Quantity Based Aggregation for Cadastral Databases

Firas Al Khalil, Alban Gabillon and Patrick Capolsini
Université de la Polynésie française
BP 6570, Faa’a Aéroport, French Polynesia
{firas.khalil, alban.gabillon, patrick.capolsini}@upf.pf

ABSTRACT
Quantity Based Aggregation (QBA) is a subject that is closely related to inference in databases. The goal is to enforce $k$ out of $N$ disclosure control. In this paper we work on QBA problems in the context of cadastral databases, and we focus on one particular problem: how to prevent a user from accessing all parcels in a region. This work is a new version of the model presented in [2, 3] where we introduce the concept of “Dominant Zones”. This concept increases the availability of the data while preserving their confidentiality. Moreover, we provide a more detailed discussion on security aspects of different choices of the model’s parameters.

Categories and Subject Descriptors
K.4.1 [Computers and Society]: Public Policy Issues -- Privacy, Regulation; H.2.7 [Database Management]: Database Administration -- Security, Integrity, and Protection

Keywords
Security, Access Control, Inference Control, Database, Collusion

1. INTRODUCTION
The inference problem [11, 12, 15, 18] in databases (e.g. [7, 27]), privacy preserving data publishing [16] and mining (e.g. [10]) and other domains (e.g. [1, 26]) has been heavily studied in the last couple of decades. The inference problem arises whenever partial or complete knowledge about some classified information can be derived (inferred, deduced) using unclassified information. In this work we address a problem that is very close to the inference problem and usually discussed with it: the aggregation problem. Actually, we are interested in a very special type of aggregation problems, specifically quantity-based aggregation problems (QBA henceforth).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGSPATIAL ’14 November 04-07 2014, Dallas/Fort Worth, TX, USA
ACM Copyright 2014 ACM 978-1-4503-3134-0/14/11 $15.00
http://dx.doi.org/10.1145/2676582.2676394

QBA problems were distinguished from inference and other aggregation problems for the first time in the work of Hinke [17], under the name “cardinality aggregation”. Lunt [20] analyzed inference and aggregation problems and showed the difference between them. She coined the term quantity-based aggregation (that we adopt in this paper), and she gave the following example to illustrate QBA: suppose that there is a phonebook of $N$ phone entries. A user has the right to know $k$ entries, at most, out of $N$. The goal of QBA control is to enforce this “$k$ out of $N$” disclosure control. The abovementioned example is known in the literature as the NSA (National Security Agency) or the SGA (Secretive Government Agency) phonebook problem. After reviewing the literature, it seems like the problem has not been addressed fully. The only work we are aware of, treating QBA directly, is that of Motro, Marks and Jajodia [21,23] twenty years ago. In a previous paper [2] we addressed two QBA problems in the context of cadastral databases. In fact, we proposed a solution to enforce the following security policy: In the cadastral database, any user has the right to know the owner’s name of any given parcel. However, this permission is constrained with the following prohibitions that represent two QBA problems: the user is forbidden to know, $Pr_1$: the list of all parcel owners in a region,

$Pr_2$: the list of all parcels belonging to the same family.

In [2, 3] we proposed a first model and its implementation. The main contribution of this paper is a new version of the model to achieve more availability on cadastral data where we introduce “Dominant Zones” (Definition 5) that requires less storage space, provides more availability, and gives more flexibility to the security administrator controlling the trade-off between confidentiality and availability of data. We also developed a prototype accessible from http://webgis.upf.pf:8080, we will concentrate in this paper on $Pr_1$. Solutions presented for $Pr_1$ can be easily applied to $Pr_2$ with minimal modifications due to the similarity of both problems. For more details about the legislative context, the reader is invited to read [2].

This paper is organized as follows (we would like to stress out the fact that all sections, except Section 3.1, are original contributions): Section 2 gives some preliminary definitions that help identifying aggregation problems. Section 3 defines the security policy and presents the new model. Section 4 compares the new model to our previous work [2, 3], and provides a detailed analysis on why dominant zones achieve...
higher availability. Section 5 describes the application workflow, the role of QBA control in that workflow, and authentication. Section 6 gives a review of QBA control. Finally, Section 7 concludes this paper.

2. DEFINITIONS
In this section we give a set of definitions (relying on earlier work described in Section 6) that helps identifying inference, aggregation and QBA problems:

Definition 1. [Inference problem] The inference problem arises whenever a collection of information can be used to derive (infer, deduce) partial or complete knowledge about information stored in the database and classified higher than the classification of each subset of the collection. This collection forms an inference channel. Inference control is a mechanism used to eliminate inference channels and prevent users from performing inferences.

