Exploring Origin and Rotation Parameters While Using Hilbert Curves in Mobile Environments

Anand Paturi* and Subhasish Mazumdar†
Department of Computer Science & Engineering
New Mexico Institute of Mining and Technology
Socorro, NM, USA
Email: *anand@cs.nmt.edu, †mazumdar@cs.nmt.edu

***Abstract*—In mobile computing, nearest-neighbor queries are of the form "find me the nearest service of type S" or "find me k nearest services of type S." It is known that such queries, while convenient for the consumer, are associated with privacy threats. For addressing such privacy threats, one of the approaches suggested by researchers is spatial transformation via Hilbert curves. A Hilbert curve fills a 2-dimensional grid with a one-dimensional sequence that may be viewed as a curve. It is thus usable as a hash function that is order-preserving, in the sense that adjacent elements in the single dimension represent physically contiguous space in two dimensions. It provides an encryption of the two-dimensional space coordinates with the parameters involved in its construction serving as the *key*. The origin of the two-dimensional grid that is conceptually overlaid on the physical space and the choice of two canonical forms of the curve are crucial elements of the key. In this paper, we examine the ramifications of these parameters on the Quality of Service (QoS) provided to mobile users and suggest that these parameters be chosen based on acceptable QoS thresholds. By considering rotation and transposition, we enhance the space of keys, thus providing more options in the choice of those parameters.**

Keywords–Mobile Privacy; Spatial Transformation; Hilbert Curves; Location-Based Queries; Location-Based Service.

I. INTRODUCTION

Privacy is a challenge in mobile environments. Users are happy with location-based queries of the form "find me the nearest service of type S" or "find me k nearest services of type S." Unfortunately, their satisfaction is reduced by the underlying threat to their privacy. One of the two main approaches suggested by researchers for addressing this threat involves spatial encoding via Hilbert curves. The idea is that the *Location Based Server (LBS)* handling these user queries can be made unaware of the actual geographical coordinates of the users, the Points of Interest (POIs), and the categories of those POIs (e.g., restaurant, gas station), by being provided encoded spatial coordinates instead of actual geographic coordinates and encrypted identifiers instead of plaintext categories by another server, a trusted one (referred to as *Trusted Server (TS)*). The spatial encryption seems infeasible to break (invert) because of the large number of possible keys, i.e., ways the curve construction parameters can be chosen. In this paper, we explore the effect on the end-user of the choice of two of those parameters: origin and rotation, by defining QoS metrics around them; and propose an optimal strategy for their selection using quantitative thresholds.

A. Introducing the Hilbert Curve

A *Hilbert curve* is a space-filling transformation of bounded 2-dimensional space. Assume that a square space is divided into 2^{2N} cells arranged in a $2^N \times 2^N$ grid. A Hilbert curve H of order N is defined by a bijective function h that maps each (x, y) pair, where x and y are integers in $0 \cdot\cdot (2^N - 1)$, into an integer in $0 \cdot\cdot (2^{2N} - 1)$. Figure 1 shows an example with $N = 3$ (i.e., an 8x8 grid); values of h for each cell is shown within it; the sequence $0 \cdot\cdot 63$ defines a curve that fills the grid passing through each cell exactly once. By abuse of notation, we will use this function h to refer to the curve H as well. The bottom left cell corresponds to the origin of the X-Y coordinates. We also refer to the map as a 2-dimensional matrix: In Figure 1, $H[0, 0] = 0$; $H[1, 0] = 1$; and $H[0, 1] = 3$. Figure 2 shows a *transposed* curve with a similar logic but starting with cells $0 \cdot\cdot 3$ numbered anticlockwise. Since $H[i, j]$ in this transposed curve is equal to $H[j, i]$ in the normal curve, this is essentially a matrix transpose operation (the original paper [10] described it as *rotated*). Hence, we are calling it a transposed curve. Rotated representations of the curve are in Figures 3 (90°), 4 (180°), and 5 (270°).

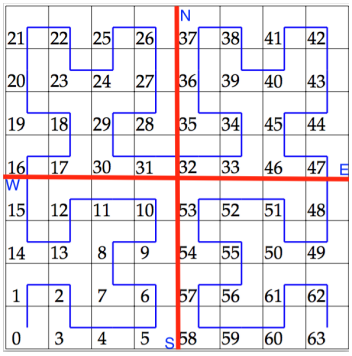

Figure 1. Normal Hilbert Curve for $N = 3$. The bottom row and leftmost column correspond to row 0 and column 0 of the corresponding matrices.

