Synthesis Tamper-evident Attestation and Molecular Provenance (STAMP): Cryptographic Molecular Barcoding for DNA Synthesizers

Anonymous Authors[1]

Abstract

As AI lowers the design barrier for dangerous biology and benchtop DNA synthesizers proliferate, the biothreat bottleneck shifts from design to physical synthesis. We introduce STAMP (Synthesis Tamper-evident Attestation and Molecular Provenance): a 120-base cryptographic barcode that an HSM-equipped synthesizer stamps into a non-coding region of every DNA it produces, attesting that the sequence originated from an untampered, registered synthesizer and was not significantly modified post-synthesis. STAMP combines HSM-anchored cryptographic attestation with a novel content-aware landmark map that enables forensic reconstruction of post-synthesis modifications. Empirically, the encoder succeeds on 100% of $N = 2000$ random plasmids, and the privacy-preserving landmark signal alone detects $\geq 95\%$ of kilobase-scale insertions. We prove that no system of this class can fully defeat a determined attacker; instead, STAMP is a cost imposer and evidence generator that converts every viable attack into a forensically suspicious artifact or a supply-chain-visible event.[1]

1. Introduction

AI-assisted biological design is approaching a threshold where the design barrier for novel proteins, optimized variants, and screening-evasion strategies is meaningfully lowered (National Academies of Sciences, Engineering, and Medicine, 2025; Brent & McKelvey, 2025; Wheeler, 2025). As that barrier falls, the bottleneck shifts to physical synthesis. Affordable benchtop synthesizers capable of printing viral-length DNA may be available within two years under aggressive projections (Anguzu, 2025), creating a decentralized synthesis ecosystem fundamentally harder to police than centralized providers. Frameworks such as the IGSC Harmonized Screen Protocol (International Gene Synthesis Consortium (IGSC), 2024) have pushed for improved synthesizer-side screening, typically backed by a tamper-resistant Hardware Security Module (HSM) (Langenkamp, 2024). This approach faces two technical limits. First, benchtop synthesizers are fluid-handling machines built on well-understood principles, so full anti-tamper hardware is implausible — a determined adversary can always jailbreak one through known exploits (Adam & McArthur, 2024; Institute for Progress, 2024). Second, post-synthesis modification of benign sequences into dangerous ones is computationally difficult to detect from the synthesizer alone.

We introduce STAMP as a supplementary primitive whose benefits are largely orthogonal to these countermeasures. A STAMP-compliant HSM signs a 120-base barcode into a non-coding region of every synthesized DNA, attesting that the molecule was produced by an uncompromised synthesizer and that no large-scale modifications have been made post hoc. The barcode also embeds plaintext synthesizer ID, run counter, and sequence length for forensic tracing and ordinary research use, and a content-aware landmark logging system enables reconstruction of post-synthesis modifications. STAMP's value scales with ecosystem compliance — most powerfully if benchtop sequencers (whose reliance on computational base calling (Oxford Nanopore Technologies; Au et al., 2021) makes them strong candidates for inextricable HSM lock) act as a downstream verification chokepoint — but even at minimal compliance, STAMP raises visibility, friction, and post-hoc forensic exposure across multiple independent attack vectors.

Our contributions are:

1. An **integrated provenance system** for benchtop-synthesized DNA, combining HSM-anchored cryptographic attestation, content-aware forensic landmarks, and supply-chain-visible primer flanks into a single 120-base barcode. The integration addresses post-synthesis modification under decentralized synthesis — a threat surface few existing systems target.

2. The **landmark map**, a novel forensic primitive that

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

[1]Research conducted at the AIxBio Hackathon, April 2026.

records sparse, content-aware sequence fingerprints in a few bases of the barcode, optionally extended via a public ledger, enabling reconstruction of cloning operations. We show empirically that the privacy-preserving variant detects $\geq 95\%$ of kilobase-scale insertions.

3. A **formal characterization of the residual attack surface**, including a proof bounding what any cryptographic system of this class can achieve (Appendix B), and a worked case study showing the residual gap can be navigated forensically with both ledger access and barcode-only data.

2. Related Work

Cryptographic watermarking of DNA has been explored intermittently both *in silico* (Heider & Barnekow, 2008) and *in vitro* (Lee, 2014; Hamad et al., 2018), but these efforts focus on watermarking proprietary sequences via synonymous codon substitution for IP protection — not biosecurity. We distinguish a watermark (codon-embedded information dispersed across a sequence) from a barcode (a single visible locus carrying information directly in its DNA). STAMP is a barcode by design: a forensic mark must be visible, PCR-amplifiable, and recoverable from a degraded sample, and codon-embedded information satisfies none of these. STAMP is more analogous to a vehicle identification number than to a watermark on a document.

3. Methods

Verification produces a three-level verdict. **Green**: clean synthesizer, no significant post-synthesis modification. **Yellow**: post-synthesis modification — common and often benign. **Red**: a compromised synthesizer or a tampering pattern strongly suggestive of malicious intent. The barcode is added at a non-coding, user-specified site (placement could be automated by a plasmid parser identifying harmless insert sites), and the verifier additionally exposes synthesizer ID, run counter, and forensic landmark data.

The 120-base barcode comprises three primary components, deliberately interspersed so that molecular tampering requires multiple independent precision operations (Figure 1). The full cryptographic specification is in Appendix A; the implementation is open-source on GitHub.[2]

3.1. Cryptographic Anti-Tamper Triad

Three interlocking cryptographic values jointly attest to the integrity of both the sequence and the synthesizing machine:

- **mech_hash** — a hash of the HSM tamper-attestation report at synthesis time. Detects machine tampering.

[2]Anonymous repository link withheld for double-blind review.

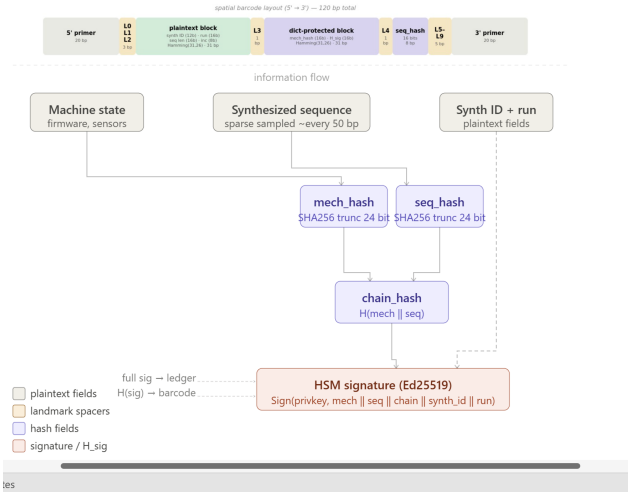

Figure 1. Spatial and informational architecture of the STAMP barcode. Plaintext fields (synth ID, run, sequence length, increment), hash fields (mech_hash, seq_hash), and the truncated h_sig anchor are interspersed with 10 landmark spacer bases between two flanking primer sites.

