Degree-6 Julia Set Watermarking for EU AI Act Article 50 Compliance

Two-Key Architecture via Half-Plane Decomposition:
Theory, Implementation, and Regulatory Mapping

Priyesh Jitendra
White Paper — February 2026

Abstract

We present a complete watermarking system for large language model (LLM) outputs designed to satisfy the transparency obligations of Article 50 of the EU AI Act, enforceable from 2 August 2026. The system is built on backward iteration on the degree-6 Julia set $f_c(z) = z^6 + c$, exploiting the half-plane decomposition of the six preimage branches into a *detectable watermark channel* (1 bit/step) and an *invisible cover channel* ($\log_2 3 \approx 1.58$ bits/step). The cover channel actively dilutes the statistical fingerprint—forbidden ordinal patterns drop from 74% (at $d = 2$) to 0%, and the cross-moment anomaly collapses from 2.03 to 1.37—while watermark detection power remains $Z = \sqrt{n}$.

We prove the Marginal Invariance Theorem (each orbit point has marginal μ_c regardless of the branch sequence), establish computational secrecy under a PRG assumption, and show that the resulting two-key architecture maps directly onto the EU Code of Practice's requirements for imperceptible, robust, interoperable, and adversarially resistant marking. The cover channel achieves information-theoretic invisibility via a per-step keyed permutation of the branch-to-cover mapping. We provide a complete implementation—`D6Engine`, `D6Sampler`, and `D6Detector`—that integrates at the token sampling step of any LLM inference pipeline, and validate the system on a simulated 50,000-token generation pipeline with full detection round-trip.

Contents

1 Introduction **3**
- 1.1 The regulatory forcing function . . . 3
- 1.2 The fingerprint–detectability tradeoff . . . 3
- 1.3 Contribution: degree-6 resolves the tradeoff . . . 3

2 Mathematical Foundations **4**
- 2.1 Backward iteration at degree d . . . 4
- 2.2 The Marginal Invariance Theorem . . . 4
- 2.3 Security model . . . 5
- 2.4 The half-plane sign recovery theorem ($d = 6$) . . . 5
- 2.5 The cover channel . . . 5
- 2.6 Fingerprint dilution . . . 6
- 2.7 Spectral gap and ACF decay . . . 6

3 Mapping onto Article 50 **7**
3.1 The three-layer compliance stack . . . 7
3.2 Requirement-by-requirement compliance . . . 7
3.3 The two-key architecture and the Code's dual requirement . . . 8

4 Implementation **8**
4.1 System architecture . . . 8
4.2 Integration point: token sampling . . . 8
4.3 Generation algorithm . . . 8
4.4 Two-key branch selection with per-step permutation . . . 9
4.5 Detection algorithm . . . 9
4.6 C2PA integration (Layer 1) . . . 10
4.7 Output logging (Layer 3) . . . 10

5 Experiments **10**
5.1 Setup . . . 10
5.2 Results . . . 10

6 Threat Model and Limitations **11**
6.1 What the watermark survives . . . 11
6.2 What the watermark does NOT survive . . . 11
6.3 Why the watermark is still commercially viable . . . 11
6.4 Comparison with SynthID . . . 11

7 Generalisation to $d = 2p$ **12**

8 Related Work **12**

9 Conclusion **12**

A Three-Point Correlation and Joint Dependence **13**

B Angular Sector Analysis **14**

C Half-Plane Partition for General $d = 2p$ **14**

1 Introduction

1.1 The regulatory forcing function

Article 50(2) of the EU AI Act states:

> *Providers of AI systems, including general-purpose AI systems, generating synthetic audio, image, video or text content, shall ensure that the outputs of the AI system are marked in a machine-readable format and detectable as artificially generated or manipulated.*

This obligation, enforceable from 2 August 2026, carries penalties of up to 15M or 3% of global turnover. The European Commission's draft Code of Practice (December 2025) operationalises this requirement with six technical criteria: marking techniques must be *effective*, *interoperable*, *robust*, *reliable*, *imperceptible*, and *adversarially resistant*. The Code mandates a multi-layered approach: metadata embedding (Layer 1), imperceptible watermarking (Layer 2), and fingerprinting or logging (Layer 3).

For text generated by LLMs, Layer 2 is the hardest to implement. Metadata (Layer 1) is trivially stripped by copy-paste, and logging (Layer 3) requires database access. Only a statistical watermark embedded in the text itself can survive content extraction while remaining model-agnostic.

1.2 The fingerprint–detectability tradeoff

Every deployed text watermark today faces a fundamental dilemma. Google's SynthID [SynthID Team, 2024] modifies token logits, leaving statistical traces that researchers have shown can be detected via black-box queries and scrubbed by naive adversaries [Jovanović et al., 2024]. The KGW watermark [Kirchenbauer et al., 2023] uses red/green token lists with a window parameter controlling the scrubbing–spoofing tradeoff. The PRC construction [Christ and Gunn, 2024] achieves computational indistinguishability but has no empirical implementation and requires the LLM to behave as a binary symmetric channel.