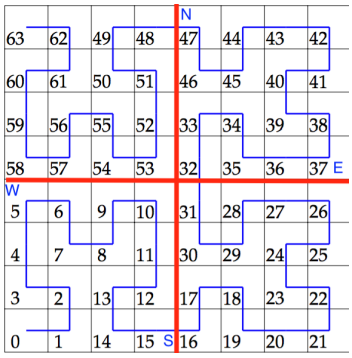

Figure 2. Transposed Hilbert Curve for $N = 3$.

This function h is contiguity-preserving (i.e., two cells mapped into i and $(i + 1)$ must represent 2-dimensional spaces that are contiguous). However, h may map contiguous

 ISBN: 978-1-61208-656-9

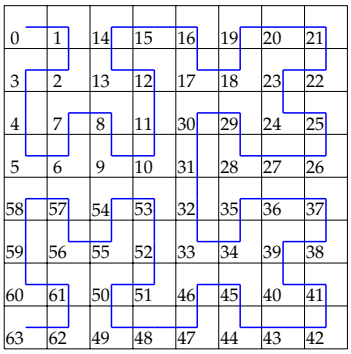

Figure 3. Hilbert Curve for N=3 rotated by 90 degrees clockwise.

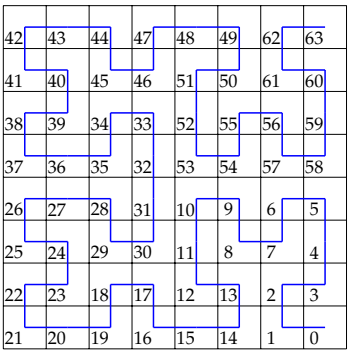

Figure 5. Hilbert Curve for N=3 rotated by 270 degrees clockwise.

cells in 2D-space into Hilbert numbers that are not close (e.g., numerically distant cells 5 and 58 in Figure 1 represent contiguous spaces).

B. Utilizing the Hilbert Curve for Location Privacy

It was first suggested in [10] that a Hilbert curve can be applied to location-based services. Such a Hilbert curve is generated by a Trusted Server (TS) by deciding the curve's parameters. They are: (1) the *order* of the curve N; (2) the (physical location of the) point of *origin* X_0, Y_0; (3) the *orientation* Θ (*normal* or *transposed* as in Figures 1 and 2 respectively); and (4) a *scaling factor* Γ that captures the number of meters that each unit cell represents (in both figures, Γ is the distance in meters covered by the grid in either the X- or Y-direction divided by 8). Using Γ and the origin, any geographic location (x_0, y_0) in the 2-D space (which could be represented by latitude and longitude), can be converted into a grid cell (x_0^*, y_0^*), where $x_0^*, y_0^* \in 0 \cdot \cdot (2^{2N} - 1)$. Thus, the transformation parameters (unknown to any adversary) are $[X_0, Y_0, \theta, N, \Gamma]$.

The parameter N is chosen in an effort to maintain a low average number of POIs per cell ($\frac{POI}{H}$ ratio); N is increased until that ratio is less than a given threshold. Some have suggested a hierarchy of curves with different N when the POI distribution varies markedly across the region [3] [15].

With knowledge of the map of the area and POIs, the TS first converts the geographical coordinates of each POI into a corresponding Hilbert cell number (in $0 \cdot \cdot (2^{2N} - 1)$); and next, using an encryption key e, it encrypts the description of the POI as well as its *category* or *domain* (e.g., restaurant) and *subcategory* (e.g., Vietnamese). Thus, the TS generates the

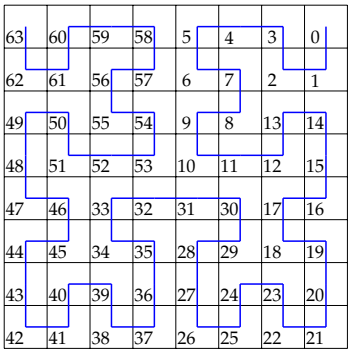

Figure 4. Hilbert Curve for N=3 rotated by 180 degrees clockwise.

curve H, creates a table of POIs of the kind shown in Table I, and sends it to the LBS, which uses it to answer location-based queries from users. Disruption of services from the TS and LBS can be avoided by using the well-known strategy of replication of servers and storage.