To illustrate an inference problem, let us consider the phonebook example: a phonebook is represented by the relation PHONEBOOK((NAME, TEL, DEPT)) where the classification of NAME and TEL (say, CONFIDENTIAL) is lower than that of DEPT (say, SECRET). A user with an UNCLASSIFIED clearance can access both NAME and TEL, and naturally DEPT is prohibited. However, if we consider that TEL depends on DEPT (e.g. one telephone per department), or numbers of the same department have the same suffix, etc.), then a user can infer, using NAME + TEL, to which department a given employee is affiliated, or even the list of employees who work in the same department.

Notice that the definition of the inference problem does not specify the source(s) of information in the collection. They could be partially derived from the database as in the inference from external knowledge, where a user combines his external knowledge with partial knowledge acquired from objects (that s/he has the appropriate clearance to read) of the database to conduct an inference, hence deduce sensitive information.

Definition 2. [General Aggregation problem] The general aggregation problem arises whenever a collection of k pieces of information, stored in the database, is higher than the classification of each subset. Aggregation control is a mechanism used to prevent users from performing aggregations.

To illustrate general aggregation problems, let us consider 3 phone entries (tuples): A, B and C labeled SECRET. The aggregation of A and B is labeled TOP SECRET while the aggregation of A and C is labeled SECRET. A user with a SECRET clearance level cannot access the aggregate A + B but s/he can access A + C.

Definition 3. [QBA problem] The QBA problem arises whenever the classification of more than k out of N items in a database is higher than the classification of that of k or less items. QBA control is a mechanism used to prevent users from aggregating more than k out of N items.

Indeed, a QBA problem arises when a user has the right to query any subset of the phonebook relation (of size N), under the condition that the size of the queried subset does not exceed k (where k < N). The key difference between Definitions 2 and 3 is that the former does not take into account the quantity of aggregated entries. If we apply these definitions to work found in the literature, we find that inferences from dependencies on schema and data [7], or inferences from external knowledge [26] or denial of access [24] fall under Definition 1. The Chinese-Wall policy [5, 22] falls under Definition 2 where the role of the policy is preventing a user from aggregating data from the same conflict of interest class, while the phonebook problem as presented by Hinke [17] and Lunt [20], and both aggregation problems of the cadastral database fall under Definition 3.

3. NEW MODEL
3.1 Security Policy
A cadastral database is a geographical database used to manage parcels of a country, state, municipality, etc. Parcels are pieces of land represented in the database by geo-referenced polygons. In addition to their geometric representation, parcels are associated with information like mutation history, taxation, and most importantly ownership information. Currently, access to the cadastral database in French Polynesia is limited to employees of the real-estate service, notaries, and surveyors. The Computer Science Department of French Polynesia wishes to make this database available online and apply the following security policy: citizens (parcel owners or not) can access ownership information of any parcel through a point and click mapping interface (similar to Google Maps or Bing Maps). However, this access is limited by the following prohibitions:

$Pr_{1}$: A user cannot get the list of all owners in a geographical region.

$Pr_{2}$: A user cannot get the list of all parcels belonging to the same family.

It should be obvious by now that both $Pr_{1}$ and $Pr_{2}$ are two separate QBA problems. Indeed, we have the following analogies:

- The list of owners of a given region is analogous to a phonebook ($Pr_{1}$).
- The list of parcels of a given family is analogous to a phonebook ($Pr_{2}$).
- The association between a parcel and an owner is analogous to a phonebook entry.

We should remind the reader that in this work we focus on $Pr_{1}$ uniquely due to space limitations and the similarity between $Pr_{1}$ and $Pr_{2}$.

3.2 QBA Enforcement
Definition 4. [Zone] A zone of a parcel $p$ is formed by $p$ itself and all its neighbors.

Notice that, according to this definition, every parcel belongs to its own zone and the zone of every direct neighbor. Moreover, using this definition we obtain a graph where parcels are nodes in the graph and 2 neighboring parcels are connected with an edge.

