Asymmetric Scalar Product Encryption for Circular and Rectangular Range Searches

Rodrigo Folha[1(✉)], Valeria Cesario Times[1], and Claudivan Cruz Lopes[2]

[1] Center of Informatics, Federal University of Pernambuco, Recife, PE, Brazil
{rbf2,vct}@cin.ufpe.br
[2] Federal Institute of Education, Science and Technology of Paraiba, Patos, PB, Brazil
claudivan@ifpb.edu.br

Abstract. Although spatial database applications and location based systems require the execution of several types of searching operations over spatial data, works related to encrypted spatial data address a limited set of searching operations, restricting their use in real applications. This article proposes an encryption scheme that enables circular range search, rectangular range search and kNN operation over encrypted spatial data. Also, we have compared the encryption functions of our scheme with other encryption schemes and, even though the results have shown a similar performance, our work allows the execution of circular and rectangular range searches by using a unique encryption scheme.

Keywords: Encrypted spatial database · Asymmetric scalar product encryption · Circular range search · Rectangular range search

1 Introduction

A solution for protecting data confidentiality is using cryptography, in which data are encrypted in the user environment before being sent to a cloud. Nevertheless, searching operations executed over encrypted data require decryption, which may cause a processing overhead or compromise data confidentiality when the decryption is carried out in the cloud. Thus, encryption techniques for spatial data are addressed in literature, and allow calculations and operations to be executed directly over encrypted spatial data. The use of these techniques aims to reduce the overhead caused by encryption on data processing and avoid data decryption in unsafe environments.

Among the proposed schemes found in the literature, there are Circular Range Search Encryption (CRSE) [1], which enable circular range search; Scalable Multidimensional Range Search (MAPLE) [2,3], both enabling the execution of rectangular range searches over encrypted data; Asymmetric Scalar Product Encryption (ASPE) [4], which allows comparisons to be made between encrypted points stored in a database and encrypted query points used as parameters in k nearest neighbor operations; Distance Preserving Transformation

J. Pokorný et al. (Eds.): ADBIS 2016, LNCS 9809, pp. 183–197, 2016.
DOI: 10.1007/978-3-319-44039-2_13

(DPT) [5], which preserves the real distance between the encrypted data; and the work in [6] that enables rectangular range searches over nodes of an encrypted R-tree by using an asymmetric scalar product encryption scheme.

These schemes represent the state-of-the-art theory in the area of spatial data encryption. They are nevertheless limited to using a single type of geometry as search predicate (i.e. either circular range or rectangular range) that restricts their use in spatial database applications. Therefore, proposing a scheme that supports the use of different predicates in searches is the focus of this paper. We introduce the following contributions to the area: two encryption schemes - with a trade-off between security strength and performance - that encrypt spatial data and enable circular and rectangular searches, named CR-ASPE (Asymmetric Scalar Product Encryption for Circular and Rectangular range search); a formalization of the correctness of the scheme's operations; a security analysis; and a performance evaluation.

This article is organized as follows: Sect. 2 presents the main concepts used in this article; Sect. 3 discusses related work; Sect. 4 explains the problem to be solved; Sect. 5 presents CR-ASPE; Sect. 6 contains a performance evaluation comparing CR-ASPE with ASPE and DPT schemes; and, finally, Sect. 7 concludes the paper and addresses future work.

2 Basic Concepts

Before introducing our work, we briefly present some concepts used through this article.

2.1 Types of Range Search over Spatial Data

Range searches over spatial data usually receive as input a set of geometric objects P and a region R in a space, and are aimed at retrieving a subset of P that is inside or intersects R. The region R may assume different formats, such as circle, halfspace, rectangle, and polygon. Thus, we introduce the preliminary concepts of the following operations.

The k Nearest Neighbor (kNN) operation searches for the closest k objects from a point of interest. kNN is frequently used in data mining, machine learning and recommendation systems [4]. The circular range search inspects all points that are within the radius of interest. The range comprises all points at the same distance r from a central point, where r represents the radius of the n-sphere.

A halfspace is a space resulting from the division of an Euclidean space by a hyperplane. In order to execute halfspace range searches, two equidistant points with respect to the hyperplane (named *anchor points*) are chosen. They must be collinear and the line formed by them must be perpendicular to the hyperplanes. Thus, halfspace range search receives a point as input and indicates in which halfspace it is located based on the distance to the closest anchor point [6].

A third concept is that of rectangular range search, which is important for spatial data because it is constantly used in r-tree operations as Minimum

Bounding Rectangle (MBR) [7]. A MBR is the minimal rectangle necessary to envelop a bi-dimensional geometric object. If we consider each rectangle's edge as the hyperplane that divides a halfspace, we can represent a rectangular range search as a conjunction of four halfspace range searches.