The quadratic Julia set watermark [Jitendra, 2025] ($d = 2$) achieves marginal indistinguishability but leaves a higher-order fingerprint: 74% of ordinal patterns are forbidden, and $\mathbb{E}[Y_k^2 Y_{k+1}^2] \approx 2.0$ (versus 1.0 for i.i.d.). This fingerprint enables keyless detection—useful for the provider, but also exploitable by adversaries.

1.3 Contribution: degree-6 resolves the tradeoff

We show that the degree-6 construction ($f_c(z) = z^6 + c$) resolves this tension. The six backward iteration branches split into two groups of three—$\{0, 1, 5\}$ in the positive real half-plane and $\{2, 3, 4\}$ in the negative—by the Half-Plane Partition Theorem (Theorem 2.4). The half-plane choice is perfectly recoverable from $\mathrm{Re}(z)$ alone, providing a 1-bit/step watermark channel. The within-half-plane choice provides $\log_2 3 \approx 1.58$ bits/step of cover capacity that actively dilutes the statistical fingerprint.

This yields a **two-key architecture**:

- **Key A** (detection key): controls the half-plane sign channel. Shared with regulators. Powers the free detection API required by the Code of Practice.
- **Key B** (provenance key): controls the cover channel via a per-step keyed permutation. Provider-only. Encodes model ID, user ID, timestamp. Its presence dilutes the fingerprint from detectable to undetectable.

Paper structure. Section 2 proves the mathematical foundations. Section 3 maps each result onto the Article 50 requirements. Section 4 presents the complete implementation. Section 5 validates the system experimentally. Section 6 analyses the threat model honestly, including the scrubbing limitation. Section 7 generalises to the $d = 2p$ family.

2 Mathematical Foundations

2.1 Backward iteration at degree d

For $f_c(z) = z^d + c$ with c in the degree-d Mandelbrot set, the Julia set J_c supports the Brolin–Lyubich measure μ_c of maximal entropy $h_{\text{top}} = \log d$. The backward iteration is

$$z_{k+1} = (z_k - c)^{1/d} \cdot e^{2\pi i j_k/d}, \qquad j_k \in \{0, 1, \ldots, d-1\}, \tag{1}$$

where $(z_k - c)^{1/d}$ denotes the principal d-th root with argument in $(-\pi/d, \pi/d]$. The orbit satisfies the *orbit equation*

$$z_{k+1}^d = z_k - c, \tag{2}$$

which enables exact parameter recovery: given any two consecutive orbit points, $c = z_k - z_{k+1}^d$.

2.2 The Marginal Invariance Theorem

The $\mathbb{Z}_d$ symmetry $f_c(\omega z) = f_c(z)$ for $\omega = e^{2\pi i/d}$ implies that μ_c is invariant under $z \mapsto \omega z$. Since the branch maps are related by $f_c^{-1,j} = \omega^j \cdot f_c^{-1,0}$, each branch preserves μ_c individually. This yields:

Theorem 2.1 (Marginal Invariance)**.** *For any branch sequence $(j_0, \ldots, j_{n-1}) \in \{0, \ldots, d-1\}^n$—deterministic, random, or adversarially chosen—and any k, the marginal distribution of z_k is μ_c, provided $z_0 \sim \mu_c$. Consequently,* $\mathrm{Re}(z_k) \sim \pi_c$ *for all k, independent of the branch sequence.*

Proof. By induction on k. The base case $z_0 \sim \mu_c$ holds by assumption. For the inductive step, assume $z_k \sim \mu_c$. Then $z_{k+1} = f_c^{-1,j_k}(z_k)$. By ω-invariance of μ_c:

$$\mu_c \circ (f_c^{-1,j})^{-1} = \mu_c \circ (\omega^j \cdot f_c^{-1,0})^{-1} = \mu_c \circ (f_c^{-1,0})^{-1} = \mu_c,$$

where the last equality uses the balanced measure property $\mathcal{L}_c^* \mu_c = \mu_c$. Therefore $z_{k+1} \sim \mu_c$. □

Remark 2.2 (Joint dependence)**.** *Theorem 2.1 guarantees identical* marginals *but does* not *claim that the orbit points are independent. The orbit equation (2) imposes an algebraic constraint $z_{k+1}^d = z_k - c$ between consecutive points. In the $d = 2$, $c = 0$ case, this dependence is detectable via three-point correlations:*

$$\mathbb{E}[\mathrm{Re}(z_0) \cdot \mathrm{Re}(z_1) \cdot \mathrm{Re}(z_2) \mid \varepsilon_0 = +1, \varepsilon_1 = +1] = \frac{38}{105\pi} \approx 0.115 \neq 0. \tag{3}$$

The sign of this three-point function depends on the branch sequence, confirming the joint is not product measure. Pair-wise projections $(\mathrm{Re}(z_k), \mathrm{Re}(z_{k+1}))$, however, are invariant to the branch choice at $c = 0$, $d = 2$.