Definition 5. [Dominant Zone] The dominant zone of a parcel $p$, $dom_{p}$, is the zone containing $p$ having the highest cardinality. A parcel can have multiple dominant zones.
Let \( \text{Dom} = \{ \text{dom}_i : i = 1, 2 \ldots M \} \) be the set of all dominant zones in the database; \( |\text{dom}_i| = N_i \). A user has the right to know the ownership of any parcel belonging to any dominant zone in the database. The **Aggregation Control Property** is: for all \( \text{dom}_i \), the number of disclosed parcels for any user, \( k_i \), should always be strictly lower than \( N_i \). Satisfying the aggregation control property, namely preventing a user from accessing all parcels in a dominant zone, implies the satisfaction of the security policy (Section 3.1) and effectively preventing a user from acquiring the knowledge of all owners in any region of any size.

Enforcing QBA control is simple: when a user requests a parcel \( p_i \), the algorithm should make sure that the number of disclosed parcels is strictly lower than \( k_i \) for the dominant zone of the requested parcel, \( \text{dom}_i \), and all dominant zones containing it, i.e. dominant zones of its direct neighbors containing the requested parcel. If this condition is satisfied, access is granted; otherwise, access is denied. We omit here the technical details of this algorithm due to space limitations. In order to satisfy the security policy: for every dominant zone \( \text{dom}_i \), \( k_i \) must satisfy: \( 0 < k_i < N_i \). This is sufficient if we consider a single user accessing the cadastral database in isolation. Let us take an example showing why this condition is not sufficient if we consider multiple users: if two users are accessing a dominant zone where \( k_i = N_i - 1 \), none of them can get the ownership information of all parcels in that dominant zone, however, they could collaborate and combine their knowledge to bypass the limit \( k_i \). Therefore \( 0 < k_i < N_i \) is a necessary but not sufficient condition in real-world applications where colluding users form an actual threat to the security of the application. An important property that should be satisfied by QBA control is collusion resistance.

### 3.2.1 \( x \)-Collusion Resistance

**Definition 6.** \([x\text{-collusion}]\) We say that \( x \) users collude to reconstruct all entries in a dominant zone if the union of accessed parcels by those \( x \) users covers the complete dominant zone.

A QBA control mechanism is \( x \)-collusion resistant if \( x \) or fewer users cannot reconstruct a complete dominant zone. To achieve \( x \)-collusion resistance the **Aggregation Control Property** should be extended to:

\[
k_i = |N_i/x| - 1
\]

This way, \( x \) users are guaranteed to never collude and reconstruct a complete dominant zone even if those \( x \) users accessed disjoint subsets of \( \text{dom}_i \). Now we should analyze \( x \)-collusion resistant QBA control. In fact, we should evaluate the effect of varying size zones. Since we proposed a solution to achieve \( x \)-collusion resistance that relies on \( N_i \), and \( N_i \) is variable due to the dynamic nature of the database (deletions, insertions, and divisions of existing parcels), then we should see which values would vary with respect to \( N_i \). At any given moment \( x \) users or fewer can never collude to get the complete set of records; therefore \( k_i \) should vary with \( N_i \).

1. \( k_i \) is dynamically computed according to the number \( N_i \) of parcels. Therefore, if \( k_i \) increases then users have access to more parcels. This means that users might collude to reconstruct the previous dominant zone \( \text{dom}_i \) before it expanded. From a security point of view this means that the previous zone has been “declassified”. If we wish to avoid this situation then the only solution would be enforcing smaller values of \( k_i \):

\[
k_i = |N_i/x| - \alpha, \alpha \geq 1
\]

With \( \alpha \) carefully chosen, expanding dominant zones \( \text{dom}_i \) do not allow colluding users to reconstruct today, information that was considered sensitive yesterday.

2. In the case where the value of \( k_i \) decreases to \( k_i' \), then users who have already accessed more than \( k_i' \) parcels might collude to reconstruct the new dominant zone \( \text{dom}_i' \). Here also, computing smaller values of \( k_i \) (i.e. choosing a proper \( \alpha \) value) would eliminate this security threat.

Now let us consider the set of all \( M \) dominant zones under \( x \)-collusion resistance. If \( x \) has the same value for all dominant zones, then \( x \) or fewer users are guaranteed not to collude on all \( M \) zones. If the security administrator sets for zones \( \text{dom}_1, \text{dom}_2 \ldots \text{dom}_M \) different values \( x_1, x_2 \ldots x_m \) then a coalition of \( x \) users, where \( \text{min}(x_1, x_2 \ldots x_m) < x < \text{max}(x_1, x_2 \ldots x_m) \) can collude to reconstruct all zones \( \text{dom}_b \) with \( x_a \)-collusion resistance such as \( x_a < x \). Therefore, we recommend setting a single value of \( x \)-collusion resistance to all zones.