2.2 Data Splitting and Addition of Artificial Dimensions

[4] proposed two techniques aiming to increase the security level of encrypted spatial data, which are based on the number of dimensions of the spatial data, such as latitude, longitude, altitude and velocity. The first technique is the Random Asymmetric Splitting, where each dimension of spatial data is split. In order to split it, we may have a random bit vector that indicates which positions should be split. For example, in a three-dimensional space, consider a bit vector $= (0, 1, 1)$, and a point $p = (5, 3, 2)$; two split vectors are randomly picked, such as $p_a = (5, -2, -7)$ and $p_b = (5, 5, 9)$; ergo, $p = p_a + p_b \cdot bitvector$. The second technique is the addition of artificial dimensions to the spatial data. This method attributes random values to artificial dimensions in a way that the scalar product of two asymmetric points over the artificial dimension value is 0, preserving the result of scalar product and increasing the number of data dimensions. As it is an asymmetric method, it splits query points by using the inverse of the bit vector when $bitvector[i] = 0$; otherwise, it does not.

2.3 Levels of Attacker's Knowledge

Regarding the security of encrypted spatial data, we assume that an attacker may obtain knowledge about encrypted spatial data stored in outsourced databases. This knowledge may enable three attack levels [4]: level 1, when the attacker has access to all encrypted data; level 2, when the attacker has access to all encrypted data and a subset of unencrypted data; and level 3, when the attacker has access to all encrypted data, a subset of unencrypted data and the correspondence between unencrypted data and equivalent encrypted data.

3 Related Work

Although there are several works that present solutions to performing operations over encrypted scalar data such as numbers, dates and keywords [8,9], such solutions are no applicable to spatial data. In plain spatial data, circular or rectangular range searches, among others, are calculated from the distance between spatial geometries. Thus, one alternative would be to encrypt the spatial data preserving the distance. However, the distance preservation is subject to attacks [10] that limit the use of some schemes, such as distance-recoverable encryption (DRE) schemes. Such DRE schemes, e.g. Distance Preserving Transformation (DPT) [5], encrypt spatial data by moving them to a different space, but preserving all distances between them. Hence, if an attacker has access to a

subset of plain data and encrypted data, he is able to discover the correspondence between plain data and encrypted data, which may reveal the encryption key.

Related work in this area propose different techniques to encrypt spatial data without revealing the distance between two points [1–4,6]. Nevertheless, to the best of our knowledge, those are able to execute only one type of search over spatial data.

Predicate encryption [1,11,12] is an encryption scheme that generates tokens as predicates, which are used to verify whether a piece of encrypted data satisfy their constraints by executing an inner product. In [2,3], the authors propose schemes based on predicate encryption to allow rectangular searches to be executed on encrypted R-trees, reducing the search complexity from $\mathcal{O}(n)$ to $\mathcal{O}(\log n)$. In [1] a predicate-based encryption scheme to execute circular range searches is presented.

ASPE was proposed in [4] to execute kNN operations over encrypted data without using any data structures. ASPE encrypts query points and database points in two different ways - by using a invertible matrix to encrypt the database points in addition to its inverted matrix to encrypt query points, avoiding an attack based on distance preservation between unencrypted data and encrypted data, hence avoiding distance recovery. In [4], two different schemes were proposed, i.e. ASPE 1 and ASPE 2. The difference between them is the insertion of additional dimensions to increase the number of variables in encrypted point compositions and a random splitting of the points to improve the security of the scheme at the expense of performance.

Our work aims to support circular range search, rectangular range search and kNN operations using a single scheme, in addition to providing security against honest-but-curious attackers with different levels of knowledge.

4 Problem Definition

The distance between encrypted spatial values should not preserve the distance between the corresponding spatial values, as it may reveal the spatial data [4,10]. Thus, for security reasons, several works have proposed encryption schemes which are not based on distance for computing searches on encrypted spatial data [1–4,6]. However, these schemes only allow for a single type of searching, limiting their functionality.

For example, consider two systems using encrypted spatial databases, namely system A and system B. Systems A and B have adopted encryption schemes that enable circular range searching and rectangular range searching on encrypted spatial data, respectively. Suppose a user wants to find restaurants within 2 km from his current location, which characterizes a case of circular range searching. By using system A, the user can find all restaurants that satisfy his condition, as it is capable of performing circular range searching on encrypted spatial data. On the other hand, using system B will return false candidates. Then, suppose a user wants to analyze a disease in a rectangular area, such as a district or a street, in order to extract the number of infected people. By using system B, the

user can obtain the exact number of infected people since system B can execute rectangular range searching on encrypted spatial data, whereas using system A will not ensure that all infected people will be selected. Finally, suppose a user wants to call a taxi, and expects that a limited number of taxi drivers can receive his call. This is a typical use of kNN computation. In this case, circular range searching and rectangular range searching cannot determine the drivers who are closest to the user's location, hence, neither system A nor system B will be able to perform this computation. Therefore, systems A and B have limited functionalities due to their encryption schemes.

