **V** belong and K_S is a secret key known to all users associated with **S**
- **DB** is the database and R is the response to the query Q
- K_V is a secret key known to **V** and **DB**

circulation de l'information



U **S** **V** **DB**

$\xrightarrow{e_{K_{\mathcal{S}}}(Q)}$

- **U** and **V** are two users (we allow **U** = **V**)
- **U** is the source of the query Q and **V** is the proxy.
- **S** is a memory space (e.g., a secure dropbox) to which **U** and **V** belong and K_S is a secret key known to all users associated with **S**
- **DB** is the database and R is the response to the query Q
- K_V is a secret key known to **V** and **DB**

circulation de l'information

$$\mathbf{U} \xrightarrow{e_{K_{\mathcal{S}}}(Q)} \mathbf{S} \xrightarrow{e_{K_S}(Q)} \mathbf{V} \qquad \mathbf{DB}$$

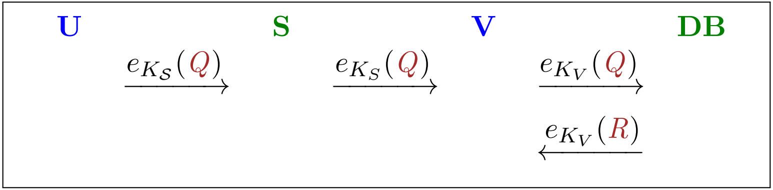
- **U** and **V** are two users (we allow $\mathbf{U} = \mathbf{V}$)
- **U** is the source of the query Q and **V** is the proxy.
- **S** is a memory space (e.g., a secure dropbox) to which **U** and **V** belong and K_S is a secret key known to all users associated with **S**
- **DB** is the database and R is the response to the query Q
- K_V is a secret key known to **V** and **DB**

circulation de l'information

$$\mathbf{U} \xrightarrow{e_{K_{\mathcal{S}}}(Q)} \mathbf{S} \xrightarrow{e_{K_S}(Q)} \mathbf{V} \xrightarrow{e_{K_V}(Q)} \mathbf{DB}$$

- $\mathbf{U}$ and $\mathbf{V}$ are two users (we allow $\mathbf{U} = \mathbf{V}$)
- $\mathbf{U}$ is the source of the query Q and $\mathbf{V}$ is the proxy.
- $\mathbf{S}$ is a memory space (e.g., a secure dropbox) to which $\mathbf{U}$ and $\mathbf{V}$ belong and K_S is a secret key known to all users associated with $\mathbf{S}$
- $\mathbf{DB}$ is the database and R is the response to the query Q
- K_V is a secret key known to $\mathbf{V}$ and $\mathbf{DB}$

circulation de l’information



$$\begin{tikzcd}
\mathbf{U} \arrow[r, "e_{K_{\mathcal{S}}}(Q)"] & \mathbf{S} \arrow[r, "e_{K_S}(Q)"] & \mathbf{V} \arrow[r, shift left, "e_{K_V}(Q)"] & \mathbf{DB} \arrow[l, shift left, "e_{K_V}(R)"]
\end{tikzcd}$$

- $\mathbf{U}$ and $\mathbf{V}$ are two users (we allow $\mathbf{U} = \mathbf{V}$)
- $\mathbf{U}$ is the source of the query Q and $\mathbf{V}$ is the proxy.
- $\mathbf{S}$ is a memory space (e.g., a secure dropbox) to which $\mathbf{U}$ and $\mathbf{V}$ belong and K_S is a secret key known to all users associated with $\mathbf{S}$
- $\mathbf{DB}$ is the database and R is the response to the query Q
- K_V is a secret key known to $\mathbf{V}$ and $\mathbf{DB}$

circulation de l'information

$$\begin{tikzcd} \mathbf{U} \arrow[r, "e_{K_{\mathcal{S}}}(Q)"] & \mathbf{S} \arrow[r, shift left, "e_{K_S}(Q)"] & \mathbf{V} \arrow[l, shift left, "e_{K_S}(R)"] \arrow[r, shift left, "e_{K_V}(Q)"] & \mathbf{DB} \arrow[l, shift left, "e_{K_V}(R)"] \end{tikzcd}$$