- **seq_hash** — a hash over a sparse sample of the synthesized sequence ($\sim$1 sample per 50 bp). Detects large-scale edits such as recombinase-based segment replacement or assembly events. Sparse sampling is deliberately insensitive to point substitutions, which routinely occur due to replication error or directed mutagenesis.

- **h_sig** — a 24-bit truncation of an HSM signature binding mech_hash and seq_hash; the full signature lives on the public ledger, with h_sig acting as a barcode-side pointer. h_sig is the cryptographic anchor preventing an attacker from independently regenerating internally consistent hashes.

A modified sequence yields invalid seq_hash and h_sig (yellow). A tampered synthesizer yields invalid mech_hash and h_sig (red). Only a narrow class of technically demanding plasmid modifications can downgrade a red flag to a yellow flag — formal proof in Appendix B; full verification truth table in Appendix D.

3.2. Landmark Map

A landmark map is a sparse forensic fingerprint of the synthesized molecule. The encoder picks ten anchor sites in the sequence based on local content (typically common restriction enzyme sites; priority-list construction in Appendix A), records each site's identity, position, and the single base immediately 5' of it, and publishes these records to the public ledger alongside the cryptographic attestation.

Unlike some proposed encrypted DNA screening systems (Gretton & Esvelt, 2024), the landmark system is deliberately not cryptographically secured. Its value is forensic: most large plasmid manipulations destroy or shift one or more landmark sites in characteristic patterns, allowing a human or ML investigator to reconstruct what kind of modification occurred and roughly where. The system also acts as a friction multiplier against the residual cryptographic attack surface — an attacker forging a valid barcode must reproduce not just the hashes but a landmark pattern consistent with their dangerous sequence.

Privacy-preserving variant. Ledger-stored landmark records leak some structural information (which restriction sites at which positions, with one base of context). Privacy-sensitive deployments may opt to publish only the cryptographic attestation to the ledger and rely on a minimal 10-base landmark fingerprint physically embedded in the barcode itself — recording only the 5' base value at each landmark, with no site or position information. Section 4.2 demonstrates this minimal representation is sufficient for meaningful forensic reconstruction.

3.3. Plaintext Metadata

Synthesizer ID (12 bits), run counter (16 bits), sequence length (16 bits), and an 8-bit resampling increment are stored unencrypted, Hamming(31, 26)-protected. These fields enable ledger lookup, forensic tracing, and routine research uses such as contamination tracking and reproducibility checks. Dual-use design aids both investigators and legitimate researchers, incentivizing voluntary adoption.

3.4. Primer Flanks

The barcode is flanked by two primer-binding sequences, ideally brand-specific and registered with compliant primer-synthesis services as flagged — analogous to existing DHHS screening for primers matching dangerous-pathogen sequences (U.S. Department of Health and Human Services, ASPR, 2023). Beyond their forensic role as PCR amplification handles, the primers act as a supply-chain choke point: an attacker ordering custom primers containing flagged motifs creates an auditable trail before any molecular work begins. Deployment primers should be GC-rich, optimized for trivial forward amplification (for investigators) and difficult inverse amplification (for would-be barcode excisers).

3.5. Constraint-Clean Encoding via Resampling

The barcode must be sanitized of forbidden motifs (the demo uses $\sim$100 common ≥ 6 bp restriction sites from BioPython's CommOnly database) and meet GC and homopolymer constraints. Constraint satisfaction uses a deterministic resampling loop seeded by the 8-bit increment: each iteration produces a fresh nucleotide-encoding dictionary and seq_hash sampling indices, and the first increment whose barcode passes all constraints is selected. Section 4.1 reports the measured success-rate distribution. Blacklisting both removes the attacker's easiest molecular handle for editing the barcode and prevents interference with downstream legitimate cloning.

4. Empirical Validation

We characterized two properties of STAMP empirically: (1) the encoder's ability to satisfy molecular constraints within the increment search, and (2) the verifier's detection power against insertion-class modifications of varying size. All experiments use random plasmids with $\mathrm{GC} \in [0.40, 0.60]$. Random sequences are an appropriate first-order proxy here because encoding success depends on length and seed-derived state rather than host-sequence content, and landmark-anchor finding depends on 6-mer frequencies, which random DNA at realistic GC reproduces well. Real-plasmid robustness is identified as future work.

4.1. Encoding Yield

Across $N = 2000$ random plasmids of length uniformly sampled in $[500, 5000]$ bp and GC fraction uniformly sampled in $[0.40, 0.60]$, all 2000 trials produced a valid encoding within 256 attempts (**100% success**). The increment-of-success distribution had median 10, mean 14.9, p95 = 46, p99 = 71, and worst case 117; increment 0 sufficed for 5.3% of trials and ≤ 5 attempts sufficed for 31.6%. The long upper tail justifies the 8-bit increment field — a 4-bit field would not have covered the p99 case — and 256 attempts provide sufficient headroom across the tested length and GC range.

4.2. Detection Power vs Modification Size

To characterize STAMP's detection power against the canonical attack class — sequence insertion via cloning — we measured the rate at which each verifier signal fires as a function of insertion length L. For each $L \in \{0, 1, 5, 10, 25, 50, 100, 250, 500, 1000, 2500\}$ bp, we generated $N = 200$ random host plasmids (length uniformly sampled in $[1500, 3500]$ bp), STAMPed each, inserted L bp of random DNA at a uniformly random host position, and verified the unchanged barcode against the modified host.