In order to fulfill the aforementioned limitations, this work proposes an encryption scheme for spatial data that allows circular range searching, rectangular range searching and kNN operations directly over encrypted spatial data.

5 CR-ASPE

We propose an asymmetric product scalar encryption for circular and rectangular searches without compromising security or losing performance, named CR-ASPE, and detailed as follows.

5.1 Basic CR-ASPE

CR-ASPE enables comparisons over the encrypted data without preserving the distance between spatial points. The CR-ASPE asymmetry consists of encrypting the data point without preserving distance between them, and encrypting a query point to allow the comparison with encrypted data points. Therefore, we can compare two encrypted data points and define which is closer to a reference point using a scalar product. The comparison is possible through the use of a invertible matricial key. We present the basic functions of CR-ASPE, as follows:

CR-ASPE Scheme 1

Key: a $(d+2)\times(d+2)$ invertible matrix M, where d is the number of dimensions of plain data, such as latitude, longitude, altitude.

Tuple encryption function E_d: Given a point from database p, the function creates a (d+2)-dimensional point $\hat{p} = (p^T, -0.5||p||^2, 1)^T$ and encrypts it, $p' = M^T\hat{p}$.

Search encryption function E_q: Given a query point q and a random number $r > 0$, the function creates a (d+2)-dimensional point $\hat{q} = r(q^T, 1, -0.5||q||^2)^T$ and encrypts it, $q' = M^{-1}\hat{q}$. The factor r makes it possible to randomize the query point, in case the user submits it twice.

Decryption function D: Given an encrypted point p' from the database, the function extracts the original point, $p = \pi_d M^{T^{-1}} p'$ where $\pi_d = (I_d, 0, 0)$ is a $d \times (d+2)$ projection matrix and I_d is a $d \times d$ identity matrix.

Distance comparison operator A_e: Given two encrypted points p'_1 and p'_2 and an encrypted query point, the function calculates whether p'_1 is closer to q' than p'_2 is, assessing if $(p'_1 - p'_2) \cdot q' > 0$. This function is sufficient to run the kNN operation, which compares the database points with a reference point two-by-two using a distance comparison operator.

To support circular and rectangular range searches, some functions must be introduced into a preprocessor module on the data owner side. The auxiliary functions are presented below for each search.

Circular Range Search. To adapt a circular range search to a comparison between encrypted points, we must select a random point in boundary circle and encrypt it as a data point b'. Thus, to a given encrypted point p' from encrypted database, it is possible to discover if q' is closer to p' or to b' using the distance comparison operator. The necessary functions to enable circular range search are listed below.

1. **Get Circle Point** (q_{center}, *distance*) $\rightarrow q_{center} + distance$. Given a point ($q_{center}$) and a distance, this function will pick a random point in the circle formed by q_{center} as center and *distance* as its radius.
2. **Circular Range Encryption** (p,q) $\rightarrow$ (p',q'). Given a circle's boundary point (p) from Function 1 and the query point (q), it encrypts them, using the encryption functions of the scheme, returning p' and q'.
3. **Circular Range Search** (p'_1,p'_2,q') $\rightarrow$ {True, False}. It is executed on the outsourced database. Given a point from an encrypted database (p'_1), the encrypted point (p'_2) and the encrypted query point (q') from Function 2, it runs a scalar product operation to verify whether p'_1 is nearer to q' than p'_2 is, using the distance comparison operator. If it is, then the point satisfies the circle range search; otherwise, the point is beyond the circle's boundaries.

The Function 1 generates a random database point even if the circle's center is the same. Therefore, in the case the same search is executed twice, encrypted query and generated database point are unlikely to recur, avoiding that an attacker recognizes that the same search is executing again. For the same reason, encrypted searches do not reveal any information about the radius. Moreover, it is not possible to distinguish whether an operation is a circular range search or a kNN operation, as they are all based on distance comparison.

Rectangular Range Search. To execute rectangular range searches, our approach uses halfspace range searches to assess whether a point is inside of a rectangle. Each one of a rectangle's edge is a line that separates the inner region of the rectangle from the outer region. Therefore, a rectangular range search will be transformed into a conjunction of halfspace range searches. This transformation is made by the following functions:

1. **Generate Anchor Points** $((r_1^A, r_2^A, ..., ..., r_n^A), (r_1^B, r_2^B, ..., ..., r_n^B)) \rightarrow ((q_1^<, q_2^<, ..., ..., q_n^<), (q_1^\geq, q_2^\geq, ..., ..., q_n^\geq))$. Given two vertices ($r^A$ and r^B)

in a rectangle, which are linked by an edge, this algorithm will choose a line perpendicular to said edge. Then, it will randomly pick two points ($q^<$ and $q^\geq$) which are in the line and equidistant from the edge ($q^<$ and $q^\geq$).

