Article

GeoVault: Leveraging Human Spatial Memory for Secure Cryptographic Key Management

Marko Corn [*,†,‡] and Primož Podržaj [‡]

Posted Date: 12 March 2026

doi: 10.20944/preprints202603.0935.v1

Keywords: cryptographic key management; spatial memory; mnemonic security; entropy modeling; memory-hard key derivation; Argon2; brainwallet security; human-centered cryptography

 doi:10.20944/preprints202603.0935.v1

Article

GeoVault: Leveraging Human Spatial Memory for Secure Cryptographic Key Management

Marko Corn [†,‡] * **and Primož Podržaj** [‡]

Faculty of mechanical engineering, University of Ljubljana
* Correspondence: marko.corn@fs.uni-lj.si
† Current address: Affiliation.
‡ These authors contributed equally to this work.

Abstract

Human-centered cryptographic key management is constrained by a persistent tension between security and usability. While modern cryptographic primitives offer strong theoretical guarantees, practical failures often arise from the difficulty users face in generating, memorizing, and securely storing high-entropy secrets. Existing mnemonic approaches suffer from severe entropy collapse due to predictable human choice, while machine-generated mnemonics such as BIP–39 impose significant cognitive burden. This paper introduces *GeoVault*, a spatially anchored key derivation framework that leverages human spatial memory as a cryptographic input. GeoVault derives keys from user-selected geographic locations, encoded deterministically and hardened using memory-hard key derivation functions. We develop a formal entropy model that captures semantic and clustering biases in human location choice and distinguishes nominal from effective spatial entropy under attacker-prioritized dictionaries. Through information-theoretic analysis and CPU–GPU benchmarking, we show that spatially anchored secrets provide a substantially higher effective entropy floor than human-chosen passwords under realistic attacker models. When combined with Argon2id, spatial mnemonics benefit from a hardware-enforced asymmetry that strongly constrains attacker throughput as memory costs approach GPU VRAM limits. Our results indicate that modest multi-point spatial selection combined with memory-hard derivation can achieve attacker-adjusted work factors comparable to those of 12-word BIP–39 mnemonics, while single-point configurations provide meaningful offline resistance with reduced cognitive burden.

Keywords: cryptographic key management; spatial memory; mnemonic security; entropy modeling; memory-hard key derivation; Argon2; brainwallet security; human-centered cryptography

1. Introduction

Secure cryptographic key management remains one of the most persistent usability challenges in modern security systems. While cryptographic primitives themselves have achieved a high degree of mathematical maturity, real-world security failures are frequently caused not by algorithmic weaknesses, but by the difficulty humans face in generating, storing, and reliably recalling high-entropy secrets. This mismatch between human cognitive capabilities and cryptographic requirements has motivated a wide range of mnemonic key storage approaches, including password-based systems, brainwallets, and standardized mnemonic phrases such as BIP-39.

Brainwallets and passphrase-derived keys attempt to eliminate the need for external storage by relying entirely on human memory. However, extensive empirical research has shown that such approaches fail under realistic offline attacker models due to severe entropy collapse caused by predictable human choice [1,2]. Large-scale studies of real-world password datasets demonstrate that users consistently select secrets with strong linguistic and cultural bias, resulting in effective entropy levels far below theoretical estimates [1,3]. Analyses of funded brainwallets further confirm

 doi:10.20944/preprints202603.0935.v1

that passphrase-derived keys are often compromised shortly after use, sometimes within minutes [2,4]. Even mnemonic-friendly strategies fail to provide durable protection once human selection bias is taken into account.

In contrast, machine-generated mnemonic schemes such as BIP-39 offer strong cryptographic guarantees by selecting words uniformly from a fixed dictionary, yielding well-defined entropy budgets. A standard 12-word BIP-39 mnemonic provides 128 bits of nominal entropy, making exhaustive brute-force attacks computationally infeasible under conventional assumptions. However, these schemes impose a significant cognitive burden on users, who must reliably store or back up the mnemonic phrase. In practice, this often leads to insecure storage practices or irreversible loss of access, effectively shifting the problem from entropy generation to key retention and recovery.

A key limitation shared by most existing mnemonic systems is their dependence on linguistic or abstract symbolic memory, which is known to be fragile and error-prone over long time horizons [5,6]. In contrast, decades of research in cognitive psychology and neuroscience have established that humans possess a robust capacity for spatial memory. Spatial representations are encoded, organized, and retrieved using dedicated cognitive mechanisms centered around landmarks and reference frames, and are retained accurately even after long periods without rehearsal [7–9]. Neurobiological evidence further supports this distinction: the hippocampus plays a central role in both spatial navigation and long-term memory formation, suggesting a deep functional link between spatial awareness and memory encoding [10].

These findings have motivated prior work exploring graphical and spatially inspired authentication mechanisms [11–13]. While such systems demonstrate improved memorability compared to textual passwords, they typically suffer from limited entropy, susceptibility to observation attacks, or strong user bias toward predictable choices. As a result, spatial memory has not yet been systematically integrated into cryptographic key derivation frameworks with explicit entropy modeling and hardware-aware offline threat assumptions.

In parallel, advances in cryptographic key derivation functions have made it possible to impose strong, hardware-enforced asymmetries between legitimate users and offline adversaries. Memory-hard functions such as Argon2 were explicitly designed to resist massively parallel GPU and ASIC attacks by binding performance to memory bandwidth and capacity rather than raw compute throughput [14,15]. Empirical studies have shown that such functions significantly reduce attacker advantage and allow defenders to trade modest increases in computation time for large increases in brute-force resistance [16].

This paper introduces *GeoVault*, a key management framework that combines these insights. GeoVault derives cryptographic keys from one or more user-selected spatial locations, encoded deterministically using geospatial encoding schemes and hardened through memory-hard key derivation. By anchoring secrets in spatial memory and amplifying their security through computational hardness, GeoVault aims to mitigate the security–usability limitations inherent in brainwallet-style systems under offline attack models.

The contributions of this work are threefold:

- We introduce a formal entropy model for spatially anchored secrets that distinguishes nominal spatial entropy from effective entropy under attacker-prioritized spatial dictionaries, capturing realistic semantic and clustering biases in human location choice.
- We analyze and quantify the security limits of spatial memory under realistic human selection behavior, showing how clustering and semantic constraints reduce effective entropy and how multi-point spatial selection mitigates this effect.
- We empirically evaluate the defender–attacker asymmetry induced by memory-hard key derivation using Argon2id, demonstrating that modest multi-point spatial selection combined with memory-hard derivation can achieve attacker-adjusted work factors that approach or, under certain configurations, exceed those of 12-word BIP–39 mnemonics, while single-point configurations provide meaningful offline resistance with reduced cognitive burden.

 doi:10.20944/preprints202603.0935.v1

The remainder of this paper is organized as follows. Section 2 reviews related work on brainwallets, mnemonic-based key storage, human memory limitations, spatial cognition, and geospatial encoding systems, highlighting the shortcomings of existing approaches and motivating the use of spatial memory as a cryptographic primitive. Section 3 introduces the theoretical foundations of the proposed approach, including formal entropy models for linguistic and spatial secrets, effective entropy under attacker-prioritized dictionaries, and the role of computational entropy boosting. Section 4 presents the design of the *GeoVault* protocol, detailing the spatial selection process, geospatial encoding, entropy-boosting mechanisms, and the attacker–defender security evaluation methodology. Section 5 reports empirical results, including spatial entropy analysis, CPU–GPU asymmetry benchmarks, and the impact of memory-hard key derivation on attacker work factors under realistic hardware constraints. Section 6 discusses the implications of the results, examines limitations and threat model assumptions, and positions GeoVault within the broader landscape of human-centered key management. Finally, Section 7 concludes the paper and outlines directions for future research.

2. Related Work

2.1. Brainwallets and Their Weaknesses

Brainwallets represent a class of deterministic cryptocurrency wallets in which private keys are derived directly from user-chosen passwords or passphrases. The underlying assumption is that users can reliably memorize sufficiently complex secrets, thereby eliminating the need for physical or digital storage of private keys [2,17]. In practice, however, extensive empirical research has demonstrated that this assumption is rarely satisfied, leading to severe security and usability limitations.

The core vulnerability of brainwallets stems from the predictable nature of human-chosen passwords and passphrases. Vasek et al. [2] conducted a large-scale empirical analysis of funded brainwallets and showed that users overwhelmingly select easily guessable phrases, resulting in a dramatic reduction of effective entropy. Their findings revealed that the majority of brainwallets were compromised shortly after being funded, often within minutes, underscoring the practicality of offline dictionary attacks against passphrase-derived keys.

Earlier work by Kuo et al. [3] identified similar weaknesses in mnemonic phrase-based passwords, demonstrating that users frequently rely on common phrases sourced from online material. Yang et al. [4] further confirmed that mnemonic-based password strategies, despite their perceived memorability benefits, produce secrets that remain highly vulnerable to statistical guessing attacks.

These vulnerabilities are compounded by fundamental cognitive limitations in human memory. Bonneau [1,18] demonstrated that, despite repeated efforts to promote stronger password practices, most users fail to achieve sufficient entropy in practice. Adams and Sasse [19,20] explained this phenomenon from a usability perspective, arguing that increasing password complexity requirements directly conflicts with human memory constraints and predictably leads users toward insecure coping strategies.

Attempts to mitigate these weaknesses through user education have shown limited effectiveness. Kävrestad and Nohlberg [21] found that context-based security training failed to produce meaningful improvements in real-world passphrase strength, highlighting the persistent tension between security requirements and human cognitive limitations.

Alternative authentication schemes that seek to exploit human visual memory have similarly encountered structural weaknesses. Tari et al. [12] and Golla et al. [13] investigated graphical and emoji-based password systems, identifying vulnerabilities such as susceptibility to shoulder-surfing and strong user bias toward predictable choices. While such approaches can improve memorability, they do not eliminate the fundamental problem of low effective entropy under targeted attack models.

To address the inherent weaknesses of brainwallets, alternative key management approaches have been proposed, including hierarchical deterministic (HD) wallets [22,23] and hardware-based wallets, commonly referred to as cold wallets [17,24]. HD wallets, standardized in BIP32, derive large key hierarchies from a single master seed, simplifying backup and recovery. However, Di Luzio et al. [22]

 doi:10.20944/preprints202603.0935.v1

identified privilege escalation vulnerabilities in multi-user settings, while Gutoski and Stebila [23] demonstrated that partial key leakage can enable recovery of the master secret, exposing the entire wallet hierarchy.

Cold wallets mitigate network-based attacks by storing private keys offline, but they introduce distinct attack surfaces. Das et al. [17] analyzed the hot/cold wallet paradigm and formalized conditions under which such systems remain secure. Nevertheless, Guri [24] experimentally demonstrated that air-gapped wallets can be compromised through physical side channels, including electromagnetic, acoustic, optical, and thermal leakage, challenging the assumption that hardware isolation alone provides absolute security.

In summary, brainwallets and mnemonic-based password schemes consistently suffer from two fundamental weaknesses: low effective entropy caused by predictable human-chosen secrets, and usability constraints rooted in cognitive limitations [2–4]. While HD wallets and cold wallets mitigate certain risks, they introduce additional complexity and new failure modes, such as master-key leakage [22,23] or physical side-channel exfiltration [17,24]. These limitations motivate the exploration of alternative approaches that align more closely with human cognitive strengths. In particular, leveraging spatial memory—an ability known for its robustness and long-term retention—offers a promising foundation for secure and user-friendly key management.

2.2. Human Spatial Memory as a Cryptographic Asset

Human memory is notoriously unreliable when it comes to memorizing abstract information such as complex alphanumeric strings or passphrases [5,6]. However, decades of cognitive psychology research have shown that humans exhibit a disproportionately strong ability to encode, recall, and retain spatial information [8,25]. This observation underpins the "method of loci," an ancient memory technique that leverages imagined spatial environments to recall information by placing memories at physical locations within a mental map [26].