- $\mathbf{U}$ and $\mathbf{V}$ are two users (we allow $\mathbf{U} = \mathbf{V}$)
- $\mathbf{U}$ is the source of the query Q and $\mathbf{V}$ is the proxy.
- $\mathbf{S}$ is a memory space (e.g., a secure dropbox) to which $\mathbf{U}$ and $\mathbf{V}$ belong and K_S is a secret key known to all users associated with $\mathbf{S}$
- $\mathbf{DB}$ is the database and R is the response to the query Q
- K_V is a secret key known to $\mathbf{V}$ and $\mathbf{DB}$

circulation de l'information

$$\begin{tikzcd}
\mathbf{U} \arrow[r, shift left=2, "e_{K_{\mathcal{S}}}(Q)"] & \mathbf{S} \arrow[l, shift left=2, "e_{K_S}(R)"] \arrow[r, shift left=2, "e_{K_S}(Q)"] & \mathbf{V} \arrow[l, shift left=2, "e_{K_S}(R)"] \arrow[r, shift left=2, "e_{K_V}(Q)"] & \mathbf{DB} \arrow[l, shift left=2, "e_{K_V}(R)"]
\end{tikzcd}$$

- $\mathbf{U}$ and $\mathbf{V}$ are two users (we allow $\mathbf{U} = \mathbf{V}$)
- $\mathbf{U}$ is the source of the query Q and $\mathbf{V}$ is the proxy.
- $\mathbf{S}$ is a memory space (e.g., a secure dropbox) to which $\mathbf{U}$ and $\mathbf{V}$ belong and K_S is a secret key known to all users associated with $\mathbf{S}$
- $\mathbf{DB}$ is the database and R is the response to the query Q
- K_V is a secret key known to $\mathbf{V}$ and $\mathbf{DB}$

objectifs et assomptions

There are three primary objectives:

1. User-anonymity WRT the database.
2. User-anonymity WRT other users.
3. Confidentiality WRT external observers.

We make the following assumptions (attack model):

- The database and the users in the scheme do not observe information being posted to or read from memory spaces (no traffic analysis).
- Users are honest-but-curious and they may collaborate in an attempt to compromise the anonymity of other users.
- The database is also honest and does not collaborate with any users.

confidentialité contre adversaires externes

- In order to provide confidentiality against external adversaries, all queries and responses are encrypted.
- All information posted to or read from a memory space $\mathbf{S}$ is encrypted with a secret key K_S known only to the users associated with $\mathbf{S}$.
- Every user shares a different secret key with the database; these secret keys are used to encrypt queries sent to the database and the database's responses.
- Due to the possibility of compromise of a memory space key $K_{\mathcal{S}}$, we do not want any individual memory space to contain "too many" users. In particular, one memory space containing all users is inadvisable. Therefore each user will be associated with multiple memory spaces, each of which contains a "small" number of users.

structure combinatoire

- We model a UPIR scheme as a set system (combinatorial design) that satisfies two regularity conditions:
    - $\mathcal{U} = \{\mathbf{U}_1, \ldots, \mathbf{U}_v\}$ denotes the set of v users.
    - $\mathcal{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_b\}$ denotes b memory spaces.
    - each memory space consists of k users.
    - each user is associated with r memory spaces
- Suppose we regard each memory space (termed blocks) as a subset of k users (termed points).
- The pair $(\mathcal{U}, \mathcal{S})$ is a (v, b, r, k)-1-design. In a (v, b, r, k)-1-design, we have $vr = bk$.
- Alternatively, we can treat the b memory spaces as points and then define v blocks, each of which contains the memory spaces to which a given user belongs. This yields the dual design $(\mathcal{S}, \mathcal{U})$, which is a (b, v, k, r)-1-design.

exemple

Suppose $v = 12$, $b = 8$ and the design obtained from the memory spaces is the following:

$$\begin{array}{lll} \mathbf{S}_1 = \{\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3\} & \mathbf{S}_2 = \{\mathbf{U}_4, \mathbf{U}_5, \mathbf{U}_6\} & \mathbf{S}_3 = \{\mathbf{U}_7, \mathbf{U}_8, \mathbf{U}_9\} \\ \mathbf{S}_4 = \{\mathbf{U}_{10}, \mathbf{U}_{11}, \mathbf{U}_{12}\} & \mathbf{S}_5 = \{\mathbf{U}_1, \mathbf{U}_4, \mathbf{U}_7\} & \mathbf{S}_6 = \{\mathbf{U}_2, \mathbf{U}_5, \mathbf{U}_{10}\} \\ \mathbf{S}_7 = \{\mathbf{U}_3, \mathbf{U}_8, \mathbf{U}_{11}\} & \mathbf{S}_8 = \{\mathbf{U}_6, \mathbf{U}_9, \mathbf{U}_{12}\} & \end{array}$$

This is a $(12, 8, 2, 3)$-1-design.