#### 3.2.2 \((x, y, z)\)-Collusion Resistance

The drawback of \( x \)-collusion resistance is that it assumes that all users are potential colluders on all dominant zones. In practice this assumption is somewhat too strong and may lead the QBA control mechanism to detect too many false positives. Another type of collusion resistance is needed where a user is assumed to be a potential colluder if his/her querying behavior is suspicious. This new type should take into account the main idea behind \( P_{R1} \), while relaxing the assumption on colluding users: recall that \( P_{R1} \) states “a user cannot get the list of all owners in a geographical region”. Therefore a group of users should be considered as potential colluders if they are trying to attack a region, the more general concept of a zone.

**Definition 7.** \([z\text{-region}]\) A \( z \)-region of a parcel is formed by \( \text{dom}_p \) and the set of dominant zones of neighbors of \( p \) of degree \( \leq z \).

**Definition 8.** \([(x, y, z)\text{-collusion}]\) We say that \( x \) users collude to reconstruct \( y \) dominant zones in a \( z \)-region if the union of accessed parcels by those \( x \) users covers those \( y \) complete dominant zones.

To achieve \((x, y, z)\)-collusion resistance, the **Aggregation Control Property** becomes: for any \( z \)-region \( R_z \), where \( |R_z| = M_z \); 1) A user is prohibited from querying more than \( k_i \) parcels, such as \( k_i = N_i - 1 \) (or \( k_i = N_i - \alpha \)), in \( y \) dominant zones of \( R_z \); 2) It enforces \( x \)-collusion resistance on the remaining \( M_z - y \) dominant zones of \( R_z \). The idea behind \((x, y, z)\)-collusion resistance is that as long as users cannot reconstruct more than \( y \) dominant zones in a given region then they should not be considered as colluders. As soon as these users can reconstruct \( y \) dominant zones in a given region then the \( x \)-collusion resistance scheme should be applied on the remaining \( M_z - y \) dominant zones. To support \((x, y, z)\)-collusion resistance, \( k_i \) should be split into 2 variables: \( k_h \) (read \( K \text{ HIGH} \)) and \( k_l \) (read \( K \text{ LOW} \)). A user has the right
to access up to \( k_b \) entries in \( y \) dominant zones in any \( z \)-region \( R_t \) of a parcel, after which \( s/he \) is considered a potential colluder. For the remaining \( M_t - y \) dominant zones \( s/he \) has the right to access \( k_t \) entries, where \( k_h > k_t \). The security administrator sets \( k_h \), for instance \( k_h = N_t - 1 \), and \( k_t \) is determined by the required level of \( x \)-collusion resistance as defined in equations 1 or 2.

3.2.3 Choosing \( x \), \( y \), and \( z \)
Setting the parameters \( x \), \( y \) and \( z \) is the responsibility of the security administrator. These values cannot be assigned arbitrarily. For instance, \( x \) defines the level of collusion resistance per dominant zone, therefore, for a given parcel \( p \), \( x \leq |\text{dom}_p| \). In Section 3.2.1 we argued that \( x \) should have the same value for all dominant zones, which implies that \( x = \min(|\text{dom}_1|, |\text{dom}_2|, \ldots, |\text{dom}_m|) \). However, practically, and for reasons explained in Section 5, \( x \) could be less than or equal to the average size of a dominant zone in the database.

If we consider \((x, y, z)\)-collusion resistance, then \( y \) is the number of parcels that are not under collusion resistance in a \( z \)-region, therefore, for a given parcel \( p \), \( y < |z \text{-regions}| \). Practically, \( y \) is lower than the average number of zones in a \( z \)-region. As for \( z \), the security administrator should keep in mind that \( z \) determines the depth in the depth-first traversal, which has a runtime complexity of \( O(b^z) \), where \( b \) is the branching factor (or the average number of neighbors per parcel). Therefore, \( z \) should be \( > 1 \), and an assessment of available computational resources should be taken into account to achieve the desired and most practical results.

