Journal of Information Security, **2012, 3, 105-112**
http://dx.doi.org/10.4236/jis.2012.32013 Published Online April 2012 (http://www.SciRP.org/journal/jis)

Privacy Preserving Scheme for Location-Based Services

Youssef Gahi[1], Mouhcine Guennoun[2], Zouhair Guennoun[1], Khalil El-Khatib[2]
[1]Laboratoire d'Electronique et de Communications—LEC, Ecole Mohammadia d'Ingénieurs—EMI, Université Mohammed V-Agdal—UM5A. BP, Rabat, Morocco
[2]University of Ontario Institute of Technology, Oshawa, Canada
Email: youssef.gahi@gmail.com, mouhcine.guennoun@uoit.ca, zouhair@emi.ac.ma, khalil.el-khatib@uoit.ca

Received November 29, 2011; revised December 31, 2011; accepted January 30, 2012

ABSTRACT

Homomorphic encryption schemes make it possible to perform arithmetic operations, like additions and multiplications, over encrypted values. This capability provides enhanced protection for data and offers new research directions, including blind data processing. Using homomorphic encryption schemes, a Location-Based Service (LBS) can process encrypted inputs to retrieve encrypted location-related information. The retrieved encrypted data can only be decrypted by the user who requested the data. The technology still faces two main challenges: the encountered processing time and the upper limit imposed on the allowed number of operations. However, the protection of users' privacy achieved through this technology makes it attractive for more research and enhancing. In this paper we use homomorphic encryption schemes to build a fully secure system that allows users to benefit from location-based services while preserving the confidentiality and integrity of their data. Our novel system consists of search circuits that allow an executor (*i.e.* LBS server) to receive encrypted inputs/requests and then perform a blind search to retrieve encrypted records that match the selection criterion. A querier can send the user's position and the service type he/she is looking for, in encrypted form, to a server and then the server would respond to the request without any knowledge of the contents of the request and the retrieved records. We further propose a prototype that improves the practicality of our system.

Keywords: Privacy; Location Based Services; Homomorphic Encryption

1. Introduction

The growth of smart phones and mobile devices in both software and hardware capabilities have resulted in the emergence of a set of new products and internet services that guarantee new promising business models. Location-Based Services (LBSs) have attracted the utmost importance in this regard. These services rely on the Global Positioning System (GPS) or Network-Based Positioning, which are mainly used to determine the current position of a user, in order to define his/her location relatively to a business or a service (banks, restaurants, universities etc). The user can enquire about that information by communicating wirelessly with an LBS server. The server uses the signal emitted by the user to locate him/her using Real-time Locating Systems (RTLS) [1]. Once the coordinates of the user are determined, the server responds with a list of all services surrounding the user's position.

LBSs have attracted the research and development community. However, LBSs suffer from a major security pitfall in terms of violating users' privacy. In other words, as the LBS server gains knowledge of the users' coordinates, this information can be manipulated by the server itself or by any malicious party to trace the movements of the users. Thereby, instead of using such a mechanism to facilitate lifestyle, it can easily turn over into an efficient tracking tool. This problematic urged the research community to find a secure way to use LBSs without disclosing users' private information.

Strong protection for users' information can be attained if the server is made capable of retrieving location-related information without being aware of the user's position or the point of interests he/she is requesting. It is challenging to achieve the latter target as the server needs to at least know this search criterion to retrieve the requested information. In this paper, we tackle this problem by using encryption schemes to retrieve data without violating the privacy of the users.

The remainder of this paper is organized as follows. In Section 2 we review the related work that aimed at securing location-based services. Section 3 provides a detailed description of the circuits that makes it possible to respond to requests in a blind fashion. Section 4 presents our prototype and the evaluation of its performance. Finally, Section 5 concludes our work and provides future research directions.

2. Related Work

There are a number of approaches in the literature to

solve the problem of privacy protection with location based services, including:

- Cloaking;
- Generation of dummies;
- Private information retrieval (PIR).