The dual design is:

$$\begin{array}{lll} \mathbf{U}_1 = \{\mathbf{S}_1, \mathbf{S}_5\} & \mathbf{U}_2 = \{\mathbf{S}_1, \mathbf{S}_6\} & \mathbf{U}_3 = \{\mathbf{S}_1, \mathbf{S}_7\} \\ \mathbf{U}_4 = \{\mathbf{S}_2, \mathbf{S}_5\} & \mathbf{U}_5 = \{\mathbf{S}_2, \mathbf{S}_6\} & \mathbf{U}_6 = \{\mathbf{S}_2, \mathbf{S}_8\} \\ \mathbf{U}_7 = \{\mathbf{S}_3, \mathbf{S}_5\} & \mathbf{U}_8 = \{\mathbf{S}_3, \mathbf{S}_7\} & \mathbf{U}_9 = \{\mathbf{S}_3, \mathbf{S}_8\} \\ \mathbf{U}_{10} = \{\mathbf{S}_4, \mathbf{S}_6\} & \mathbf{U}_{11} = \{\mathbf{S}_4, \mathbf{S}_7\} & \mathbf{U}_{12} = \{\mathbf{S}_4, \mathbf{S}_8\} \end{array}$$

This is an $(8, 12, 3, 2)$-1-design.

questions reliées

- Suppose there is a series of linked queries on a similar, esoteric topic.
- Assuming that the linked queries all have the same source, it might be possible to deduce the source by means of an intersection attack.
- For example, suppose that there are three linked queries Q_1, Q_2, Q_3 having proxies $\mathbf{U}_2$, $\mathbf{U}_{11}$ and $\mathbf{U}_8$, respectively.
- If the proxy is $\mathbf{U}_2$, then the source is in
$$\mathbf{S}_1 \cup \mathbf{S}_6 = \{\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_5, \mathbf{U}_{10}\}.$$
- If the proxy is $\mathbf{U}_{11}$, then the source is in
$$\mathbf{S}_4 \cup \mathbf{S}_7 = \{\mathbf{U}_3, \mathbf{U}_8, \mathbf{U}_{10}, \mathbf{U}_{11}, \mathbf{U}_{12}\}.$$
- If the proxy is $\mathbf{U}_8$, then the source is in
$$\mathbf{S}_3 \cup \mathbf{S}_7 = \{\mathbf{U}_3, \mathbf{U}_7, \mathbf{U}_8, \mathbf{U}_9, \mathbf{U}_{11}\}.$$
- Because the three queries are linked, the source is in the intersection of these three sets, so the source is $\mathbf{U}_3$.

configurations

- It has been suggested to use a special type of design known as a configuration to realise UPIR.
- A configuration is a (v, b, r, k)-1-design satisfying the additional property that any two distinct blocks intersect in at most one point (equivalently, every pair of points occur in at most one block).
- In a (v, b, r, k)-configuration, $v \geq r(k-1)+1$ and $b \geq k(r-1)+1$.
- Configurations with $v > r(k-1)+1$ are susceptible to the intersection attack.
- If $v = r(k-1)+1$ in a configuration, then every pair of points occur in exactly one block; such a design will resist the intersection attack.

BIBDs

- We define a more general class of designs that resist the intersection attack.
- A (v, b, r, k, λ)-balanced incomplete block design (or BIBD) is a (v, b, r, k)-1-design in which every pair of points occurs in exactly λ blocks.
- A (v, b, r, k)-configuration with $v = r(k-1)+1$ is a $(v, b, r, k, 1)$-BIBD.
- A $(v, b, r, k, 1)$-BIBD having parameters $(n^2+n+1, n^2+n+1, n+1, n+1, 1)$ is a finite projective plane of order n.

résultats précédents et commentaires

- In previous work on UPIR, it has mainly been suggested to use configurations (especially, projective planes [3]) to implement the schemes.
- Configurations were proposed as key rings in wireless sensor networks by Lee and Stinson due to memory constraints of sensor nodes – a configuration maximises network connectivity when k and r are “small” relative to v and b. However, this is not so much an issue in UPIR.
- Initial protocols for UPIR did not allow any user to submit his or her own query to the database (i.e., the proxy is never the source). Note that this already gives the database some partial information about the source.
- It was also observed in [4] that the existence of sufficiently many linked queries in a projective plane scheme could allow the source to be identified, since every user except the source will eventually act as a proxy for the source.