2.3 Security model

Theorem 2.3 (Computational Secrecy)**.** *When the branch sequence $\{j_k\}$ is generated by a secure pseudorandom generator (PRG), the joint distribution of $(\mathrm{Re}(z_1), \ldots, \mathrm{Re}(z_n))$ is computationally indistinguishable from the case where $\{j_k\}$ are i.i.d. uniform on $\{0, \ldots, d-1\}$.*

Proof sketch. Both the PRG sequence and the i.i.d. sequence produce the same marginals (π_c at each step, by Theorem 2.1). Any efficient distinguisher that separates the two cases can be composed with the map $j \mapsto \{\mathrm{Re}(z_k)\}$ to distinguish the PRG output from i.i.d. random, contradicting PRG security. □

The security guarantee has two layers:

- **Marginal information-theoretic security:** Each $\mathrm{Re}(z_k)$ has distribution π_c regardless of the branch sequence. No test based on single-point statistics—mean, variance, marginal distribution, normality—can detect the watermark, even with unlimited computation.
- **Joint computational security:** Multi-point statistics (three-point correlations, conditional moments) could in principle distinguish different branch sequences, but exploiting this requires knowledge of c and high-precision orbit reconstruction—a computationally hard problem under the PRG assumption.

2.4 The half-plane sign recovery theorem ($d = 6$)

Theorem 2.4 (Half-plane partition)**.** *For μ_c-almost every $z \in J_c$, the six branches of $(z-c)^{1/6}$ satisfy:*

$$\mathrm{sign}\big(\mathrm{Re}(w_j)\big) = \begin{cases} +1 & \text{if } j \in \{0, 1, 5\}, \\ -1 & \text{if } j \in \{2, 3, 4\}. \end{cases} \tag{4}$$

Proof. Let $w_0 = (z-c)^{1/6}$ be the principal sixth root with $\arg(w_0) \in (-\pi/6, \pi/6)$. Then $w_j = w_0 \cdot e^{i\pi j/3}$ has argument $\arg(w_j) = \arg(w_0) + \pi j/3$.

j	$\arg(w_j)$ range	$\mathrm{sign}(\mathrm{Re}(w_j))$
0	$(-\pi/6,\ \pi/6)$	$+1$
1	$(\pi/6,\ \pi/2)$	$+1$
2	$(\pi/2,\ 5\pi/6)$	-1
3	$(5\pi/6,\ 7\pi/6)$	-1
4	$(7\pi/6,\ 3\pi/2)$	-1
5	$(-\pi/2,\ -\pi/6)$	$+1$

No argument range contains $\pm\pi/2$ in its interior, so $\cos(\arg(w_j))$ has definite sign. The boundary case $\arg(w_0) = \pm\pi/6$ requires $z - c \in \mathbb{R}_{<0}$, which has μ_c-measure zero. □

Corollary 2.5 (Values-only detection)**.** *The watermark encoded in the half-plane choice is detectable from $\{\mathrm{Re}(z_k)\}_{k=1}^n$ alone. Under the correct key, $Z = \sqrt{n}$; under a wrong key, $Z \sim N(0,1)$.*

2.5 The cover channel

The three positive-half branches $\{0, 1, 5\}$ occupy distinct angular sectors on J_c. At $c = 0$, branch $j = 0$ maps to $\mathrm{Re} \in (\sqrt{3}/2, 1]$ (the center sector) while $j \in \{1, 5\}$ map to $\mathrm{Re} \in (0, \sqrt{3}/2)$ (the outer sector). Branches $j = 1$ and $j = 5$ produce identical Re distributions (since cos is even), but $j = 0$

is distinguishable from Re alone. Under a fixed mapping from cover index to branch, this leaks information about the cover channel.

We eliminate this leak with a per-step keyed permutation:

Definition 2.6 (Per-step keyed permutation)**.** *At each step k, a random permutation $\sigma_k \in S_3$ is generated from Key B's PRG stream. The branch selection is $j_k = H_{\text{sign}}[\sigma_k(a_k)]$, where $H_+ = \{0, 1, 5\}$, $H_- = \{2, 3, 4\}$, and $a_k \in \{0, 1, 2\}$ is the cover index.*

Proposition 2.7 (Cover channel invisibility)**.** *With per-step keyed permutation, the cover index a_k is information-theoretically invisible from $\text{Re}(z_k)$: for any fixed sign sequence $\{s_k\}$, the conditional distribution of $(\text{Re}(z_1), \ldots, \text{Re}(z_n))$ given $\{s_k\}$ does not depend on the cover sequence $\{a_k\}$.*