TABLE I. TABLE T SENT FROM THE TS TO LBS.

Cell	POI description	Category	Subcategory
43	05A4C3BB02F568489	9A4027D	4715
...	...	...	...
16	47923CC19B6C71AA0	7399BBA	02AA

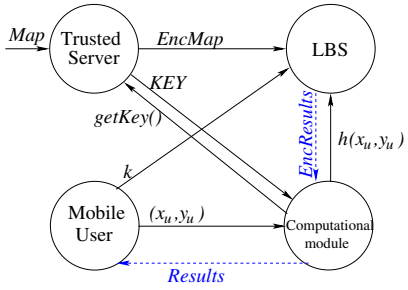

Figure 6. Interaction among actors to process a user query.

A mobile user queries the LBS using the Hilbert cell number corresponding to his/her location's two-dimensional coordinates; the LBS searches for the numerically closest Hilbert cell number that contains appropriate POIs and returns them to the user. A computational module obtains the transformation parameters from the TS by sending a *getKey()* request; transforms the user's geographic location (x_u, y_u) (e.g., latitude, longitude) into a grid cell (x_u^*, y_u^*), and then applies $h()$ to obtain a Hilbert cell number $h(x_u^*, y_u^*)$; and also decrypt the returned POI descriptions using the inverse of e. To perform the above mentioned steps, this module needs a *KEY* from the TS:

$$KEY = \{[X_0, Y_0, \theta, N, \Gamma], \ \ e^{-1}\}. \quad (1)$$

The practical implementation of the system is feasible in terms of performance because (a) the generation of the curve is done offline; (b) at the LBS, the POI retrieval is a range query that can exploit a B+ tree; [8] and (c) at the computational module, the computation of a Hilbert cell number can be done in time $O(n)$ or faster [2].

C. The Problem

We have found it useful to represent the overall scheme using Figure 6; by exploring the knowledge available to the four agents, we have argued elsewhere [16] that the computational module must reside in the user's device while remaining opaque to the user.

Now, the restriction on $\frac{POI}{H}$ mentioned above often dictates the determination of N and Γ, i.e., the placement of the $2^N \times 2^N$ grid on the physical map, making the number of choices of the origin and rotation crucial for a robust encryption. So, the question we ask is the following: assuming the grid is decided and the only variables are the origin and θ, what are the ramifications of the choices? Can the origin be moved to any cell in the grid? This leads to the question, are there cells in the grid that can remain unused, i.e., unmapped by the Hilbert curve? The answer is *no* when every grid cell is a potential POI location and cannot be ignored; the answer is *yes* when the fringe areas of a city consist of uninhabited open spaces or mountains or streams.

When the answer is *no*, we explore a wraparound scheme for re-indexing unmapped cells. Since this has the potential of degrading the nearness of returned POIs, we have used a QoS measure and found a correlation between the defined QoS measure and the number of contiguity violations.

When the answer is *yes*, the number of unmapped cells is a QoS measure; it can increase dramatically for many choices of origin. Here, we have outlined an opportunistic shift-plus-rotation scheme that reduces the degradation considerably. Thus, in both cases, origin shifts can be chosen based on a quantifiable acceptability criteria as well as considerations regarding the importance of those POIs either left unmapped or mapped violating contiguity.

The paper is structured as follows. In section 2, we present related work. In sections 3 and 4 we explore the *no* and *yes* answers respectively. Finally, we present our conclusions and future work in section 5.