Gruteser and Grunwald [2] and Chow *et al*. [3] have based their approaches on K-anonymity [4-7]. The latter concept relies on hiding the user's location among K-1 neighbors. The main idea behind this concept is to send a box of locations instead of only the true one, whereby the probability to guess the user's location is always less than 1/K. Most of techniques relying on K-anonymity [2-7] use a middleware (the anonymizer). This anonymizer is a third party responsible for creating a Cloaking Region (CR), which contains the true user's location, as well as K-1 other neighbors. With such a technique, a typical scenario can be a user trying to localize the nearest bank. The user sends his/her requests (including his/her credentials) to the anonymizer through a wireless network. Thereafter, the anonymizer, which keeps the locations of all current users, authenticates the requester first and chooses a set of K-1 neighbors to create a CR that can be sent instead of the user's position. This way, the risk of violating the user's privacy is reduced by making it difficult to locate the position that has triggered the process (since the server is answering the whole CR). However, this approach suffers from several drawbacks. Firstly, the users' data is still revealed to a third party (the anonymizer) and thus the problem of preserving the user's privacy has not been solved. That is, we still have no guarantees that the anonymizer cannot be misused if a malicious hacker gains access to it. Secondly, the anonymizer needs to update the current location of all the subscribed users repeatedly, which will require a permanent communication and remote monitoring of the users, which is a clear violation of the users' privacy. Finally, the robustness of these approaches depends totally on having a relatively big number of neighbors at the time of receiving the requests. Therefore, depending on a middleware is far from being a perfect solution to secure location-dependent queries and hence any secure solution need to communicate directly with the Location Based Server without any intermediate parties.

Kido *et al*. [8] and You *et al*. [9] have proposed a new technique to hide users' location and trajectory by sending several queries instead of only one. The technique depends on creating several fake queries with fake identities in addition to the real query, thereby; the LBS server will not be able to identify. Apparently, the perfection of this mechanism depends on the number of fake requests generated; the more fake queries generated the more robust and secure the system becomes. The problem with this technique is that as the number of requests sent out by a user grows, the LBS may suspect that it is under an attack and thus the requests may be ignored. Moreover, receiving a big number of requests can slow down the server's response time significantly.

Ghinita *et al*. [10] have proposed a novel approach based on the Private Information Retrieval (PIR) scheme [11] as well as Grid Cells (GCs). The PIR scheme is used to retrieve data from a database without revealing the content of the queries or the identity of the user. GCs technique is used to request a reduced set of LBS which represents the area of interest to the user. The GCs firstly enquire from the server about the appropriate cells, and then retrieve anonymously suitable objects. Ghinita's technique succeeds in solving some of the issues associated with the abovementioned techniques. However, it relies on unguaranteed expectations like extensive data processing on the user's side. In most cases, the user is submitting the request through a mobile phone that has very limited processing capabilities.

Rebollo-Monedero and Forné [12] have proposed a mathematical model to minimize the risk of privacy violations in PIR's queries. They presented a promising system to enhance LBS exchange protocol and make communication more secure, despite using a TTP server as middleware between the users and the LBS server.

3. Secure Location-Based Services

To preserve the privacy of the user while interacting with the LBS server, we present in this section a novel approach based on homomorphic encryption scheme to preserve the privacy of the user while interacting with the LBS server.

Homomorphic encryption schemes allow performing arithmetic operations (additions and multiplications) over encrypted data, meaning that the result of an arithmetic operation would be the same whether applied over plain bits or encrypted bits. Our work uses a symmetric encryption scheme as a basis to request LBS services anonymously and guarantee retrieving only suitable data.

Figure 1 depicts a high level architecture of the proposed Location Based Service. A user encrypts the request, which consists mainly of the user's geographical position and the category of the service (Bank, University, etc.) he/she is looking for, and then sends the encrypted request to the LBS server. This latter performs a search on the location database and produces an encrypted result that matches the search criterion. The encrypted records are returned to the user and upon decryption, the requesting party gets the location of the nearest services.

The encryption scheme is defined as: $c = m + 2r + s_k * Q$ where $c = \varepsilon_{sk}(m)$ is the cipher text of a bit m encrypted under the secret key s_k; r and Q are two random integers. 2r is called the noise of the cipher text. The decryption scheme uses the relationship $m = (c \bmod s_k) \bmod 2$. By carefully choosing the size of the secret key s_k and the

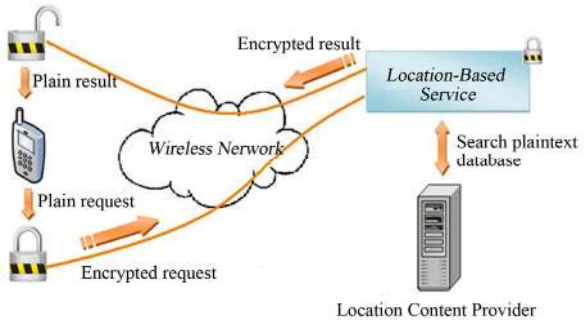

Figure 1. Architecture of secure Location-Based Service.