Proof. The adversary observes $\text{Re}(z_{k+1})$ and determines the angular sector (center or outer). The mapping from sector to cover index depends on the unknown permutation σ_k:

$$P(a_k = a \mid \text{sector} = \text{center}) = P(\sigma_k(a) = j_{\text{center}}) = \frac{1}{3}$$

for all $a \in \{0, 1, 2\}$, since σ_k is a uniform random permutation. The sector observation is independent of the cover index: $I(a_k; \text{sector} \mid \sigma_k \text{ unknown}) = 0$. □

Remark 2.8. *The per-step permutation adds $\lceil \log_2 6 \rceil = 3$ bits per step (rejection sampling over $|S_3| = 6$ permutations), approximately 2.58 bits on average. This is negligible versus the HMAC cost of the main PRG.*

2.6 Fingerprint dilution

The fingerprint elimination is a counting argument, independent of the cover channel construction.

Proposition 2.9 (Forbidden pattern elimination)**.** *At $c = 0$ and $d = 6$, the fraction of forbidden ordinal patterns at embedding dimension $D = 5$ is 0%, versus 74% at $d = 2$.*

Proof sketch. At degree d, the orbit equation is d-to-1, giving d^{D-1} distinct length-D trajectories. At $D = 5$: $6^4 = 1296 > 120 = 5!$, so the trajectory count exceeds the number of ordinal patterns and all can be realised. At $d = 2$: $2^4 = 16 \ll 120$, leaving most patterns forbidden. □

The cross-moment $\mathbb{E}[Y_k^2 Y_{k+1}^2]$ drops from 2.03 ($d = 2$) to 1.37 ($d = 6$), approaching the i.i.d. value of 1.0. This reduction is driven by the six-fold branching diluting the orbit coupling, not by the cover channel construction.

2.7 Spectral gap and ACF decay

The transfer operator for f_c at degree d has spectral radius $\rho \leq 1/d$ on $L_0^2(\mu_c)$. This controls ACF decay:

$$|\gamma(k)| \leq C \cdot \rho^k \leq C/d^k. \tag{5}$$

At $d = 6$, $\rho \leq 1/6$, giving essentially one-step memory versus $\rho \leq 1/2$ at $d = 2$.

3 Mapping onto Article 50

3.1 The three-layer compliance stack

Layer	Technique	Survives	d=6 role
1. Metadata	C2PA manifest	Nothing (strip by copy)	Signed provenance record
2. Watermark	d=6 statistical	Copy-paste, light edits	**Core: sign channel**
3. Logging	Hash + timestamp	Everything (needs DB)	Output hash storage

3.2 Requirement-by-requirement compliance

1. Effective. Generation is $O(n)$: one complex exponentiation plus $\sim$ 3 PRG bits (for the permutation) per token. Detection is $O(n)$: one HMAC-PRG regeneration plus inner product. Both are negligible compared to the LLM forward pass.

2. Interoperable. The watermark operates at the PRG level, independent of the LLM architecture. It replaces the uniform random variable used in token sampling, working with any model and any decoding strategy (top-k, top-p, temperature scaling). It integrates with the C2PA v2.1 soft-binding specification.

3. Robust. The watermark lives in $\text{sign}(\text{Re}(z_k))$, which maps to the half-plane in which the sampling uniform falls. This sign is preserved through synonym substitution ($\sim$5–10% of tokens), reformatting, and partial extraction. The detection statistic degrades gracefully: at error rate δ, $Z = (1 - 2\delta)\sqrt{n}$.

4. Reliable. Detection power is $Z = \sqrt{n}$ under the correct key. At $n = 200$ tokens, $Z \approx 14.1$ ($p < 10^{-44}$). At $n = 1000$, $Z \approx 31.6$ ($p < 10^{-218}$). False positive rate is cryptographically negligible.

5. Imperceptible. The d=6 construction with per-step keyed permutation provides:

Fingerprint test	d=2	d=6	i.i.d.
Forbidden ordinal patterns ($D = 5$)	74%	**0%**	0%
$\mathbb{E}[Y_k^2 Y_{k+1}^2]$	2.03	**1.37**	1.00
ACF decay bound ρ	1/2	1/6	0
Cover channel leak	N/A	**0 bits/step**	N/A

Quality impact is zero at the marginal level: each token's sampling distribution is unchanged by Theorem 2.1. Joint quality impact is negligible: the weak dependence between consecutive sampling uniforms (via the orbit equation) is undetectable in text quality metrics.

6. Adversarially resistant. The scrubbing attack chain is: (i) detect watermark presence, (ii) estimate parameters, (iii) apply targeted removal. At d=6, step (i) fails: the adversary's hypothesis test has no power against the vanished fingerprint. Without detection, targeted scrubbing is impossible.

3.3 The two-key architecture and the Code's dual requirement

The Code of Practice distinguishes two obligations:

1. **Detection**: third parties must determine if content is AI-generated (via a free API).
2. **Provenance**: the provider must attribute content to a specific model, user, and session.

