

A Geometry-Grounded Data Perimeter in Azure

Christophe Parisel*

Abstract

While *Data Perimeter* is ubiquitous in cybersecurity speak, it rarely defines how boundary points are arranged. In this paper we show how Azure’s blast radius ultrametric provides the distance, and how solving the Traveling Salesman Problem in this ultrametric space provides the ordering, yielding a true geometric contour: an actionable perimeter measure for Service Principals (SPNs) data plane operations de-escalation.

1 Introduction

In cybersecurity and data privacy management, the notion of a Data Perimeter is rarely tied to a concrete geometric construction, because perimeter implies a *distance* function and a *circular ordering* of the data, both of which are usually either missing or don’t make sense for abstract datasets. To define a meaningful Data Perimeter in Azure, we leverage the *blast radius* ultrametric[7] to propose a geometry-grounded perimeter which is both efficiently computable and minimal.

We show how our Data Perimeter acts as a secondary sort key after blast radius stratification, to provide fine-grained risk prioritization of SPNs.

2 Background

2.1 Related Work

2.1.1 Attack Surface Model

Manadhata and Wing’s attack surface[1] pioneered the formalization of ”entry/exit points” and ”channels” as quantitative risk contributors. While their model provides a useful abstraction for OS-level systems, it is less suited to the structured permission sets typical of cloud platforms IAM like Azure RBAC.

In contrast, our approach treats the action set geometry directly, producing shape-aware summaries rather than quantitative counts.

2.1.2 Ultrametrics and Hierarchical Models

Ultrametric spaces arise naturally in settings when items are grouped by resource hierarchy. Such metrics obey the strong triangle inequality and induce a natural tree structure. Previous works on hierarchical clustering have used ultrametrics to model efficient machine learning algorithms[2], phylogenetic trees[3], and linguistic evolution[4].

Our use of ultrametrics follows this tradition, but we leverage their properties for efficient tour construction (not just visualization).

*Email: ch.parisel@gmail.com

2.1.3 Seriation and metric TSP

Seriation, the process of ordering elements to reveal latent structure, has applications in archaeology, heatmap visualization, and gene expression analysis. In computational geometry, finding an optimal ordering that minimizes some distance (e.g., via the Traveling Salesman Problem) has been proposed as a generic sorting principle[5].

We contribute a novel application of this idea to the Cloud context, in Azure RBAC, showing that TSP solutions over ultrametrics are not only efficient to compute but also yield meaningful secondary metrics.

2.1.4 Perimeter-Like Measures in Security

Few existing metrics capture the notion of a perimeter over privileges. Graph-based approaches like privilege escalation trees (e.g., BloodHound[6]) focus on path enumeration and control flow, not data-action layout. Similarly, role-mining algorithms often compress permissions but ignore spatial structure.

Our notion of a Data Perimeter is grounded in metric space geometry, enabling fine-grained differentiation within otherwise identical blast-radius bands.

2.2 SPNs and Data Actions in Azure

Service Principals in Azure hold sets of fine-grained data actions (e.g., ReadBlob, WriteSecret) scoped to individual resources or to resource containers (resource groups, subscriptions, management groups). Ultimately, this design means that Azure Data Perimeter is not a global, Tenant-wide metric: it depends on each SPN.

SPN permissions can be assigned either directly, or via groups membership. Excessive read permission may lead to a data leakage, whereas excessive write permissions may lead to data forgery. The concentration of read and write permissions across a large set of resources or resource containers under the same SPN makes risks assessment related to lateral motion particularly challenging.

2.3 Blast Radius Ultrametric

The Blast radius was introduced to address the challenge of measuring lateral motion, data leakage and data forgery of individual SPNs in Azure. It is fully implemented in Azure Silhouette[8].

Define an ultrametric distance $d(a, b) = \frac{\text{impact}(a, b)}{2^{2D(a, b)+1}}$, where D is the Least Common Ancestor of a and b in Azure’s native clustering hierarchy, and impact is a parameter depending on data actions[7].