The $L = 0$ control yielded zero false positives across all signals. Any non-zero insertion was caught by either the length or seq_hash signal at $\geq 97\%$ from $L = 1$ bp onward — the length field is deterministic and seq_hash sample indices shift under any indel, so even single-base insertions trigger seq_hash on the vast majority of trials. **Of central interest is the privacy-preserving landmark signal**, which

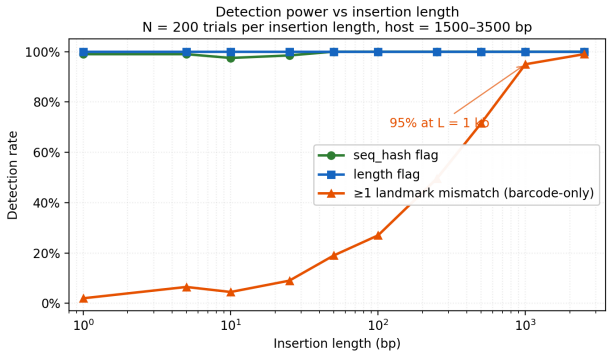

Figure 2. Detection rate as a function of insertion length L, across three verifier signals. Length and seq_hash detect $\geq$ 97% of insertions from $L = 1$ bp upward. The barcode-only landmark signal scales monotonically with L, exceeding 95% by $L = 1$ kb. $N = 200$ trials per insertion length; host plasmid length sampled uniformly in $[1500, 3500]$ bp.

uses only the 10-base barcode-embedded fingerprint with no ledger landmark publication. This signal climbs monotonically with insertion length: 19% at $L = 50$ bp, 27% at $L = 100$ bp, 71.5% at $L = 500$ bp, 95% at $L = 1000$ bp, and 99% at $L = 2500$ bp. At the $\sim$700 bp scale of the case study (Section 5), the expected detection rate is approximately 80–90%, consistent with the case study's two observed landmark mismatches.

5. Case Study: Forensic Reconstruction of a Transplant Attack

We present a worked case study illustrating STAMP's central thesis: the combination of seq_hash mismatch, length mismatch, and landmark forensics produces a structured evidence trail that constrains an investigator's interpretation to a narrow set of plausible modification histories.

5.1. The Attack

An attacker generates a legitimate plasmid (GFP) from an accredited synthesizer, producing a STAMP-attested 1062 bp construct (Figure 3, center). In parallel, the attacker obtains — from a black-market jailbroken synthesizer or naturally-derived source — a copy of a dangerous gene (in our demo, harmless mCherry as a stand-in payload). They use restriction cloning to ligate the payload into the stamped backbone, producing a 1776 bp chimeric plasmid (Figure 3, right). The barcode itself is unedited; the host has gained $\sim$714 bp of new sequence.

Table 1. Landmark displacement, sorted by original position. Six landmarks displaced by exactly +714 bp localize the insert; the +2 bp shift at landmark 0 identifies the EcoRI cut site; landmark 5's +1032 bp displacement reflects a novel anchor inside the insert.

LM #	Orig. Pos.	New Pos.	Move	Interpretation
3	44	44	0	Upstream of insert
2	53	53	0	Upstream of insert
0	60	62	+2	Likely EcoRI cut site
1	91	805	+714	Displaced by insert
6	178	892	+714	Displaced by insert
5	587	1619	+1032	Novel site within insert
8	828	1542	+714	Displaced by insert
7	845	1559	+714	Displaced by insert
4	861	1575	+714	Displaced by insert
9	893	1607	+714	Displaced by insert

5.2. Forensic Reconstruction with Ledger Access

The verifier returns MODIFIED with three independent signals: seq_hash mismatch, length mismatch (declared 942, actual 1776), and 2/10 landmark feature mismatches. The barcode also exposes synthesizer ID, run counter, and timestamp via the ledger entry. Because $942 + 120 + 714 = 1776$ exactly, the investigator can already infer that $\sim$714 bp of new sequence has been inserted.

To localize the insertion, the investigator compares stored vs. found landmark positions (Table 1). The pattern is unambiguous: six landmarks downstream of position $\sim$60 are all displaced by exactly +714 bp. Two upstream landmarks (positions 44, 53) are unmoved. Landmark 0 (an EcoRI site at position 60) shifted +2 bp and changed its 5' base from C to T — consistent with EcoRI being the 5' cut/ligate site. Landmark 5 anchors to a novel AAAGTC site at position 1619, most parsimoniously a higher-priority match created by the inserted material.

The investigator concludes: a single $\sim$714 bp insert was added at approximately position 60 using EcoRI as the 5' cut site, the backbone is largely unmodified outside the insertion, and other common cut sites (HindIII, BamHI, XhoI) were not used. The 3' cut site is not directly identified but can be narrowed to a short candidate list (Acc65I, KpnI, SacII, PspOMI, etc.). All conclusions match the blinded ground truth: a 714 bp restriction-cloning insertion