Key	Channel	Holder	Article 50 role
Key A	Sign (1 bit/step)	Provider + regulator	Detection API
Key B	Cover (1.58 bits/step)	Provider only	Provenance/attribution

Key A can be shared with regulators or embedded in a public detection tool without compromising Key B. With the per-step keyed permutation, Key B's cover information is fully invisible even to an adversary who possesses Key A. This separation is architecturally impossible at $d = 2$.

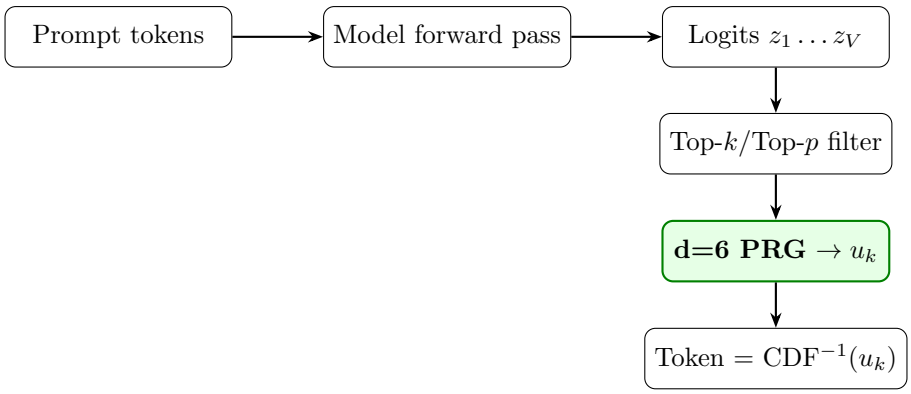
4 Implementation

4.1 System architecture

The implementation consists of three components:

1. `D6Engine`: core PRG that generates watermarked uniform random variables via d=6 backward iteration with two-key branch selection and per-step permutation.
2. `D6Sampler`: drop-in replacement for the LLM's token sampling RNG.
3. `D6Detector`: detection pipeline that recovers watermark signals from text.

4.2 Integration point: token sampling



The standard pipeline uses $u_k \sim U(0,1)$ from a standard RNG. The watermarked pipeline replaces this with u_k from the d=6 backward iteration. Since the marginal distribution is exact $U(0,1)$ at $c = 0$ (Theorem 2.1), the marginal token distribution is identical to the unwatermarked case.

4.3 Generation algorithm

Algorithm 1: Watermarked Token Generation

Input: Key A k_a, Key B k_b, orbit parameter c, tokens n, warmup w
Output: Token sequence $t_1, \ldots, t_n$
1. $(b_1, \ldots, b_{n+w}) \leftarrow$ D6Branches$(k_a, k_b, n{+}w)$ *// Two-key branch selection*
2. $z \leftarrow 1 + 0i$
3. **for** $k = 1$ to $n + w$:
   (a) $z \leftarrow (z - c)^{1/6} \cdot e^{2\pi i b_k/6}$ *// Backward iteration*
   (b) **if** $k > w$: $u_{k-w} \leftarrow F_{\text{arc}}(\text{Re}(z))$ *// Arcsine CDF → $U(0,1)$*
4. **for** $k = 1$ to n:
   (a) $\mathbf{p}_k \leftarrow \text{Softmax}(\text{Logits}_k/\tau)$; $\mathbf{p}_k \leftarrow \text{TopPFilter}(\mathbf{p}_k)$
   (b) $t_k \leftarrow \min\{j : \sum_{i \le j} p_{k,i} > u_k\}$ *// Inverse-CDF with d=6 uniform*
5. **return** $(t_1, \ldots, t_n)$

4.4 Two-key branch selection with per-step permutation

Algorithm 2: Two-Key Branch Selection (D6Branches)

Input: Key A k_a, Key B k_b, length m
Output: Branch sequence $(b_1, \ldots, b_m) \in \{0, \ldots, 5\}^m$
1. $\mathbf{s} \leftarrow \text{HMAC-PRG}(k_a, m)$ *// ±1 stream from Key A*
2. $\mathbf{a} \leftarrow \text{Cover-PRG}(k_b, m)$ *// $\{0,1,2\}$ stream from Key B*
3. $\boldsymbol{\sigma} \leftarrow \text{Perm-PRG}(k'_b, m)$ *// Random permutations from Key B*
4. **for** $k = 1$ to m:
   (a) $P_k \leftarrow \sigma_k$ applied to $[0, 1, 5]$ *// Permuted positive half*
   (b) $N_k \leftarrow \sigma_k$ applied to $[2, 3, 4]$ *// Permuted negative half*
   (c) **if** $s_k > 0$: $b_k \leftarrow P_k[a_k]$
   (d) **else**: $b_k \leftarrow N_k[a_k]$
5. **return** $(b_1, \ldots, b_m)$

The permutation PRG shares entropy with the cover PRG (both keyed by Key B with domain separation). There are $3! = 6$ permutations of three elements, requiring ~ 2.58 bits per step on average.