random values r and Q, this encryption scheme is proved to be semantically secure [13]. The size of each of these elements is based on a security parameter called λ, whereby s_k is an odd λ^5-bit number, and r and Q are λ-bit and λ^2-bit numbers, respectively. As a consequence, each bit encrypted using this scheme would be represented in at most $\left(1+\mathrm{Log}_{10}(\mathrm{P})\right)$ decimal digits, where P is λ^7-bit number. Furthermore, this scheme can support a finite number of arithmetic operations over the same ciphertexts, since it depends mainly on the ratio s_k/r. Thereby, we are able to decrypt successfully a bit c as long as the noise value is less than s_k. It is obvious to notice that this value doubles after each addition and squares after each multiplication. Therefore, our proposed process must be carefully executed such that it can terminate tasks before reaching the upper limit of the noise value. It is worth noting that the ability to support a high number of operations depends mainly on the parameter of security λ. If the latter is relatively large, then the ratio s_k/r will be large enough to support a considerable number of computations. However, an encryption scheme that uses a big value of λ produces big encrypted values, whereby the time needed to perform operations will be relatively long. In our system we use the Karatsuba algorithm [14] to manipulate big integer values. This algorithm allows performing more than 10^6 integer operations in less than one second (tested on a personal computer with 2 GB memory with security parameter $\lambda = 5$). This is a practical situation, especially if we consider the size (order of 2^{78125}) of the encrypted values generated by λ.

In our system, processing a user's request goes through the following four main steps:

1) Localizing category;
2) Localizing services;
3) Filtering services;
4) Generating results.

We demonstrate how these steps work by the following example. Assuming that the user needs to enquire about the nearby hospitals, he/she sends an encrypted request that represent both his/her current location (x,y) and the enquired category (hospital in this case). Once the server receives the request, it uses the user's position to calculate distances and localize nearby services. Thereafter, it selects only the objects enquired about, and sends them back to the user in encrypted form. We note that encryption scheme, described, allows performing operations between plain and encrypted bits and the resulting record becomes encrypted.

In the next sub-sections, we provide complete details on each of these main processing steps.

3.1. Localizing Categories

The LBS database is structured as a tree, as shown in **Figure 2**. Thus, localizing suitable objects must be preceded by localizing the associated category. This process requires an exhaustive search, since the server doesn't have access to the content of the user's request since it's encrypted. The server needs to compare, bit by bit, the encrypted category, requested by the user, to all available categories in the database. The following formula is used for that comparison:

$$\forall C \in \mathrm{LBS}\ I_C = \prod_{i=0}^{size-1} \left[\!\left[\left(1 \oplus c_i\right) \oplus v_i \right]\!\right] \qquad (1)$$

where *size* is the number of bits used to encode one category, v_i is the i^{th} bit in the enquired category, and c_i is the i^{th} bit in category C that is available in the LBS's database.

This formula focuses on comparing the two categories c and v by verifying whether their sequences of bits are similar or not, knowing that c is encrypted. Towards that end, we compare separately each couple (c_i,v_i) by calculating $1 \oplus c_i \oplus v_i$. The latter results in an encrypted value that either equals to $\varepsilon_{sk}(1)$, if c_i is an encrypted form of v_i or $\varepsilon_{sk}(0)$ otherwise. Then, it is possible to verify whether the compared categories are the same, by checking if all generated values are $\varepsilon_{sk}(1)$. Therefore, we calculate the product I_C of these encrypted bits and if we get $\varepsilon_{sk}(1)$, then the categories are the same. Otherwise, we confirm that at least one pair (c_i,v_i) exists such that $v_i \neq c_i$, meaning that the compared sequences are not the same. The values of I_C are used by the server to filter out the objects that belong to the enquired category.

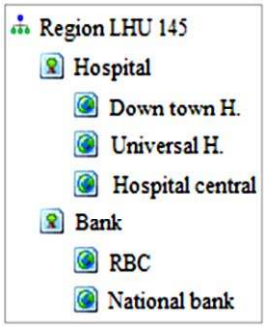

Figure 2. LBS's structure.

We should finally mention that each entry in I_C will have a noise in the order of $\lceil size \times (\lambda + 1) \rceil$-bit value, since it is the product of size encrypted bits, which are in the order of $(\lambda + 1)$-bit.

In **Figure 3** we show the noise values, produced from localizing categories, with respect to the bit sizes of these categories.

3.2. Localizing Services

In this sub-section we describe the mechanism that locates objects that surround the user's position.