II. Related Work

Abel et al. [1] compare Hilbert curves with four other spatial transformation orderings (row, row prime, Morton and Gray code) for geographical data processing. They conclude that Hilbert curves are poorer than Morton ordering in terms of oversearch for windowing operations but are much better in preserving contiguity. Dai et al. in [4], present an analytical study on the locality properties of Hilbert curves. Moon et al. [14] focus on the curve rotation concept while evaluating the clustering properties of Hilbert curves. They show that the clustering properties of Hilbert curves are better than z-order curves for range queries. A shift operation is implemented on the Hilbert curve in [13] and [5] to address the loss of proximity in the Hilbert space introduced by the gap in the *U-pattern*. Liao et al. in [13] introduce multiple shifted copies of the Hilbert curve that are indexed and stored in $(d+1)$ B-trees where d is the dimensionality of the space. Dai et al. [5], uses multiple B^+-trees for storage of the curves and evaluate range query performance in terms of number of pages accessed.

k-anonymity and spatial encoding have been the two primary approaches to address location privacy. In the k-anonymity-based approach, the goal is to send an obfuscated region instead of an exact user location ensuring that it contains at least $(k-1)$ other users. Such a region is constructed either by relying on a Trusted Third Party (TTP) with which all users share their exact locations or through a Peer-to-Peer (P2P) approach collaborating directly with $(k-1)$ other users nearby.

Khoshgozaran et al. [11] use a Private Information Retrieval (PIR) technique: the untrusted server does not have complete knowledge of the user's request and related response. But this approach is computationally very expensive.

Kalnis et al. [9] proposed a k-anonymity-based Hilbert Cloaking (HC) scheme that generates a cloaked (obfuscated) region using a Hilbert Curve. In [12], Lee et al. address the Hilbert curve's drawback while generating a cloaked region i.e., an extended cloaked region is generated (while using Hilbert curves) due to the disconnect in adjacent cells resulting from the *U-pattern*. Ghinita et al. [7] use Hilbert curves in a P2P setting. Damiani et al. in [6] propose Hilbert curve-based obfuscation. The map knowledge of the adversary and user's privacy preferences are taken into account while creating the obfuscated region over the target encoded space. Niu et al. [15] introduced the FGcloak algorithm that generates Hilbert curves with finer orders based on the query probability in the target region. Thereby, customizing the curve for high query probability regions. Cui et al. [3] propose a Hierarchical Hilbert Curve-based spatial k-anonymity that uses the average query density and constructs a hierarchical index to fill layers of the region with a curve.

III. Origin Shift via Wraparound

Here, we explore the question, can any of the 2^{2N} cells be chosen as the origin when no cell can be ignored? This would provide a very large key space — a desirable feature. We refer to this as shifting the origin.

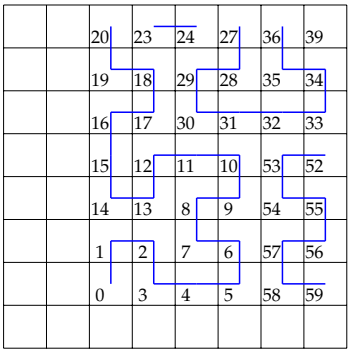

Figure 7. Hilbert curve for $N = 3$ with origin shifted to $(2, 1)$. Unmapped cells are shown blank.

Suppose the origin is shifted to, say, $(2, 1)$; the situation is as shown in Figure 7. The result is that a portion of the grid is empty, i.e., a portion of the geographical area is unmappable and hence inaccessible. Assuming that this is unacceptable, the easiest solution would be a wraparound of the grid that conceives the space as continuous by merging the left boundary with the right and the top with the bottom. In other words, if cell j maps to (x, y) where $x \geq 2^N$, replace x by $x \bmod 2^N$ (similarly y). The result is shown in Figure 8 for the origin at $(4, 4)$ and $(2, 1)$. For $(4, 4)$, only three pairs of cells ((15, 16), (47, 48) and (31, 32)) become discontinuous (red lines). These violate the contiguity assurance given by the Hilbert curve. Since the LBS is blind to the physical space, such

 ISBN: 978-1-61208-656-9

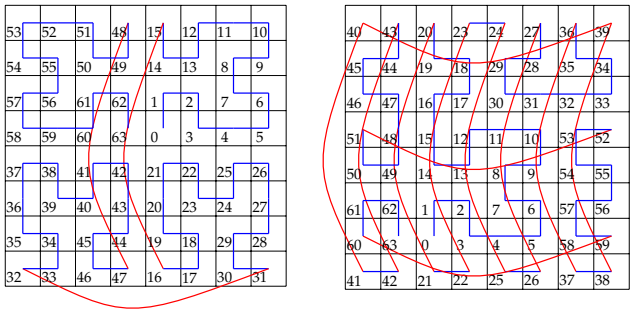

Figure 8. Hilbert curve for $N = 3$ with origin shifted to $(4, 4)$ and $(2, 1)$. Unmappable cells are re-mapped by a wraparound and connected.

discontinuities can result in poor search quality. For example, it may return a POI in cell 48 when the user is in cell 47, perhaps the other end of the city. However, the number of violations is small (only 3). On the other hand, for $(2, 1)$, there are many more red lines indicating multiple discontinuities. Multiple discontinuities indicate a higher probability of poor QoS for the mobile user.