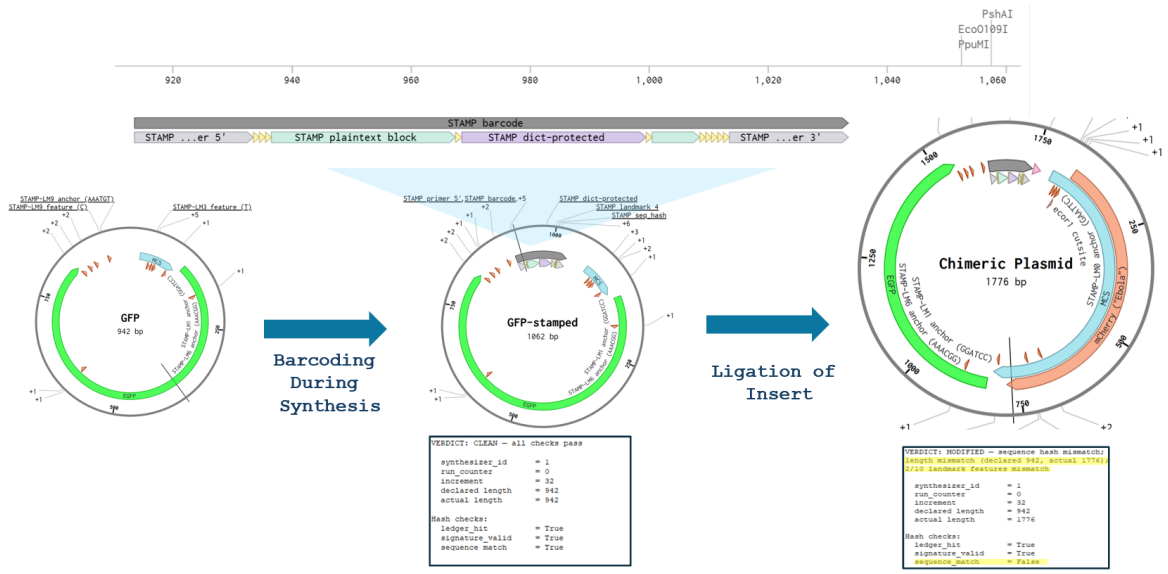

Figure 3. Case study workflow. Left: original 942 bp GFP plasmid. Center: GFP-stamped construct after barcode insertion (1062 bp; verifier returns CLEAN). Right: chimeric plasmid after attacker ligates the payload via EcoRI (1776 bp; verifier returns MODIFIED with seq_hash mismatch, length mismatch, and 2/10 landmark feature mismatches). The attacker has not edited the barcode itself — they have inserted ∼714 bp of new sequence into the host construct.

via EcoRI- and Acc65I-mediated cuts, with minimal cloning scar.

5.3. Forensic Reconstruction Without Ledger Access

A privacy-preserving deployment may forbid the ledger from storing landmark site or position information, leaving only the 10 base values physically embedded in the barcode. We replicate the analysis under this constraint (full reasoning in Appendix E) and recover the same modification class, approximate location, and most likely cloning enzyme. Confidence is genuinely lower, but **the modification class and approximate location are recovered from a 10-base fingerprint alone** — without any ledger landmark publication. Forensic richness scales with ledger access; it does not collapse without it.

6. Discussion

Compliance regimes. STAMP's value scales with ecosystem participation but does not require it. In a *low-compliance* regime where only synthesizers participate, STAMP still imposes meaningful costs: barcode excision is slow against adversarially designed primer flanks; excised barcodes leave a substantial post-hoc forensic trail (synthesizers produce picomole quantities of barcoded molecules per run, recoverable from wastewater and benchtop swabs); barcoded plasmids on Addgene narrow distribution channels for dangerous constructs; and the landmark + sparse-hash forensic toolkit eases the path to warrants and prosecution. In a *high-compliance* regime where third-party sequencing and primer-synthesis services also enforce STAMP, iterative AI-assisted threat development is either logged or rejected at every cycle. We identify sequencing as the most compelling chokepoint: most threat workflows still require sequencing for verification, and benchtop sequencers' computational dependence on base calling makes them strong candidates for inextricable HSM lock (Oxford Nanopore Technologies; Au et al., 2021; Stillman et al., 2025). Detailed cost-imposition arguments per regime are in Appendix G.

Limitations and future work. STAMP requires a minimal HSM in benchtop synthesizers, which does not yet ship. The two-synthesizer transplant attack — an attacker stamping a benign plasmid and ligating the barcode onto a dangerous backbone — is the residual cryptographic gap formalized in Appendix B; ledger landmark publication tightens it dramatically. STAMP is unsuited to fine-grained engineering detection, and the 120 bp footprint recommends enforcement only for sequences $\geq$ 1 kb. Full limitations and dual-use risks are in Appendix F. Natural follow-ups: broader attack benchmarking, confidence intervals for barcode-only reconstruction, ML interpretation of landmark patterns, and richer encodings. The governance work needed to reach high-compliance sequencing — STAMP's load-bearing assumption — determines how much the technical primitive

matters in practice.

7. Conclusion

STAMP is a 120-base barcode that establishes provenance and forensic documentation of synthesized DNA via a cryptographic triad, a content-aware landmark map, and plaintext research metadata. Its value is best understood not as a lock but as a *cost imposer and evidence generator* — converting every viable attack into either a forensically suspicious artifact or a supply-chain-visible event, with the case study showing the evidence trail is meaningful even from the 10-base privacy-preserving fingerprint alone. We acknowledge the ethical gray zone of cryptographically tagging genetic material under universal compliance: we have built the technical primitive; whether it becomes a forensic tool, a compliance infrastructure, or something more concerning depends on policy decisions explicitly outside this work.

Code and Data

- GitHub: *Anonymous link withheld for double-blind review.*
- Walkthrough video: *Anonymous link withheld for double-blind review.*
- Interactive Streamlit demo: *Anonymous link withheld for double-blind review.*

Impact Statement

This work develops a technical primitive for cryptographic provenance of synthesized DNA, with explicit dual-use considerations. A full discussion of limitations, surveillance and IP-control risks, compliance-theater concerns, concentration-of-trust risks, adversarial use of forensic methods, and recommended ethical guardrails for any deployment regime appears in Appendix F. Whether STAMP becomes a forensic tool, a compliance infrastructure, or something more concerning depends on policy decisions explicitly outside the scope of this work.

References

Adam, L. and McArthur, G. H. Substitution attacks: a catalyst to reframe the DNA manufacturing cyberbiosecurity landscape in the age of benchtop synthesizers. *Applied Biosafety*, 29(3):172–180, 2024.

Anguzu, S. Preventing biosecurity risks posed by next-gen benchtop DNA synthesizers. *Health Economics and Management Review*, 6(1):126–143, 2025.

Au, K. F. et al. Nanopore sequencing technology, bioinformatics and applications. *Nature Biotechnology*, 39: 1348–1365, 2021.

Brent, R. and McKelvey, Jr., G. Contemporary foundation AI models increase biological weapons risk. Technical Report PE-A3853-1, RAND Corporation, 2025.

Carter, S. R., Yassif, J., and Isaac, C. Benchtop DNA synthesis devices: Capabilities, biosecurity implications, and governance. Technical report, Nuclear Threat Initiative (NTI), 2023.

Diggans, J. and Leproust, E. Next steps for access to safe, secure DNA synthesis. *Frontiers in Bioengineering and Biotechnology*, 7:86, 2019.

Gretton, D. and Esvelt, K. M. Exact-match search with functional variant prediction enables automated DNA screening. *bioRxiv*, 2024. 2024.03.20.585782.

Hamad, S., Elhadad, A., and Khalifa, A. DNA watermarking using codon postfix technique. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(5): 1605–1610, 2018.

Heider, D. and Barnekow, A. DNA watermarks: A proof of concept. *BMC Molecular Biology*, 9(1):40, 2008.

Institute for Progress. Securing benchtop DNA synthesizers. Technical report, Institute for Progress, 2024.