Neurological evidence also supports the unique role of spatial memory in human cognition. The hippocampus, a brain structure critical for memory formation, is also central to spatial navigation [10]. The dual-use of this brain region suggests a strong evolutionary and functional link between spatial awareness and memory encoding. Studies in cognitive neuroscience confirm that people can recall spatial configurations with high accuracy even after long periods, particularly when these memories are grounded in visual or map-based stimuli [27,28].

More recently, spatial interfaces have been proposed in password systems and digital authentication [11]. Although graphical passwords do not directly translate to geographic locations, their effectiveness demonstrates that spatially anchored memory cues are more durable than abstract password recall. In usability studies, users retained spatially-based authentication secrets for longer durations and with fewer errors compared to textual passphrases [29].

These findings suggest that spatial memory may provide a superior foundation for cryptographic key storage and retrieval mechanisms. Unlike traditional brainwallets that rely on memorized strings, a system that anchors a seed derivation process to a spatial coordinate or map location would naturally exploit this cognitive strength. Such systems could offer both better security and usability, reducing the cognitive load while maintaining high entropy—especially when combined with computational functions that bind access to the correct location and effort.

2.3. Geospatial Encoding Systems

Geospatial encoding systems provide a structured way to represent geographic locations using codes or identifiers, facilitating applications such as navigation, logistics, and, in the context of this research, secure mnemonic storage based on spatial memory. These systems translate physical locations into a format that can be easily shared and processed, making them a critical component of the proposed *GeoVault* protocol.

A widely recognized example is What3Words (w3w), which partitions the globe into 3m x 3m squares, each assigned a unique three-word identifier. This system excels in usability due to its intu-

 doi:10.20944/preprints202603.0935.v1

itive word-based format, reducing errors compared to traditional coordinates, and supports multiple languages [30]. However, its fixed resolution limits its flexibility for applications requiring finer granularity or three-dimensional encoding, such as indoor environments [31]. Moreover, recent critiques highlight potential confusion between similar word triplets, posing risks in critical applications like emergency response or secure storage [32]. These limitations are particularly relevant to *GeoVault*, where precise and unambiguous location encoding directly impacts mnemonic security.

Alternative systems include Geohash, Google's Open Location Code (OLC), and Google S2. Geohash employs a hierarchical grid with alphanumeric codes, offering variable precision but suffering from complexity and potential user confusion due to its base32 encoding [33]. Open Location Code, designed for offline use, provides a more accessible alphanumeric approach, balancing usability and scalability [33]. Google S2, optimized for spatial indexing, uses a cell-based structure that supports efficient querying and could enhance the protocol's computational efficiency [33]. Each system presents distinct characteristics: Geohash and OLC prioritize flexibility, while S2 emphasizes performance in large-scale applications.

The choice of a geospatial encoding system for *GeoVault* involves several trade-offs. Usability is critical for enabling users to recall and input locations accurately, favoring systems like What3Words [30]. Granularity, or the size of the encoded area, affects the entropy of the system—a finer grid increases the number of possible locations, strengthening security against brute-force attacks. Encoding precision also influences the integration with computation-hard derivation functions like Argon2, as higher precision demands greater computational effort to protect the mnemonic seed. Additionally, reliability and accuracy are paramount, as geocoding errors can compromise the protocol's integrity. Studies reveal significant variability in geocoding accuracy across services and regions, necessitating robust validation frameworks [34,35].

In this research, geospatial encoding systems underpin the spatial anchoring of mnemonic seeds. What3Words offers a compelling starting point due to its balance of usability and precision, yet its fixed resolution and potential for confusion [32] suggest the need for enhancements, such as variable resolution or three-dimensional support [30,31]. Alternatives like Google S2 could improve scalability and precision, while custom schemes might better align with the protocol's security requirements [33]. Furthermore, address extraction and matching techniques from web data, as explored in [36], could enhance location selection, ensuring users choose memorable yet secure burial sites.

Ultimately, the selected system must integrate seamlessly with the protocol's threat model, resisting attacks such as map scanning or brute-forcing while maintaining user-friendliness. Future work will explore hybrid approaches, potentially combining What3Words' simplicity with S2's flexibility, and evaluate their performance in real-world mnemonic recovery scenarios.

2.4. Encryption Techniques for Burying Secrets

Secure storage of mnemonic seeds in the *GeoVault* protocol relies on cryptographic techniques that impose significant computational barriers to unauthorized recovery while ensuring practical access for legitimate users. This section examines four mechanisms—memory-hard key derivation functions (KDFs), verifiable delay functions (VDFs), time-lock puzzles, and proofs of sequential work (PoSW)—that protect against brute-force attacks by leveraging memory, time, or sequential computation. Each technique is illustrated with a scientific example to demonstrate its role in securing spatially anchored mnemonic seeds.

Memory-hard KDFs, such as Argon2, transform inputs like spatial coordinates into cryptographic keys through processes that require substantial memory resources, thereby resisting optimization by hardware accelerators such as ASICs or GPUs [14,15]. For instance, Argon2d with parameters set to use 1 GB of memory and 10,000 iterations generates a 256-bit key from coordinates (e.g., 40.7128° N, 74.0060° W) by filling a 1 GB array with pseudorandom values and iteratively referencing it in a dependent pattern. An adversary with only 100 MB of memory must recompute missing blocks, increasing their runtime by a factor of 10,000 compared to a system with the full 1 GB. This memory-computation trade-off ensures that brute-forcing keys across multiple locations is resource-intensive,

 doi:10.20944/preprints202603.0935.v1

making large-scale attacks impractical. However, emerging compute-capable memory technologies, such as near-data processing, may reduce memory access costs, potentially weakening memory-hard functions like scrypt [16], underscoring the need for adaptable KDFs.

Verifiable delay functions (VDFs) enforce a fixed number of sequential computational steps, with efficient verification of the result [37,38]. Consider a VDF based on repeated squaring: computing $y = x^{2^t} \mod N$, where N is a 2048-bit RSA modulus, x is derived from the coordinates, and $t = 10^6$. On a 3 GHz CPU, each squaring takes approximately 1 microsecond, totaling 1 second for 1 million squarings. Even with 1,000 parallel CPUs, the sequential dependency ensures the computation still takes 1 second. Verification, however, is a single modular exponentiation, completed in milliseconds. This property is ideal for *GeoVault*, as it imposes a mandatory delay on seed derivation, preventing rapid attacks while allowing quick validation. Recent VDF constructions achieve tight efficiency and reduce prover storage via incremental computation [39,40], though their reliance on groups of unknown order may complicate implementation [38].

Time-lock puzzles encrypt data, such as a mnemonic seed, to be unlocked only after a predefined computational effort, typically through sequential operations [41]. For example, a puzzle requiring $t = 10^9$ squarings modulo a 2048-bit N takes approximately 17 minutes on a high-end CPU performing 1 million squarings per second. The encryption might be structured as $c = m + H(k)^{2^t} \mod N$, where m is the seed and $H(k)$ is a hash of the coordinates. Decryption requires computing $H(k)^{2^t}$ sequentially, with no parallelization benefit. This ensures that the seed remains inaccessible until the full computational effort is expended, mirroring the burial of a secret in *GeoVault*. While constructions using repeated squaring or bilinear pairings offer provable security, they are vulnerable to advances in factoring or quantum attacks [41,42]. Non-interactive timed-release encryption schemes avoid server dependency but often rely on identity-based encryption, introducing additional complexity [43,44].

Proofs of sequential work (PoSW) enable verification of sequential computational effort, with applications in blockchain consensus and time-stamping [45,46]. Consider a PoSW where a sequence of hashes is computed: $h_0 = H(\text{coordinates})$, $h_1 = H(h_0)$, ..., $h_{10^6} = H(h_{10^5})$, using SHA-256, taking approximately 1 second. A Merkle tree over this sequence allows the prover to present a compact proof (the root and a path to h_{10^6}), which can be verified in microseconds. This proves that the solver performed 1 million sequential hashes tied to the coordinates, ensuring that only users at the correct location who complete the required work can retrieve the seed. Efficient PoSW designs reduce prover space to logarithmic levels using hash-based structures in the random oracle model [45,46], enhancing *GeoVault*'s defenses against map-scanning attacks.

Collectively, these techniques address the cryptographic needs of *GeoVault*: memory-hard KDFs resist hardware-accelerated attacks, VDFs enforce sequential delays, time-lock puzzles provide temporal protection, and PoSW verify location-specific computational effort. Challenges remain, including calibrating computational difficulty to match the 128-bit entropy of a 12-word BIP-39 mnemonic and mitigating risks from hardware advancements [16,47]. Future work could explore hybrid approaches that combine these mechanisms to optimize security, usability, and resistance to parallelized attacks in spatially anchored mnemonic storage.

3. Theoretical Background

3.1. Entropy in Mnemonic and Password-Based Secure Storage

Cryptographic key management systems frequently derive secret keys from human-memorable inputs, including machine-generated mnemonics, user-chosen passwords, and passphrases, collectively referred to as brainwallet-based constructions. The security of such systems is commonly expressed in terms of entropy, which characterizes the size of the underlying search space available to an offline attacker. Shannon entropy [48] provides a fundamental measure of uncertainty in this context and is defined in equation 1.

$$H(\mathcal{A}) = -\sum_{a_i \in \mathcal{A}} P(a_i) \log_2 P(a_i), \tag{1}$$

 doi:10.20944/preprints202603.0935.v1

where $\mathcal{A}$ denotes the alphabet of possible secrets and $P(a_i)$ is the probability of selecting element a_i.

While Shannon entropy provides a useful conceptual measure, resistance to offline guessing attacks is ultimately governed by the effective size of the attacker's prioritized search space. In practice, this corresponds to the entropy remaining once human selection bias and attacker-adaptive guessing strategies are taken into account.

In the BIP–39 standard, mnemonic phrases are generated algorithmically by selecting words uniformly and independently from a fixed dictionary. A standard 12-word BIP–39 mnemonic uses a dictionary of 2048 entries and yields exactly 128 bits of nominal entropy, corresponding to 2^{128} possible mnemonic combinations. Under the assumption of uniform random generation, this construction provides strong cryptographic resistance against exhaustive brute-force attacks.

To illustrate the scale implied by 128 bits of nominal entropy, consider an offline attacker equipped with modern GPU hardware optimized for hashing operations. A high-end consumer GPU can evaluate on the order of 10^9 candidate keys per second under lightweight hashing assumptions [49–51]. Under this assumption, the expected time required to exhaustively search the full mnemonic space is given by equation 2.

$$T = \frac{2^{128}}{10^9} \text{ seconds} \approx 3.4 \times 10^{29} \text{ seconds}, \tag{2}$$

This corresponds to approximately 1.08×10^{22} years, vastly exceeding the age of the universe. Consequently, uniformly generated BIP–39 mnemonics are effectively immune to brute-force attacks under realistic computational assumptions.

In contrast, password- and passphrase-based brainwallets derive cryptographic keys directly from user-chosen secrets. From a theoretical perspective, if a password or passphrase were selected uniformly at random from a dictionary of size N, each symbol would contribute $\log_2 N$ bits of nominal entropy. The minimum length w required to reach a target of 128 bits of nominal entropy is given by equation 3.

$$w = \left\lceil \frac{128}{\log_2 N} \right\rceil. \tag{3}$$

Table 1 summarizes this relationship for several representative selection sets.

Table 1. Required length to achieve at least 128 bits of nominal entropy under uniform random selection.

Selection Set (N)	Entropy per Unit	Length Required (L)
Alphanumeric (62 chars)	5.95 bits/char	22 characters
Standard ASCII (94 chars)	6.55 bits/char	20 characters
BIP–39 Wordlist (2048 words)	11.0 bits/word	12 words
Diceware (7776 words)	12.9 bits/word	10 words

While Table 1 describes the entropy achievable under idealized uniform selection, extensive empirical evidence shows that human-generated passwords and passphrases violate these assumptions in practice. Large-scale analyses of real-world password datasets reveal that user choice introduces significant linguistic, cultural, and structural biases, resulting in a dramatic reduction of effective entropy under realistic attacker models.