To quantify the effect of the contiguity violations introduced by the origin shift on the mobile user, we use a parameter we call Distance Penalty (D.P.) which was introduced in [10] as Displacement Measure. Assume the user is at location l_u and desires k nearest POIs; D.P. is defined as follows:

$$\text{D.P.} = \sum_{i=1}^{k} \frac{\mid l_u - o_i \mid}{k} - \sum_{i=1}^{k} \frac{\mid l_u - o_i^{'} \mid}{k} \tag{2}$$

where $\mid l_u - o_i \mid$ is the Euclidean distance between the user location l_u and the i^{th} closest POI returned when the LBS searches in Hilbert space, i.e., is kept blind; and $\mid l_u - o_i^{'} \mid$ is the Euclidean distance between l_u and the i^{th} closest POI that would be returned if the LBS was searching in Cartesian space, i.e., was *not* kept blind.

Even when the curve origin is at $(0, 0)$, i.e., is not shifted, and $k = 2$, a certain D.P. is introduced since the Hilbert curve does not exactly preserve the distance between the Cartesian and Hilbert space. We take this to be an acceptable amount based on the POI distribution, user location and k values. We want to measure how much worse it gets as the origin is shifted. We took $N = 3$ and conceived a situation where there was exactly one POI in each Hilbert cell, and the user posed a query from the center of each cell in turn (a total of 64 queries), each time asking for $k = 2$ POIs. (Since there is one POI in each cell, $k = 1$ would be trivial.) Setting up our origin at $(0, 0)$, we computed the D.P. for each query and averaged over all queries. This was δ_0, our baseline D.P. Then we repeated the experiment for k = 3 and 4. Next, we repeated the above for each cell as the origin. After normalizing all D.P. values by dividing by δ_0, we plotted the three D.P. results (for $k = 2, 3, 4$) in Figure 9.

Next, for each origin, we counted the number of pairs of cells that violated the contiguity guarantee (e.g., 3 for the origin at $(4, 4)$); these counts form the last bar in Figure 9. Visually discernible in that figure is our finding that the count of contiguity violations correlates with the D.P. The figure shows that the distance penalty is maximum when the origin is shifted to (3,1) and (5,1), least when the origin shift is (0,0) and (4,0) and varies for different origin shifts. Thus, the D.P. is a QoS metric that the TS can use to control the degradation by rejecting origin shifts for which it is beyond an acceptable value. Moreover, since it is correlated with the number of contiguity violations which is intrinsic to the Hilbert curve, the TS can reject the origin shifts for which those actual violations are unacceptable because they involve high-interest POIs.

IV. Without Wraparound

Next, we consider the situation where fringe areas can remain unmapped. The question is, how many cells become unmapped as we move the origin? It is possible to show from the rectangular geometry that when the origin is at (i, j), the number of unmapped cells is given by is given by

$$U = (i + j) * M - ij, \text{ where } M = 2^N. \tag{3}$$

Clearly, this naive method is not feasible when the origin moves to any other quadrant. For example, when the origin is at $(5, 5)$, only 9 cells are mapped and the unmapped cells occupy the central core of the grid not the fringes. Obviously, it would be more practical to rotate the grid. Since POIs inside the unmapped cells become inaccessible as a consequence, their number relates to the QoS.

