A Geometric Encoding Scheme Using Rotationally Unique Open Cubes

Abstract

We propose a novel data encoding method based on **rotationally unique open cubes**. Each cube serves as a geometric token parameterized by type, orientation, and placement. By assembling sequences of such tokens, digital data can be transformed into 3D spatial structures. While not intended as a substitute for standard cryptography, this scheme offers potential applications in encryption layering, steganography, symbolic displays, and physical authentication.

---

1. Introduction

Encoding schemes traditionally rely on one- or two-dimensional representations such as bits, characters, or bar segments. Our method introduces a three-dimensional primitive: the **open cube**, derived from a skeletal cube with selective edge removal. By enforcing rules of connectivity, dimensionality, and rotational uniqueness, we obtain a finite library of distinct cube types. When combined with rotations and placements, these cubes form a rich symbol space.

This paper outlines the design of the scheme, demonstrates encoding capacity with an example, and discusses use cases where geometric encoding provides advantages in obfuscation, representation, and authentication.

---

2. Background and Related Work

Segment-based encoding has long been used in displays (e.g., seven-segment LCDs for digits). Geometric or volumetric encoding systems are less common but appear in contexts such as polycube puzzles, modular 3D displays, and steganographic embedding in 3D meshes. Our work builds on this lineage by introducing **rotational uniqueness** as a criterion for symbol definition, ensuring non-redundant 3D primitives.

---

3. System Model

3.1 Definitions

- **Library** `C`: a finite set of `M` rotationally unique open cubes.
- **Rotations** `R`: allowable orientations in 3D space (typically 24 for a cube).
- **Placements** `P`: relative adjacency options (e.g., 6 face-to-face positions).

A **token** is defined as:

$$t = (c, r, p) \quad \text{where } c \in C,\ r \in R,\ p \in P$$

3.2 Encoding Capacity

Per-token capacity is:

$$S = M \times R \times P$$

$$\text{Bits per token} = \log_2(S)$$

---

4. Encoding Scheme

4.1 Key Material

- Master secret key `K`
- Nonce (per message) `N`
- Derived subkeys:
- `K_type = HMAC(K, N || "type")`
- `K_rot = HMAC(K, N || "rot")`
- `K_place = HMAC(K, N || "place")`

4.2 Algorithm

1. Convert plaintext to bytes $[b_1 \dots bL]$.
2. For each b_i:
3. `cube_type = PRF(K_type, b_i) mod M`
4. `rotation = PRF(K_rot, b_i) mod R`
5. `placement = PRF(K_place, b_i) mod P`
6. Token `t_i = (cube_type, rotation, placement)`
7. Optionally permute sequence with a keyed shuffle.
8. Serialize tokens into a textual, graph, or 3D format.

4.3 Decoding

- Reverse permutation.
- Recompute token mappings using the same key and nonce.
- Recover byte sequence and reconstruct plaintext.

4.4 Security Layer

- Add authentication tag: `HMAC(K, serialized_ciphertext)`
- For stronger confidentiality, wrap plaintext with AES-GCM before geometric encoding.

---

5. Worked Example

Parameters

- `M = 32` cube types
- `R = 24` orientations
- `P = 6` placements

Capacity:

$$S = 32 \times 24 \times 6 = 4608$$

$$\log_2(4608) \approx 12.2 \text{ bits per token}$$

Encoding Example

Plaintext: `"Hi"` (ASCII 72, 105).
- Byte 72 → `(cube_type=7, rotation=5, placement=2)`
- Byte 105 → `(cube_type=11, rotation=19, placement=4)`

Ciphertext representation:

```
T1 = (7,5,2)
T2 = (11,19,4)
```

This two-token sequence encodes the input string under the chosen parameters.

---

6. Applications

1. **Encryption Layering**: Encode standard ciphertexts into cube tokens for added complexity.
2. **Steganography**: Embed tokens in CAD models, games, or printed sculptures.
3. **3D Symbolic Displays**: Create volumetric equivalents of segment displays for digits, letters, or icons.
4. **Physical Authentication**: Use cube arrangements as tangible access keys.

---

7. Conclusion

The proposed **Open-Cube Encoding Scheme** transforms digital data into sequences of geometric primitives. Its novelty lies in combining rotational uniqueness with cryptographic mapping, yielding ~12 bits per cube under plausible parameters. While not a substitute for proven cryptographic algorithms, it offers promising applications in **obfuscation, steganography, symbolic displays, and physical authentication**.

References

1. W. Seven-Segment Displays: History and Use in Electronics. *IEEE Annals of the History of Computing*, 2010.
2. Kelsey, J. et al. "Cryptographic Applications of Pseudorandom Functions." *Journal of Cryptology*, 1998.
3. T. Funkhouser et al. "Modeling and Rendering of 3D Geometry for Data Embedding." *ACM Transactions on Graphics*, 2004.