International Gene Synthesis Consortium (IGSC). Harmonized screening protocol, version 3.0. Technical report, International Gene Synthesis Consortium, 2024.

Langenkamp, M. Securing benchtop DNA synthesizers. Institute for Progress, December 2024.

Lee, S. H. DWT-based coding DNA watermarking for DNA copyright protection. *Information Sciences*, 273:263–286, 2014.

National Academies of Sciences, Engineering, and Medicine. The age of AI in the life sciences: Benefits and biosecurity considerations. Technical report, National Academies, 2025.

Oxford Nanopore Technologies. Sequencing devices and platforms. https://nanoporetech.com.

Stillman, C., Bravo, J. E., Boucher, C., and Rampazzi, S. Toward security-aware portable sequencing. *Nature Communications*, 16:9829, 2025.

U.S. Department of Health and Human Services, ASPR. Screening framework guidance for providers and users of synthetic nucleic acids. Technical report, U.S. Department of Health and Human Services, 2023.

Wheeler, N. Responsible AI in biotechnology: balancing discovery, innovation and biosecurity risks. *Frontiers in Bioengineering and Biotechnology*, 13:1537471, 2025.

A. Detailed Technical Specification

A.1. Resampling Seed

All STAMP encoding is seeded by an 8-bit increment value (4 bases) stored in the barcode plaintext. Each candidate barcode is scanned for sequences that may interfere with synthesis or downstream cloning: GC-rich regions, homopolymer repeats, and motifs from an updatable blacklist library. The demo uses BioPython's CommOnly database of 622 commercial restriction enzymes, filtered to $\sim$100 common cutters of length ≥ 6 bp. The encoder tries increments 0..255 and accepts the first that produces a constraint-clean barcode.

A.2. Cryptographic Anti-Tamper Triad

Three cryptographic values, two stored in the barcode and one on the public ledger, jointly defend the sequence, machine, and the barcode itself.

1. **mech_hash.** A 24-bit truncation of SHA-256 over the machine's internal state at synthesis time (firmware version, hardware sensor readings). The demo simulates HSM attestation with a deterministic 256-bit mock report; the truncation is barcode-budget-driven (8 bases). Detects machine tampering.

2. **seq_hash.** A 24-bit truncation of SHA-256 over a sparse sample of the synthesized sequence, with sampling indices derived from the seed ($\sim$1 sample per 50 bp). 8 bases. The sparse design deliberately ignores point mutations — noise from synthesis error and routine site-directed mutagenesis — and is sensitive to translocation-scale modifications such as recombinase-based segment replacement, assembly events, and large insertions or deletions. A side-effect benefit: frameshift indel errors that destroy plasmid function also invalidate seq_hash, providing a voluntary-adoption early warning.

3. **h_sig.** A 24-bit truncation of SHA-256 over the full HSM signature, which itself signs (mech_hash $\|$ seq_hash $\|$ chain_hash) where chain_hash binds mech and seq together. The full signature lives on the public ledger; h_sig in the barcode acts as a pointer that lets the verifier confirm the recovered hashes match what the registered HSM signed. This is the cryptographic anchor — without it, an attacker could regenerate internally consistent hashes locally.

A.3. Landmark Mark

Ten landmarks are localized independently using ten dedicated priority lists. Priority lists are constructed by enumerating all $4^6 = 4096$ 6-mers, canonicalizing each (taking the lexicographically smaller of itself and its reverse complement), deduplicating, and sorting — yielding $\sim$2048 canonical 6-mers. These are split evenly into 10 mutually exclusive priority lists of $\sim$200 6-mers each. The first four lists have common restriction enzyme recognition sites (EcoRI, BamHI, HindIII, XhoI) bubbled to top priority; the remaining six retain natural ordering, so all ten landmarks do not collapse to a single dense MCS region. For each landmark slot the encoder walks its priority list top-down, stopping at the highest-priority 6-mer present in the construct (forward or reverse-complement). Multiple occurrences of the same site are tiebroken alphabetically on the 10 bases immediately downstream. Once an anchor is chosen, the base immediately 5' of the site is recorded as the 2-bit landmark feature.

A.4. Plaintext Metadata

Plaintext fields (Hamming-protected, standard A=0/C=1/G=2/T=3 mapping so they decode without prior knowledge of the seed-derived dictionary):

- synthesizer ID — 12 bits (4096 unique synthesizers)
- run counter — 16 bits (wraps at buffer limit)
- sequence length — 16 bits (0–65535 bp original construct length)
- increment — 8 bits, split across the plaintext block to escape both head-of-barcode and tail-of-barcode constraint failures during the resampling search

A.5. Hamming Correction

All Hamming-protected fields use Hamming(31, 26): each block of 26 data bits is encoded into a 31-bit codeword that can correct any single-bit error. The plaintext block (52 data bits → 62 codeword bits = 31 bases) and dict-protected block (mech_hash + h_sig = 48 bits → 62 codeword bits = 31 bases) each consume two Hamming(31, 26) codewords. The cost is 12 of 124 barcode bases dedicated to error correction; the benefit is buffer against synthesizer error and sequencing noise.

B. Proof of Impossibility

There exists an irreducible gap in any cryptographic verification system that permits backbone edits at all. In practice, public-ledger landmark records severely restrict the attack surface, but a fully self-contained cryptographic solution does not exist.

Theorem B.1. *Let V be any verification system that (1) permits legitimate post-synthesis sequence modification and (2) operates without access to the original sequence at verification time. Then V cannot distinguish legitimate modification from a transplant attack.*

Proof. By condition (1), V must output "legitimate" for some state $S = (sig, seq)$ where sig is valid against sequence X and $seq \neq X$. This state must exist or legitimate modification is impossible.

By condition (2), V has no access to X at verification time. V's input is therefore $(sig, seq, ledger\_entry)$ where $ledger\_entry$ contains at most $H(X)$ and metadata.