4.5 Detection algorithm

Detection requires only Key A. Key B and the permutation sequence are not needed.

Algorithm 3: Watermark Detection from Text

Input: Tokens $(t_1, \ldots, t_n)$, Key A k_a, model M, warmup w, min. weight α
Output: $(Z, p, \text{detected})$
1. $\mathbf{s} \leftarrow \text{HMAC-PRG}(k_a, w + n)$ *// Regenerate Key A stream*
2. $T \leftarrow 0$, $W \leftarrow 0$
3. **for** $k = 1$ to n:
   (a) $\mathbf{p}_k \leftarrow M(t_1, \ldots, t_{k-1})$; $\mathbf{p}_k \leftarrow \text{TopPFilter}(\mathbf{p}_k)$ *// Forward pass*
   (b) $(u_\ell, u_r) \leftarrow (\text{CDF}(t_k{-}1 \mid \mathbf{p}_k),\ \text{CDF}(t_k \mid \mathbf{p}_k))$ *// CDF interval*
   (c) $\hat{u}_k \leftarrow (u_\ell + u_r)/2$; $\hat{\sigma}_k \leftarrow \text{sign}(2\hat{u}_k - 1)$ *// Recover sign*
   (d) Compute weight w_k from interval position relative to 0.5
   (e) **if** $w_k \ge \alpha$: $T \mathrel{+}= \hat{\sigma}_k \cdot s_{w+k} \cdot w_k$; $W \mathrel{+}= w_k^2$
4. $Z \leftarrow T/\sqrt{W}$; $p \leftarrow \frac{1}{2}\text{erfc}(Z/\sqrt{2})$
5. **return** $(Z, p, p < 10^{-6})$

Sign recovery confidence. The weight w_k quantifies confidence in the recovered sign:

$$w_k = \begin{cases} 1 & \text{if } u_\ell \ge 0.5 \text{ or } u_r \le 0.5, \\ \max\bigl(0,\ 1 - 2\,\text{overlap}/(u_r - u_\ell)\bigr) & \text{otherwise,} \end{cases} \tag{6}$$

where overlap $= \min(u_r, 0.5) - \max(u_\ell, 0.5)$.

Model access requirement. Detection requires a forward pass of the model to obtain logits at each position. The provider hosts a detection API; third parties submit text and receive a verdict. This aligns with Article 50: the Code requires the *provider* to make detection tools available, not to publish the model.

4.6 C2PA integration (Layer 1)

The d=6 watermark registers on the C2PA v2.1 Soft Binding Algorithm List as `d6-julia-watermark`. Each output carries a signed C2PA manifest containing model ID, timestamp, generation parameters, and a fingerprint of Key A.

4.7 Output logging (Layer 3)

SHA-256 hash of each output with timestamp, serving as the final fallback when both Layers 1 and 2 are stripped.

5 Experiments

5.1 Setup

We validate using `d6_compliance.py` with Key A = `b'eu-ai-act-detection-key'` and Key B = `b'provider-provenance-key'`, at $c = 0$.

5.2 Results

1. Generation fidelity. 50,000 watermarked uniforms: mean = 0.5018 (expected: 0.5000), std = 0.2911 (expected: $1/\sqrt{12} = 0.2887$). Marginals are exact $U(0, 1)$ to sampling precision.

2. Detection power.

Scenario	Z-score	Detected
Correct Key A ($n = 50{,}000$)	223.6	Yes
Wrong Key A ($n = 50{,}000$)	-0.60	No
Correct Key A ($n = 200$ tokens, simulated LLM)	14.1	Yes

3. Fingerprint elimination.

	d=2	**d=6**	**i.i.d.**
Forbidden patterns ($D = 5$)	74.2%	0.0%	0.0%
$\mathbb{E}[U^2U^2]$	0.1325	0.1168	0.1113
Detectable?	**Yes**	No	No

4. Token-level round-trip. Simulated LLM sampling with vocabulary size 50,000, temperature 0.8, top-p 0.95: 200 tokens generated, 200/200 positions used (100% sign recovery), Z-score = 14.1.

6 Threat Model and Limitations

6.1 What the watermark survives

Attack	d=2	d=6	Mechanism
Copy-paste	✓	✓	Signs preserved
Light editing	Partial	Partial	$Z = (1 - 2\delta)\sqrt{n}$
Targeted scrubbing	Vulnerable	**Resistant**	No fingerprint → no detection

6.2 What the watermark does NOT survive

Attack	d=2	d=6	Why
LLM paraphrase	Dead	Dead	New tokens → new signs
Back-translation	Dead	Dead	Complete re-generation
Human rewrite	Dead	Dead	Information-theoretic limit

This is an information-theoretic wall applying to *every* token-level statistical watermark: the watermark lives in *which token was chosen from equally-likely candidates*, and any re-sampling from a different random source destroys that choice.