notre stratégie

- In previous schemes, the proxy for a query is just the "next person" to visit a given memory space.
- We propose that each source designates the proxy for each query. This enables us to balance the proxies for each possible source.
- We present a scheme based on a (v, b, r, k, λ)-BIBD. Here is the protocol for user $\mathbf{U}_i$ to submit a query:
  1. With probability $1/v$, user $\mathbf{U}_i$ acts as his own proxy and transmits his own query to the database.
  2. Otherwise, user $\mathbf{U}_i$ chooses uniformly at random one of the r memory spaces (blocks) with which he is associated, say $\mathbf{S}_h$, and then he chooses uniformly at random a user $\mathbf{U}_j \in \mathbf{S}_h \backslash \{\mathbf{U}_i\}$. User $\mathbf{U}_i$ requests that user $\mathbf{U}_j$ acts as his proxy using the memory space $\mathbf{S}_h$.

exemple

Suppose we use a $(13, 13, 4, 4, 1)$-BIBD (a projective plane of order 3). The memory spaces are as follows:

$$\begin{array}{ll}
\mathbf{S}_0 = \{\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_3, \mathbf{U}_9\} & \mathbf{S}_1 = \{\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_4, \mathbf{U}_{10}\} \\
\mathbf{S}_2 = \{\mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_5, \mathbf{U}_{11}\} & \mathbf{S}_3 = \{\mathbf{U}_3, \mathbf{U}_4, \mathbf{U}_6, \mathbf{U}_{12}\} \\
\mathbf{S}_4 = \{\mathbf{U}_4, \mathbf{U}_5, \mathbf{U}_7, \mathbf{U}_0\} & \mathbf{S}_5 = \{\mathbf{U}_5, \mathbf{U}_6, \mathbf{U}_8, \mathbf{U}_1\} \\
\mathbf{S}_6 = \{\mathbf{U}_6, \mathbf{U}_7, \mathbf{U}_9, \mathbf{U}_2\} & \mathbf{S}_7 = \{\mathbf{U}_7, \mathbf{U}_8, \mathbf{U}_{10}, \mathbf{U}_3\} \\
\mathbf{S}_8 = \{\mathbf{U}_8, \mathbf{U}_9, \mathbf{U}_{11}, \mathbf{U}_4\} & \mathbf{S}_9 = \{\mathbf{U}_9, \mathbf{U}_{10}, \mathbf{U}_{12}, \mathbf{U}_5\} \\
\mathbf{S}_{10} = \{\mathbf{U}_{10}, \mathbf{U}_{11}, \mathbf{U}_0, \mathbf{U}_6\} & \mathbf{S}_{11} = \{\mathbf{U}_{11}, \mathbf{U}_{12}, \mathbf{U}_1, \mathbf{U}_7\} \\
\mathbf{S}_{12} = \{\mathbf{U}_{12}, \mathbf{U}_0, \mathbf{U}_2, \mathbf{U}_8\} &
\end{array}$$

As an example, suppose that $\mathbf{U}_0$ is the source. $\mathbf{U}_0$ uses $\mathbf{S}_0$, $\mathbf{S}_4$, $\mathbf{S}_{10}$ and $\mathbf{S}_{12}$ each with probability $3/13$. Then every user is the proxy with probability $1/13$.

propriétés de notre plan

- We analyse the situation from the point of view of the database.
- First, the scheme ensures that

$$\Pr[\mathbf{P} = \mathbf{U}_j | \mathbf{O} = \mathbf{U}_i] = \frac{1}{v}$$

  for all $\mathbf{U}_i, \mathbf{U}_j$ ($\mathbf{P}$ denotes the proxy and $\mathbf{O}$ denotes the source).
- For all $\mathbf{U}_j$, it follows that

$$\Pr[\mathbf{P} = \mathbf{U}_j] = \frac{1}{v}.$$

- Now we have

$$\begin{aligned} \Pr[\mathbf{O} = \mathbf{U}_i | \mathbf{P} = \mathbf{U}_j] &= \frac{\Pr[\mathbf{P} = \mathbf{U}_j | \mathbf{O} = \mathbf{U}_i] \Pr[\mathbf{O} = \mathbf{U}_i]}{\Pr[\mathbf{P} = \mathbf{U}_j]} \\ &= \Pr[\mathbf{O} = \mathbf{U}_i], \end{aligned}$$

  so the identity of the proxy gives no information about the identity of the source.

propriétés de notre plan (cont.)