4. COMPARISON WITH OUR PREVIOUS APPROACH
We tackled QBA problems in the cadastral database in an earlier work \[2\] where we used zones uniquely, and defined different levels of collusion resistance \((x, (x, y)\)- and \((x, y, z)\)-collusion resistance) to prevent users from colluding and bypassing \( P_{R1} \) and/or \( P_{R2} \). Our approach to implement these levels required tracking the query history of every user. This history was used to track collusions on the user level, i.e. maintaining lists of who is colluding with whom. This tracking required \( O(b^z) \) space to maintain the list of colluding users, while searching for a potential collusion on a single parcel level was an exhaustive search requiring \( O(x^n) \) time, where \( U \) is the number of users in the system, \( n \) is the number of users who has accessed a parcel, and \( z \) is the value from \( x \)-, \((x, y)\)- or \((x, y, z)\)-collusion resistance. In addition, this implementation was described in terms of a graph database.

The work presented in \[3\] provided an alternative and more efficient implementation, using the same model, namely with zones only. In this second implementation, we were not tracking any collusion in the first place. Indeed, we were defining the number of accessible parcels in a region \((P_{R1})\) or belonging to a given family \((P_{R2})\) beforehand and then simply counting the number of actually accessed parcels and making sure it does not exceed a given threshold. We also dropped a level of collusion resistance, namely \((x, y)\)-collusion resistance, and changed some definitions in order to gain performance enhancements without compromising their security properties. Moreover, our solution was described in the relational model, facilitating the integration with the existing French Polynesian cadastral database (Algorithmic details can be found in \[3\]).

In Section 3 we provide a detailed discussion on security aspects and the intuition behind collusion resistance levels -- which was omitted from \[3\] -- and we extend the model from \[2\] with dominant zones. The motivation behind this new model is to achieve a better availability while preserving the same level of confidentiality. Our goal is to ensure data availability for authorized users only, however we want to allow them to get the possibility to know more data, using the new model, without violating the security policy. Section 4.1 provides a comparison between the old model, the new model, and a naive attempt to achieve the aforementioned objective. Section 4.2 explains why “Dominant Zones” work and allow us to raise the availability while preserving the confidentiality.

4.1 Availability without Dominant Zones
QBA control as presented in \[2,3\] relies on its collision resistance scheme. The choice of \( x \), \( y \) and \( z \) defines the number of available parcels per user. Figure 1a shows the average number of parcels available for a user for different values of \( y \) and \( z \) (\( x \) is set to 2); we used the cadastral database of the island of Maupiti provided to us by the French Polynesia Computer Science Department, containing 960 parcels. In fact, every point in Figure 1a is the average accessible number of parcels per user: for every value of \( y \) and \( z \) we created 100 users and made them traverse the complete database randomly. As expected, incrementing the value of \( y \) renders the data more available, while incrementing \( z \) assures more confidentiality for cadastral data.

Figure 1a shows that, even with relaxed security settings (high \( y \) and low \( z \)), the number of available parcels is very low. In order to achieve higher availability, the simplest (naive) solution that we thought about initially would be to change the definition of a zone to reach 2\(^{nd} \) degree neighbors. We will call this new definition a 2-zone. This solution provides more availability as shown in Figure 1b, however it presents a major drawback in processing time and storage. Indeed, in Maupiti’s database, the relation storing parcel neighbors for 2-zones increased to 306.86% when compared to its size with normal zones of Definition 4. Another major drawback is the fact that the range of available parcels shrinks with 2-zones; we mean by availability range the difference between the lowest amount of available parcels to the greatest amount. In Figure 1, \( x = 2, y = 2 \) to 8 and \( z = 2 \) to 7; the availability’s range is 7.18% for zones (32.23% to 39.41%), while 2-zones reduced the range to 4.81% (38.62% to 43.43%). On the other hand, dominant zones increased availability’s range to 23.13% (44.72% to 67.85%). This range of availability gives the security administrator more control and flexibility over the tradeoff between availability and confidentiality.