Bonneau [1], analyzing an anonymized corpus of nearly 70 million passwords, showed that typical user-chosen passwords provide fewer than 10 bits of resistance against online guessing attacks and on the order of 20 bits against optimized offline dictionary attacks. Vasek et al. [2] similarly observed that brainwallet passphrases are highly predictable and frequently compromised shortly after funding. Earlier studies by Kuo et al. [3] and subsequent work by Yang et al. [4] further corroborate that mnemonic and passphrase-based strategies, despite increased length or apparent complexity, remain vulnerable to statistical guessing due to human selection bias. Collectively, these results establish entropy collapse as a fundamental limitation of password- and passphrase-based brainwallets and

 doi:10.20944/preprints202603.0935.v1

motivate security models that explicitly distinguish nominal entropy from effective entropy under attacker-adaptive guessing strategies.

3.2. Entropy in Spatial Memory-Based Systems

The cryptographic strength of spatially anchored mnemonic systems, such as *GeoVault*, derives from the discretization of a continuous spatial domain into a finite set of selectable cells. Each cell constitutes a mnemonic element analogous to a word in a linguistic dictionary. The entropy provided by such systems is therefore determined by the resolution of the spatial discretization and by the effective size of the attacker's prioritized spatial search space.

3.2.1. Nominal Spatial Entropy

Let $\mathcal{M}$ denote a spatial domain with total surface area $A_{\mathcal{M}}$, discretized into cells of area A_{cell}. Under uniform random selection, the nominal entropy of a single spatial choice is defined in equation 4.

$$H_{\text{nominal}} = \log_2\left(\frac{A_{\mathcal{M}}}{A_{\text{cell}}}\right). \tag{4}$$

For the Earth's surface ($A_{\mathcal{M}} \approx 510.1 \times 10^{12}\,\text{m}^2$) and a discretization with $A_{\text{cell}} = 9\,\text{m}^2$, the nominal entropy of a single spatial cell is given by equation 5.

$$H_{\text{global}} = \log_2\left(\frac{510.1 \times 10^{12}}{9}\right) \approx 45.7\ \text{bits}. \tag{5}$$

Achieving a target of 128 bits of entropy via a single spatial cell would require an unrealistically small cell area, as shown in equation 6.

$$A_{\text{cell}} = \frac{A_{\mathcal{M}}}{2^{128}} \approx 1.5 \times 10^{-24}\,\text{m}^2, \tag{6}$$

which is infeasible in practice. Consequently, spatial systems must rely on either multiple spatial selections or computational entropy-boosting mechanisms.

3.2.2. Effective Entropy and Spatial Dictionaries

Nominal spatial entropy assumes uniform random selection across the spatial domain $\mathcal{M}$. However, extensive research in spatial cognition shows that human spatial memory is hierarchically structured around landmarks, reference frames, and semantically meaningful regions, rather than represented as a uniform metric space [7,9]. As a result, realistic offline attackers can prioritize large portions of the search space by exploiting predictable human spatial behavior.

To model such attacks, we introduce the concept of *spatial dictionaries*, defined as attacker-prioritized subsets $\mathcal{D} \subseteq \mathcal{M}$ that reflect semantic, geographic, or contextual constraints on likely user choices. This abstraction captures coarse-grained attacker strategies, such as restricting guesses to landmasses, urban areas, coastlines, or other salient regions, without assuming knowledge of user-specific preferences or fine-grained popularity rankings.

Let $A_{\text{eff}}(\mathcal{D})$ denote the total area covered by a spatial dictionary $\mathcal{D}$. The effective entropy of a single spatial selection under this attacker prior is given by equation 7.

$$H_{\text{eff}}(\mathcal{D}) = \log_2\left(\frac{A_{\text{eff}}(\mathcal{D})}{A_{\text{cell}}}\right). \tag{7}$$

The corresponding entropy collapse relative to the global nominal case is defined in equation 8.

$$\Delta H(\mathcal{D}) = H_{\text{global}} - H_{\text{eff}}(\mathcal{D}). \tag{8}$$

 doi:10.20944/preprints202603.0935.v1

3.2.3. Clustering Bias and Localized Collapse

A particularly severe form of entropy collapse arises when users select multiple locations within a familiar local context. If an initial anchor point p_1 is selected freely, but subsequent points p_i are constrained to lie within a radius r of p_1, the entropy contribution of each additional point is bounded by the area of the surrounding neighborhood. Under this proximity constraint, the entropy contribution per additional point is given by equation 9.

$$H_r(r) = \log_2\left(\frac{\pi r^2}{A_{\text{cell}}}\right). \tag{9}$$

The total effective entropy for n spatial selections under this proximity-constrained model can then be approximated as shown in equation 10.

$$H_{\text{spatial}}(n, r) \approx H_{\text{anchor}} + (n-1)\, H_r(r). \tag{10}$$

For clustering at city scale (e.g., $r = 20$ km), this model shows that even multiple spatial selections fail to approach a 128-bit entropy target under realistic attacker assumptions. This observation demonstrates that spatial entropy alone is insufficient to guarantee cryptographic-strength security in the presence of predictable clustering behavior and motivates the integration of computational entropy-boosting mechanisms.

3.3. Entropy Boosting Techniques

Given the practical limits of spatial entropy, spatially anchored mnemonic systems such as *GeoVault* rely on entropy-boosting techniques to increase resistance against offline brute-force attacks. Rather than increasing the number or resolution of spatial elements, these techniques amplify security by increasing the computational cost of each guess.

Key derivation functions (KDFs) transform low-entropy or biased inputs into cryptographically strong keys by imposing configurable time and memory costs per evaluation. Memory-hard KDFs such as Argon2 are particularly effective in this role, as they bind attacker performance to memory bandwidth and capacity rather than raw parallel compute throughput, thereby reducing the advantage of GPUs and ASICs. When combined with spatial mnemonics, memory-hard KDFs allow systems to achieve high attacker-adjusted work factors despite predictable human selection behavior.

4. Materials and Methods

4.1. Protocol Design

The GeoVault protocol derives cryptographic keys from one or more user-chosen spatial locations by combining deterministic geospatial encoding with memory-hard key derivation. The design goal is to anchor secrets in human spatial memory while ensuring that key derivation remains fully reproducible and verifiable in an offline setting.

At a high level, the protocol consists of four conceptual stages: spatial selection, geospatial encoding, entropy boosting, and key extraction. First, the user selects a finite set of spatial points within a predefined spatial domain. These selections are made deliberately and are assumed to be recallable by the user without external storage. Second, each selected point is deterministically mapped to a discrete spatial cell using a geospatial encoding scheme, transforming continuous coordinates into a finite and reproducible representation that can be evaluated offline.

Third, the encoded spatial identifiers are combined and processed using a memory-hard key derivation function. This entropy-boosting stage is implemented as a single invocation of a memory-hard KDF with configurable time and memory parameters. It does not introduce additional information, but instead increases the computational and memory cost associated with evaluating each candidate secret, thereby reducing the feasibility of large-scale offline brute-force attacks, particularly on GPU-class hardware.

 doi:10.20944/preprints202603.0935.v1

Finally, the output of the key derivation function is used directly as a cryptographic key or as input to a higher-level key management mechanism, depending on the application context.

The protocol makes no assumptions about secrecy of the geospatial encoding scheme or obscurity of the spatial discretization. All security derives from the unpredictability of the user's spatial choices and the computational hardness imposed by the key derivation function. By separating human-memorable input selection from computational hardening, GeoVault allows the security–usability trade-off to be adjusted through two independent parameters: the number of spatial selections and the cost parameters of the key derivation function.

4.2. Spatial Selection

GeoVault assumes that a user selects a finite set of geographic points. Formally, the set of selected locations is defined in equation 11.

$$\mathcal{P} = \{p_1, p_2, \ldots, p_n\}, \qquad p_i = (\varphi_i, \lambda_i) \in \mathcal{M}, \tag{11}$$

Here, $\mathcal{M} \subset \mathbb{R}^2$ denotes the spatial domain of the application, and each point p_i is represented by its coordinates within that domain.

In the baseline instantiation, $\mathcal{M}$ corresponds to the WGS-84 model of the Earth's surface. However, the definition is deliberately abstract: $\mathcal{M}$ may also represent a fictional continent, a virtual environment, or any other two-dimensional world equipped with a well-defined coordinate system. No randomness, salting, or grid quantization is imposed at this stage; each point p_i is treated as an exact element of $\mathcal{M}$ prior to encoding.

The cardinality n of $\mathcal{P}$ is user-defined and represents a primary degree of freedom in the security-usability trade-off. Selecting more points increases the available spatial entropy but also increases the cognitive burden during recall. This trade-off is analyzed quantitatively in later sections through entropy modeling and attacker-adjusted work factor evaluation.

4.3. Geospatial Encoding

After the user selects a set of spatial points $\mathcal{P} = \{p_1, p_2, \ldots, p_n\} \subset \mathcal{M}$, GeoVault deterministically maps each point to a discrete spatial cell and produces a corresponding identifier. The purpose of geospatial encoding is to transform continuous spatial coordinates into a finite, reproducible representation that can be evaluated offline by both legitimate users and adversaries.

We formalize a geospatial encoding scheme as the composition defined in equation 12.

$$E = \mathsf{code} \circ \mathsf{snap}, \tag{12}$$

Here, $\mathsf{snap} : \mathcal{M} \to \mathcal{C}$ partitions the spatial domain into equal-area cells and returns the cell containing a point p_i, while $\mathsf{code} : \mathcal{C} \to \Sigma^*$ assigns each cell a deterministic identifier over a finite alphabet. The mapping E is applied independently to each $p_i \in \mathcal{P}$, yielding a multiset of identifiers

$$\mathcal{I} = \{E(p_1), E(p_2), \ldots, E(p_n)\}. \tag{13}$$

The identifier strings themselves are not assumed to contribute cryptographic entropy; GeoVault relies exclusively on the unpredictability of the underlying spatial selections. Consequently, the encoding scheme must satisfy three core properties. First, *injectivity*: distinct spatial cells must map to distinct identifiers. Second, *determinism*: the same spatial point must always yield the same identifier. Third, *offline resolvability*: the encoding must be computable without access to a trusted online service. Beyond these requirements, the encoding need not be cryptographically random or secret and may be designed primarily for human usability.

 doi:10.20944/preprints202603.0935.v1

Baseline Encoder (What3Words)

In the reference implementation, both snap and code are instantiated using the *What3Words* grid and lexicon. The spatial domain is partitioned into 3×3 m cells, each mapped to a unique three-word phrase. For a single unconstrained cell, substituting the Earth's surface area $A_{\mathcal{M}} \approx 5.1 \times 10^{14}\,\mathrm{m}^2$ and cell area $A_{\mathrm{cell}} = 9\,\mathrm{m}^2$ yields a nominal entropy of approximately 45.7 bits, as derived previously in equation 5. This value reflects the size of the global cell index and is independent of the linguistic structure of the identifier.

Because a single encoded cell does not provide sufficient entropy for high-security applications, GeoVault supports aggregation of multiple encoded spatial points and introduces additional computational hardening in subsequent stages. Importantly, the cryptographic core of GeoVault remains encoder-agnostic: any geospatial encoding scheme that satisfies the above properties may replace the What3Words system without altering the security model or attacker assumptions.

4.4. Entropy Boosting

The entropy provided by spatial selection alone is insufficient to guarantee cryptographic-strength security under realistic offline attacker models, particularly in the presence of semantic bias and spatial clustering (Section 3.2). GeoVault therefore incorporates an explicit entropy-boosting stage based on a memory-hard key derivation function (KDF), which increases the computational cost of each candidate evaluation without introducing additional information entropy.

This mechanism does not increase Shannon entropy, but instead amplifies the *computational hardness* faced by an attacker by enforcing substantial time and memory costs per guess. As a result, brute-force resistance depends jointly on the effective spatial entropy and the hardware-bound cost of key derivation.

4.4.1. Key Derivation Function

GeoVault employs Argon2id, the hybrid variant of the Argon2 memory-hard KDF [14]. Argon2id was selected for its resistance to GPU- and ASIC-accelerated attacks, combining data-independent memory access (Argon2i) with data-dependent access (Argon2d) to mitigate both side-channel and time-memory trade-off attacks.