- The mathematics is analogous to Shannon's analysis of perfect secrecy for an encryption scheme (e.g., the one-time pad).
- Because we have achieved a perfect anonymity property, it follows that there is no information obtained by analysing linked queries.
- Observe that this analysis is independent of any computational assumptions, so the security is unconditional.

extensions

We consider some extensions and generalisations of the basic approach in the remaining time:

1. Using less structured types of designs than BIBDs.
2. Techniques for dynamic UPIR schemes, where new users can join and old users can leave the scheme.
3. Investigate anonymity WRT other users. Note that perfect anonymity is not possible, since any user in a scheme knows that a query posted to a memory space must have a source who is associated with that memory space (i.e., the source is one of only $k-1$ possible users).

covering designs

- The anonymity proof works provided that $\Pr[\mathbf{P} = \mathbf{U}_j | \mathbf{O} = \mathbf{U}_i] = 1/v$ for all $\mathbf{U}_i, \mathbf{U}_j$.
- We do not need a BIBD in order to ensure this property holds.
- We can use any covering design, i.e., a set system in which every pair of points occurs in at least one block.
- Here is a covering design with 5 points, and 4 blocks of size 3:

$$\begin{array}{ll} \mathbf{S}_1 = \{\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3\} & \mathbf{S}_2 = \{\mathbf{U}_1, \mathbf{U}_4, \mathbf{U}_5\} \\ \mathbf{S}_3 = \{\mathbf{U}_2, \mathbf{U}_4, \mathbf{U}_5\} & \mathbf{S}_4 = \{\mathbf{U}_3, \mathbf{U}_4, \mathbf{U}_5\} \end{array}$$

- Here is a generalised protocol for user $\mathbf{U}_i$ to submit a query, based on an arbitrary covering design:
  1. User $\mathbf{U}_i$ chooses the designated proxy $\mathbf{U}_j$ uniformly at random.
  2. User $\mathbf{U}_i$ chooses uniformly at random one of the memory spaces (blocks) that contains both $\mathbf{U}_i$ and $\mathbf{U}_j$, say $\mathbf{S}_h$ (note that there is at least one such memory space). User $\mathbf{U}_i$ asks user $\mathbf{U}_j$ to act as his proxy using memory space $\mathbf{S}_h$.

retirer un utilisateur

- Using covering designs provides additional flexibility, while retaining the desirable property of perfect anonymity.
- This approach also facilitates a dynamic scheme.
- In order to delete a user $\mathbf{U}_i$ from a UPIR scheme, we simply delete $\mathbf{U}_i$ from all memory spaces of which he/she is a member (a rekeying mechanism would be required to update the keys associated with these memory spaces).
- The result is still a covering design defined on a set consisting of one fewer users than before.

ajouter un utilisateur

- To add a new user $\mathbf{U}_{new}$ to a UPIR scheme based on a covering design, it is first necessary to find any set of memory spaces whose union contains all current users:

$$\mathbf{S}_{h_1} \cup \mathbf{S}_{h_2} \cup \cdots \cup \mathbf{S}_{h_\ell} = \mathcal{U}.$$

- This could be done using a greedy algorithm (although it would not likely be optimal).
- Finding the minimum set of memory spaces is NP-hard.
- Then we add $\mathbf{U}_{new}$ to these ℓ memory spaces by giving it the ℓ keys associated with them.
- The result is still a covering design defined on a set consisting of one more user than before.

anonymat contre les autres utilisateurs

- Assumptions
  1. When source $\mathbf{U}_i$ requests that proxy $\mathbf{U}_j$ makes a query to the DB, everyone in the associated memory space knows that this request has been made, but no one (except for $\mathbf{U}_i$) knows who the source is.
  2. When a source is its own proxy, the request is still posted to the relevant memory space.
- For example, suppose $\mathbf{S}_0 = \{\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_3, \mathbf{U}_9\}$ and $\mathbf{U}_1$ is requested to act as proxy by $\mathbf{U}_0$.
- $\mathbf{U}_3$ only knows that the source is $\mathbf{U}_0, \mathbf{U}_1$ or $\mathbf{U}_9$.
- $\mathbf{U}_3$ and $\mathbf{U}_9$, acting as a "passive coalition", can deduce that the source is $\mathbf{U}_0$ or $\mathbf{U}_1$.

