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

Nicole Lai-Lopez
Independent

Ethan Lai
Independent

With
Apart Research

Abstract

As AI systems become more capable at biological design and benchtop DNA synthesizers more affordable, the biothreat bottleneck shifts from design to physical synthesis – beyond the reach of centralized customer screening. We introduce STAMP (Synthesis Tamper-evident Attestation and Molecular Provenance), a 120-base barcode an HSM-equipped synthesizer stamps into a non-coding region of every DNA it produces, attesting that a sequence originated from a registered, untampered synthesizer and was not significantly modified post-synthesis. STAMP combines cryptographic anchoring with a novel content-aware "landmark map" enabling forensic reconstruction of post-synthesis modifications. Empirically, the encoder achieves success across $N = 2000$ random plasmids and the privacy-preserving barcode landmark signal detects $\geq 95\%$ of kilobase-scale insertions. We do not claim to defeat attackers with jailbroken synthesizers — we prove this is irreducible. Instead, STAMP is a cost imposer and evidence generator: it converts every viable attack into a forensically suspicious artifact or a supply-chain-visible event.

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[12, 13,14]. As that barrier falls, the bottleneck shifts to physical synthesis. Affordable benchtop synthesizers

capable of printing viral-length DNA may be available in as few as two years under aggressive projections[1], creating a decentralized synthesizer ecosystem that is potentially uncontrolled and untrackable as a bioterrorism platform.

In response, frameworks like the IGSC Harmonized Screen Protocol[8], have pushed for improved synthesizer screening capabilities, typically backed by a tamper-resistant Hardware Security Module (HSM)[2]. This approach faces two technical limits. First, benchtop synthesizers are fundamentally 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 [15, 17] to produce a harmful sequence. Second, post-synthesis modification of benign sequences into dangerous ones is computationally difficult to detect.

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 the non-coding region of every synthesized DNA, attesting that this specific sequence 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: in an ecosystem where third-party services (assembly, sequencing, primer synthesis) also enforce STAMP, iterative AI-assisted development becomes dependent on detectable barcode tampering or visible non-compliance. We identify sequencing as the most compelling chokepoint. Unlike synthesizers, benchtop sequencers (typically Nanopore-based) rely on delicate computational infrastructure for base calling[3, 16], making them strong candidates for inextricable HSM lock. Even at minimal compliance, STAMP raises visibility, friction, and post-hoc forensic exposure across multiple independent attack vectors; it is designed to make misuse harder to conceal, easier to investigate, and riskier to operationalize.

Our main 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 records sparse, content-aware sequence fingerprints in a few bases of the barcode, optionally extended via a public ledger, allowing reconstruction of cloning operations. We empirically demonstrate that the privacy-preserving variant detects $\geq$95% of kilobase-scale insertions.
3. **A formal characterization of the residual attack surface**, including a proof of impossibility (See Appendix 2) bounding what any cryptographic system of this class can achieve, and a

worked case study demonstrating that the residual gap can be navigated forensically with both ledger access and barcode-only data.

2. Related Work

Cryptographic watermarking of DNA is not a new concept and has been explored intermittently both *in silico* [4] and *in vitro* [5, 7]. These efforts focus on watermarking proprietary sequences via synonymous codon substitution for IP protection, not biosecurity. We distinguish a watermark — typically codon-embedded information dispersed across a sequence — from a barcode, which occupies a single visible locus and carries 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; codon-embedded information satisfies none of these and conflicts with constrained codon choice required for many biological workflows. STAMP is more analogous to a vehicle identification number than a watermark on a document.

3. Methods

Verification produces three flag levels. A **green flag** indicates a clean synthesizer and no significant post-synthesis modification. A **yellow flag** indicates post-synthesis modification — common, often benign. A **red flag** indicates a compromised synthesizer or tampering pattern strongly suggestive of malicious intent. STAMP also exposes synthesizer ID, run counter, and supports forensic reconstruction of specific modifications. The barcode is added at a non-coding, user-specified site; placement could be automated by a plasmid parser identifying harmless insert sites.

The 120-base STAMP barcode comprises three primary components, deliberately interspersed to make molecular tampering require multiple independent precision operations (Figure 1). Detailed cryptographic specification is in Appendix 1; full implementation is on GitHub.