Unlike legacy KDFs such as PBKDF2, Argon2id binds attacker performance to memory bandwidth and capacity rather than raw compute throughput, enabling strong defender-attacker asymmetry through tunable time, memory, and parallelism parameters.

4.4.2. Input Construction

Let $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ denote the set of deterministic identifiers produced by geospatial encoding (Section 4.3). These identifiers are concatenated in a fixed canonical order:

$$S = I_1 \parallel \texttt{|} \parallel I_2 \parallel \texttt{|} \parallel \cdots \parallel \texttt{|} \parallel I_n, \tag{14}$$

and hashed to obtain a fixed-length KDF input

$$X = H(S), \tag{15}$$

where H denotes SHA-256 in the reference implementation.

4.4.3. Argon2id Hardening

The spatial hash X is processed using Argon2id as defined in Equation 16:

$$K = \mathsf{Argon2id}(X, \mathrm{salt}, t, m, p), \tag{16}$$

where t is the iteration count, m the memory cost, and p the degree of parallelism. The salt is fixed and public and does not contribute entropy.

 doi:10.20944/preprints202603.0935.v1

All parameters (t, m, p) are assumed to be known to the attacker, consistent with Kerckhoffs' principle. Security arises from the enforced memory footprint of Equation 16, which bounds attacker parallelism by available high-bandwidth memory. The resulting defender-attacker asymmetry and its impact on offline brute-force resistance are quantified empirically in Section 5.

4.5. Security Evaluation Methodology

The goal of our evaluation is to quantify how much practical protection a user gains from a given key-derivation cost when the defender derives keys on a CPU while the attacker mounts an offline brute-force attack using GPU-class hardware. In particular, we seek to answer questions of the form: *"if a user is willing to spend T_{CPU} seconds on a memory-hard key derivation function, what level of brute-force resistance does this provide against a realistic GPU-equipped attacker?"*

Our methodology decomposes the problem into three components: a defender (user) cost model, an attacker cost model, and a derived security metric that combines both into a single, comparable quantity.

4.5.1. Defender (User) Cost Model

On the defender side, we assume that the user derives keys using Argon2 on a commodity CPU. For a given choice of Argon2 parameters (t, m, p), where t denotes the number of iterations, m the memory cost, and p the degree of parallelism, the wall-clock time required to compute a single key-derivation invocation is defined in equation 17.

$$T_{\mathrm{CPU}}(t, m, p) = \text{runtime per Argon2 invocation on the user CPU.} \tag{17}$$

This quantity directly captures the usability cost: larger values of $T_{\mathrm{CPU}}(t, m, p)$ increase resistance to brute-force attacks but also increase the delay experienced by legitimate users during key derivation or recovery. In our evaluation, we restrict attention to parameter sets for which T_{CPU} remains within an acceptable latency budget (e.g., below one second) for interactive use.

4.5.2. Attacker (GPU) Cost Model

On the attacker side, we assume access to parallel computing resources typical of modern GPUs and distinguish between two fundamentally different classes of operations.

- **Fast hash evaluations** (e.g., SHA-256), which are compute-bound and highly parallelizable, and are relevant for attacking mnemonic schemes without additional computational hardening, such as raw BIP–39 verification.
- **Memory-hard KDF evaluations** (Argon2id), which are constrained by memory bandwidth and capacity rather than raw compute throughput, and are relevant for attacking spatial mnemonics protected by a memory-hard key derivation function. In this case, the attacker is assumed to evaluate the same Argon2 parameter set (t, m, p) as the defender.

For each class of operations, we characterize the attacker by an effective guess rate, defined in equations 18 and 19.

$$R_{\mathrm{GPU,hash}} = \text{hash evaluations per second for a fast hash function (e.g., SHA-256),} \tag{18}$$

$$R_{\mathrm{GPU,Argon2}}(t, m, p) = \text{Argon2 evaluations per second with parameters } (t, m, p). \tag{19}$$

A key distinction between the two cases lies in their scalability under parallelism. Fast hash functions scale primarily with available compute resources and can exploit massive parallelism efficiently. In contrast, parallelism for memory-hard KDFs is intrinsically bounded by available high-

 doi:10.20944/preprints202603.0935.v1

bandwidth memory. For an attacker with total memory capacity M, the maximum number of fully independent parallel KDF instances is upper-bounded by

$$P_{\max} = \left\lfloor \frac{M}{m} \right\rfloor,$$

where m denotes the per-instance memory cost. Once this bound is reached, additional compute units cannot be exploited to increase throughput, and attacker performance becomes memory-bound rather than compute-bound.

4.5.3. Derived Security Metric: Attacker-Adjusted Work Factor

To compare the brute-force resistance of mnemonic- and spatial-based schemes under realistic adversarial capabilities, we employ a unified security metric termed the *attacker-adjusted work factor*, denoted $\mathcal{W}_{\text{attacker}}$. This metric captures the expected time required for an offline adversary to exhaust the effective search space of candidate secrets under the assumed attacker model.

Let H denote the effective entropy of the scheme, so that the total number of candidates is $N = 2^H$. Let C represent the computational cost required to evaluate a single guess, expressed in hash-equivalent operations, and let R denote the attacker's sustained guess rate in evaluations per second. The expected attacker work factor is given by equation 20.

$$\mathcal{W}_{\text{attacker}} = \frac{N \cdot C}{R} = \frac{2^H \cdot C}{R}. \tag{20}$$

Empirical Instantiation

In practice, the attacker's guess rate R is instantiated using measured GPU throughput, while C is determined by the cost of the key derivation function (KDF) or hash evaluation used by the scheme. For constructions employing a memory-hard KDF such as Argon2 with parameters (t, m, p), the attacker-adjusted work factor is approximated by equation 21.

$$\mathcal{W}_{\text{attacker}}(H, t, m, p) \approx \frac{2^H}{R_{\text{GPU,Argon2}}(t, m, p)}. \tag{21}$$

For baseline schemes without computational hardening—such as BIP–39 mnemonics or single-point spatial secrets verified using a single cryptographic hash—the work factor reduces to equation 22.

$$\mathcal{W}^{\text{baseline}}_{\text{attacker}}(H) \approx \frac{2^H}{R_{\text{GPU,hash}}}. \tag{22}$$

Interpretation via Security Zones

To aid interpretation of attacker-adjusted work factors, we categorize security outcomes into three qualitative zones based on the expected offline attack time. These zones are not intended as strict security guarantees, but as interpretive reference points commonly used in password security and cryptographic practice [52].

- **Insecure zone.** Configurations for which $\mathcal{W}_{\text{attacker}}$ corresponds to attack times on the order of hours, days, or weeks on commodity GPU hardware.
- **Human-safe zone.** Configurations for which $\mathcal{W}_{\text{attacker}}$ corresponds to attack times on the order of years to centuries under the assumed attacker model, reflecting security levels commonly considered sufficient for systems relying on human-memorable secrets.
- **Cryptographic-strength zone.** Configurations for which $\mathcal{W}_{\text{attacker}}$ approaches or exceeds the effective brute-force resistance of uniformly generated 128-bit secrets, such as standard BIP–39 mnemonics, and are widely regarded as computationally infeasible even for highly resourced adversaries.

 doi:10.20944/preprints202603.0935.v1

The Results section uses this attacker-adjusted work factor and the associated security zones to interpret empirical measurements across different spatial entropy levels and key-derivation parameter choices.

4.6. Construction of Spatial Dictionaries

All spatial dictionaries use a fixed cell resolution of 3×3 m, corresponding to a cell area of $A_{\text{cell}} = 9 \times 10^{-6}\,\text{km}^2$. To account for tolerance in human spatial recall and imprecision in location memory, semantic features are expanded using an orientable buffer of width $w = 100$ m. This buffer models the fact that users may recall locations approximately rather than with exact geometric precision.

For areal regions, including the global surface, terrestrial land, habitable land, and urban boundaries, the effective area A_{eff} is set directly to the reported surface area of the corresponding region, as obtained from established geographic datasets [53**?**]. These dictionaries represent attacker strategies that restrict guesses to broad but semantically meaningful regions.

For linear features such as coastlines and rivers, the effective area is approximated by buffering the feature along its total length. For a feature of total length L (in km), the effective area is given by equation 23.

$$A_{\text{eff}}^{\text{linear}} \approx L \cdot w_{\text{km}}, \tag{23}$$

where $w_{\text{km}} = 0.1$ km denotes the buffer width expressed in kilometers. For rivers, buffers are applied symmetrically to both banks, yielding the effective area defined in equation 24.

$$A_{\text{eff}}^{\text{rivers}} \approx L_{\text{rivers}} \cdot 2w_{\text{km}}. \tag{24}$$

For point-like semantic anchors such as UNESCO World Heritage sites or major mountain peaks, each anchor is modeled as a disk of radius w. For a dictionary containing N such anchors, the effective area is approximated by equation 25.

$$A_{\text{eff}}^{\text{points}} \approx N \cdot \pi w_{\text{km}}^2. \tag{25}$$

Overlaps between buffered regions are ignored in all cases, yielding conservative, attacker-favoring estimates of effective area and corresponding entropy.

4.7. Experimental Setup

All measurements in this paper, including BIP–39 and Argon2id, were obtained on a single controlled workstation and software environment to ensure methodological consistency. The purpose of this setup is to empirically characterize both defender-side computation costs and attacker-side throughput under realistic CPU and GPU execution models. No parameter tuning beyond vendor- or implementation-recommended defaults was applied unless explicitly stated.

Table 2 summarizes the hardware platform, operating system, GPU driver stack, and cryptographic benchmarking tools used throughout the study. Hashcat was used for both PBKDF2 (mode 12100) and Argon2id (mode 8200) benchmarks to ensure comparability across key-derivation functions.

 doi:10.20944/preprints202603.0935.v1

Table 2. Hardware and software configuration used for all KDF benchmarks (PBKDF2–HMAC–SHA512 and Argon2id).

Category	Specification
Hardware	
CPU	Intel Xeon Gold 6338 (Ice Lake), 32 cores @ 2.0 GHz
GPU	NVIDIA RTX A6000 (GA102GL), 48 GB GDDR6
System Memory	128 GB DDR4 ECC
Storage	NVMe SSD
Operating System & Drivers	
OS	Ubuntu 24.04 LTS
Kernel	Linux 6.x (distribution default)
NVIDIA Driver	CUDA 12.4 / driver 550.xx
OpenCL (CPU)	PoCL 5.0 (LLVM 16)
Software Tools	
Hashcat	v6.2.6 (benchmark mode)
Benchmark mode (PBKDF2)	`-b -m 12100`
Benchmark mode (Argon2id)	`-b -m 8200`
OpenSSL	v3.x (reference PBKDF2 implementation)
Shell environment	GNU bash 5.x
Benchmark Configuration	
PBKDF2 algorithm	PBKDF2–HMAC–SHA512 (BIP–39 standard)
PBKDF2 iteration baseline	999 (hashcat default)
PBKDF2 target scaling	2048 iterations (BIP–39 specification)
Argon2 algorithm	Argon2id
Argon2 parameters	(t, m, p) selected as described in Section **??**
Salt (PBKDF2)	`"mnemonic"` (BIP–39 standard)
Password input	Fixed test vectors for reproducibility
Backend selection	CPU-only: `-D 1`; GPU: default CUDA device

Attacker-side benchmarks were conducted on a high-end GPU representative of a realistic, well-provisioned offline adversary, consistent with the attacker model defined in Section 4.5. For fast hash functions such as SHA-256, attacker throughput is primarily compute-bound and scales efficiently with available parallel execution units. In contrast, for memory-hard key derivation functions such as Argon2id, scalability is fundamentally constrained by available high-bandwidth device memory. Given a per-instance memory cost m and total GPU memory M_{GPU}, the maximum number of fully independent parallel Argon2 evaluations is upper-bounded by

$$P_{\max} = \left\lfloor \frac{M_{\text{GPU}}}{m} \right\rfloor.$$

Once this bound is reached, additional compute resources cannot be exploited to increase throughput, and attacker performance becomes memory-bound rather than compute-bound. All attacker-side measurements therefore reflect configurations chosen to maximize effective guess throughput, corresponding to a near-worst-case attacker from the defender's perspective.