Now construct the attack state $S' = (sig, seq')$ where seq' is an arbitrary dangerous sequence and sig is transplanted from a legitimate synthesis of X. V receives $(sig, seq', ledger\_entry)$, which is informationally identical to S from V's perspective, since:

- sig validates correctly → same in both cases
- $H(seq) \neq H(X)$ → same in both cases
- $ledger\_entry$ points to X → same in both cases
- seq and seq' are both $\neq X$ → same in both cases

V cannot distinguish S from S' without additional information not available under condition (2). Therefore V outputs the same result for both. □

Corollary B.2. *Closing this gap requires relaxing either condition (1) — prohibiting post-synthesis modification — or condition (2) — providing the verifier access to original sequence content, either directly or via a trusted oracle such as a sequence registry.*

C. Information-Theoretic Lower Bound for Modification Description

A thought experiment: how few bits suffice to characterize a translocation-scale modification on a plasmid? The result suggests the lower bound is far smaller than expected — under 10 bases for many modification classes — providing theoretical motivation for the landmark system's minimal physical footprint.

Minimal Description of Segment Replacement

We wish to describe minimally the replacement of Segment A with Segment B, with a broad estimate for the size and position of these sequences. Barring uncommon edits such as inversions, all modifications can be decomposed into insertions, excisions, and two classes of substitution (specifying which fragment is inserted and which is removed, with an arbitrary priority-determining algorithm). These four operation types can be stored as **2 bits**.

We assume location resolution to an octad ($\frac{1}{8}$) of the plasmid, requiring **3 bits**. Four size ranges are defined for each segment (stored as raw value, proportion of total plasmid size, etc.), requiring **2 bits** per segment. Each plasmid is assigned one of eight landmark-based IDs, requiring **3 bits** per landmark.

Field	Encoding	Bits
Operation type	4 types	2
Location	$\frac{1}{8}$ of plasmid	3
Size of Segment A	4 ranges	2
Size of Segment B	4 ranges	2
Landmark, Segment A	1 of 8	3
Landmark, Segment B	1 of 8	3
Total		**15 bits**

A 15-bit descriptor corresponds to $\lceil 15/2 \rceil = 8$ bases (rounding up to the nearest base, since each base encodes 2 bits).

$$2 + 3 + 2 + 2 + 3 + 3 = 15 \text{ bits} = 7.5 \text{ bases} \approx 8 \text{ bases}$$

Note that the gap between theoretical minimum description length and practical detection reliability remains uncertain.

D. STAMP Verification Truth Table

Individual pass or fail by a hash or landmark test is neither inherently good nor bad. The relevant signal is the pattern of failures, which uniquely identifies attack classes. Only an unmodified plasmid from a secure synthesizer generates a green flag. Benign post-synthesis modifications generate a yellow flag; most barcode manipulations generate a red flag. There is a theoretically irreducible gap (Appendix B) through which a perfect attacker could, in principle, spoof a yellow-flag signal. In the absence of a public landmark ledger this is technical but achievable; with a full ledger this requires both perfect molecular reproduction of a forged STAMP and design of a functional, harmful plasmid that still presents the correct ten landmark features. We treat this as a primarily theoretical vulnerability; the rational attacker response is full barcode removal, which is itself a red-flag signal in a compliant ecosystem.

Table 2. STAMP verification truth table mapping verifier output patterns to attack classes. Pass/fail of any individual check is not the verdict — the specific pattern of passes and failures is diagnostic.

Case	Mech	Seq	Chain	H-sig	Landmarks	Flag
Baseline						
Clean plasmid	pass	pass	pass	pass	consistent	GREEN
Legitimate modification						
Large modification	pass	fail	pass	fail	not consistent	YELLOW
Indel	pass	fail	pass	fail	consistent	YELLOW
Attack — Partial replacement						
Jailbroken machine	fail	pass	pass	fail	inactive	RED
Switched mech hash	pass	fail	fail	fail	inactive	RED
Switched seq hash	pass	fail	fail	fail	inactive	RED
Switched mech + seq	pass	pass	fail	fail	inactive	RED
Attack — Transplant						
Barcode transplant from safe sequence	pass	fail	pass	fail	consistent	RED
Irreducible gap — theoretical						
All hashes switched + landmarks faked	pass	pass	pass	fail	not consistent	YELLOW
Removal						
Barcode removed	N/A	N/A	N/A	N/A	inactive	RED

E. Forensic Reconstruction from Barcode-Embedded Landmark Data Only

Without ledger access, forensic reconstruction proceeds from barcode-embedded information alone. The sequence length field indicates the modified plasmid has grown approximately ∼714 bp since synthesis. Landmark data consists only of the

5’ flanking base at each of 10 anchor sites recorded at synthesis time; neither anchor position nor priority queue rank is stored in the barcode.

Reconstruction relies on three probabilistic priors:

1. The highest-priority 6-mer in each priority queue is statistically most likely to have been the encoded anchor, especially when it is a common restriction enzyme site (high prior on plasmids).
2. If the current 5’ base at a landmark slot matches the barcode-recorded value, the landmark is likely unchanged.
3. If the 5’ base at a landmark slot has changed and the priority anchor is a restriction site, the most parsimonious explanation is restriction-mediated cloning at that site.

Applying these priors, 8 of 10 landmarks show unchanged flanking bases and are treated as positionally stable. Two landmarks are perturbed. Landmark 0 is currently a restriction enzyme site, consistent with its use as a ligation point. Landmark 5 is not a restriction enzyme site and has no proximal restriction site, but shows a changed flanking base and falls within 700 bp of the presumed cut site; the 8 stable landmarks collectively span the remainder of the plasmid (Figure 4, right). The largest gap is 650 bp, placing landmark 5 within the region consistent with the insert. We therefore map the insertion as originating at landmark 0 and extending 714 bp, encompassing landmark 5.

We conclude that the construct derives from the original GFP plasmid with a single insertion event, that the backbone is largely unmodified outside the insert, and that EcoRI was likely the cloning enzyme while HindIII, BamHI, and XhoI were not. These conclusions are consistent with ground truth and with ledger-assisted analysis, though without a formal confidence interval.