The aim of our approach is to allow users to find the nearest targets while preventing LBS from identifying their positions. Therefore, we use the encrypted position $(\varepsilon_{sk}(X), \varepsilon_{sk}(Y))$ and calculate, based on the Manhattan distance [15， 16], the distance separating them from the stored targets. This distance, depicted in **Figure 4**, is calculated between two points $A = (X_A, Y_A)$ and $B = (X_B, Y_B)$ as follows:

$$d(A,B) = |X_B + (1 \oplus X_A)| + |Y_B + (1 \oplus Y_A)| \quad (2)$$

The relevant positions are presented as a set of bits, and therefore, binary addition is mandatory to calculate the distance. This arithmetic operation results in noise values ranging from λ-bit to $(\text{sizeXY} \times \lambda)$-bit values, which are produced when using the same encrypted bits for calculating both the current bit S_i and the carry bit S_{i+1}. Here, λ, sizeXY, and S_i are the original noise, the bit length of the coordinate (X,Y), and the i^{th} bit in the resulted addition S. In other words, this distance produces a noise value confined between 2λ-bit and ($\text{sizeXY}^2 \times \lambda$)-bit, since it is the addition of two sequences with ($\text{sizeXY} \times \lambda$)-bit noise value.

In **Figure 5** we show the noise caused by this step in terms of the coordinate's bit length.

3.3. Filtering Targets

The distance that is separating the targets from the user

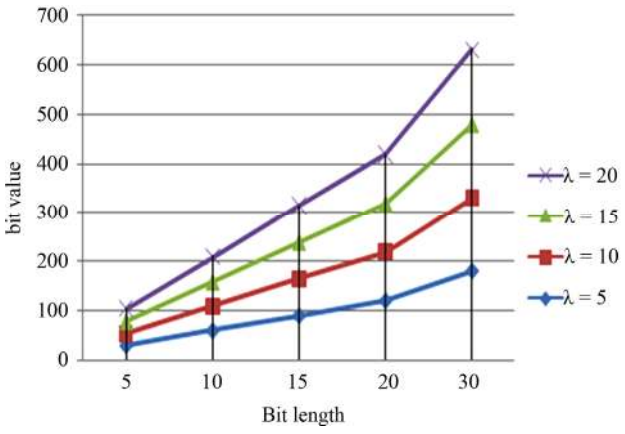

Figure 3. Noise value produced while localizing categories.

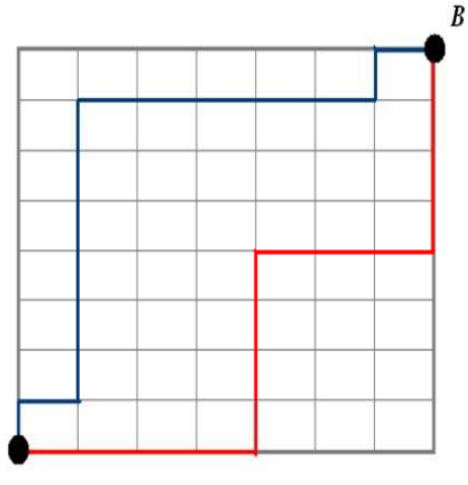

Figure 4. Manhattan distance between two points.

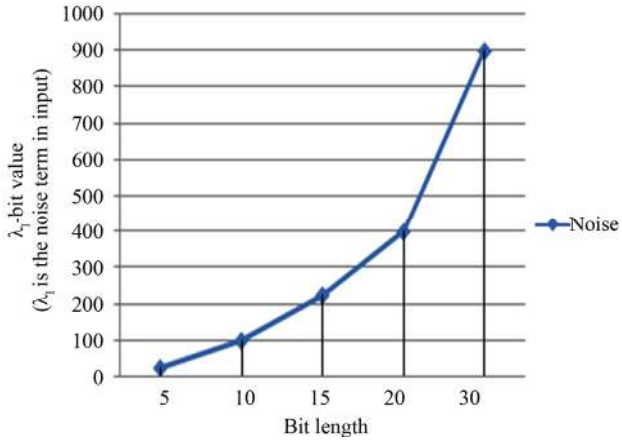

Figure 5. Noise value when calculating distances.

allows the server to decide which records to suggest for that user. Although the trivial solution is to send back all of the available targets and leave it for the user to filter these targets, this behavior may lead to an excessive processing time and transmission bandwidth. This is because the user must decrypt the results before filtering them which is a time-consuming procedure. Moreover, forcing a user to receive and process many objects irrelevant to his/her search may unnecessarily waste the resources of the user. Therefore, we propose here two novel approaches to mitigate this situation. in one approach, we use a blind sorting process that arranges the targets based on their encrypted distance, and then chooses the closest ones (number of records is known). In the second approach, we blindly localize the points that belong to a coverage area. In what follows we discuss these two approaches, the details about their functionality, and we highlight their benefits and drawbacks.