Benchmark configurations were selected to reflect canonical settings of the evaluated schemes while ensuring reproducibility and a clear separation between defender and attacker constraints. For PBKDF2–HMAC–SHA512, the BIP–39 specification was followed using 2,048 iterations and the fixed salt string `"mnemonic"`. Benchmark tools that internally employ reduced iteration counts were used only to measure raw throughput, and reported attacker rates were scaled linearly to correspond to the full BIP–39 iteration count.

For Argon2-based spatial mnemonics, the Argon2id variant was evaluated using parameter triples (t, m, p), where t denotes the number of passes, m the memory cost per instance, and p the degree of parallelism. Defender-side measurements were obtained using fixed parameter sets to characterize user-perceived latency under interactive constraints. Attacker-side measurements, in

 doi:10.20944/preprints202603.0935.v1

contrast, were obtained under maximal-throughput configurations in which available GPU parallelism and memory allocation were selected to saturate device resources and maximize the measured guess rate $R_{\text{GPU,Argon2}}(t, m, p)$. All benchmarks were executed under sustained load and repeated across multiple runs to mitigate transient system effects, yielding optimistic attacker throughput estimates that are used in subsequent sections to compute attacker-adjusted work factors.

5. Results

In this section, we instantiate the analytical framework of Section 4.5 for concrete mnemonic schemes. We quantify resistance against offline brute-force attacks by separating (i) the effective search space size, expressed via input entropy H (so $N = 2^H$), from (ii) empirically measured attacker throughput R on representative GPU hardware.

5.1. Effective Spatial Entropy Under Attacker Priors

We first establish the effective search space (N) for spatial inputs under attacker-prioritized priors. Using the spatial dictionary construction defined in Section 4.3, we evaluate the effective entropy H_{eff} for increasingly pessimistic spatial dictionaries.

Table 3 summarizes the results. The "Omnibus HPZ" dictionary represents a pessimistic upper bound on attacker knowledge, assuming the user selects a semantically salient region within a broad human-preferred zone (HPZ), operationalized as the union of habitable land, coastlines, rivers, and dense urban regions.

Table 3. Effective entropy of a single spatial cell under attacker-prioritized spatial dictionaries.

Dictionary	Area (km^2)	Entropy (bits)	Collapse (bits)
Global Nominal	5.10×10^8	45.69	0.00
Habitable HPZ	2.50×10^7	41.34	4.35
Global Coastline	1.16×10^5	33.59	12.10
Global Rivers	7.00×10^5	36.18	9.51
UNESCO Sites	3.77×10^1	22.00	23.69
Major Peaks	3.14×10^3	28.38	17.31
Omnibus HPZ	$\mathbf{2.58 \times 10^7}$	**41.38**	**4.30**
Urban (London)	1.57×10^3	27.38	18.31

Crucially, even under the pessimistic Omnibus prior, the effective entropy remains ≈ 41.4 bits, substantially higher than typical human-chosen passwords, which empirical studies frequently place below ≈ 20 effective bits under adaptive guessing models [1].

Radius Constraints on Multi-Point Selection

For scenarios involving multiple spatial points, the total spatial entropy is constrained by geographic clustering. Figure 1 visualizes the total effective entropy $H_{\text{spatial}}(n, r)$ as a function of the clustering radius r under the proximity model defined in Section 3.2.3. For small radii, additional points contribute limited independent entropy due to spatial correlation, whereas increasing dispersion increases the effective search space.

The figure further illustrates that for $n = 2$, the achievable entropy saturates below the 128-bit target even under maximal geographic dispersion, as the second point contributes at most the global single-cell entropy.

 doi:10.20944/preprints202603.0935.v1

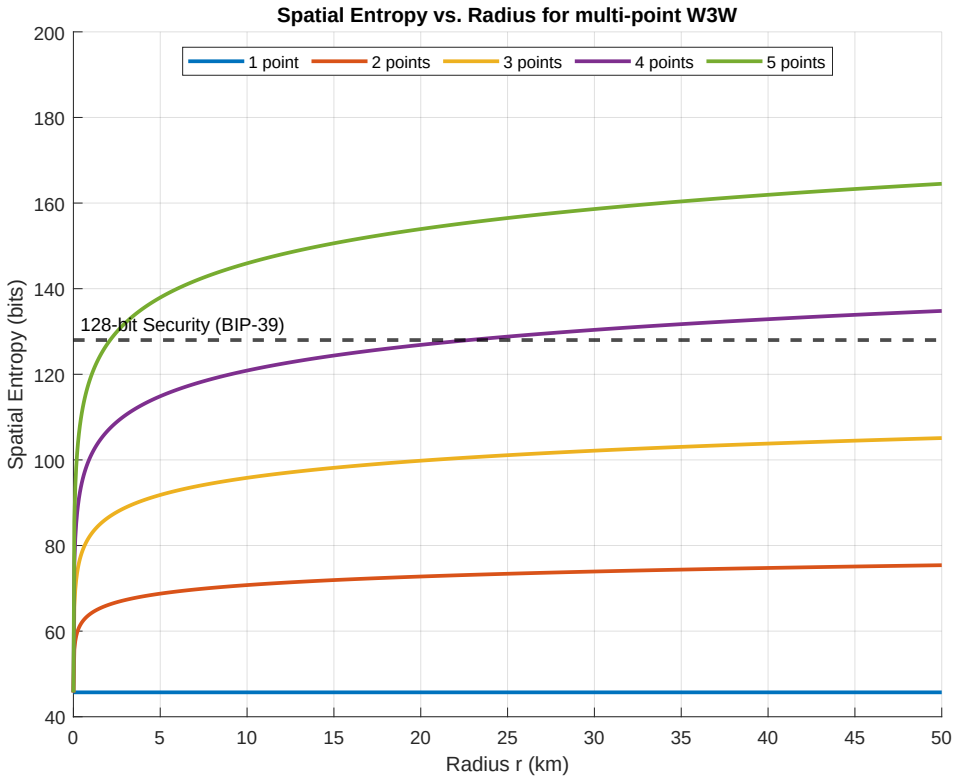

Figure 1. Spatial entropy $H_{\text{spatial}}(n, r)$ for multi-point selection under the proximity model.

Table 4 reports the raw entropy limits and the dispersion radius required to reach the 128-bit spatial entropy target defined in Section 4.5 for increasing numbers of selected points under the proximity model of Section 3.2.3.

Table 4. Raw radius and entropy calculations for multi-point spatial selection under the proximity model.

Points (n)	**Radius for 128-bit (r) (km)**	**Max Entropy (bits)**
1	∞	45.70
2	4.13×10^9	92.69
3	2,643.86	139.68
4	22.79	186.67
5	2.12	233.65

The entropy values in Table 4 are bounded by a maximum feasible dispersion radius corresponding to the Earth's circumference (approximately 20,000 km). For configurations in which the required radius exceeds this bound, the 128-bit entropy target defined in Section 4.5 cannot be achieved through geographic dispersion alone. This Earth-scale constraint defines a hard upper limit on the entropy contribution of additional spatial points under the proximity model of Section 3.2.3 and explains the saturation behavior observed for small n in Figure 1.

5.2. Key-Derivation Cost: Defender Latency vs. Attacker Throughput

To instantiate attacker-adjusted work factors (Section 4.5), we measured defender-side latency on CPU and attacker-side throughput on GPU. These measurements instantiate the attacker rates $R_{\text{GPU,hash}}$ and $R_{\text{GPU,Argon2}}(t, m, p)$ in equations 22 and 21.

 doi:10.20944/preprints202603.0935.v1

5.2.1. Defender-Side Key-Derivation Latency

A legitimate user (defender) performs a single key derivation event during vault decryption. To quantify the user-visible cost, we measured key-derivation latency on the CPU platform specified in Section 4.7.

For the unhardened baseline (SHA-256), computational overhead was negligible (< 0.01 ms). The BIP–39 baseline, which uses PBKDF2–HMAC–SHA512 with 2,048 iterations, exhibited a median latency of 3.57 ms, a delay that is effectively imperceptible to users. The full latency distribution is reported in Table 5.

Table 5. Empirical defender benchmark for BIP–39 PBKDF2–HMAC–SHA512 (CPU).

Iterations	Median (ms)	P90 (ms)	P95 (ms)	P99 (ms)
2,048	3.57	3.68	3.74	4.18

In contrast, GeoVault applies Argon2id to deliberately impose a hardware-bound delay on key derivation. Table 6 reports defender-side latency as a function of the Argon2id memory parameter m (with $t = 1$ and $p = 1$). Configurations up to 1,024 MiB remain within a few seconds and are suitable for interactive use, whereas extreme configurations at 16–32 GiB incur minute-scale delays consistent with archival or cold-storage usage.

Table 6. Argon2id CPU latency percentiles ($t = 1, p = 1$).

Memory setting	Median (ms)	P95 (ms)	P99 (ms)
64 MiB	140.7	147.5	152.1
256 MiB	621.7	643.9	658.2
1024 MiB	2,429.5	2,488.1	2,510.4
8192 MiB	20,538.2	20,877.7	21,145.0
16,384 MiB (16G)	46,943.7	51,201.4	54,822.1
32,768 MiB (32G)	103,986.2	119,436.5	124,105.8

5.2.2. Attacker-Side Guess Throughput

Attacker-side capabilities were benchmarked using Hashcat under the attacker model defined in Section 4.5. We distinguish between *compute-bound* primitives, which scale efficiently with GPU parallelism (SHA-256, PBKDF2), and *memory-bound* primitives, for which throughput is fundamentally constrained by available high-bandwidth memory (Argon2id).

BIP-39 Throughput (PBKDF2)

For linguistic mnemonics, an offline attacker evaluates candidate phrases using PBKDF2–HMAC–SHA512 as specified by the BIP-39 standard. Hashcat benchmarks were scaled to the full 2,048-iteration configuration. The measured sustained throughput is

$$R_{2048} \approx 6.40 \times 10^5 \text{ H/s}.$$

Table 7. Empirical attacker benchmark for BIP–39 PBKDF2–HMAC–SHA512 (GPU).

Runs (N)	Mean Time per Guess	Throughput R_{2048} (H/s)
30	1.56 μs	6.40×10^5

 doi:10.20944/preprints202603.0935.v1

Fast Hash Throughput (SHA-256)

For unhardened secrets verified using a single cryptographic hash, the attacker employs raw SHA-256. Benchmarks indicate a sustained throughput of approximately

$$R_{\text{GPU,hash}} \approx 1.3 \times 10^7 \text{ H/s},$$

with only minor degradation as input length increases. This rate serves as the baseline attacker capability for raw spatial and password-based secrets without computational hardening.

Argon2id Throughput (Memory-Hard Regime)

To characterize attacker performance against GeoVault-style hardened secrets, we measured Argon2id throughput across a wide range of memory costs m with fixed parameters $(t = 1, p = 1)$. Table 8 reports the resulting attacker throughput as a function of m.

Table 8. Empirical attacker Argon2id throughput as a function of memory cost m ($t = 1$).

Memory (MiB)	Median (H/s)	P95 (H/s)	P99 (H/s)
64	4985.50	4988.50	4988.50
128	2339.00	2341.00	2341.00
256	802.00	803.00	803.00
512	212.00	212.00	212.00
1024	53.66	53.80	53.80
2048	13.56	13.56	13.56
4096	2.70	2.70	2.70
8192	0.54	0.54	0.54
16384	0.11	0.11	0.11
32768	0.03	0.03	0.03

The results show a sharp, non-linear collapse in attacker throughput as the memory cost approaches and exceeds the GPU's available VRAM. At $m = 1024$ MiB, throughput is already reduced by more than five orders of magnitude relative to SHA-256. At 16–32 GiB, attacker throughput falls below one guess per second, indicating entry into a hardware-capacity regime in which GPU parallelism is no longer exploitable and brute-force attacks become effectively serialized.