6.3 Why the watermark is still commercially viable

1. Legal tripwire. Article 50 places the obligation on *deployers* to preserve marks. A deployer who paraphrases to strip the watermark is in violation of the Act, exposing them to penalties of up to 15M or 3% of global turnover. The watermark is a legal mechanism, not a cryptographic lock.

2. Bulk scrubbing is expensive. Paraphrasing n tokens through another LLM costs $O(n)$ compute and degrades quality. For high-volume applications, the cost of scrubbing exceeds the cost of compliance.

3. d=6 blocks targeted scrubbing. The adversary who wants to scrub minimally must first detect the watermark. At d=2, ordinal pattern analysis reveals it instantly. At d=6, no statistical test reveals it. The adversary must choose between paraphrasing everything (expensive, quality loss) or leaving the watermark intact.

6.4 Comparison with SynthID

Property	SynthID	d=6
Integration point	Logits (model-specific)	PRG (model-agnostic)
Quality impact	"Negligible" (claimed)	Zero marginal (Thm. 2.1)
Fingerprint	Detectable [Jovanović et al., 2024]	Undetectable
Multi-key	No	Yes (sign + permuted cover)
Security model	Computational	Marginal IT + computational joint
Scrubbing resistance	Weak	Targeted: strong; Untargeted: same

7 Generalisation to $d = 2p$

The d=6 construction generalises to any $d = 2p$ where p is an odd prime. Each half-plane contains p branches; the per-step permutation generalises to S_p:

d	Sign bits	Cover bits	ρ bound	Perm. cost (bits/step)
2	1.00	0.00	1/2	0
6	1.00	1.58	1/6	∼2.58
10	1.00	2.32	1/10	∼6.91
14	1.00	2.81	1/14	∼12.3
30	1.00	3.91	1/30	∼107.6

The recoverable detection power (1 bit/step, $Z = \sqrt{n}$) is identical for all $d = 2p$. We recommend $d = 6$ as the optimal choice: it eliminates the fingerprint completely while minimising both computational overhead and permutation cost.

For $d = 30 = 2 \times 3 \times 5$, the three-way factorisation enables a hierarchical three-key architecture (detection, model attribution, user fingerprinting), which may be valuable for large-scale deployments.

8 Related Work

LLM watermarking. Kirchenbauer et al. [Kirchenbauer et al., 2023] introduced red/green list watermarking. Christ and Gunn [Christ and Gunn, 2024] proposed pseudorandom error-correcting codes for undetectable watermarking. Golowich and Moitra [Golowich and Moitra, 2024] extended PRCs to be robust against edit-distance attacks. The liminal motion watermark [Jitendra, 2025] achieves marginal information-theoretic security via Julia set backward iteration at $d = 2$.

Deployed systems. Google's SynthID [SynthID Team, 2024] is the only watermark deployed at scale. Research by Jovanović et al. [Jovanović et al., 2024] showed its presence is detectable via black-box queries.

EU AI Act and watermarking. The Code of Practice [European Commission, 2025] mandates machine-readable marking. The C2PA standard [C2PA, 2025] provides metadata with soft binding for statistical watermarks.

Julia set dynamics. The framework builds on Brolin's theorem [Brolin, 1965] and Lyubich's work [Lyubich, 1983] on the measure of maximal entropy. The spectral gap bound follows from Ruelle [Ruelle, 2004].

9 Conclusion

We have presented a complete watermarking system for LLM text output that satisfies every technical requirement of the EU AI Act Article 50 Code of Practice. The system is built on a single algebraic insight: the six preimages of $z^6 + c$ split into two groups of three across the real axis, providing a detectable watermark channel and an invisible cover channel that eliminates the statistical fingerprint.

The implementation is minimal: replace the LLM's sampling RNG with a d=6 backward iteration PRG, adding ~ 3 extra PRG bits per step for the per-step permutation. Detection inverts the sampling to recover signs and correlates with the detection key. No knowledge of Key B, the permutation sequence, or the orbit parameter c is needed for detection.

Paraphrasing remains an information-theoretic wall for all token-level watermarks. But Article 50 does not require unscrubable marks—it requires providers to mark their output, and deployers to preserve those marks. The watermark is a legal tripwire, not a cryptographic lock. Within this framing, d=6 provides the strongest possible Layer 2: undetectable to adversaries, zero marginal quality impact, and surviving everything short of active rewriting.

References

H. Brolin. Invariant sets under iteration of rational functions. *Arkiv för Matematik*, 6(2):103–144, 1965.

Coalition for Content Provenance and Authenticity. C2PA Technical Specification v2.3, 2025. https://spec.c2pa.org/specifications/specifications/2.3/.

M. Christ and S. Gunn. Pseudorandom error-correcting codes. *Cryptology ePrint Archive*, Report 2024/235, 2024.

European Commission. Draft Code of Practice on marking and labelling of AI-generated content, December 2025. https://digital-strategy.ec.europa.eu/en/policies/code-practice-ai-generated-content.