3.3.1. Blind Sorting

We exploit the PIR methodology and propose a novel

circuit that sorts encrypted values. That is, our circuit allows us to arrange the available locations based on their encrypted distance. Our novel circuit uses the principle of blind comparison. This principle compares two sequences of bits, of length n by performing binary subtraction between them. The n^{th} bit resulting from this subtraction is used to check the nature of the comparison, since this latter is negative whether the bit value is $\varepsilon_{sk}(1)$ and positive otherwise. Blind comparison, however, suffers from a major drawback related to the value of noise that grows rapidly before finishing the process. Therefore, we enhance the model by proposing a novel technique that divides the set of bits into two parts, namely, low and high, and then compares these parts separately, as shown in **Figure 6**.

The partial comparisons conducted on the low and high parts of the targeted sequences are used to finalize the comparison between the two sequences using the following equation:

$$B_c = B_c^{''} + B_c^{'} \times \overline{B_c^{''}} \tag{3}$$

where, B_c, $B_c^{'}$ and $B_c^{''}$ are the resulting encrypted bits of the comparison, the low part, and the high part, respectively. In **Table 1** we show the relation between these values, and **Figure 7** shows how our technique stabilizes the value of the noise, since in both low and high parts we reset the noise value.

Our comparison circuit is then used to arrange the selected objects in a certain order. We compare the two first elements to see if their associated bit B_c is equal to $\varepsilon_{sk}(1)$. If so, we permute these elements, otherwise we keep them in their original order. This process is repeated for each adjacent pair of elements until all of the elements are compared. After that, the full comparison is repeated N times from the beginning, where N is the number of items available in R the set of targets belonging to a category.

We should mention here that having B_c in an encrypted form may hinder the swap of bits described above. This problem is overcome by using a passkey formula that allows the executor to use the encrypted B_c to apply the comparison. The passkey formula is defined as follows:

$\forall R1 \in R \; and \; R2 \in R$:

$$R1_{new} = B_c \times R1_{old} + \overline{B_c} \times R2_{old} \tag{4}$$

$$R2_{new} = \overline{B_c} \times R1_{old} + B_c \times R2_{old} \tag{5}$$

where $R1_{old}$, $R2_{old}$ and $R1_{new}$, $R2_{new}$ are the old and new values of the compared entries. **Figure 8** shows the performance of the latter technique in terms of the noise values it achieves. The technique generates high values of noise because of the fact that the same encrypted values are used N^2 times before arranging all of the items.

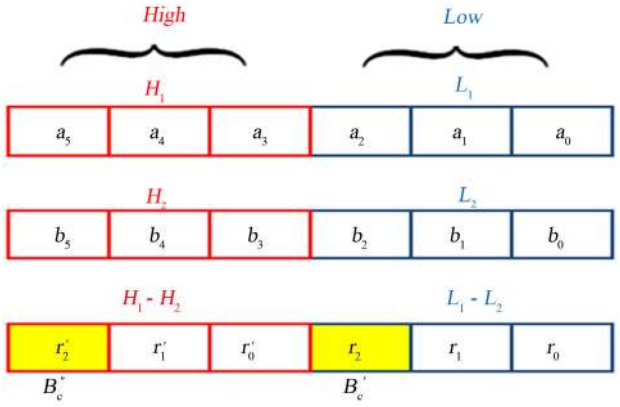

Figure 6. Translating the comparison of two set of bits to their low and high parts.

Table 1. Nature of comparaison from calculated bit value.

$B_c^{'}$	$B_c^{''}$	B_c	Order
$\varepsilon_{sk}(1)$	$\varepsilon_{sk}(1)$	$\varepsilon_{sk}(1)$	$\geq$
$\varepsilon_{sk}(1)$	$\varepsilon_{sk}(0)$	$\varepsilon_{sk}(1)$	$\geq$
$\varepsilon_{sk}(0)$	$\varepsilon_{sk}(1)$	$\varepsilon_{sk}(1)$	$\geq$
$\varepsilon_{sk}(0)$	$\varepsilon_{sk}(0)$	$\varepsilon_{sk}(0)$	$<$

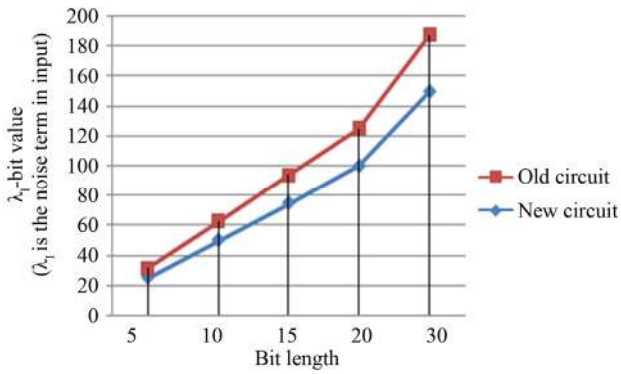

Figure 7. Noise value resulting from the new circuit compared to the old circuit.