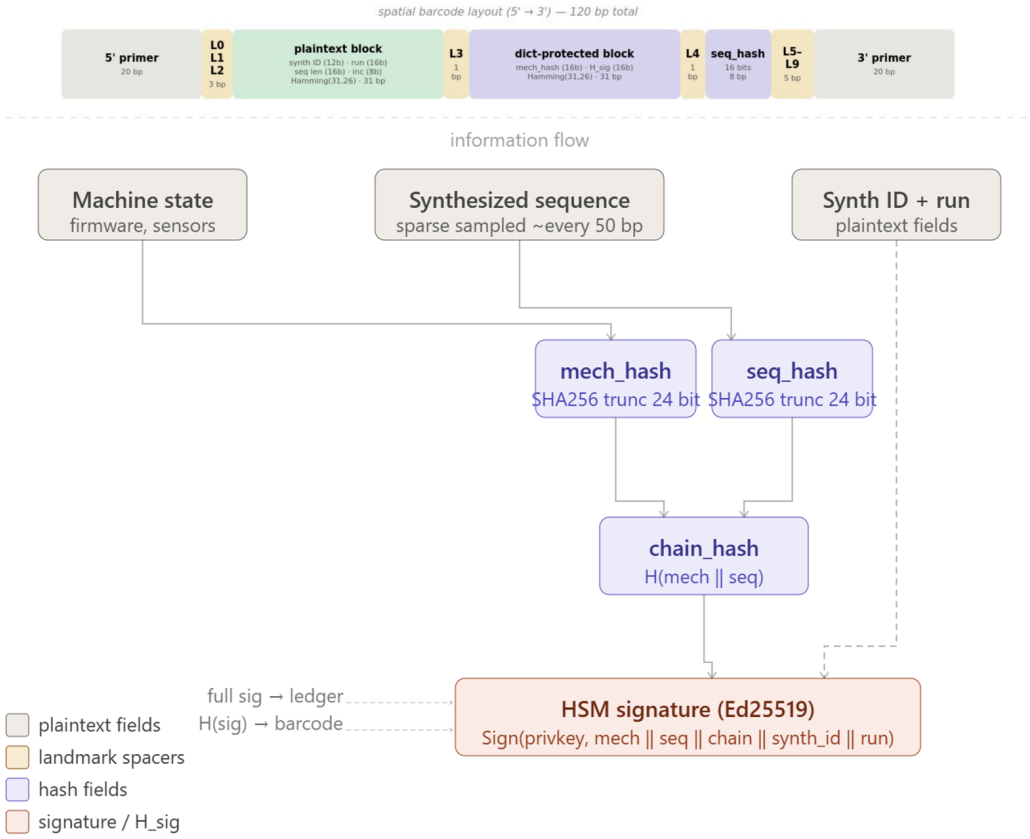

Figure 1: Spatial and informational architecture of the STAMP barcode. Plaintext fields (synth ID, run, sequence length, increment), hash fields (mech_hash, seq_hash) and the truncated h_sig anchor are interspersed with 10 landmark spacer bases between two flanking primers sites.

3.1 Cryptographic Antitamper Triad

Three interlocking cryptographic values together 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.
- **seq_hash** — a hash over a sparse sample of the synthesized sequence (~1 sample per 50 bp). Detects large-scale edits such as recombinase-based segment replacement or assembly events. Sparse-sampling renders seq_hash deliberately insensitive to point substitutions which routinely occur due to replication error or deliberate mutagenesis.
- **h_sig** — a 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 that prevents an attacker from independently regenerating internally consistent hashes; the ledger entry can also publish non-proprietary synthesis data such as total sequence length and (optionally) full landmark records.

A modified sequence yields an invalid seq_hash and h_sig — a yellow flag. A tampered synthesizer yields an invalid mech_hash and h_sig — a red flag. Only a narrow class of technically demanding

plasmid modifications can downgrade a red flag to a yellow flag **(See Appendix 2 for the formal proof; See Appendix 4 for the verification truth table)**. Landmark maps, described next, meaningfully constrain these attacks.

3.2 Landmark Map

A landmark map is a sparse forensic fingerprint of the synthesized molecule. The encoder runs a simple deterministic algorithm that picks ten anchor sites in the sequence based on local content (typically common restriction enzyme sites), 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 [6], 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 about the sequence (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 single 5' base value at each landmark, with no site or position information. Section 4 demonstrates that meaningful forensic reconstruction is possible from this minimal barcode-only representation, providing a privacy-preserving fallback when ledger landmark publication is unacceptable.

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-protected. These fields enable ledger lookup, forensic tracing, and routine research uses such as contamination tracking and reproducibility checks. The dual-use design aids both investigators and legitimate researchers, incentivizing voluntary adoption.

3.4 Primer-Flanked Architecture

The barcode is flanked by two primer-binding sequences, ideally brand-specific and registered with compliant primer-synthesis services as flagged, similar to DHHS screening for primers matching the sequences of dangerous pathogens [9]. Beyond their forensic role as PCR amplification handles, the primers act as supply-chain choke points: 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 Sequence Motif Blacklist and Resampling

The barcode is sanitized of forbidden motifs from a curated, user-modifiable blacklist (the demo uses ~100 common $\geq$6 bp restriction sites from BioPython's CommOnly database). Constraint satisfaction uses a deterministic resampling loop seeded by the 8-bit increment: each iteration produces a fresh nucleotide-encoding dictionary and 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 an attacker's easiest molecular handle for editing the barcode and prevents interference with downstream legitimate cloning.