2. **Encrypt Rectangle**. For each pair of linked vertices of the rectangle, two query points are generated by the function in item 1 and encrypted using the same random number (r).
3. **Rectangle Search Operator** $(((q_1^{<'}, q_1^{\geq'}), ..., (q_4^{<'}, q_4^{\geq'})), p') \rightarrow \{\text{True, False}\}$. Given four pairs of anchor points encrypted by Function 2 and a point from an encrypted database (p'), for each pair ($q_i^{<'}$, $q_i^{\geq'}$), it will run a scalar product operation in the outsourced database to verify whether p' is nearer to $q_i^{<'}$ or to $q_i^{\geq'}$ using the distance comparison operator. If it is always near to $q_i^{<'}$, then the point satisfies the rectangular range search; otherwise, the point is beyond the rectangle's boundaries.

Function 1 randomly picks two anchor points. Therefore, in the case the same search is executed twice, the anchor points are unlikely to recur, avoiding that an attacker will link the search with a previously executed search.

Correctness. The operations are calculated using the scalar product over the encrypted data, without including false results. We present the Theorem 1 for kNN and circular range search and Theorem 2 for rectangular range search to guarantee their results.

Theorem 1. *Let p'_1 and p'_2 be encrypted points of the database and q' the encrypted reference point. Thus, the scheme determines whether p_1 or p_2 is closer to q by evaluating if $(p'_1 - p'_2) \cdot q' > 0$.*

Proof. Note that,

$$\begin{aligned}(p'_1 - p'_2) \cdot q' &= (p'_1 - p'_2)^T q' \\ (p'_1 - p'_2) \cdot q' &= (M^T \hat{p}_1 - M^T \hat{p}_2)^T M^{-1} \hat{q} \\ (p'_1 - p'_2) \cdot q' &= (\hat{p}_1 - \hat{p}_2)^T \hat{q}\end{aligned}$$

This scalar product can be represented by

$$\begin{aligned}&= (p_1 - p_2)^T (rq) + (-0.5||p_1||^2 + 0.5||p_2||^2)r + 0.5||q||^2 - 0.5||q||^2 \\ &= 0.5r(||p_2||^2 - ||p_1||^2 + 2(p_1 - p_2)^T q) \\ &= 0.5r(||p_2||^2 - 2p_2^T q + ||q||^2 - ||p_1||^2 + 2p_1^T q - ||q||^2) \\ &= 0.5r(d(p_2, q) - d(p_1, q))\end{aligned}$$

where d is the Euclidean distance between two points. Thus,

$$0.5r(d(p_2, q) - d(p_1, q)) > 0 \Leftrightarrow d(p_2, q) > d(p_1, q)$$

Therefore, if the condition is satisfied, p_1 is closer to the reference point q. □

In case of a circular range search, the search preprocessor transformation ensures *radius* = d(p_2,q), hence $0.5r(d(p_2,q)-d(p_1,q))>0 \Leftrightarrow radius > d(p_1,q)$. Therefore, if the condition is satisfied, p_1 is inside the circle range search.

Theorem 2. *Let p' be an encrypted point of the database, q'_1 and q'_2 be the two encrypted anchor points, $q^{<}$' and $q^{\geq}$' respectively. Therefore, the scheme determines whether p is inside the rectangle by evaluating if $p' \cdot (q'_1 - q'_2) > 0$.*

Proof. Note that,

$$\begin{aligned} p' \cdot (q'_1 - q'_2) &= p'^T(q'_1 - q'_2) \\ p' \cdot (q'_1 - q'_2) &= (M^T\hat{p})^T(M^{-1}\hat{q}_1 - M^{-1}\hat{q}_2) \\ p' \cdot (q'_1 - q'_2) &= \hat{p}^T(\hat{q}_1 - \hat{q}_2) \end{aligned}$$

Since r is the same in q_1 and q_2, this scalar product can be represented by

$$\begin{aligned} &= p^T r(q_1 - q_2) + (-0.5||p||^2 + 0.5||p||^2)r + (-0.5||q_1||^2 + 0.5||q_2||^2)r \\ &= 0.5r(||q_2||^2 - ||q_1||^2 + 2p^T(q_1 - q_2)) \\ &= 0.5r(||p||^2 - 2p^Tq_2 + ||q_2||^2 - ||p||^2 + 2p^Tq_1 - ||q_1||^2) \\ &= 0.5r(d(p,q_2) - d(p,q_1)) \end{aligned}$$

where d is the Euclidean distance between two points. Hence,

$$0.5r(d(p,q_2) - d(p,q_1)) > 0 \Leftrightarrow d(p,q_2) > d(p,q_1)$$

Accordingly, if the condition is satisfied, p is inside the rectangle. □

Security Analysis

Theorem 3. *If a level-3 attacker knows $d+2$ plain points $P = \{x_1, x_2, ..., x_{d+2}\}$ and their corresponding encrypted points $E(P) = \{x'_1, x'_2, ..., x'_{d+2}\}$, he can recover the key K.*