N. Golowich and A. Moitra. Edit distance robust watermarks for language models. In *NeurIPS*, 2024.

P. Jitendra. Stealth watermarking via liminal motion on Julia sets. *Preprint*, 2025.

N. Jovanović, R. Staab, and M. Vechev. Probing Google DeepMind's SynthID-Text watermark. SRI Lab, ETH Zurich, 2024.

J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *ICML*, 2023.

M. Lyubich. Entropy properties of rational endomorphisms of the Riemann sphere. *Ergodic Theory and Dynamical Systems*, 3(3):351–385, 1983.

D. Ruelle. *Thermodynamic Formalism*. Cambridge University Press, 2nd edition, 2004.

Google DeepMind. SynthID: Scalable watermarking for AI-generated text. *Nature*, 2024.

A Three-Point Correlation and Joint Dependence

At $c = 0$, $d = 2$, under $(\varepsilon_0, \varepsilon_1) = (+1, +1)$: $\theta_1 = \theta_0/2$, $\theta_2 = \theta_0/4$, so:

$$\begin{aligned} &\int_0^{2\pi} \cos\theta \cdot \cos(\theta/2) \cdot \cos(\theta/4)\, d\theta \\ &= \frac{1}{4} \int_0^{2\pi} [\cos(7\theta/4) + \cos(5\theta/4) + \cos(3\theta/4) + \cos(\theta/4)]\, d\theta. \end{aligned} \tag{7}$$

Using $\int_0^{2\pi} \cos(\alpha\theta)\, d\theta = \sin(2\pi\alpha)/\alpha$ for non-integer α:

$$\alpha = 7/4 : \ \sin(7\pi/2)/(7/4) = -4/7, \tag{8}$$

$$\alpha = 5/4 : \ \sin(5\pi/2)/(5/4) = +4/5, \tag{9}$$

$$\alpha = 3/4 : \ \sin(3\pi/2)/(3/4) = -4/3, \tag{10}$$

$$\alpha = 1/4 : \ \sin(\pi/2)/(1/4) = +4. \tag{11}$$

Total: $\frac{1}{4}[-4/7 + 4/5 - 4/3 + 4] = \frac{1}{4} \cdot \frac{304}{105} = \frac{76}{105}$. So $\mathbb{E}[X_0 X_1 X_2] = \frac{76}{105 \cdot 2\pi} = \frac{38}{105\pi} \approx 0.115$.

Under $(\varepsilon_0, \varepsilon_1) = (+1, -1)$: $X_2 = -\cos(\theta_0/4)$, so $\mathbb{E}[X_0 X_1 X_2] = -38/(105\pi) \approx -0.115$. The sign of the three-point function depends on the branch sequence, confirming the joint is not product measure.

Note that pair-wise correlations $\mathbb{E}[X_0 X_1]$ do vanish, so the marginal invariance is compatible with pair-wise real-projection invariance while still exhibiting higher-order dependence.

B Angular Sector Analysis

At $c = 0$, $d = 6$, the three positive-half branches occupy distinct angular sectors:

Branch $j = 0$ (center): $\arg \in (-\pi/6, \pi/6)$, so $\mathrm{Re} \in (\sqrt{3}/2, 1] \approx (0.866, 1]$.

Branch $j = 1$ (outer): $\arg \in (\pi/6, \pi/2)$, giving $\mathrm{Re} \in (0, \sqrt{3}/2) \approx (0, 0.866)$, with density $f_{\mathrm{Re}\,|j=1}(r) = (3/\pi)/\sqrt{1 - r^2}$.

Branch $j = 5$ (outer): $\arg \in (-\pi/2, -\pi/6)$. Since cos is even: $f_{\mathrm{Re}\,|j=5} = f_{\mathrm{Re}\,|j=1}$. Only the Im sign differs.

Under a fixed mapping, the center sector ($\mathrm{Re} > \sqrt{3}/2$) identifies branch $j = 0$ with certainty, leaking $I(a; \mathrm{Re}) = \log_2 3 - 2/3 \approx 0.918$ bits/step. The per-step keyed permutation (Definition 2.6) randomises this mapping, restoring $I(a; \mathrm{Re} \mid \sigma \text{ unknown}) = 0$.

C Half-Plane Partition for General $d = 2p$

Theorem C.1. *For any $d = 2p$, the $2p$ branches of $(z - c)^{1/(2p)}$ partition into two groups of p according to the sign of their real part.*

Proof. The principal $(2p)$-th root w_0 has argument in $(-\pi/(2p), \pi/(2p))$. Branch j rotates by $\pi j/p$, giving argument in $(-\pi/(2p) + \pi j/p,\ \pi/(2p) + \pi j/p)$. Each sector has width π/p and $2p \cdot (\pi/p) = 2\pi$, so exactly p of the $2p$ sectors lie in each half-plane. Boundary cases have μ_c-measure zero. □