As with all ultrametric distances, it satisfies the strong triangle inequality:

$$\forall x, y, z \in X, \quad d(x, z) \leq \max\{d(x, y), d(y, z)\}$$

The *blast radius* of an SPN is the diameter of $d()$: $\max_{i, j} d(x_i, x_j)$. It ranges from 0.0 (no permissions) to 1.0 (Tenant-wide permissions).

3 Problem Statement

The Blast Radius assigned to each SPN is a sort key to quickly identify the most risky SPNs. Due to the nature of ultrametry and the use of the *impact* parameter, SPNs are grouped into up to 22 bands of identical Blast radius values:

- 2 bands for Tenant-wide radius
- 12 bands for Management groups
- 2 bands for subscriptions
- 2 bands for resource groups
- 2 bands for resources
- 2 bands for resource parts

The Blast radius stratifies SPNs into coarse bands, so many SPNs tie on the same bands, leaving no intra-band ranking.

We seek a secondary sorting key that leverages the ultrametric structure to distinguish SPNs further.

4 Data Perimeter

4.1 Formulation

Given data actions x_1, \dots, x_n with ultrametric d , the Data Perimeter is defined as the minimal circular path covering all actions under ultrametric distance.

$$P = \min_{\pi \in S_n} \sum_{i=1}^n d(x_{\pi(i)}, x_{\pi(i+1)}), \quad \text{with } x_{\pi(n+1)} = x_{\pi(1)}$$

Where π is a permutation of the data action indices.

This amounts to solving a metric Traveling Salesman Problem (metric TSP). A major benefit of using the ultrametric TSP formulation for computing the Data Perimeter is that the problem, typically NP-hard in general, becomes efficiently solvable. In ultrametric spaces, the hierarchical structure imposed by the strong triangle inequality enables polynomial-time solutions to the metric TSP. This stems from the fact that shortest tours respect the natural tree structure of the space. That is, the tour effectively follows a traversal of the leaves of the ultrametric tree with minimal backtracking.

As will be shown in the ultrametric TSP pseudo-code below, the Data Perimeter is not only geometrically grounded but also computationally tractable in our setting.

4.1.1 Ultrametric TSP pseudo-code

Input: A set of data actions $A = \{a_1, \dots, a_n\}$ with distance function $d(a, b)$

Output: Perimeter length L

1. Let $P \leftarrow []$ (Tour list)
2. Pick $a_{\text{start}} \in A$ at random
3. Append a_{start} to P
4. Let $U \leftarrow A \setminus \{a_{\text{start}}\}$ (Unvisited actions)
5. **while** $U \neq \emptyset$ **do**
 - (a) Let $a_{\text{current}} \leftarrow$ last element of P
 - (b) Find $a_{\text{next}} \in U$ such that $d(a_{\text{current}}, a_{\text{next}})$ is minimized
 - (c) Append a_{next} to P
 - (d) Remove a_{next} from U
6. Append a_{start} to P to complete the cycle
7. Initialize $L \leftarrow 0$
8. **for** $i = 1$ to $|P| - 1$ **do**
 - (a) $L \leftarrow L + d(P_i, P_{i+1})$
9. **return** L

4.1.2 Notes

- Ultrametric TSP admits multiple optimal tours, so the circular ordering required for constructing a perimeter is not unique, but all tour lengths are equal. They form an equivalence class.
- Because of the wraparound, a tour length doesn't depend on its starting point.

4.1.3 Known Limitations

Silhouette implements Blast radii calculation over the native Azure Tenant hierarchy, as well as on a family of alternate hierarchies. This option lets risk officers tailor blast radii to the local needs of their organization: they can for instance use a special alternate hierarchy to represent the state of a reorganization, a merger or acquisition, a carve-out, ...

When hierarchical families are defined, Silhouette computes the Blast radius as the pointwise infimum in all hierarchies to yield a tighter, smaller Blast Radius than the native one. In doing so, it must be noted that the pointwise infimum is not ultrametric, hence the data Perimeter should be treated with care in this situation. Falling back to the native tenant hierarchy is highly recommended.