Landmark #	Current Position (bp)	Neighbor Change (Sorted asc.)	Type
3	44	No	Random 6-mer
2	53	No	Random 6-mer
0	62	C -> T	EcoRI Cut
1	805	No	Random 6-mer
6	892	No	Random 6-mer
8	1542	No	Random 6-mer
7	1559	No	Random 6-mer
4	1575	No	Random 6-mer
9	1607	No	Random 6-mer
5	1619	G-> A	Random 6-mer

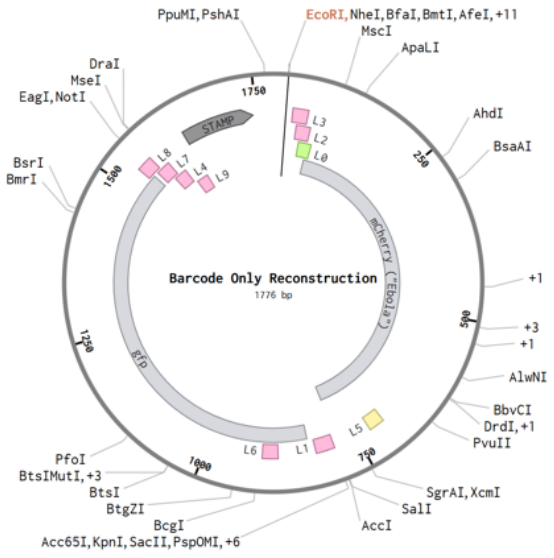

Figure 4. Barcode-only reconstruction. Left: landmark table sorted by current position; only landmark 0 (EcoRI cut site) and landmark 5 (novel site, 5’ base changed) deviate from baseline. Right: chimeric plasmid map. Pink = unchanged landmarks; green = EcoRI cut site; yellow = novel landmark in insert; orange = displaced. The single sufficient gap to fit a ∼714 bp insert overlaps landmark 5’s new position, confirming the inferred insertion locus.

Confidence is genuinely lower than under ledger-based analysis, and we do not provide formal confidence intervals. We leave this demonstration as proof-of-concept that even a 10-base embedded fingerprint can support meaningful forensic reconstruction. We remain uncertain whether similar logic chains generalize to all modification classes, and how far probabilistic chains can be extended before collapsing under ambiguity. Formal characterization is left to future work.

E.1. Supporting Verifier Output (with Ledger Access)

For completeness, the verifier panel and per-slot landmark forensic readout supporting Section 5.2 are reproduced below. The displacement table (Table 1, main text) is the central evidence; these panels show the raw verifier output a forensic investigator would see.

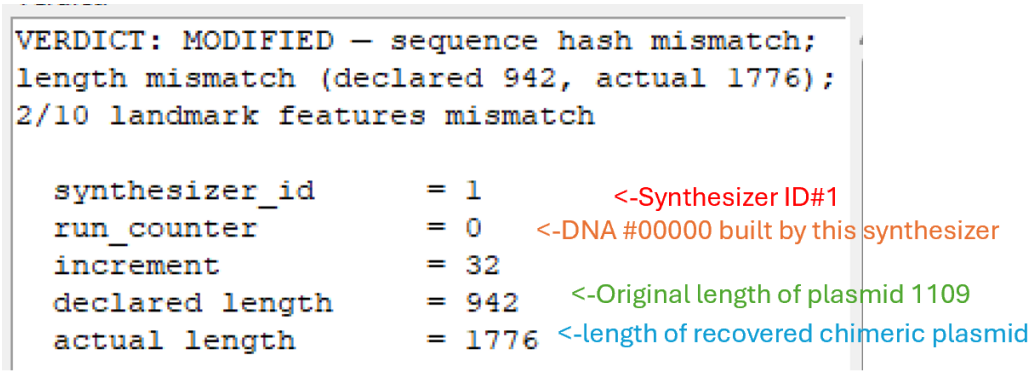

Figure 5. Verifier output panel for the chimeric plasmid. The barcode reports synthesizer #1, run counter 0, original length 942 bp; the recovered suspect sequence is 1776 bp, and 2/10 landmarks show feature mismatches. Note that $942 + 120 + 714 = 1776$.

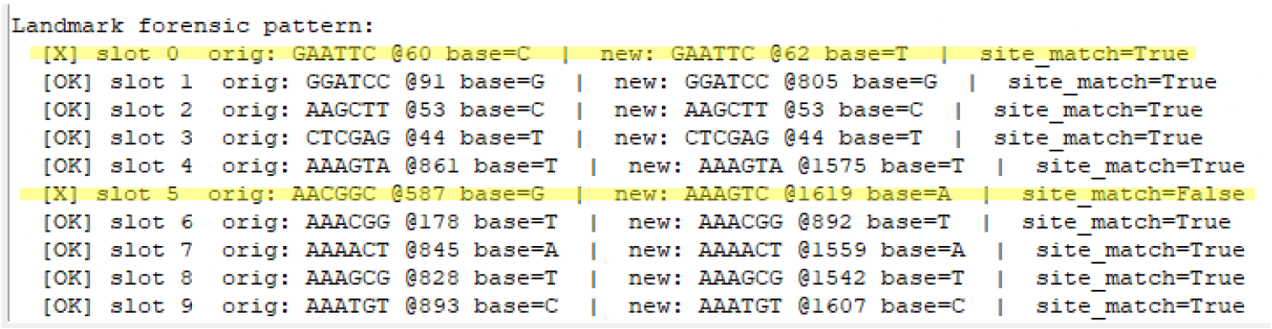

Figure 6. Landmark forensic readout. Slot 0 (EcoRI, GAATTC) retained its anchor site at a near-original position ($60 \rightarrow 62$) but its 5' base changed from C to T, suggesting EcoRI was used to cut and ligate. Slot 5 changed identity entirely (AACGGC $\rightarrow$ AAAGTC) at a position deep inside the insert region.

F. Limitations and Dual-Use Considerations

F.1. Limitations and Edge Cases

Our limitations are stated in Section 6; we elaborate here on edge cases relevant to deployment.

- **False negatives.** Sub-50 bp modifications often pass seq_hash undetected, by design. A determined attacker with knowledge of the sparse-sampling pattern (computable from the public increment) could in principle engineer modifications to fall between sample positions; mitigated in practice by the seed-derivation hash being a one-way function over synth_id, run, and increment, making per-construct sampling positions hard to predict in advance.
- **False positives.** Legitimate large modifications (sub-cloning, deletion of dispensable regions, multi-fragment assembly) all generate yellow flags. The system is not designed to distinguish "yellow because attack" from "yellow because legitimate research"; that judgment requires human or ML interpretation of the forensic readout.
- **Hash truncation collisions.** At 24 bits, birthday-bound collision resistance is $\sim 2^{12}$. Acceptable for forensic anchoring at single-construct scale, unacceptable for cryptographic non-repudiation. Production deployment should use longer truncations or full 256-bit hashes on the ledger side.