Proof. As the attacker knows the plain points and the corresponding encrypted points, he can set up a system of equations to solve K, $K\hat{x}_i = x'_i$ for $i = 1$ to $d+2$, where $\hat{x}_i = (x_i, -0.5||x_i||^2, 1)^T$. □

Theorem 4. *Scheme 1 is resistant to brute force attacks with level-2 knowledge.*

Proof. As a level-2 attacker does not know the correspondence among the points in P and the encrypted points in $E(DB)$, he may try finding it using a brute-force attack. As presented in Theorem 3, at least $d+2$ points are necessary to discover the key of our scheme. Thus, if $|P| > d+2$, a subset of P may be selected to discover the key, dividing P into two sets, a validating set (P_v) and a training set (P_t) where $|P_t| = d+2$. The initial step is to randomly pick $d+2$ encrypted points from $E(DB)$ to set up equations with P_t in order to discover the key. Then, the result key K_i is verified against points in P_v; if

submitting P_v to an encryption function with K_i generates points from $E(DB)$, K_i is valid; otherwise, K_i is not valid. However, a brute-force attack may test all combinations of correspondences of P_t and $E(DB)$, i.e. A^n_{d+2} tries, where $n = |E(DB)|$. For an example with 50000 encrypted bi-dimensional pieces of data, if an attacker is able to set up and solve 1 million systems of equations per second, it would take over 300 years to compute all combinations. □

Besides brute force attacks, Principle Component Analysis (PCA) [10] may be used to link the correlation of dimensions of known points in P and the correlation of dimensions of encrypted points in $E(DB)$. However, CR-ASPE does not preserve the correlation of dimensions, since each encrypted dimension is a linear combination of all dimensions of original data. An attack based on duplicate analysis [13] retrieves information from repeated occurrences of data in small domains. CR-ASPE is also resistant to duplicate analysis, due to linear combination of dimensions, i.e. even if a dimension is from a small domain, the domain of an encrypted dimension will not be the same.

5.2 Enhanced CR-ASPE Scheme

In Sect. 5.1, we proposed the trivial solution for executing kNN operations, rectangular range search and circular range search. However, CR-ASPE 1 is not secure against an attacker who knows a subset of unencrypted spatial points, the set of encrypted spatial points and the correspondence between them, as shown in Theorem 3, since the attacker may set up and solve the system of equations to recover the key. Therefore, we proposed an enhanced CR-ASPE scheme, CR-ASPE 2, which uses the two techniques of Sect. 2: random asymmetric splitting and adding artificial dimensions, increasing the difficulty to crack.

Key: two $d' \times d'$ invertible matrices M_1 and M_2, a bit vector S with d' elements and a vector w with $d' - (d + 2)$ random numbers, where d is the number of dimensions of plain data and d' is the number of dimensions of encrypted data.

Tuple encryption function E_d: Given a point from database p, the function creates a d'-dimensional point where the first $d + 2$ dimensions are $\hat{p} = (p^T, -0.5||p||^2, 1)^T$. Then, for $i = d + 2$ to d', if $S_i = 1$, $\hat{p}[i] = w_{i-(d+2)}$; otherwise, $\hat{p}[i] = randomnumber$. For the last dimension, where $S_i = 0$, the result of the scalar product of artificial dimensions $\hat{p}$ by w must be equal to zero; consequently, $\hat{p}[i]$ is a number whose value makes this result true. This creates a pair of points $(\hat{p}_a, \hat{p}_b)$. For $i = 1$ to d', if $S_i = 1$, it randomly splits $\hat{p}[i]$ into $\hat{p}_a[i]$ and $\hat{p}_b[i]$; otherwise, $\hat{p}_a[i] = \hat{p}[i]$ and $\hat{p}_b[i] = \hat{p}[i]$ too. Lastly, it encrypts them, returning a pair $(p'_a = M_1^T \hat{p}_a, p'_b = M_2^T \hat{p}_b)$.

Query encryption function E_q: Given a query point q and a random number $r > 0$, the function creates a d'-dimensional point where the first $d + 2$ dimensions are $\hat{q} = r(q^T, 1, -0.5||q||^2)^T$. Then, for $i = d + 2$ to d', if $S_i = 0$, $\hat{q}[i] = w_{i-(d+2)}$; otherwise, $\hat{q}[i] = randomnumber$. For the last dimension, where $S_i = 1$, the result of the scalar product of $\hat{q}$ by w must be equal to

one; consequently, $\hat{q}[i]$ is a number whose value makes this result true. This creates a pair of points $(\hat{q}_a, \hat{q}_b)$. For $i = 1$ to d', if $S_i = 0$, it randomly splits $\hat{q}[i]$ into $\hat{q}_a[i]$ and $\hat{q}_b[i]$; otherwise, $\hat{q}_a[i] = \hat{q}[i]$ and $\hat{q}_b[i] = \hat{q}[i]$ too. Lastly, it encrypts them, returning a pair $(q' = M_1^{-1}\hat{q}_a, q' = M_2^{-1}\hat{q}_b)$.