Here is a small counter-example showing the lack of ultrametry:

Let $X = \{x, y, z\}$, and define two ultrametries d_1 and d_2 on X as follows:

$$\begin{aligned} d_1(x, y) &= 2, & d_1(y, z) &= 1, & d_1(x, z) &= 2, \\ d_2(x, y) &= 1, & d_2(y, z) &= 2, & d_2(x, z) &= 2. \end{aligned}$$

Each d_i satisfies the ultrametric inequality:

$$d_i(a, c) \leq \max\{d_i(a, b), d_i(b, c)\} \quad \text{for all } a, b, c \in X.$$

Now define d_3 as the pointwise minimum:

$$d_3(a, b) := \min\{d_1(a, b), d_2(a, b)\}.$$

This gives:

$$\begin{aligned} d_3(x, y) &= \min\{2, 1\} = 1, \\ d_3(y, z) &= \min\{1, 2\} = 1, \\ d_3(x, z) &= \min\{2, 2\} = 2. \end{aligned}$$

However, the strong triangle inequality fails:

$$d_3(x, z) = 2 \not\leq \max\{d_3(x, y), d_3(y, z)\} = \max\{1, 1\} = 1.$$

5 Mean distance (μ_{mean})

The average distance between any two data actions is the metric that springs to mind for complementing the Blast radius. Unfortunately, if it captures global dispersion, it ignores local structure and chaining. In this section, we explain why this doesn't make it a good secondary key candidate for sorting SPNs.

5.1 Comparison with Data Perimeter

To understand the benefits of the Data Perimeter over the mean, we sample 1285 SPNs that we sort by Blast band:

1. For each SPN we compute a *spread_ratio* defined as $\frac{P}{n \cdot \mu_{\text{mean}}}$, where n is the data actions count held by this SPN. The perimeter P is normalized by the action count n to maintain dimensional consistency with μ_{mean} .
2. We group SPNs by band
3. We calculate the average spread ratio in each band.

Table 1 summarizes our findings. Blast radii are first shuffled, then anonymized into Band IDs.

5.1.1 Discussion

The in-band `spread_ratio` values (Table 1) fall into two distinct regimes:

Tightly clustered SPNs (blast radius $\lesssim 10^{-4}$) In bands VII-XIV, and other small radius groups, the normalized Data Perimeter and the mean coincide (`spread_ratio` ≈ 1.0). Here, each SPN's permission set is nearly point-like, so any reasonable ordering of actions (whether implicit in the mean or made explicit via TSP) produces the same minimal contour length.

Band ID	SPNs count	Spread Ratio	Regime
I	33	1.00000	Tight
II	69	0.998993	Tight
III	4	0.820245	Dispersed
IV	2	1.00000	Tight
V	110	0.99186	Tight
VI	8	0.978792	Dispersed
VII	228	1.00000	Tight
VIII	19	1.00000	Tight
IX	3	1.00000	Tight
X	505	0.997459	Tight
XI	6	1.00000	Tight
XII	70	0.987850	Tight
XIII	2	1.00000	Tight
XIV	16	1.00000	Tight
XV	28	0.581303	Dispersed
XVI	66	0.806384	Dispersed
XVII	2	1.00000	Tight
XVIII	75	0.950056	Dispersed
XIX	5	0.956395	Dispersed
XX	2	1.00000	Dispersed
XXI	32	0.888986	Dispersed

Table 1: In-band spread ratios (Data Perimeter vs. mean) per shuffled and anonymized blast radius band.

Dispersed SPNs (blast radius $\gtrsim 10^{-4}$) As blast radii grow, we see a systematic divergence (spread_ratio < 1). For example:

- Band XV: spread_ratio 0.58
- Band XVI: spread_ratio 0.81
- Band XVIII: spread_ratio 0.95

This reflects the fundamental difference between our two metrics.