anonymat avec questions reliées

- Anonymity becomes more difficult if there are linked queries.
- Suppose that $\mathbf{U}_0$ makes two linked queries, using memory spaces

  $\mathbf{S}_0 = \{\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_3, \mathbf{U}_9\}$ and $\mathbf{S}_{12} = \{\mathbf{U}_{12}, \mathbf{U}_0, \mathbf{U}_2, \mathbf{U}_8\}$.
- Suppose $\mathbf{U}_3$ and $\mathbf{U}_8$ are a coalition.
- From the first query, $\mathbf{U}_3$ knows that the source is in $\{\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_9\}$, and from the second query, $\mathbf{U}_8$ knows that the source is in $\{\mathbf{U}_0, \mathbf{U}_2, \mathbf{U}_{12}\}$.
- Therefore $\mathbf{U}_0$ can be identified as the source by this coalition; this is just another intersection attack.

garanties d'anonymat

- Consider a sequence of q linked queries made by the same (unknown) user, and a coalition of c users trying to identify the source of the q queries.
- If there are always at least κ users who could with probability > 0 be the source (regardless of the queries and coalition) then we say that the scheme provides (q, c, κ)-anonymity.
- Of course we want $\kappa \geq 2$ because the source might be identified if $\kappa = 1$.
- In the case $q = 1$, we always achieve $(1, c, k - c)$-anonymity.
- If any two memory spaces intersect in at least μ users, then we achieve $(2, c, \mu - c)$-anonymity (this is useful only when $c \leq \mu - 2$).

exemple

- The classical result known as Fisher's Inequality asserts that $b \geq v$ in any (v, b, r, k, λ)-BIBD.
- If $b = v$, then $r = k$ and the BIBD is termed symmetric.
- In a symmetric BIBD, it can be shown that any two blocks intersect in exactly λ points.
- We obtain the following result.

Theorem

Suppose there exists a symmetric (v, v, k, k, λ)-BIBD. Then the resulting UPIR scheme provides $(2, c, \lambda - c)$-anonymity for any $c \leq \lambda - 2$.

meilleur anonymat

- Here is one possible approach to providing anonymity in the presence of $q > 2$ linked queries.
- Suppose the set of users $\mathcal{U}$ is partitioned into t-anonymity sets $\mathcal{V}_1, \ldots, \mathcal{V}_g$, where each $\mathcal{V}_i$ consists of at least t users.
- Suppose that the set system has the property that

$$\mathcal{V}_i \cap \mathbf{S}_j = \emptyset \text{ or } \mathcal{V}_i \tag{1}$$

  for all i, j.
- then the resulting UPIR scheme provides $(q, c, t - c)$-anonymity for any positive integers q and c.
- However, notice that all members in a given anonymity set have access to each other's queries, so there is no confidentiality among members of an anonymity set.

construction de systèmes d'ensembles convenable

- First, construct a set system (a covering design) on a set of g points, say $x_1, \dots, x_g$.
- Then define a bijection between the set of g points and the g anonymity sets.
- Finally, in every block, replace the point x_i by the anonymity set $\mathcal{V}_i$.
- This yields a covering design satisfying the desired property (1).

sommaire

- The combinatorial methods described in this talk provide an elegant way of ensuring anonymity against the database, even in the case of linked queries.
- There is a fundamental tradeoff:
  1. As mentioned earlier, we do not want memory spaces to be too large in case of key compromise (better security against external adversaries).
  2. However, small memory spaces require users to store more keys if the scheme is going to be secure.
- Anonymity against other users is more difficult to ensure, especially in the case of linked queries (and "perfect" anonymity is impossible).

références

[1] J. Domingo-Ferrer and M. Bras-Amorós. Peer-to-peer user-private information retrieval. *Lecture Notes in Computer Science* **5262** (2008), 315–323 (PSD 2008).

[2] J. Domingo-Ferrer, M. Bras-Amorós, Q. Wu and J. Manjón. User-private information retrieval based on a peer-to-peer community. *Data & Knowledge Engineering* **68** (2009), 1237–1252.

[3] K. Stokes and M. Bras-Amorós. Optimal configurations for peer-to-peer user-private information retrieval. *Computers and Mathematics with Applications* **59** (2010), 1568–1577.

[4] K. Stokes and M. Bras-Amorós. On query self-submission in peer-to-peer user-private information retrieval. In *PAIS 2011*.

merci pour votre attention!