Decryption function D: Given a pair of encrypted points (p'_a, p'_b) from the database, the function extracts the original points, $p_a = \pi_d M_1^{T^{-1}} p'_a$ and $p_b = \pi_d M_2^{T^{-1}} p'_b$, where $\pi_d = (I_d, 0, 0)$ is a $d \times d'$ projection matrix and I_d is a $d \times d$ identity matrix. After that, if $S_i = 0$, $p[i] = p_a[i]$; otherwise $p[i] = p_a[i] + p_b[i]$.

Distance comparison operator A_e: Given two pairs of encrypted points (p'_{1a}, p'_{1b}) and (p'_{2a}, p'_{2b}), and a pair of encrypted query points (q'_a, q'_b), the function calculates whether p'_1 is closer to q' than p'_2 is, assessing if $(p'_{1a} - p'_{2a}) \cdot q'_a + (p'_{1b} - p'_{2b}) \cdot q'_b > 0$.

Security Analysis. The use of Random Asymmetric Splitting generates 2^d possible configurations, since a bit vector is used to split an original point. In addition to that, adding artificial dimensions will increase the number of dimensions of encrypted data. Therefore, both techniques may be combined to increase the number of possible configurations to $2^{d'}$ in relation to Scheme 1. Thus, a CR-ASPE with 128 dimensions is equivalent to an AES with a 128 bits key size.

Theorem 5. *The CR-ASPE 2 scheme is resistant to a level-3 attacker.*

Proof. Although the attacker has a knowledge $H = \{E(DB), P, I\}$, he does not know the splitting configuration of encrypted points. Hence, for each point p_i in P, he has to suppose a random pair of encrypted point (p'_{ia}, p'_{ib}) in order to set up two systems of equations, $M_1^T \hat{p}_{ia} = p'_{ia}$ and $M_2^T \hat{p}_{ib} = p'_{ib}$, where M_1 and M_2 are unknown matrices from the key. Thus, the attacker does not have sufficient equations to discover the matrices, rendering the scheme resistant to a level-3 attack. □

The incorporated techniques (i.e. random asymmetric splitting and adding artificial dimensions) do not affect the correctness of search functions. However, by raising the security strength, they impact performance, since the complexity of these operations vary according to the number of dimensions.

6 Performance Evaluation

We compared our schemes in terms of performance with the ASPE schemes 1 and 2 proposed in [4] and the DPT scheme presented in [5], since [4] proposed the asymmetric scalar product encryption, and [5] is able to execute searches based on distance. We have conducted the experiments on a computer with 2.60 GHz i7 Intel Core processor, 16 GB RAM and Windows 8.1. All schemes where implemented using Python version 2.7.10. The performance evaluation was based on common functions of all schemes: encryption, decryption, kNN, circular

range search and rectangular range search. As [4,5] cannot perform circular range search and rectangular range search functions on encrypted spatial data, we have to decrypt all data on ASPE and DPT schemes before running the search.

For the experiments, we have firstly generated two sets of random data. The first set generated n-data points with four dimensions, where n varied from 10,000 to 100,000. The second set was generated with d dimensions and 50,000 data points, where d varied from 10 to 100. On CR-ASPE 2 and ASPE 2, we adopted $d' = 80$ to secure our data; however, when $d \geq 80$, we adopted $d' = d+2$ on our scheme and $d' = d+1$ on ASPE. Secondly, we used a real dataset, Shuttle, which may be found in UCI repository [14], containing 58,000 spatial data points with 9 dimensions. We have executed each operation on the schemes 100 times and calculated an arithmetic average with them.

6.1 Experimental Results

We have evaluated the encryption, decryption, kNN, circular range search and rectangular range search functions varying the number of data dimensions and collecting the time in seconds in order to analyze the overhead caused by the artificial dimensions on the enhanced scheme (CR-ASPE 2) in relation to our simplest scheme (CR-ASPE 1). In order to analyze the complexity of the proposed functions, we also varied the number of spatial objects encrypted, collecting the time consumed for each case in seconds.

In Fig. 1b, it becomes clear that even when the number of dimensions of plain data is close to the number of dimensions used to encrypt data by enhanced schemes, the time consumed by the encryption function of both CR-ASPE 1 and ASPE 1 were around 11 % and 12 % of the time consumed by CR-ASPE 2 and ASPE 2, respectively. The results presented in Fig. 1a have shown that the cost grew linearly. Moreover, the time consumed by the encryption function of our simplest scheme was around 60 % higher than that of DPT's function in Fig. 1a. Nevertheless, the tendency of encryption functions of the ASPE schemes in comparison to ours was the same (around 3 % of difference) in Figs. 1a and b. Such result was expected because both use asymmetric scalar product encryption. In Fig. 1d, we have observed that the time consumed by the decryption function of the DPT scheme was higher than that of the first ASPE scheme and our first proposed scheme, because it has to invert a rotation matrix, multiply it by the encrypted point and subtract the result by a translation matrix. That means an extra operation when compared to the decryption functions of our schemes and ASPE schemes. Furthermore, we notice that even when the number of data dimensions is close to the number of dimensions used to encrypt data by schemes that use additional dimensions to encrypt data, the time consumed by the encryption function of CR-ASPE 1 and ASPE 1 was around 19 % and 20 % of the time consumed by CR-ASPE 2 and ASPE 2 respectively. Figure 1c shows that the cost had grown linearly. Moreover, the time consumed by the encryption function of our simplest scheme was around 8 % bigger than DPT's decryption function in Fig. 1c. Nevertheless, the tendency of decryption functions of the ASPE schemes in comparison to ours was the same (around 2 % of difference)