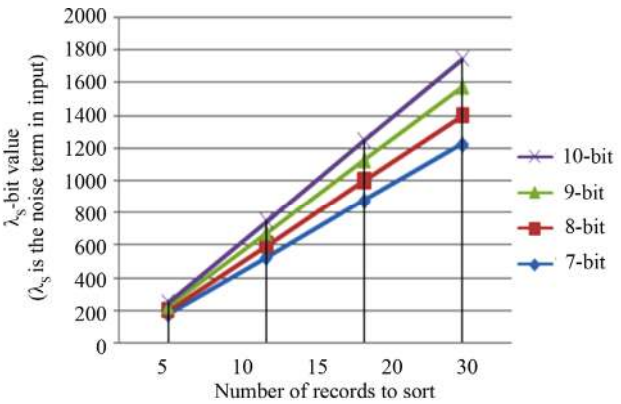

Figure 8. Noise value generated from arranging entries.

Therefore, unless the security parameter λ is made big enough (*i.e.*, $\lambda > 15$), this circuit does not perform well due to the maximum noise that is reached before finishing the process. For that reason, we propose a lighter technique that focuses on finding the targets that belong to a specific area, instead of arranging all the objects and then choosing the nearest one, in such a way to filter out the suitable entries while keeping reduced noise values.

3.3.2. Coverage area

Objects can be chosen based on the specific area, surrounding the user's location within a radius P, as shown in **Figure 9**. For that purpose, we compare distances, calculated in the third step to P and then all associated B_C will form a set of encrypted bits called L_R. Thus, the location belongs to the specified area only if $L_{R,i}$ is equal to $\varepsilon_{sk}(1)$. Furthermore, each category in the database T will have a corresponding L_R that indicates anonymously whether a target is suitable or not.

The fact that the content of L_R is encrypted forces us to find a special process to extract only $L_{R,i}$ that are equal to $\varepsilon_{sk}(1)$. Therefore, we need first to calculate the sum of L_R using the equation:

$$\forall R \in \mathrm{T} : S_R = \sum_{i \le R} L_R \tag{6}$$

This sum is calculated using elementary symmetric polynomials since this technique keeps the noise value at the order of n, whereby n is the size of L_R. It is then possible to localize the i^{th} valid target by calculating the new sequence $L_R^{'}$ as:

$$\forall R \in TL_R^{'} = L_R \times \prod_{i=0}^{n} \left(1 \oplus \eta_i \oplus S_{R,i}\right) \tag{7}$$

where η_i is the binary representation of the index of the element to select. Moreover, $L_R^{'}$ contains only one bit value equal to $\varepsilon_{sk}(1)$ (the index to localize) while all others are equal to $\varepsilon_{sk}(0)$. This sequence leads to constructing a new database $T^{'}$ of rows $R^{'}$ that contains only the targets that belong to the coverage area. The new database $T^{'}$ can be formed as follows:

$$\forall R \in T, \forall R^{'} \in T^{'} : R^{'} = \sum_{i \le R} (L_{R,i}^{'} \times R_i) \tag{8}$$

where R_i is the i^{th} target available in the enquired category.

Looking for objects in a specific area produces an acceptable noise value. This is achieved due to the fact that an important number of records can be supported and the circuit can be terminate before reaching the noise's limit. In **Figure 10** we show the noise produced in this stage.

3.4. Generating Results

The last step enables the server to select only the rows $R^{'}$ that belong to the appropriate category. For that purpose

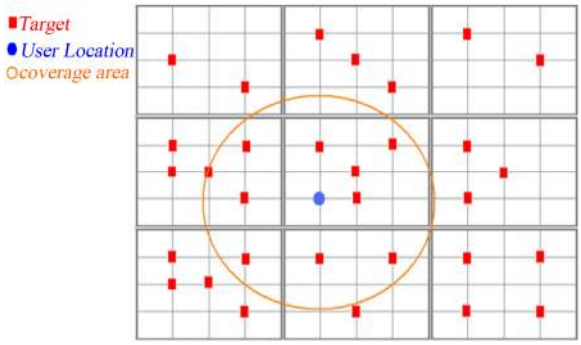

Figure 9. Coverage area.