- The *mean* sums large, disjoint jumps and thus overestimates overall spread.
- The *TSP-based perimeter* constructs a continuous tour, chaining cluster centers and “short-cutting” around outliers. By exploiting the strong triangle inequality, it compresses lengthy detours into shorter, hierarchical paths.

Implications for SPN prioritization When many SPNs share the same blast radius, the Data Perimeter provides a fine-grained, geometry-aware tiebreaker. It surfaces principals whose permission sets exhibit branching or elongated ultrametric structures whereas the mean alone would treat them identically to more compact SPNs.

This richer signal enables security teams to target reviews and policy adjustments not just based on “how far” an SPN can reach, but on “how” its permissions are organized. It confirms that TSP-derived perimeter is not just another average: it captures the global connectivity structure of data actions.

6 Conclusion

We introduced the Data Perimeter as a geometry-grounded refinement of the blast radius, capturing the spatial layout and redundancy of data actions assigned to service principals (SPNs). Unlike scalar averages such as the ultrametric mean, the perimeter is sensitive to seriation, clustering, and the underlying topology of permission sets.

By leveraging ultrametric distance and a tractable TSP solver, we demonstrated that the Data Perimeter can be computed efficiently and yields a non-arbitrary circular ordering of actions. This ordering is meaningful, not merely heuristic: it reflects a class of equivalent perimeters that respect the hierarchical structure of permissions, compressing global dispersion into a single tour.

Our small comparative analysis across blast radius bands revealed that while ultrametric mean and perimeter agree in tightly scoped SPNs, they diverge significantly as permission sets grow complex. The Data Perimeter remains stable and informative even in the presence of branching or fragmented access patterns, highlighting its suitability as a secondary sorting key for SPNs sharing similar blast magnitude.

This work bridges abstract privilege metrics with actionable geometric intuition. It enables analysts to go beyond “how much power” toward “how this power is organized,” providing a new lens to prioritize, visualize, and reduce risk in large-scale cloud environments.

References

- [1] Pratyusa Manadhata et al, “A Formal Modale for a System’s Attack Surface”, <https://www.cs.cmu.edu/~wing/publications/ManadhataKaynarWing07.pdf>
- [2] Fionn Murtagh and Pedro Contreras, “Clustering through High Dimensional Data Scaling: Applications and Implementations”, <https://d-nb.info/118565108X/34>
- [3] Amaury Lambert, “Random Ultrametric Trees and Applications”, <https://www.esaim-proc.org/articles/proc/pdf/2017/05/proc186003.pdf>
- [4] Marc E.Canby et al, “Addressing Polymorphism in Linguistic Phylogenetics”, <https://onlinelibrary.wiley.com/doi/10.1111/1467-968X.12289>
- [5] Aliyev DA, Zirbel CL. Seriation Using Tree-penalized Path Length. Eur J Oper Res. 2023 Mar 1;305(2):617-629. doi: 10.1016/j.ejor.2022.06.026. Epub 2022 Jun 17. PMID: 36385922; PMCID: PMC9642984.
- [6] BloodHound, Attack Paths For All, <https://bloodhound.specterops.io/home>
- [7] Christophe Parisel, “Scoring Azure Permissions with Metric Spaces”, <https://arxiv.org/abs/2504.13747>
- [8] Christophe Parisel, “Silhouette, a SPN sorter and minimizer”, <https://github.com/labyrinthinesecurity/silhouette>

Appendix: Minimal Data Perimeter

In this work, we have argued that the Data Perimeter in Azure security models behaves not just as a metaphor, but as a measurable geometric quantity. To further give substance to this geometric view, we identify the shapes corresponding to the smallest Data Perimeter.

Ultracycle and Minimal Perimeter

Given an ultrametric space (X, d) with n points, an *ultracycle* is a configuration where all pairwise distances are equal: $d(x_i, x_j) = \delta$ for all distinct $i, j \in \{1, 2, \dots, n\}$ and some constant $\delta > 0$.