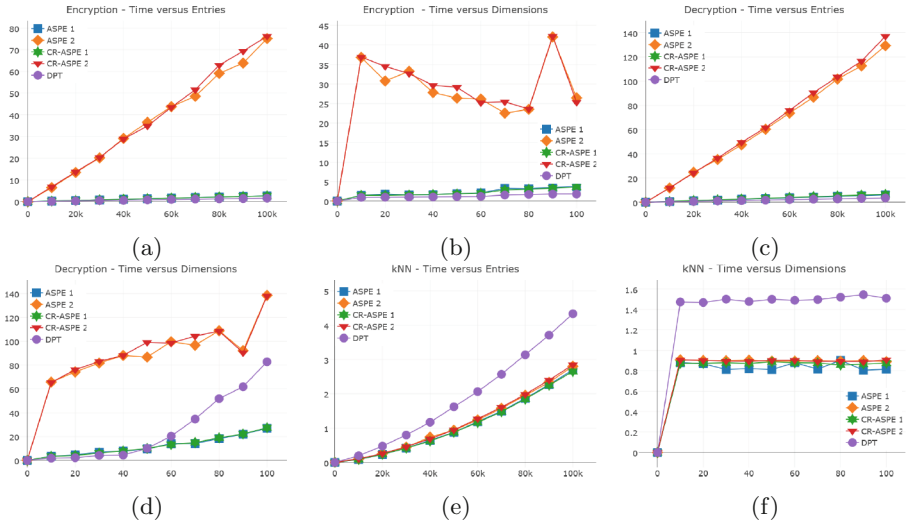
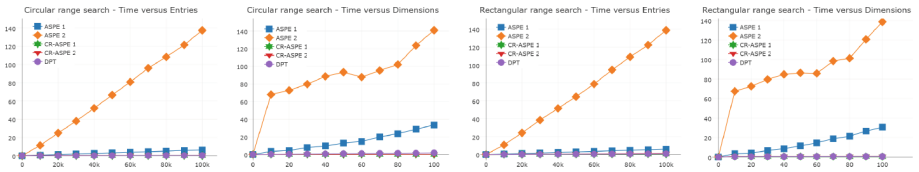


Fig. 1. Time consumed by encryption, decryption and kNN operations.

in Figs. 1c and d, as both use asymmetric scalar product encryption. Regarding the kNN operation in Figs. 1e and f, we noticed that the time consumed by all schemes did not change when dimensions varied. Furthermore, in Fig. 1f, the difference between the time consumed by kNN operation of CR-ASPE 1 and CR-ASPE 2 was around 3 %. We suppose it happened due to the optimized *NumPy* function to multiply matrices. The kNN function has a $\mathcal{O}(n \log n)$ complexity, which is detailed in Fig. 1e. Moreover, the time consumed by CR-ASPE 1's kNN function was around 55 % smaller than DPT's kNN function (Fig. 1e). It happened because the DPT scheme calculates the distance between the encrypted database point and the encrypted reference point, while CR-ASPE 1 executes one scalar product.

Due to ASPE schemes' limitation to execute circular search over encrypted data, ASPE 1 and ASPE 2 schemes must decrypt all data before running the circular range search. Thus, the time consumed by them is bigger than CR-ASPE and DPT schemes. The time consumed by circular range search of CR-ASPE 1 and CR-ASPE 2 is around 13 % and 27 % respectively of the time consumed by DPT in Fig. 2b, since CR-ASPE schemes execute a scalar product to verify the condition. Figure 2a has shown that the cost linearly grew. Moreover, the time consumed by circular search in CR-ASPE 1 and CR-ASPE 2 schemes was around 10 % and 1 % of the time consumed by circular search in ASPE 1 and ASPE 2 respectively. Figure 2d depicts the advantage of CR-ASPE schemes over ASPE schemes. As the ASPE schemes must decrypt all data to execute rectangular range searches while CR-ASPE and DPT schemes search over the encrypted data, the time consumed by them is evidently bigger. The time consumed by rectangular range search in CR-ASPE 1 and CR-ASPE 2 is around 70 % and

114 % respectively of the time consumed by DPT. Figure 2c indicates that the cost grew linearly. Moreover, the time consumed by rectangular search in CR-ASPE 1 and CR-ASPE 2 schemes was around 2 % and 0.5 % of that consumed in ASPE 1 and ASPE 2, respectively.