4.2 Availability with Dominant Zones
Let us take the example of Figure 2: we have 6 parcels \( \{P_1, P_2, P_3, P_4, P_5, P_6\} \). According to Definition 4, the zone of \( P_1 \), \( \text{zone}_{P_1} \) is formed by \( P_1 \) and \( P_6 \). Similarly, \( \text{zone}_{P_2} = \{P_2, P_5, P_6\} \), \( \text{zone}_{P_3} = \{P_3, P_5\} \), \( \text{zone}_{P_4} = \{P_4, P_5\} \), \( \text{zone}_{P_5} = \{P_1, P_2, P_3, P_4, P_5, P_6\} \), and \( \text{zone}_{P_6} = \{P_2, P_3, P_4, P_5, P_6\} \). We suppose that we are not trying to achieve any level of collusion resistance. Every user has the right to access, in every zone, all parcels except 1 (see equation 1; \( z = 1 \)). We consider a user, Alice, who has never queried any parcel. If Alice decides to access parcel \( P_5 \), then access would be granted. However, access to \( \{P_1, P_2, P_3\} \) will be automatically blocked because \( k_t \) is
reached for all zone, i = {1, 4, 5}. Alice can finally query P6 (or P2), thus acquiring the knowledge of 2 parcels out of 6. However, Alice could have queried these parcels in a different order: P1, P2, P3, P4 then P6, thus acquiring the knowledge of the owners of 5 out of 6 parcels. Notice that querying behavior changed drastically the number of accessible parcels; QBA control went from very restrictive to very permissive. Even if we try to apply x- or (x, y, z)-collusion resistance, the problem persists: these levels of collusion resistance change the quantity of accessible parcels, and actually render the QBA control enforcement stricter. This issue comes from the fact that we give all zones equal importance and we enforce collusion resistance on every single zone.

Figure 3 shows the distribution of parcels attached to zones of different sizes in Maupiti’s database: every point represents the percentage of parcels in the database (ordinate) that are attached to a zone (resp. 2-zone and dominant zone) of a given size (abscissa). Let us take the example of Figure 2 to explain what we mean by attachment: if we are using zones, then P2 is attached to 1 zone of size 5 (zoneP5), 1 zone of size 3 (zoneP2) and 1 zone of size 2 (zoneP4); if we are considering 2-zones, then P2 is attached to 3 2-zones of size 5 (2-zoneP2, 2-zoneP3 and 2-zoneP4), 2 2-zones of size 6 (2-zoneP2 and 2-zoneP3), 1 2-zone of size 3 (2-zoneP4); if we are considering dominant zones, then P2 is attached to 1 dominant zone of size 5 (domP5 = zoneP5). Notice that dominant zones reduce the number of attached parcels.
drastically: for instance, while 67.1%, 71.2%, and 98.9% of parcels are attached to zones of sizes 3, 4 and 5 respectively, dominant zones reduces these percentages to 52.2%, 26.4% and 28% respectively. On the other hand, 2-zones reduces these percentages but the number of 2-zones of different sizes is higher and 20 to 40% of parcels are attached to 12 2-zones of different sizes (bigger number of peaks in Figure 3). This distribution explains why dominant zones provide more availability than zones and 2-zones: by giving priority on every parcel to the largest zone it is attached to, they reduce the number of parcels attached to zones of small sizes while keeping the sizes of (dominant) zones intact.

While Figure 3 shows how many parcels are attached to zones of different size, Figure 4 shows how many parcels are attached to how many zones, i.e. how many parcels are attached to 1 zone (resp. 2-zones, dominant zones) uniquely, 2, 3, . . . etc. Let us take the example of Figure 2: if we are using zones, then $P_2$ is attached 3 zones, namely $zone_{P_21}$, $zone_{P_23}$ and $zone_{P_25}$; if we're using 2-zones, then $P_2$ is attached to all 2-zones of the graph; if we're using dominant zones, then $P_2$ is attached to 2 dominant zones, namely $zone_{P_21}$ and $zone_{P_25}$ (because both contain $P_2$ and they are dominant zones thus considered by the QBA control enforcement algorithm). Notice that, for dominant zones, 1) 85.2% of parcels are attached to 1, 2 and 3 dominant zones respectively, and 2) all parcels are attached to 1 to 9 dominant zones only, unlike zones (and 2-zones) that can be attached to 1 to 39 zones (and 1 to 57 2-zones). This distribution explains the fact that dominant zones provide a bigger range of availability: they reduce the number of zones that could influence the disclosure decision on a parcel, which, when combined with the fact that they reduce parcel attachment to zones of small sizes (Figure 3), allows for a greater margin of flexibility when applying $(x, y, z)$-collision resistance in QBA control.

The introduction of dominant zones has multiple advantages: 1) it requires less storage space: we only have to store dominant zones instead of every single zone, thus improving execution time of the QBA control enforcement algorithm (390 dominant zones vs 960 zones for the island of Maupiti); 2) it provides more availability (see Figure 1) by giving importance to zones of big sizes; 3) it lowers the effect of the user's querying behavior on the range of available parcels.