In any ultrametric space, ultracycles uniquely minimize the TSP tour length among all configurations with the same number of points.

Proof

Let (X, d) be an ultrametric space and consider n points $\{x_1, x_2, \dots, x_n\} \subset X$.

Case 1: Ultracycle Configuration

Suppose all pairwise distances are equal: $d(x_i, x_j) = \delta$ for all $i \neq j$.

For any TSP tour π visiting all points exactly once, the tour length is:

$$P_{\text{ultra}} = \sum_{i=1}^n d(x_{\pi(i)}, x_{\pi(i+1)}) = \sum_{i=1}^n \delta = n\delta \quad (1)$$

where $x_{\pi(n+1)} := x_{\pi(1)}$.

Since all edges have the same weight δ , every possible tour has identical length $n\delta$.

Case 2: Non-Ultracycle Configuration

Now suppose the configuration is not an ultracycle. Then there exist distinct indices i, j, k, ℓ such that:

$$d(x_i, x_j) \neq d(x_k, x_\ell) \quad (2)$$

Let $d_{\min} = \min_{i \neq j} d(x_i, x_j)$ and $d_{\max} = \max_{i \neq j} d(x_i, x_j)$.

By assumption, $d_{\min} < d_{\max}$.

Lemma: in any ultrametric space, if $d(a, b) < d(a, c)$, then $d(b, c) = d(a, c)$.

Proof of Lemma: By the strong triangle inequality: $d(a, c) \leq \max\{d(a, b), d(b, c)\}$.

So we must have $d(a, c) \leq d(b, c)$.

Similarly, $d(b, c) \leq \max\{d(b, a), d(a, c)\}$. But $d(a, b) < d(a, c)$, so $d(b, c) \leq d(a, c)$.

Therefore, $d(b, c) = d(a, c)$. This proves the Lemma.

Now, let $S_{\min} = \{(i, j) : d(x_i, x_j) = d_{\min}\}$ be the set of point pairs with minimum distance.

Claim: Any TSP tour must include at least one edge with distance $> d_{\min}$.

Proof of Claim: Suppose, for contradiction, that there exists a tour using only edges of distance d_{\min} . This would mean we can connect all n points using only edges from S_{\min} , forming a Hamiltonian cycle.

However, by the ultrametric property (Lemma), if $d(x_i, x_j) = d_{\min}$ and $d(x_j, x_k) = d_{\min}$, then either:

- $d(x_i, x_k) = d_{\min}$ (all three points are equidistant), or

- $d(x_i, x_k) > d_{\min}$ (the points form a non-equidistant triple)

If all points were connected by minimum-distance edges, then by transitivity of the equidistance relation in ultrametric spaces, all pairwise distances would equal d_{\min} , contradicting our assumption that the configuration is not an ultracycle.

Therefore, any Hamiltonian cycle must include at least one edge with distance $> d_{\min}$.

Completing the Proof:

Let $P_{\text{non-ultra}}$ be the length of an optimal TSP tour in the non-ultracycle configuration.

Since the tour must include at least one edge of length $> d_{\min}$, and the remaining $(n - 1)$ edges have length $\geq d_{\min}$:

$$P_{\text{non-ultra}} > d_{\min} + (n - 1) \cdot d_{\min} = n \cdot d_{\min} \tag{3}$$

For an ultracycle with the same minimum distance d_{\min} (i.e., $\delta = d_{\min}$):

$$P_{\text{ultra}} = n \cdot d_{\min} \tag{4}$$

Therefore: $P_{\text{non-ultra}} > P_{\text{ultra}}$.

This proves that ultracycles uniquely minimize the TSP tour length in ultrametric spaces.

Security Implications

In Azure’s permission model, a Principal with n data actions all scoped to the same resource (forming an ultracycle at depth d in the hierarchy) achieves the minimal Data Perimeter, representing optimal privilege containment.

This geometric result aligns with the theoretical foundation for the least-privilege principle: permissions should be clustered as tightly as possible in the resource hierarchy to minimize potential lateral movement exposure.