5.3. Baseline Brute-Force Resistance Under Fast Hashing

We analyze the security of linguistic and spatial mnemonics in their *raw* state, assuming an attacker utilizes a standard fast-hash primitive (SHA-256) without memory-hard protection. In this setting, entropy is treated as a fixed input parameter, and security is evaluated exclusively via the attacker-adjusted work factor $\mathcal{W}$ defined in Equation 20. Following Section 4.5, we categorize security into three qualitative zones based on $\mathcal{W}$: **Insecure** ($< 10^{10}$ s), **Human-Scale Secure** (10^{10}–10^{32} s), and **Super Secure** ($\geq 10^{32}$ s).

5.3.1. Brute-Force Resistance of Linguistic Secrets

For typical human passwords, brute-force resistance under fast hashing is determined by the effective entropy implied by alphabet size and length. As established in Section 2.1, human-chosen passwords frequently collapse to approximately 20 bits of effective entropy under adaptive guessing models. Using the baseline attacker work factor formulation for fast hashing given in Equation 22, the expected time to compromise such a secret is

$$\mathcal{W}_{\text{password}} \approx \frac{2^{20}}{1.32 \times 10^7} \approx 0.08 \text{ seconds}. \tag{26}$$

 doi:10.20944/preprints202603.0935.v1

To transition from the **Insecure** zone into the **Super Secure (BIP-39)** regime without computational hardening, a user would need to memorize a truly random ASCII-94 string. The required string length L follows directly from the entropy target:

$$L = \left\lceil \frac{128}{\log_2(94)} \right\rceil = \left\lceil \frac{128}{6.55} \right\rceil = 20 \text{ characters.} \tag{27}$$

Such strings (e.g., `7&y#B9@q!x2$LpZ*5mW1`) substantially exceed typical human cognitive limits for reliable recall, illustrating the well-known usability barrier of high-entropy linguistic secrets.

5.3.2. Brute-Force Resistance of Spatial Secrets

In contrast, spatial mnemonics exhibit substantially higher baseline brute-force resistance due to their higher effective entropy. Applying the same baseline work factor model of Equation 22 to a single unhardened What3Words location ($n = 1$) yields an expected attack time of approximately 4.34×10^6 seconds (about 50 days). While this remains within the **Insecure** zone, it represents roughly a seven-order-of-magnitude improvement over typical human-chosen passwords.

Figure 2 visualizes this equivalence in brute-force resistance between spatial and linguistic secrets across the three defined security zones under fast hashing.

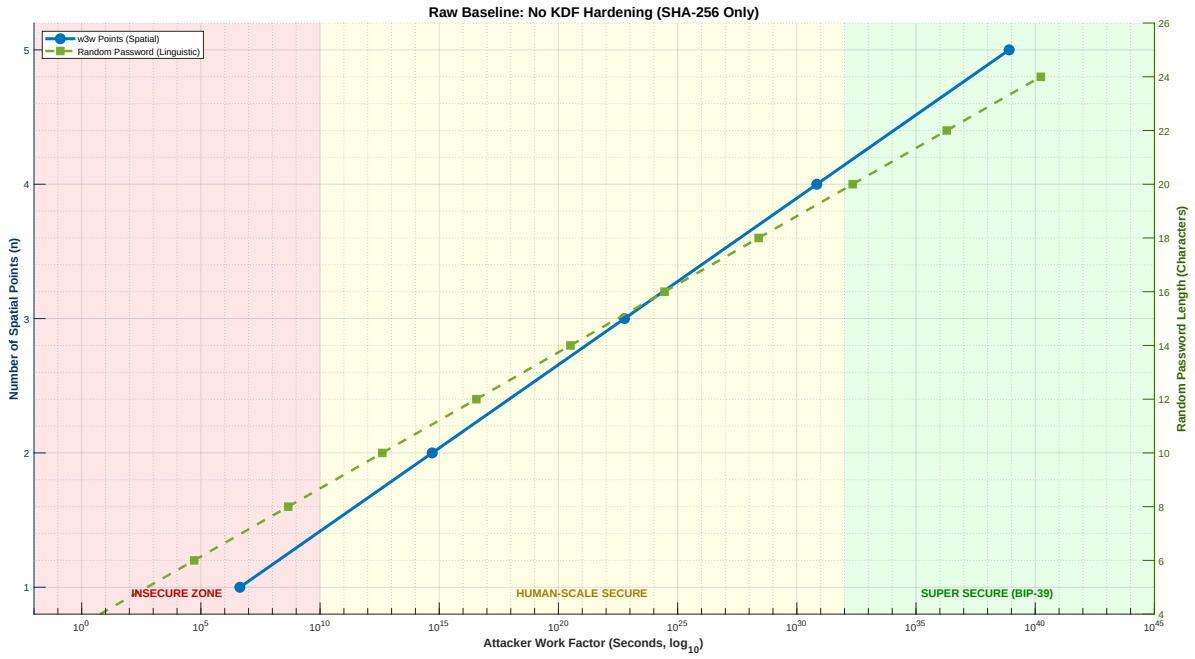

Figure 2. Raw baseline brute-force resistance under SHA-256 hashing: spatial points versus random password length. Shaded regions indicate the defined security zones: Insecure (red), Human-Scale Secure (yellow), and Super Secure/BIP-39 (green).

As the number of spatial points increases, the total search space expands sufficiently for spatial mnemonics to exceed the 128-bit security threshold even without computational hardening. Table 9 reports the geographic dispersion required for the baseline work factor of Equation 22 to enter the **Super Secure** regime under SHA-256.

 doi:10.20944/preprints202603.0935.v1

Table 9. Breakeven analysis: required spatial dispersion to achieve 128-bit brute-force resistance using SHA-256.

Points (n)	Radius (r)	Feasibility	Security Zone
1	Impossible	Exceeds Earth's area	Insecure
2	Impossible	Exceeds Earth's area	Insecure
3	5,396 km	Continental scale	Human-Scale
4	**36.4 km**	**City scale**	**Super Secure**
5	3.0 km	Neighborhood scale	Super Secure

These results highlight the fundamental advantage of spatial mnemonics under fast hashing: achieving super-secure brute-force resistance requires either memorizing approximately 20 random characters (cognitively impractical) or recalling a small number of geographically meaningful locations (cognitively feasible).

5.4. Brute-Force Resistance Under Memory-Hard KDFs

We analyze the impact of memory-hard key derivation on brute-force resistance by applying Argon2id across a wide range of memory costs, including extreme configurations (16 GiB and 32 GiB) that approach or exceed the physical VRAM limits of professional-grade GPU hardware. As in the previous subsection, entropy is treated as a fixed input parameter, and security is evaluated via the attacker-adjusted work factor $\mathcal{W}$ defined in Equation 20. Throughout this analysis, **BIP-39–equivalent security** is defined as a time-to-compromise exceeding 10^{32} seconds, consistent with Section 4.5.

Figure 3 illustrates the attacker-adjusted brute-force resistance for both linguistic and spatial secrets as a function of the Argon2id memory parameter. The figure instantiates Equation 21 using empirically measured Argon2id throughput values.

 doi:10.20944/preprints202603.0935.v1

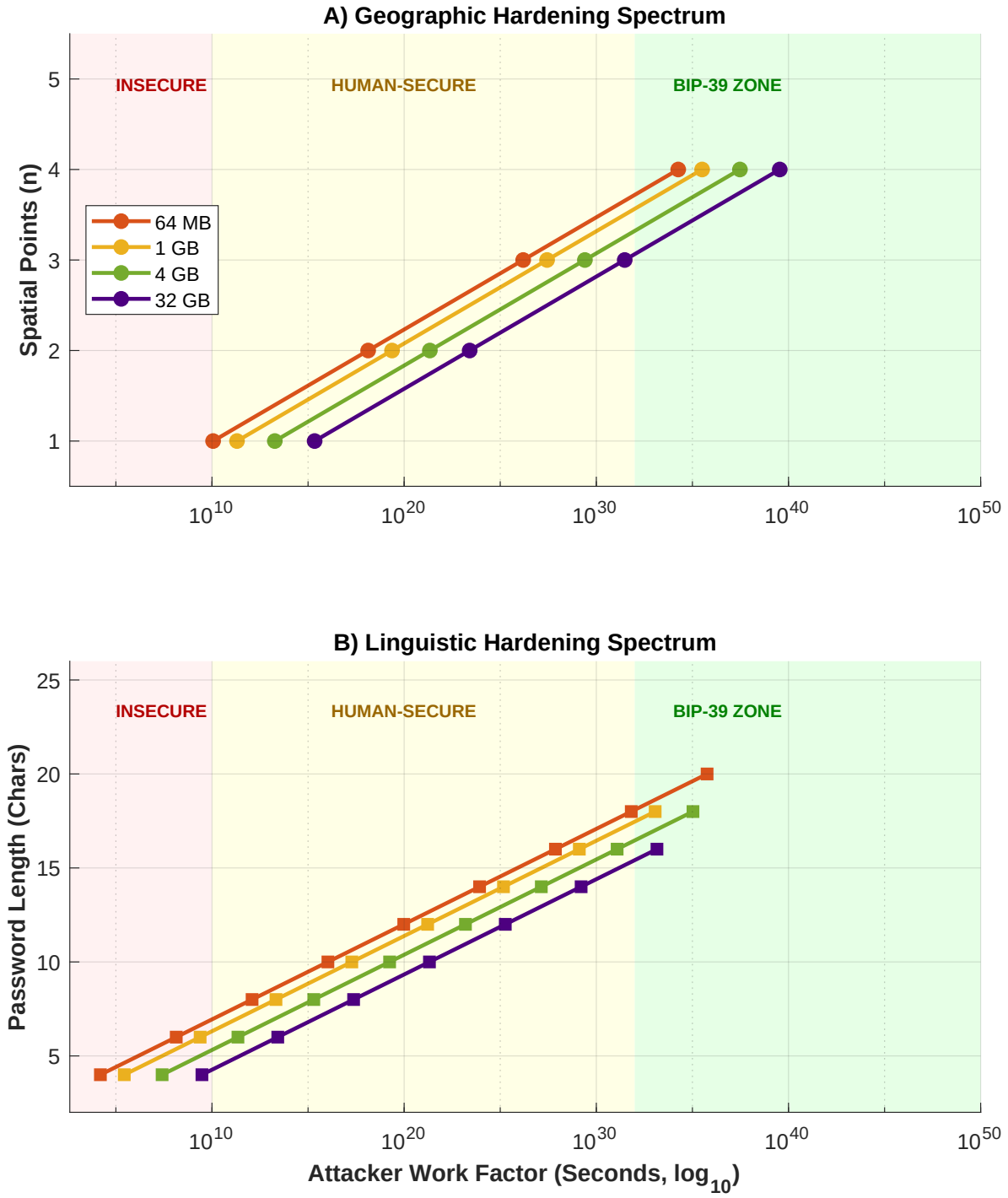

Figure 3. Attacker-adjusted brute-force resistance under Argon2id hardening for spatial and linguistic secrets, computed using Equation 21 across increasing memory costs.

Table 10 reports the corresponding time-to-compromise values for a representative linguistic secret (approximately 20 bits of effective entropy) and a single spatial point ($n = 1$, approximately 45.7 bits), derived from the attacker throughput measurements reported in Section 5.2. All values are computed using the Argon2id work factor model of Equation 21.

 doi:10.20944/preprints202603.0935.v1

Table 10. Estimated time-to-compromise under Argon2id hardening for linguistic and spatial secrets ($n = 1$). Attacker throughput R corresponds to empirical Argon2id benchmarks, and time-to-compromise is computed via Equation 21.