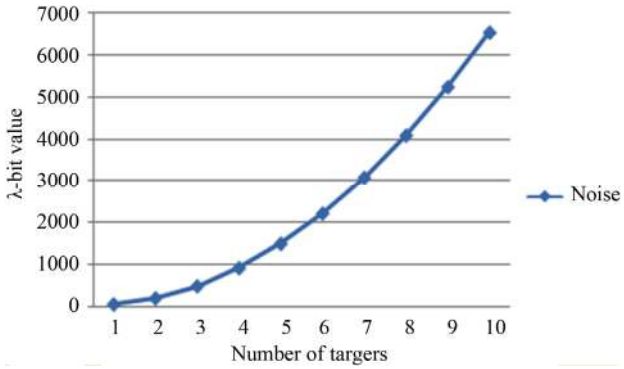

Figure 10. Noise value generated while filtering objects in a specific category.

we use the sequence of bits I_C that is calculated during the first step, and then multiply it to the locations $R^{'}$ that are available in the fixed perimeter. After that, we calculate the resulting location as follows:

$$\forall R^{'} \in T^{'} : R^{''} = I_C \times R^{'} \tag{9}$$

4. Implementation and Results

In this section we study the performance of our proposed system. In **Figure 11** we show the data flow of our proposed system. A user can auto-locate himself/herself (the region where he belongs as well as his position) using smart phone capabilities. Then, the user's software encrypts both his coordinate and the type of service he/she is targeting, and sends them to the server. The server retrieves the requested targets depending on the encrypted information. Thereafter, it sends these encrypted targets to the client to be decrypted and viewed by the user.

Our system has a limitation in terms of the number of records it can support. As the number of the stored targets grows, the system needs to conduct a significant number of arithmetic operations. We may even reach the upper limit of the noise value before extracting all the targets, and a successful decryption will not be guaranteed. We can overcome this problem by using large values of security parameter λ.

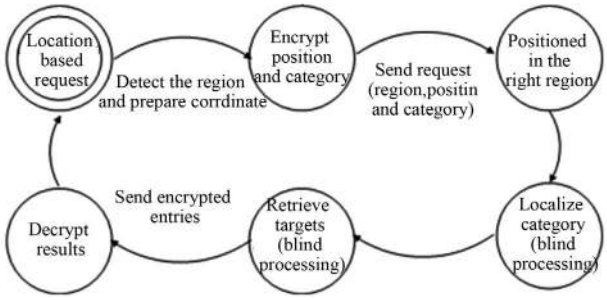

Figure 11. Data flow.

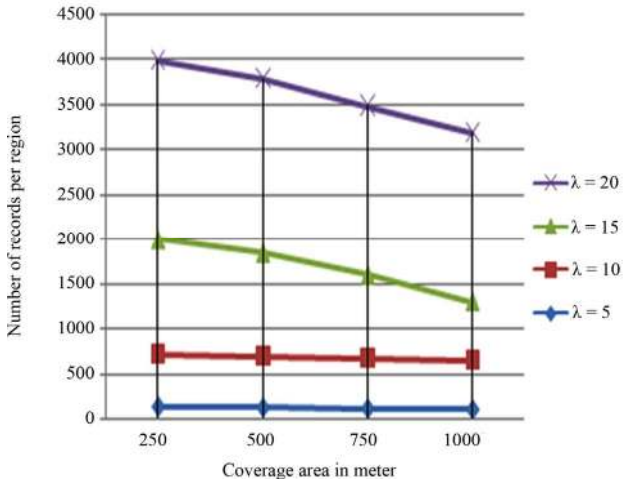

Figure 12. Number of records supported for different values of λ.

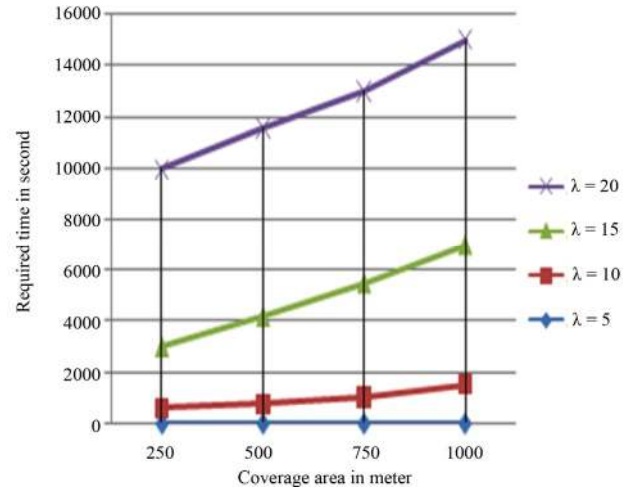

Figure 13. Processing time.