A. Combining Shift with Rotation

The original paper [10] had introduced two canonical forms of the curve: the normal and transposed (they called the latter curve *rotated*). Moon et al. [14] introduced two more canonical forms of the curve. Their idea can be explained by flipping the normal curve (Figure 1), which they called 1^+, about a vertical (horizontal) line running through its middle, i.e., the red line NS (WE) in that figure. After a flip about the horizontal (WE), the resulting shape is changed but after a flip about the vertical (NS), it is not. They called the new shape 1^-. Similarly, flipping the transposed curve 2^+ around the vertical line yielded a new curve they called 2^-. We use their idea but we make use of the unchanged shapes as well because they do yield a different hash function or numbering scheme and that is what we care about. To make this process precise, we propose two orthogonal transformation operators:

1) Transposition, creating H^T from H;
2) Rotation by 90°; creating H^{90} from H; repeated composition of this provides H^{180} and H^{360} respectively.

The results of rotating the normal curve by 90, 180 and 270 degrees are shown in Figures 3 (H^{90}), 4 (H^{180}), and 5 (H^{270}) above. Similar rotations for the transposed curve (H^T) are not shown for shortage of space.

The following can be proved (though we omit the proofs for shortage of space). For a Hilbert curve H $(= H^0)$, any integer $n \geq 0$, $\theta \in \{0°, 90°, 180°, 270°\}$ and where raising to a power n means repeated composition:

$$((H^\theta)^T)^T = H^\theta \tag{4}$$

$$(H^\theta)^n = H^{n\theta \mod 360} \tag{5}$$

$$(H^T)^\theta = (H^\theta)^T \tag{6}$$

$$(H^T)^\theta \sim H^{\theta+90} \tag{7}$$

where $H \sim H'$ means that H, H' are related (by shape symmetry) through the following:

$$H'[i, j] = (2^N - 1) - H[i, j] \tag{8}$$

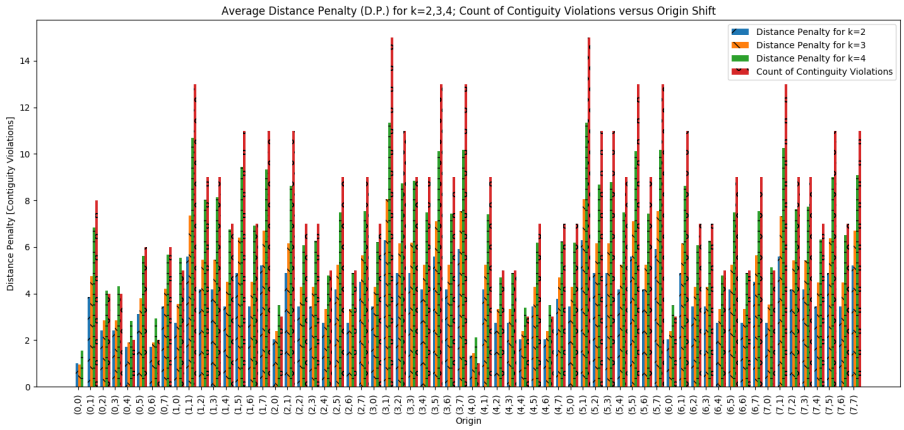

Figure 9. Contiguity Violations and (Normalized) Distance Penalty versus Origin Shift.

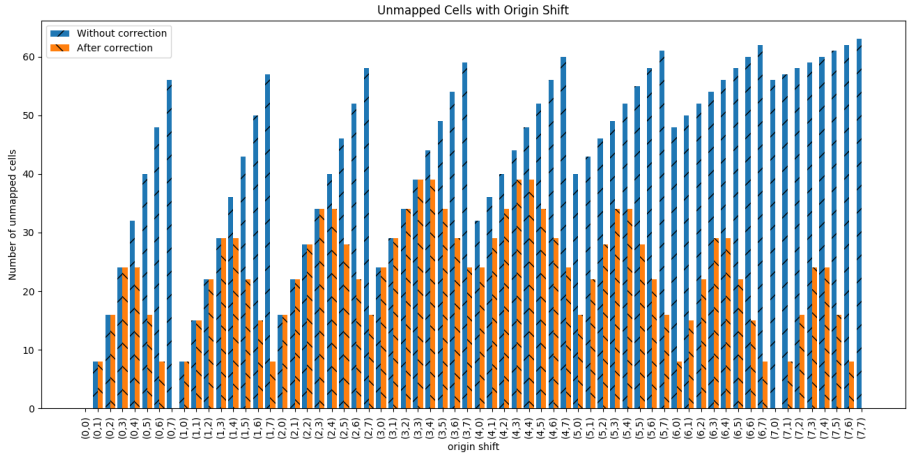

Figure 10. Number of unmapped cells with naive shift and opportunistic shift+rotation.