Memory (MiB)	**Attacker Rate (R)**	**Password Time (≈ 20 bits)**	**Spatial Time ($n = 1$) (≈ 45.7 bits)**
64	4986 H/s	3.5 min	360 years
128	2339 H/s	7.5 min	768 years
256	802 H/s	22.0 min	2,240 years
512	212 H/s	1.4 hours	8,480 years
1024	53.7 H/s	5.4 hours	33,400 years
2048	13.6 H/s	21.5 hours	132,000 years
4096	2.7 H/s	4.5 days	665,000 years
8192	0.54 H/s	22.5 days	3.3 million years
16384 (16G)	0.115 H/s	105 days	15.5 million years
32768 (32G)	0.026 H/s	1.2 years	68.6 million years

The results demonstrate a pronounced hardware-enforced non-linearity in attacker cost as the Argon2id memory parameter approaches the GPU's available VRAM. In particular, at 16–32 GiB, attacker throughput collapses due to the inability to parallelize memory-bound computations, effectively serializing brute-force attempts.

For linguistic secrets, even extreme 32 GiB hardening—while imposing substantial defender-side latency—remains vulnerable to compromise on the order of one year under the attacker model considered. In contrast, a single spatial point ($n = 1$) under the same hardening parameters achieves an expected time-to-compromise of approximately 6.9×10^7 years. Although this remains below the 10^{32} second BIP-39 threshold, it comfortably exceeds human and geological timescales, corresponding to the **Human-Scale+** regime.

These results confirm that combining spatial entropy with high-memory KDF hardening fundamentally alters the brute-force landscape by enforcing hardware-level constraints on the attacker, thereby breaking the traditional security–usability trade-off observed for linguistic secrets.

5.5. Brute-Force Resistance via Combined Spatial and Computational Hardening

We analyze the combined effect of spatial multi-point selection and memory-hard key derivation on brute-force resistance. As in the preceding subsections, spatial entropy is treated as a fixed input parameter (Section 3.2.3), and security is evaluated exclusively via the attacker-adjusted work factor $\mathcal{W}$ defined in Equation 20. Time-to-compromise values are computed using the empirical Argon2id work-factor formulation of Equation 21, with attacker throughput measurements taken from Section 5.2.

Table 11 reports defender-side latency and attacker time-to-compromise for spatial secrets with increasing numbers of points n, evaluated across Argon2id memory settings ranging from 64 MiB to 8 GiB. Spatial entropy values for each point count are taken from Table 4. Defender latency corresponds to CPU-side Argon2id execution, while attacker time-to-compromise reflects single-GPU brute-force attempts under the assumed attacker model.

 doi:10.20944/preprints202603.0935.v1

Table 11. Combined brute-force resistance for multi-point spatial secrets under Argon2id hardening. Cell shading indicates attacker work-factor regime as defined in Section 4.5.

Memory (MiB)	Defender Latency (s)	Time-to-Compromise (s)			
		$n = 1$	$n = 2$	$n = 3$	$n = 4$
64	0.14	1.1×10^{10}	2.3×10^{24}	4.1×10^{37}	7.5×10^{50}
128	0.29	2.3×10^{10}	4.9×10^{24}	8.6×10^{37}	1.6×10^{51}
256	0.62	6.7×10^{10}	1.4×10^{25}	2.5×10^{38}	4.6×10^{51}
512	1.22	2.5×10^{11}	5.3×10^{25}	9.6×10^{38}	1.8×10^{52}
1024	2.43	9.9×10^{11}	2.1×10^{26}	3.8×10^{39}	7.1×10^{52}
2048	4.92	3.9×10^{12}	8.3×10^{26}	1.5×10^{40}	2.8×10^{53}
4096	10.0	1.6×10^{13}	3.4×10^{27}	6.0×10^{40}	1.1×10^{54}
8192	20.5	6.5×10^{13}	1.4×10^{28}	2.4×10^{41}	4.4×10^{54}
16384	46.9	2.6×10^{14}	5.6×10^{28}	9.5×10^{41}	1.8×10^{55}
32768	104.0	1.0×10^{15}	2.2×10^{29}	3.8×10^{42}	7.2×10^{55}

Shading legend: Insecure ($< 10^{10}$ s), Human-Scale (10^{10}–10^{32} s), Super-Secure / BIP-39 ($\geq 10^{32}$ s).

The reported values show that increasing the number of spatial points rapidly dominates brute-force resistance, such that modest Argon2id memory settings already yield extremely large attacker work factors for $n \geq 3$. At the same time, defender-side latency increases only linearly with the Argon2id memory parameter, remaining below one second for memory settings up to 256 MiB and below five seconds at 2 GiB.

Relative to the BIP-39 equivalence criterion defined in Section 4.5 (time-to-compromise exceeding 10^{32} seconds), the table demonstrates that combined spatial and computational hardening achieves cryptographic-strength brute-force resistance without requiring extreme KDF parameters. This illustrates how spatial expansion of the search space reduces reliance on aggressive computational hardening, enabling strong security guarantees while preserving practical usability.

6. Discussion

The results presented in this work show that spatial memory, when combined with modern memory-hard key derivation functions, can meaningfully reshape the traditional security–usability trade-off in cryptographic key management. Rather than attempting to force users to reliably generate or retain abstract high-entropy secrets, GeoVault leverages a cognitive domain in which humans demonstrate strong and durable recall. The empirical and analytical results indicate that spatially anchored secrets exhibit a substantially higher effective entropy floor than human-chosen passwords under realistic offline attacker models, and that this advantage can be amplified through hardware-bound computational hardening.

6.1. Effective Entropy of Spatial Mnemonics

A central finding of this study is that spatial mnemonics provide a higher baseline of effective entropy than linguistic secrets selected by humans. As shown in Section 5.1, even pessimistic attacker models that restrict guesses to semantically meaningful spatial dictionaries, such as habitable land, coastlines, or urban regions, retain effective entropy levels on the order of 40–42 bits for a single spatial cell. This stands in contrast to human-chosen passwords and passphrases, which empirical studies consistently place well below 20 effective bits under realistic offline guessing strategies.

This higher entropy floor alters the practical boundary of what can be considered usable for human-memorable secrets. While linguistic recall rapidly collapses under selection bias, spatial recall preserves a larger and more evenly distributed search space. The results indicate that a small number of spatial selections, even under conservative attacker assumptions, provide substantially greater resistance to offline attacks than traditional password-based constructions, while remaining compatible with human memory capabilities.

 doi:10.20944/preprints202603.0935.v1

6.2. Impact of Memory-Hard Key Derivation

While spatial entropy alone is insufficient to guarantee cryptographic-strength security under all attacker assumptions, the application of memory-hard key derivation functions substantially increases resistance to brute-force attacks. The Argon2id benchmarks reported in Section 5.2 demonstrate a pronounced asymmetry between defender and attacker capabilities as memory costs increase. Attacker throughput decreases non-linearly once per-instance memory requirements approach available GPU memory, while remaining feasible, though intentionally costly, for legitimate users operating on commodity CPUs.

This asymmetry arises because Argon2id binds performance to memory bandwidth and capacity rather than raw compute throughput. Under the evaluated attacker model, increasing parallelism does not efficiently compensate for memory exhaustion, resulting in sharp reductions in effective guess rates. Consequently, GeoVault converts modest user-side latency into a disproportionate increase in attacker work, allowing spatially anchored secrets to achieve levels of brute-force resistance that approach or exceed established cryptographic security baselines when combined with sufficient effective entropy.

6.3. Security Gains from Multi-Point Spatial Selection

The combination of multiple spatial selections and computational hardening provides a robust mechanism for increasing attacker-adjusted work factors. As shown in Section 5.5, selecting a small number of spatial points within a realistic geographic dispersion radius, when combined with modest Argon2id memory settings, yields attacker-adjusted work factors that approach or exceed those associated with 128-bit security under the evaluated configurations. Importantly, these gains can be achieved with sub-second to second-scale user-side latency, making the approach suitable for interactive key derivation and recovery.

This result highlights the flexibility of the GeoVault design. Users may trade spatial complexity, expressed through the number and dispersion of selected locations, against computational hardness, expressed through key-derivation parameters, according to their threat model and usability constraints. Within the evaluated parameter ranges, these trade-offs remain above commonly accepted security thresholds, decoupling strong offline resistance from extreme computational cost.

6.4. Threat Model and Limitations

The security properties of GeoVault are evaluated under a strong offline attacker model in which the adversary has unrestricted access to high-end GPU hardware but no auxiliary side-channel information beyond attacker-prioritized spatial dictionaries. The system does not rely on secrecy of the geospatial encoding scheme or obscurity of the spatial domain. Security derives solely from the effective entropy of user-selected locations and the enforced computational cost of key derivation.

The analysis does not account for targeted social engineering, coercion, or leakage of user-specific spatial preferences. As with all mnemonic-based systems, knowledge of a user's habits, routines, or personal history could reduce the effective search space if spatial choices are overly predictable. Additionally, this work assumes accurate user recall within the resolution of the spatial encoding scheme. While tolerance buffers are incorporated into the entropy analysis, future implementations may benefit from explicit error-correction or fuzzy matching mechanisms to accommodate imperfect recall without expanding the attacker's advantage.

6.5. Implications for Human-Centered Key Management

The findings of this study suggest that spatial memory constitutes a viable and underutilized resource for cryptographic key management. By aligning security mechanisms with a cognitive domain in which humans demonstrate strong long-term retention, GeoVault shows that strong offline resistance need not be achieved at the expense of usability. Unlike traditional brainwallets, which rely on abstract linguistic recall and suffer severe entropy collapse, spatially anchored mnemonics provide a higher baseline of effective entropy under realistic attacker models.

 doi:10.20944/preprints202603.0935.v1

More broadly, GeoVault illustrates how hardware-enforced asymmetry and cognitive ergonomics can be combined to design secure systems that remain usable under real-world constraints. This perspective opens new directions for key management, particularly in scenarios where physical storage of secrets is undesirable or impractical, and where long-term memorability is a primary concern.

7. Conclusions

This work introduced *GeoVault*, a spatially anchored key management framework that leverages human spatial memory in combination with memory-hard cryptographic primitives to address persistent security–usability limitations in mnemonic-based secret storage. By grounding key derivation in spatial locations rather than purely linguistic inputs, GeoVault exploits a cognitive capability that is well-supported by empirical research, while remaining compatible with standard offline cryptographic threat models.

Through information-theoretic analysis, we showed that spatially anchored secrets exhibit a substantially higher effective entropy floor than human-chosen passwords under realistic attacker priors. Empirical benchmarking further demonstrated that when this spatial entropy is combined with a memory-hard key derivation function such as Argon2id, brute-force resistance becomes strongly influenced by hardware constraints. In particular, increasing memory costs toward GPU VRAM limits results in a pronounced, non-linear reduction in attacker throughput, creating a favorable asymmetry between legitimate users and offline adversaries.

The results indicate that GeoVault can achieve attacker-adjusted work factors that approach or, under certain configurations, exceed those associated with the 128-bit security baseline of BIP–39 mnemonics. These protection levels can be attained either through modest multi-point spatial selection with low computational overhead, or through single-point spatial selection combined with higher memory costs, allowing users to balance cognitive load and computational expense according to their threat model and usage context.

Beyond the specific instantiation evaluated in this paper, the GeoVault design highlights a broader principle for key management: aligning cryptographic mechanisms with inherent human cognitive strengths can improve practical security without relying on purely abstract secrets. Future work will explore extensions to three-dimensional and indoor spatial environments, error-tolerant spatial matching mechanisms, multi-user or shared-vault scenarios, and formal analysis under stronger adversarial knowledge assumptions.

In summary, GeoVault demonstrates that spatial memory can serve as a viable cryptographic input when combined with hardware-enforced computational asymmetry, offering a promising direction for the design of secure and usable human-centered key management systems.