5. APPLICATION WORKFLOW AND AUTHENTICATION

Our model is secure with “strong” authentication. However QBA control in the context of this cadastral application is only preventive as we shall show in this section. In fact, currently, in order to acquire information about any given parcel, a citizen of French Polynesia needs to visit the facilities of the French Polynesian real-estate service. There, s/he will stand in a queue waiting for her/his turn, and then s/he will meet an employee who will accept the citizen’s query. The citizen needs to provide the requested parcel’s ID, or its address. Moreover, s/he can query multiple parcels at the same time. The citizen needs not to provide any identification (no driver’s license, nor passport, etc).

Once provided with the parcel’s ID or its address, the employee will perform a check on the query itself, the number of requested parcels and the rate at which the citizen has been issuing queries:

1. Is the requested information classified? (e.g. owned by the military, the president, etc.)
2. Is the citizen requesting a lot of parcels? (e.g. the owners of a complete neighborhood)
3. Has the citizen been asking for excerpt of parcels regularly and in a suspicious manner?

Obviously, the employee is enforcing an internal policy constraining citizens’ requests. If the employee accepts the request, the citizen must pay a fee before getting the excerpt of the requested parcel(s). There are two main issues with this workflow:

1. Citizens should be physically present at the real-estate service. This is especially problematic in countries such as French Polynesia that are formed uniquely by archipelagos (118 islands and atolls; Exclusive Economic Zone (EEZ) of over 5 million $km^2$, while Metropolitan France's EEZ is around 330 thousand $km^2$ only).

2. Employees enforcing the service’s internal policy are themselves human, therefore error-prone. Moreover, there is not a single employee, and they do change with time.

The real-estate service wishes to make the cadastral database available online, making it easier for citizens to acquire excerpts of parcels, while adapting the original workflow as follows:

1. A user is presented with a mapping interface where s/he has the option to select a single parcel.
2. Once selected, the user has the option to “preview” the parcel’s ownership information, as long as this “preview” does not violate the service’s policies – namely $Pr_{1}$ and $Pr_{2}$.
3. If the preview was successful, the user can either cancel his order or proceed and place the order for the excerpt where s/he is required to pay a predefined fee.
4. If the preview was not successful – due to the violation of either $Pr_{1}$, $Pr_{2}$, or both – the user can still proceed and place the order for the excerpt and pay the required fee.

This “preview” feature acts as a guard for the user himself: online data can be out of date, or incorrect. $Pr_{1}$ and $Pr_{2}$ are required to limit the abuse of this feature.

The first obvious security issue that could arise comes from authentication. Indeed, if we suppose that users are not authenticated – using any authentication mechanism (e.g. [19]) that could tie their virtual identity with their physical one – they can forge multiple identities to circumvent the QBA control mechanism. This is known in the literature as “The Sybil Attack” [9]. Sybil attacks were discussed by Doucet in the context of peer-to-peer networks, but they appear in a lot of application domains, such as mobile networks, cash economies, reputation systems, etc. There is no general solution for such an attack, and every case should be considered separately. Among the solutions proposed in the literature, we find trusted certifications, trusted devices, and others.

The service explicitly mentioned that any form of “strong” authentication is unnecessary and might discourage users
from using the service, especially that access to the internet on small islands is available uniquely through the municipality, and users are not necessarily tech-savvy. They want to replicate the current workflow found at their offices. Therefore, users are authenticated with their IP addresses, which seems to be sufficient — from the service’s point of view — to enforce QBA and manage collisions. It follows that collusion resistance, as described in this paper, is also meant to prevent users from constantly changing their IP addresses (e.g. disconnecting their ADSL modem then reconnecting it) to circumvent QBA control.