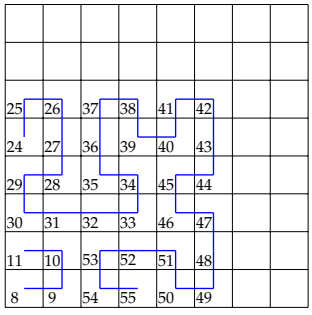

Figure 11. Instead of shifting the origin to $(5,5)$ and leaving 86% of the cells unmapped, an opportunistic shift strategy allows better coverage. Notice that this effectively shifts the origin to $(-2,-2)$.

Equation 6 allows us to replace both $(H^T)^\theta$ and $(H^\theta)^T$ by $H^{T,\theta}$. In equation 7, the $\sim$ is especially interesting. It shows that pairs of these transformed curves are similar (capturing the similarity in shape) though not identical. Moreover, they involve a swap of origins: the equation shows that 0 is swapped with 63. Such curves, which are similar but not identical are very useful to us because they provide another hash function, i.e., another key combination for encryption. Thus, we obtain 8 canonical forms:

$$\{H, H^T\} \cup \{H^\theta, H^{\theta,T} \mid \theta \in \{90°, 180°, 270°\}\}. \quad (9)$$

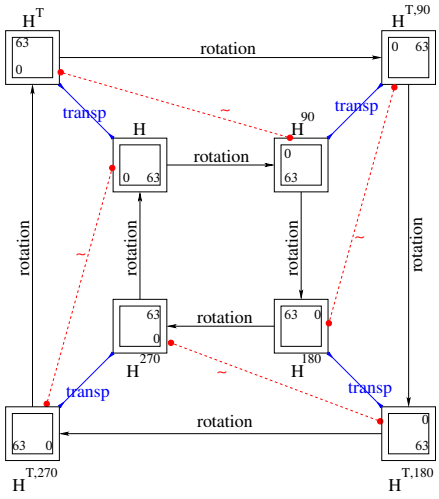

Figure 12. Transformations and Relations among 8 canonical forms.

Figure 12 indicates the relationships (rotation, transposition, and $\sim$) among all eight. Within each square is a thumbnail of the 8×8 matrix showing only the positions of the two end cells: 0 and 63. It is important to note that these eight canonical forms incorporate origin shift: each of the four corner cells is an origin in exactly two of the eight.

B. Opportunistic Shifting with rotation

Now we wish to apply the above operators for arbitrary origin shifts in order to make the shift useful in leaving only the fringe unmapped and in reducing the number of unmapped cells. We propose an opportunistic origin shift. Suppose the origin is to be shifted to (i, j). Depending on the quadrant (i, j) occupies, we choose any one of the 8 canonical curves, find the cell in its outermost corner that is nearest to (i, j), and drag (translate the grid) it to (i, j). For example (Figure 11), when the origin is shifted to $(5, 5)$, instead of leaving out 86% of the grid unmapped, let us choose the normal curve H as the canonical curve; since cell 42 is the outermost corner of H nearest to $(5, 5)$, we translate H dragging cell 42 to $(5, 5)$. Note that the origin is now effectively $(-2, -2)$. (If we had chosen H^{180} or $H^{T,180}$ instead of H, the origin would have been at $(5, 5)$.) This along with the choice of any of the 8 forms demonstrates that we have expanded on the space of possible keys. The key now includes boolean T and W, transposition and wraparound flags:

$$KEY = \{[X_0, Y_0, \theta, T, W, N, \Gamma], \ \ e^{-1}\}. \qquad (10)$$

Figure 10 shows the number of unmapped cells with and without this scheme (labeled *correction* in the figure). It shows that this scheme is very effective at reducing the number of unmapped cells. The TS can reject certain origin shifts in either of two ways: if the number of unmapped cells exceeds an acceptable threshold, or if those unmapped cells contain high-value POIs. The opportunistic scheme clearly gives a much better set of options in this choice.

V. Conclusion and Future Work

In this paper, we explored two parameters underlying the Hilbert curve: the origin and 'rotation'. The ability to choose among many values of these two parameters is crucial because we are often constrained in our choice of the others. We considered two scenarios: in the first, it is *not* acceptable to leave some cells unused, i.e., every cell needs to be used; and in the second, it *is*, e.g., when fringe areas of a city do not have useful POIs.