Funding: Slovenian Research Agency: P2-0270; Ministry of Higher Education , Science and Technology of the Republic of Slovenia: 100-15-0510

References

1. Bonneau, J. Statistical metrics for individual password strength (Transcript of discussion). *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2012**, *7622 LNCS*, 87–95. https://doi.org/10.1007/978-3-642-35694-0_11.
2. Vasek, M.; Bonneau, J.; Castellucci, R.; Keith, C.; Moore, T. The bitcoin brain drain: Examining the use and abuse of bitcoin brain wallets. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2017**, *9603 LNCS*, 609–618. https://doi.org/10.1007/978-3-662-54970-4_36.
3. Kuo, C.; Romanosky, S.; Cranor, L.F. Human selection of mnemonic phrase-based passwords. *ACM International Conference Proceeding Series* **2006**, *149*, 67–78. https://doi.org/10.1145/1143120.1143129.
4. Yang, W.; Li, N.; Chowdhury, O.; Xiong, A.; Proctor, R.W. An empirical study of mnemonic sentence-based password generation strategies. *Proceedings of the ACM Conference on Computer and Communications Security* **2016**, *24-28-October-2016*, 1216–1229. https://doi.org/10.1145/2976749.2978346.

 doi:10.20944/preprints202603.0935.v1

5. Pals, F.F.; Tolboom, J.L.; Suhre, C.J.; van Geert, P.L. Memorisation methods in science education: tactics to improve the teaching and learning practice. *International Journal of Science Education* **2018**, *40*, 227–241. https://doi.org/10.1080/09500693.2017.1407885;PAGE:STRING:ARTICLE/CHAPTER.
6. Akaygun, S.; Jones, L.L. Words or Pictures: A comparison of written and pictorial explanations of physical and chemical equilibria. *International Journal of Science Education* **2014**, *36*, 783–807. https://doi.org/10.1080/09500693.2013.828361.
7. Montello, D. A New Framework for Understanding the Acquisition of Spatial Knowledge in Large-Scale Environments **1998**.
8. McNamara, T.P. How Are the Locations of Objects in the Environment Represented in Memory? *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* **2003**, *2685*, 174–191. https://doi.org/10.1007/3-540-45004-1_11.
9. McNamara, T.P. Spatial memory: Properties and organization. *Handbook of spatial cognition.* **2012**, pp. 173–190. https://doi.org/10.1037/13936-010.
10. McNaughton, B.L.; Chen, L.L.; Markus, E.J. "Dead reckoning," landmark learning, and the sense of direction: A neurophysiological and computational hypothesis. *Journal of Cognitive Neuroscience* **1991**, *3*. https://doi.org/10.1162/JOCN.1991.3.2.190.
11. Bauer, M.I.; Johnson-Laird, P.N. How Diagrams Can Improve Reasoning. *Psychological Science* **1993**, *4*, 372–378. https://doi.org/10.1111/j.1467-9280.1993.tb00584.x.
12. Tari, F.; Ozok, A.A.; Holden, S.H. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. *ACM International Conference Proceeding Series* **2006**, *149*, 56–66. https://doi.org/10.1145/1143120.1143128.
13. Golla, M.; Detering, D.; Dürmuth, M. EmojiAuth: Quantifying the Security of Emoji-based Authentication **2017**. https://doi.org/10.14722/usec.2017.23024.
14. Biryukov, A.; Dinu, D.; Khovratovich, D. Argon2: New generation of memory-hard functions for password hashing and other applications. In Proceedings of the Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016, 2016, pp. 292–302. https://doi.org/10.1109/EuroSP.2016.31.
15. Wetzels, J. Open Sesame: The Password Hashing Competition and Argon2. *IACR Cryptology ePrint Archive* **2016**, [1602.03097].
16. Choe, J.; Moreshet, T.; Bahar, R.I.; Herlihy, M. Attacking Memory-Hard scrypt with Near-Data-Processing. *ACM International Conference Proceeding Series* **2019**, pp. 33–37. https://doi.org/10.1145/3357526.3357570.
17. Das, P.; Faust, S.; Loss, J. A formal treatment of deterministic wallets. *Proceedings of the ACM Conference on Computer and Communications Security* **2019**, pp. 651–668. https://doi.org/10.1145/3319535.3354236.
18. Bonneau, J. The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In Proceedings of the Proceedings - IEEE Symposium on Security and Privacy, 2012, pp. 538–552. https://doi.org/10.1109/SP.2012.49.
19. Adams, A.; Sasse, M.A. Users Are Not The Enemy. *Communications of the ACM* **1999**, *42*, 40–46. https://doi.org/10.1145/322796.322806.
20. Sasse, M.A.; Brostoff, S.; Weirich, D. Transforming the 'weakest link' - A human/computer interaction approach to usable and effective security. *BT Technology Journal* **2001**, *19*, 122–131. https://doi.org/10.1023/A:1011902718709.
21. Kävrestad, J.; Nohlberg, M. Assisting Users to Create Stronger Passwords Using ContextBased MicroTraining. *IFIP Advances in Information and Communication Technology* **2020**, *580 IFIP*, 95–108. https://doi.org/10.1007/978-3-030-58201-2_7.
22. Di Luzio, A.; Francati, D.; Ateniese, G. Arcula: A secure hierarchical deterministic wallet for multi-asset blockchains. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2020**, *12579 LNCS*, 323–343, [1906.05919]. https://doi.org/10.1007/978-3-030-65411-5_16.
23. Gutoski, G.; Stebila, D. Hierarchical deterministic bitcoin wallets that tolerate key leakage. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2015**, *8975*, 497–504. https://doi.org/10.1007/978-3-662-47854-7_31.
24. Guri, M. Beatcoin: Leaking private keys from air-gapped cryptocurrency wallets. In Proceedings of the Proceedings - IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/GreenCom/CPSCom/SmartData/Blockchain/CIT 2018, 2018, pp. 1308–1316, [1804.08714]. https://doi.org/10.1109/Cybermatics_2018.2018.00227.

 doi:10.20944/preprints202603.0935.v1

25. Montello, D.R. A New Framework for Understanding the Acquisition of Spatial Knowledge in Large-Scale Environments. *Spatial And Temporal Reasoning In Geographic Information Systems* **2023**, pp. 143–154. https://doi.org/10.1093/oso/9780195103427.003.0011.
26. Yates, F.A. Selected works: Volume III: Art of memory. *Selected Works: Volume III: Art of Memory* **2013**, *2*, 1–400. https://doi.org/10.4324/9781315010960/ART-MEMORY-YATES/RIGHTS-AND-PERMISSIONS.
27. Konkle, T.; Brady, T.F.; Alvarez, G.A.; Oliva, A. Scene memory is more detailed than you think: The role of categories in visual long-term memory. *Psychological Science* **2010**, *21*, 1551–1556. https://doi.org/10.1177/0956797610385359.
28. Lukavský, J.; Děchtěrenko, F. Visual properties and memorising scenes: Effects of image-space sparseness and uniformity. *Attention, Perception, and Psychophysics* **2017**, *79*, 2044–2054. https://doi.org/10.3758/S13414-017-1375-9/FIGURES/4.
29. Garden, S.; Cornoldi, C.; Logie, R.H. Visuo-spatial working memory in navigation. *Applied Cognitive Psychology* **2002**, *16*, 35–50. https://doi.org/10.1002/acp.746.
30. Jiang, W.; Stefanakis, E. What3Words Geocoding Extensions. *Journal of Geovisualization and Spatial Analysis* **2018**, *2*. https://doi.org/10.1007/s41651-018-0014-x.
31. Lee, J. GIS-based geocoding methods for area-based addresses and 3D addresses in urban areas. *Environment and Planning B: Planning and Design* **2009**, *36*, 86–106. https://doi.org/10.1068/b31169.
32. Arthur, R. A critical analysis of the What3Words geocoding algorithm. *PLoS ONE* **2023**, *18*, e0292491, [2308.16025]. https://doi.org/10.1371/journal.pone.0292491.
33. Mai, G.; Janowicz, K.; Hu, Y.; Gao, S.; Yan, B.; Zhu, R.; Cai, L.; Lao, N. A review of location encoding for GeoAI: methods and applications. *International Journal of Geographical Information Science* **2022**, *36*, 639–673, [2111.04006]. https://doi.org/10.1080/13658816.2021.2004602.
34. Goldberg, D.W.; Ballard, M.; Boyd, J.H.; Mullan, N.; Garfield, C.; Rosman, D.; Ferrante, A.M.; Semmens, J.B. An evaluation framework for comparing geocoding systems. *International Journal of Health Geographics* **2013**, *12*. https://doi.org/10.1186/1476-072X-12-50.
35. Zandbergen, P.A. A comparison of address point, parcel and street geocoding techniques. *Computers, Environment and Urban Systems* **2008**, *32*, 214–232. https://doi.org/10.1016/j.compenvurbsys.2007.11.006.
36. Efremova, J.; Endres, I.; Vidas, I.; Melnik, O. A geo-tagging framework for address extraction from web pages. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2018**, *10933 LNAI*, 288–295. https://doi.org/10.1007/978-3-319-95786-9_22.
37. Boneh, D.; Bonneau, J.; Bünz, B.; Fisch, B. Verifiable delay functions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2018**, *10991 LNCS*, 757–788. https://doi.org/10.1007/978-3-319-96884-1_25.
38. Wesolowski, B. Efficient verifiable delay functions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2019**, *11478 LNCS*, 379–407. https://doi.org/10.1007/978-3-030-17659-4_13.
39. Döttling, N.; Garg, S.; Malavolta, G.; Vasudevan, P.N. Tight verifiable delay functions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2020**, *12238 LNCS*, 65–84. https://doi.org/10.1007/978-3-030-57990-6_4.
40. Döttling, N.; Lai, R.W.; Malavolta, G. Incremental proofs of sequential work. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2019**, *11477 LNCS*, 292–323. https://doi.org/10.1007/978-3-030-17656-3_11.
41. Mahmoody, M.; Moran, T.; Vadhan, S. Time-lock puzzles in the random oracle model. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2011**, *6841 LNCS*, 39–50. https://doi.org/10.1007/978-3-642-22792-9_3.
42. Cheon, J.H.; Hopper, N.; Kim, Y.; Osipkov, I. Provably secure timed-release public key encryption. *ACM Transactions on Information and System Security* **2008**, *11*, 4. https://doi.org/10.1145/1330332.1330336.
43. Cathalo, J.; Libert, B.; Quisquater, J.J. Efficient and non-interactive timed-release encryption. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2005**, *3783 LNCS*, 291–303. https://doi.org/10.1007/11602897_25.
44. Choi, G.; Vaudenay, S. Timed-Release Encryption with Master Time Bound Key. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2020**, *11897 LNCS*, 167–179. https://doi.org/10.1007/978-3-030-39303-8_13.

 doi:10.20944/preprints202603.0935.v1

45. Cohen, B.; Pietrzak, K. Simple proofs of sequential work. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2018**, *10821 LNCS*, 451–467. https://doi.org/10.1007/978-3-319-78375-8_15.
46. Abusalah, H.; Kamath, C.; Klein, K.; Pietrzak, K.; Walter, M. Reversible proofs of sequential work. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2019**, *11477 LNCS*, 277–291. https://doi.org/10.1007/978-3-030-17656-3_10.
47. Nguyen, H.A.D.; Yu, J.; Lebdeh, M.A.; Taouil, M.; Hamdioui, S. A computation-in-memory accelerator based on resistive devices. *ACM International Conference Proceeding Series* **2019**, pp. 19–32. https://doi.org/10.1145/3357526.3357554.
48. Shannon, C.E. A Mathematical Theory of Communication. *Bell System Technical Journal* **1948**, *27*, 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.
49. Choi, H.; Choi, S.J.; Seo, S.C. Parallel Implementation of Lightweight Secure Hash Algorithm on CPU and GPU Environments. *Electronics (Switzerland)* **2024**, *13*, 896. https://doi.org/10.3390/electronics13050896.
50. Qiu, W.; Gong, Z.; Guo, Y.; Liu, B.; Tang, X.; Yuan, Y. GPU-based high performance password recovery technique for hash functions, 2016.
51. Dürmuth, M.; Kranz, T. On password guessing with GPUs and FPGAs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2015**, *9393 LNCS*, 19–38. https://doi.org/10.1007/978-3-319-24192-0_2.
52. Bonneau, J.; Herley, C.; Van Oorschot, P.C.; Stajano, F. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. *Proceedings - IEEE Symposium on Security and Privacy* **2012**, pp. 553–567. https://doi.org/10.1109/SP.2012.44.
53. Venter, O.; et al. Sixteen years of change in the global terrestrial human footprint. *Nature Communications* **2016**, *7*, 12558.