Our experimental results are shown in **Figures 12** and **13**. Our experiments are conducted on a personal computer with 2 GB memory and dual core CPU of 2 GHz. **Figure 12** depicts the performance in terms of the number of records per region we can support as a function of λ. **Figure 13** illustrates the performance in terms of the processing time consumed in each case. This figure shows that whenever the coverage area gets bigger, the number of operations increases, while the number of records supported gets lesser.

5. Conclusions and Perspectives

The concept of answering location-related information for encrypted positions is promising to improve security needs. Indeed, such a mechanism can strongly attract the attention of researchers as it supports the preservation of the users' privacy.

In this paper we developed a novel fully secure location-based mechanism based on a homomorphic encryption scheme. We described the circuits that allow a LBS server to process encrypted inputs to retrieve targeted records that match the user's request. We also discussed the limitations and drawbacks of our proposed system and suggested some solutions to make it more practical. The performance of our system was tested through extensive experiments to extract useful results related to the noise generated and the processing time consumed.

As future work, we are planning to improve the performance of the encryption scheme to be able to support a large number of services. This step is mandatory to make it possible for a commercial deployment of our LBS system.

REFERENCES

[1] Clarinox Technologies Pty Ltd., "Real Time Location Systems," 2009. http://www.clarinox.com/docs/whitepapers/RealTime_main.pdf

[2] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking," *Proceedings of the* 1*st International Conference on Mobile Systems*, *Applications and Services*, San Francisco, 5-8 May 2003, pp. 31-42. doi:10.1145/1066116.1189037

[3] C. Y. Chow, M. F. Mokbel and X. Liu, "A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-Based Services," *Proceedings of the* 14*th Annual ACM International Symposium on Advances in Geographic Information Systems*, Arlington, 10-11 November 2006, pp. 171-178. doi:10.1145/183471.1183500

[4] B. Gedik and L. Liu, "Location Privacy in Mobile Systems a Personalized Anonymization Model," *Proceedings of the* 25*th International Conference on Distributed Computing System of the IEEE ICDCS*, Columbus, 10 June 2005, pp. 620-629. doi:10.1109/ICDCS.2005.48

[5] M. F. Mokbel, C. Y. Chow and W. G. Aref, "The New Casper: Query Processing for Location Services without Compromising Privacy," *Proceedings of the VLDB* 2006, Seoul, 12-15 September 2006, pp. 763-774.

[6] D. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Transactions on Automatic Control*, Vol. 24, No. 6, 1979, pp. 843-854. doi:10.1109/TAC.1979.1102177

[7] B. Gedik and L. Liu, "A Customizable k-Anonymity Model for Protecting Location Privacy," *Technical Report*, Georgia Institute of Technology, Atlanta, 2004.

[8] H. Kido, Y. Yanagisawa and T. Satoh, "An Anonymous Communication Technique Using Dummies for Location-Based Services," *Proceedings of the International Conference on Pervasive Services of the IEEE ICPS* 05, Santorini, 11-14 July 2005, pp. 88-97. doi:10.1109/PERSER.2005.1506394

[9] T. You, W. Peng and W. Lee, "Protect Moving Trajectories with Dummies," *Proceedings of the International Conference on Mobile Data Management*, Mannheim, 1 May 2007, pp. 278-282. doi:10.1109/MDM.2007.58

[10] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi and K.-L. Tan, "Private Queries in Location-Based Services: Anonymizers Are Not Necessary," *Proceedings of the SIGMOD* 08, Vancouver, 9-12 June 2008, pp. 121-132.

[11] C. Gentry and Z. Ramzan, "Single-Database Private Information Retrieval with Constant Communication Rate," *Proceedings of the* 32*nd International Colloquium on Automata, Languages and Programming*, Lisboa, 11-15 July 2005, pp. 803-815.

[12] D. Rebollo-Monedero and J. Forne, "Optimized Query Forgery for Private Information Retrieval," *IEEE Transactions on Information Theory*, Vol. 56, No. 9, 2010, pp. 4631-4642. doi:10.1109/TIT.2010.2054471

[13] C. Gentry, "A Fully Homomorphic Encryption Scheme," Ph.D. Thesis, Stanford University, Stanford, 2009.

[14] http://en.wikipedia.org/wiki/Karatsuba_algorithm

[15] http://en.wikipedia.org/wiki/Taxicab_geometry

[16] Y. Gahi , M. Guennoun and K. El-khatib, "A Secure Database System Using Homomorphic Encryption Schemes," *Proceedings of the* 3rd *International Conference on Advances in Databases*, *Knowledge*, *and Data Applications*, St. Maarten, 23-28 January 2011, pp. 54-58.