For the first scenario, we introduced a wraparound strategy for varying the origin parameter (X_0, Y_0). This works quite well for some choices of origin but not for others. We quantified this QoS issue using a D.P. metric and found that it correlates with the number of contiguity violations resulting from the origin shift. For the second scenario, the number of unmapped cells is our QoS metric. We introduced transformation primitives that clarified the process of combining rotation with origin shift, enlarged the number of canonical curves from four to eight, and enabled our opportunistic shift with rotation which greatly improved the number of unmapped cells.

In both cases, we enable the TS to choose the origin shifts not only by thresholding the QoS metric, but also by checking if high-interest POIs are among the cells suffering contiguity violations or being left unmapped. Further, by increasing the number of canonical forms and allowing the origin to move outside the grid, we have expanded the space of keys, thereby giving more options in choosing feasible origin shifts.

For future work, we will test the robustness of our approach by taking subsets of maps of actual cities and simulating attacks. Other avenues of inquiry include a user in motion and a user specifying a cloaking region.

Acknowledgment

We thank our anonymous reviewers for their careful reading and constructive comments.

References

[1] D. J. ABEL and D. M. Mark. A comparative analysis of some two-dimensional orderings. *International Journal of Geographical Information Systems*, 4(1):21–31, 1990.

[2] N. Chen, N. Wang, and B. Shi. A new algorithm for encoding and decoding the hilbert order. *Software - Practice and Experience*, 37:897–908, 2007.

[3] N. Cui, X. Yang, and B. Wang. A novel spatial cloaking scheme using hierarchical hilbert curve for location-based services. In *Proc. Part II of the 17th International Conference, WAIM 2016, 2016*, pages 15–27. 2016.

[4] H. K. Dai and H. C. Su. On p-norm based locality measures of space-filling curves. In R. Fleischer and G. Trippen, editors, *Algorithms and Computation*, pages 364–376. Springer Berlin Heidelberg, 2005.

[5] J. Dai. Efficient range query using multiple hilbert curves. In C. Turcu, editor, *Current Trends and Challenges in RFID*, chapter 19. InTech, 2011.

[6] M. L. Damiani, E. Bertino, and C. Silvestri. Protecting location privacy against spatial inferences: The probe approach. In *Proc. 2Nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, pages 32–41, 2009.

[7] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: Anonymous location-based queries in distributed mobile systems. In *Proc. 16th International Conference on World Wide Web*, pages 371–380, 2007.

[8] Y. Jin, J. Dai, and C.-T. Lu. Efficient range query using multiple hilbert curves. In C. Turcu, editor, *Current Trends and Challenges in RFID*, chapter 19. InTech, 2011.

[9] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Trans. Knowledge & Data Engineering*, 19(12):1719–1733, Dec 2007.

[10] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *Transactions on Large-Scale Data- and Knowledge-Centered Systems*, pages 239–257, 2007.

[11] A. Khoshgozaran and C. Shahabi. Private information retrieval techniques for enabling location privacy in location-based services. In *Privacy in Location-Based Applications: Research Issues and Emerging Trends*, pages 59–83. Springer Berlin Heidelberg, 2009.

[12] H.-J. Lee, S.-T. Hong, M. Yoon, J.-H. Um, and J.-W. Chang. A new cloaking algorithm using hilbert curves for privacy protection. In *Proc. 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 42–46, 2010.

[13] S. Liao, M. A. Lopez, and S. T. Leutenegger. High dimensional similarity search with space filling curves. In *Proceedings 17th International Conference on Data Engineering*, pages 615–622, 2001.

[14] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE Trans. Knowledge & Data Engineering*, 13(1):124–141, Jan 2001.

[15] B. Niu, Q. Li, X. Zhu, and H. Li. A fine-grained spatial cloaking scheme for privacy-aware users in location-based services. In *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8, Aug 2014.

[16] A. Paturi and S. Mazumdar. Can spatial transformation-based privacy preservation compromise location privacy? In *Transactions on Large-Scale Data- and Knowledge-Centered Systems*, 2018.

 ISBN: 978-1-61208-656-9