6. STATE OF THE ART

Hinke [17] identifies 2 types of aggregation problems: 1) Cardinality aggregation, and 2) inference aggregation; the former is what we now call QBA and the latter is the “classical” inference problem. He argues that both cardinality aggregation and inference aggregation are subclasses of the aggregation problem. He did not work on cardinality aggregation problems because he noted that “[inference aggregation problem] appear to be more tractable. With cardinality aggregation, it is not always clear why “N” elements of a set, such as a phonebook, are classified at one level, while “M” elements are less classified, where cardinality of N ≥ cardinality of M”. The work of Lunt [20] analyses inference and aggregation problems found in multilevel relational databases. According to Lunt, the inference problem arises whenever some data x can be used to derive partial or complete information about some other data y, where y is classified higher than x. The aggregation problem arises whenever some collection of facts has a classification strictly greater than that of the individual facts forming the aggregate. To qualify as an aggregation problem, it must be the case that the aggregate class strictly dominates the class of every subset of the aggregate. Under aggregation problems, she identified quantity-based aggregations (known earlier as cardinality aggregations); they occur whenever a collection of up to k items of a given type is not sensitive, but a collection of greater than k items is sensitive (in the original work she used N). The most relevant work is that of Motro, Marks and Jajodia [23] (MMJ). They developed a model to handle QBA in relational databases. The first key difference between their work and ours is in the hypothesis: they assume that a user can execute “arbitrary queries”: i.e. a user can select and project tuples from a phonebook relation on any set of attributes he desires, while we only consider single queries selecting a single tuple (point and click). They have also extended their work [21] to include multi-query attacks (i.e. Join and Complementary queries). Jajodia and Meadows [18] give another definition of inference problems while surveying the literature on inference control problems in multilevel secure databases. They first introduce the notion of an inference channel, which is a mean by which one can infer data classified at a high level from data classified at a low level. At the end of their paper, they briefly talk about aggregation problems and mentioned that they are similar to inference problems but not identical. They also showed how different strategies could be adopted to control different aggregation problems. Bewer and Nash [5] presented the Chinese-Wall policy. They might be the first to identify a real-world aggregation problem. In fact, the main motivation for the work was to prevent a user from aggregating knowledge in a domain that would help him learn sensitive information and conduct malicious behavior. In a different work, Meadows [22] give another definition of the aggregation problem and she says that aggregation issues arise in database security when two or more data items are considered more sensitive together than they are separately. She extended the Bewer-Nash model in order to generalize it to multilevel databases. She presents a formal model that is able to handle the Chinese-Wall security policy and other types of aggregation problems. Staddon [25] presented in her paper a dynamic inference control scheme that does not depend (directly) on user query history, which implies fast processing time, and ensures a crowd-control property: a strong collusion resistance property that not only prevents c collaborating users (where c is the degree of collusion-resistance) from issuing complementary queries to complete an inference channel, but also guarantees “if a large number of users have queried all but one of the objects in an inference channel, then no one will be able to query the remaining object regardless of the level of collusion resistance provided by the scheme”. c-collusion resistance is not desirable in QBA control because it implies that at least one object out of N can never be read by any user. Chen and Wei [6] extended the work of Staddon on dynamic inference control. They present a scheme that is resilient to what they call a “block an object” attack where a malicious user can exhaust a channel therefore blocking access to the last object for all other database users. The important thing to take from this paper is what they noticed about blocking users and how effectively a time-based key-refreshing scheme should be enforced to prevent not only “block an object” attacks, but also blocking users on a set of accessible objects, which might render the application useless after a given period of time. Bezzi et al. [4] also tackled QBA to prevent statistical inferences: their goal is to perform a k-out of N disclosure control such that the distribution of these k records does not resemble the distribution of the sensitive information. Cuppens [8] studied the aggregation problem and proposed a model based on modal logic to control aggregation problems and the work of Foley [13,14] addresses the aggregation problem with information flow policies, however these papers do not address the specific case of QBA problems.

7. CONCLUSION

In summary, we identified two quantity based aggregation problems in the cadastral database emerging from the need to publish the database online. We started by defining inference, aggregation and quantity based aggregation problems, and showed the difference between them. We focused on Pr1, the prohibition that states “a user should never know the list of all parcels in a region”. We presented a new model based on “Dominant Zones” to achieve higher availability. We compared dominant zones to our old model and a naive approach, and we justified why dominant zones achieve its objective. We presented all explanations and justifications for all different parameters used to achieve a collusion resistant “k out of N” disclosure control. We argued that performing a simple “k out of N disclosure control is not appropriate for real-life applications due to the risk of colluding users, and we proposed 2 distinct types of collusion resistance (namely x- and (x,y,z)-collusion resistance) to throttle the effect of collisions. We also explained in detail the objective of the whole application, and the purpose of the online publication of the cadastral database by explaining the current (physical) workflow, and the required online workflow. Furthermore, we
argued that, for the specific case of the cadastral application of French Polynesia, a “strong authentication” mechanism is not required, and even seen as unnecessary. Finally, we gave a review of the literature on QBA problems.

8. REFERENCES