(a) (b) (c) (d)

Fig. 2. Time consumed by circular and rectangular range searches.

Our experiment results confirm that the encryption and decryption functions of CR-ASPE schemes have similar performance to ASPE schemes' functions, despite being more costly than encryption and decryption functions of DPT schemes. On the other hand, the circular range search, rectangular range search and kNN operation of CR-ASPE schemes were faster than the kNN function of DPT schemes.

Table 1. Execution times in seconds using real data ($n = 58,000$ and $d = 9$).

	ASPE 1	ASPE 2	CR-ASPE 1	CR-ASPE 2	DPT
ENC	1.71799	39.37519	1.61879	39.80141	1.08499
DEC	3.93868	75.82168	3.94564	77.95755	2.16277
KNN	1.05159	1.13221	1.10023	1.116808	1.81489
CRS	4.21793	77.40803	0.11913	0.24631	0.28225
RRS	3.91887	76.56150	0.46885	0.80529	0.54054

For the real dataset, we have obtained the results of time consumed in seconds by encryption (ENC), decryption (DEC), kNN, circular range search (CRS) and rectangular range search (RRS) functions for each scheme shown in Table 1. The results present the same behavior as the experiment over artificial datasets, evidencing the schemes do not lose performance when handling real data.

7 Conclusion

We proposed two encryption schemes for spatial data. CR-ASPE 2 is secure against attackers that have knowledge of a subset of plain spatial data, a set

of encrypted spatial data and the correspondence between them. While the encryption functions of CR-ASPE 1 scheme were not resistant to level-3 attacks, but approximately six times faster. Furthermore, in both CR-ASPE schemes, searches are executed over encrypted spatial data, an improvement on [1–4,6], reducing the functional gap between spatial databases and encrypted spatial databases.

We have compared our work with other encryption schemes and concluded that although our work supports more types of searches, its encryption functions have a similar performance to other ASPE schemes. Moreover, we presented proofs showing that each scheme correctly performs the searches.

The proposed schemes will be used to encrypt data structures as R-trees [7] or spatial indexes [15] in future works. Another work could implement these schemes in an EDBMS-like model [8] in order to support encrypted spatial data.

References

1. Wang, B., Li, M., Wang, H., Li, H.: Circular range search on encrypted spatial data. In: 2015 IEEE Conference on Communications and Network Security (CNS), pp. 182–190. IEEE (2015)
2. Wang, B., Hou, Y., Li, M., Wang, H., Li, H.: Maple: scalable multi-dimensional range search over encrypted cloud data with tree-based index. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, pp. 111–122. ACM (2014)
3. Wang, B., Hou, Y., Li, M., Wang, H., Li, H., Li, F.: Tree-based multi-dimensional range search on encrypted data with enhanced privacy. In: Tian, J., Jing, J., Srivatsa, M. (eds.) SecureComm 2014, Part I. LNICST, vol. 152, pp. 374–394. Springer, Heidelberg (2015)
4. Wong, W.K., Cheung, D.W.L., Kao, B., Mamoulis, N.: Secure kNN computation on encrypted databases. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, pp. 139–152. ACM (2009)
5. Oliveira, S.R., Zaiane, O.R.: Privacy preserving clustering by data transformation. J. Inf. Data Manag. **1**(1), 37 (2010)
6. Wang, P., Ravishankar, C.V.: Secure and efficient range queries on outsourced databases using rp-trees. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE), pp. 314–325. IEEE (2013)
7. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: vol. 14. ACM (1984)
8. Popa, R.A., Redfield, C., Zeldovich, N., Balakrishnan, H.: Cryptdb: protecting confidentiality with encrypted query processing. In: Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, pp. 85–100. ACM (2011)
9. Lopes, C.C., Times, V.C., Matwin, S., Ciferri, R.R., de Aguiar Ciferri, C.D.: Processing OLAP queries over an encrypted data warehouse stored in the cloud. In: Bellatreche, L., Mohania, M.K. (eds.) DaWaK 2014. LNCS, vol. 8646, pp. 195–207. Springer, Heidelberg (2014)
10. Liu, K., Giannella, C.M., Kargupta, H.: An attacker's view of distance preserving maps for privacy preserving data mining. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 297–308. Springer, Heidelberg (2006)

11. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
12. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (2009)
13. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Order preserving encryption for numeric data. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pp. 563–574. ACM (2004)
14. Lichman, M.: UCI machine learning repository (2013)
15. Lopes Siqueira, T.L., Ciferri, R.R., Times, V.C., de Aguiar Ciferri, C.D.: A spatial bitmap-based index for geographical data warehouses. In: Proceedings of the 2009 ACM Symposium on Applied Computing, pp. 1336–1342. ACM (2009)