- **Mock cryptography in the demo.** Our HSM signing is a deterministic pseudorandom function; production deployment requires real Ed25519 (or equivalent) under HSM-protected keys, with manufacturer-maintained public-key registries.
- **Scalability.** Assuming order-of-magnitude $\sim 10^9$ synthesis events globally per year, the ledger requires $\sim$200 GB/year at the current schema. Tractable for transparency-log infrastructure but non-trivial for global federation.
- **Two-synthesizer transplant attack.** An attacker with both clean and jailbroken synthesizers can stamp a benign plasmid, harvest the barcode, and ligate it onto a dangerous backbone, yielding a yellow flag indistinguishable from legitimate post-synthesis modification. Only when the public ledger records landmark information does STAMP become informationally robust against this attack — the attacker must then design a host plasmid whose landmarks match the ledger record, an extremely difficult constraint.
- **Size limitations.** A $\sim$120 bp tag is non-trivial overhead for very small constructs. We recommend STAMP enforcement only for sequences ≥ 1 kb, accepting that smaller sequences may also be dangerous. For length-constrained constructs such as lentiviral preps (8–14 kb), STAMP overhead is $< 1.5\%$; institutional virus cores can validate and document barcode removal where necessary.
- **Economic cost.** As previously noted in a screening context (Diggans & Leproust, 2019), control measures including STAMP impose economic costs on laboratories and biotechnology companies, which often operate on narrow margins. We mitigate this cost by keeping the STAMP itself deliberately small and integrating it with useful metadata features that aid voluntary adoption.

F.2. Dual-Use Risks

STAMP is itself dual-use. We catalog the principal risks:

- **Surveillance and IP control.** A universal compliance regime that cryptographically tags every synthesized DNA creates infrastructure for surveillance of legitimate research and for corporate IP tracking. This is the most serious risk and we do not minimize it. Mitigations: privacy-preserving deployment (Section 5.3), seq_hash sparse sampling that reveals nothing reconstructive about content, and explicit policy work distinguishing forensic ledger access from sequence registries.
- **Compliance theater.** A poorly governed STAMP regime could become security theater — visible compliance without functional verification. Mitigation: open-source verifier tooling (this project), public truth tables (Appendix D), and documented limitations.
- **Concentration of trust.** Whoever runs the public ledger has substantial power over which synthesis events are visible. Mitigations: transparency-log architectures with verifiable append-only properties, federated multi-jurisdiction operation, and public-key registries.
- **Adversarial use of forensic methods.** Landmark forensics could in principle be used by malicious parties to identify which lab synthesized a particular construct from environmental DNA samples — a privacy concern for legitimate research, especially in academic settings. We do not have a clean mitigation; this trades off with the wastewater-monitoring forensic value.

F.3. Responsible Disclosure

We did not discover deployment vulnerabilities in existing systems during this work. The closest item: STAMP's impossibility-theorem gap (Appendix B) is a property of any cryptographic verification system permitting backbone edits, not a flaw specific to a deployed system. We disclose this openly because it shapes correct deployment expectations. STAMP should not be marketed as a perfect prevention system.

F.4. Ethical Considerations

We acknowledge the ethical gray zone described in the conclusion. Cryptographically tagging genetic material has implications for tracking and copyrighting genetic constructs and engineered organisms. We have built the technical primitive; whether it becomes a forensic tool, a compliance infrastructure, or something more concerning depends on policy decisions explicitly outside the scope of this work. We recommend any deployment regime explicitly exclude organism tracking outside formal regulatory contexts and include sunset provisions on ledger entries for academic research.

F.5. Suggested Future Improvements

- Real cryptographic primitives (Ed25519 + transparency log) replacing the demo mocks.
- Formal characterization of barcode-only forensic reconstruction confidence (Appendix E).
- ML-based landmark interpretation and encoding optimization.
- Empirical sensitivity analysis: barcode budget vs forensic recovery rate, hash truncation vs collision risk, landmark count vs reconstruction accuracy.
- Federated multi-jurisdiction ledger architecture for governance robustness.

G. Detailed Compliance-Regime Analysis

G.1. The High-Compliance Regime

In an ecosystem where sequencing services or benchtop sequencers check for and flag plasmids with suspicious STAMP readouts, and primer-synthesis services flag sequences with strong specific binding to STAMP's diagnostic primers, STAMP imposes substantial costs on attackers attempting iterative AI-assisted threat development. We identify sequencing as the key downstream chokepoint: most workflows — whether successful genetic modification of a disease model or a successful bioterrorist attack — still require sequencing for verification. If valid STAMP attestation becomes a precondition for sequencing, an attacker faces three options:

- Submit with a missing barcode and accept rejection.
- Submit with a forensically suspicious barcode and accept investigation.
- Forge a valid barcode molecularly — we estimate this requires $\sim$3 weeks of skilled cloning or custom primer orders against flagged sequences, and produces a junction-scarred construct still likely to fail landmark consistency checks.

Iterative AI-assisted sequence design — the dominant near-term concern — requires repeated synthesis-test-redesign cycles. In a compliant ecosystem, each cycle is either logged or rejected, substantially constraining the iterative capability that makes AI-assisted design dangerous in the first place.

G.2. The Low-Compliance Regime

Even when only synthesizers participate, STAMP imposes costs along independent vectors:

1. Barcode excision is desirable to scrub ledger pointer and metadata, but takes time, especially with adversarially designed primer flanks.
2. Excised barcodes still leave a substantial post-hoc forensic trail. Excision and amplification operate on a single molecule, but synthesizers produce picomole quantities of barcoded molecules per run. Unmodified plasmids can survive intact for years in dry environments; wastewater monitoring, benchtop swabs, or post-hoc raids can recover incriminating attested sequences, forcing illicit operations into expensive waste-stream sanitization or constant data-leakage risk.
3. Barcoded plasmids deposited on Addgene or the open market act as a stamp of legitimate origin, narrowing channels through which dangerous plasmids can be obtained or sold.
4. Landmark mapping and sparse sequence hashing produce a workable forensic toolkit for probing translocation-scale modifications, easing the path to warrants, prosecution, and therefore deterrence.

G.3. LLM Usage Statement

We used Claude to brainstorm algorithm design, implement computational methods, and revise writeup draft structure. All implementation was stress-tested with independent test cases, and all algorithms were verified by hand.