3.6 Security Architecture

Together, these mechanisms defend against most attacks. Any adversary attempting to produce a non-red-flag barcode for an unauthorized plasmid faces a layered set of obstacles. While it is mathematically impossible to prevent an attacker from forging a yellow-flag-only construct (See Appendix 2), in practice the attacker must reproduce not only a valid barcode sequence but a host plasmid whose landmarks match the embedded — and ideally, ledger-recorded — landmark pattern. See Appendix 4 for the truth table mapping attack class to verifier signal.

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 GC content $\in$ [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 testing is identified as future work.

4.1 Encoding Yield

We measured the rate at which the encoder finds a constraint-clean barcode within the 256-attempt increment search defined in Section 3.5. 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 $\leq$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 is sufficient headroom across the tested length and GC range. The empirical absence of failures argues for the constraint set being practically satisfiable as designed; whether stricter sequence constraints or extreme GC stress this further is a natural follow-up.

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. Three signals were measured independently: the plaintext length flag, the sparse-sample seq_hash flag, and the landmark-mismatch flag.

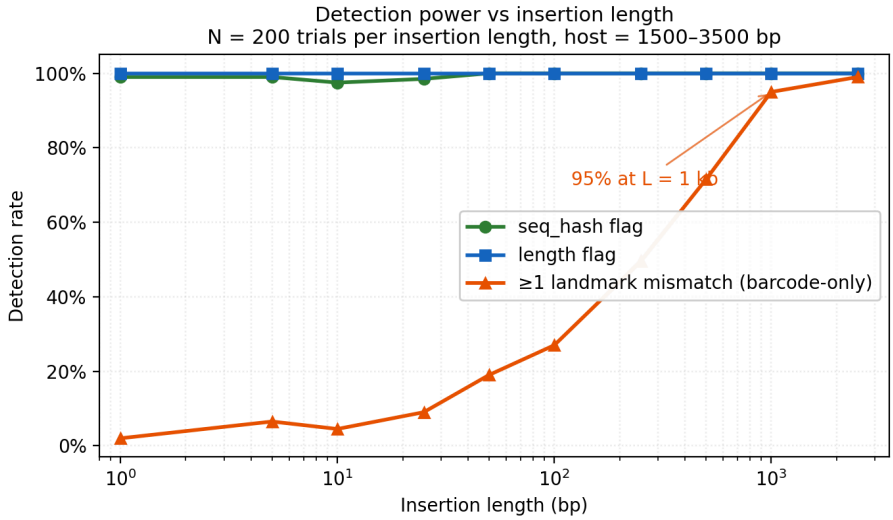

Figure 2: Detection rate as a function of insertion length L, across three verifier signals. Length and seq_hash detect ≥97% of insertions from L = 1 bp upward. The barcode-only landmark signal scales monotonically with L, exceeding 95% by L = 1 kb. N = 200 trials per insertion length; host plasmid length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

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 ≥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: this rate climbs monotonically with insertion length, reaching 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 (Figure 2). This validates the central empirical claim in Section 5.5 — that meaningful forensic detection of transplant-scale modifications is possible from the 10-base barcode-embedded fingerprint alone, without any ledger landmark publication. At the ~700 bp scale of the case study (Section 5.1), the expected detection rate from this signal alone 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 in which an attacker attempts to disguise a dangerous payload behind a legitimate STAMP attestation, then walk through how the verifier — a forensic investigator — interprets the resulting evidence. The attack illustrates STAMP's central thesis: STAMP barcodes present a combination of seq_hash mismatch, length mismatch, and landmark forensics, producing a structured evidence trail that constrains an investigator's interpretation to a narrow set of plausible modification histories.

5.1 Setup

An attacker generates a legitimate plasmid (GFP) from an accredited synthesizer, producing a STAMP-attested construct. The legitimate barcode (gray) and landmark features (orange chevrons) are visible on the center plasmid maps in Figure 3. In parallel, the attacker obtains — from a black-market jailbroken synthesizer or a naturally-derived source — a copy of a dangerous gene (in our demo, harmless mCherry standing in for a dangerous payload). The attacker uses restriction cloning to ligate the payload into the stamped GFP backbone, producing a chimeric plasmid (Figure 3, right). This immediately invalidates the barcode on multiple overlapping levels: sequence length changes, the sequence hash is invalidated, and the insertion distorts landmark features.

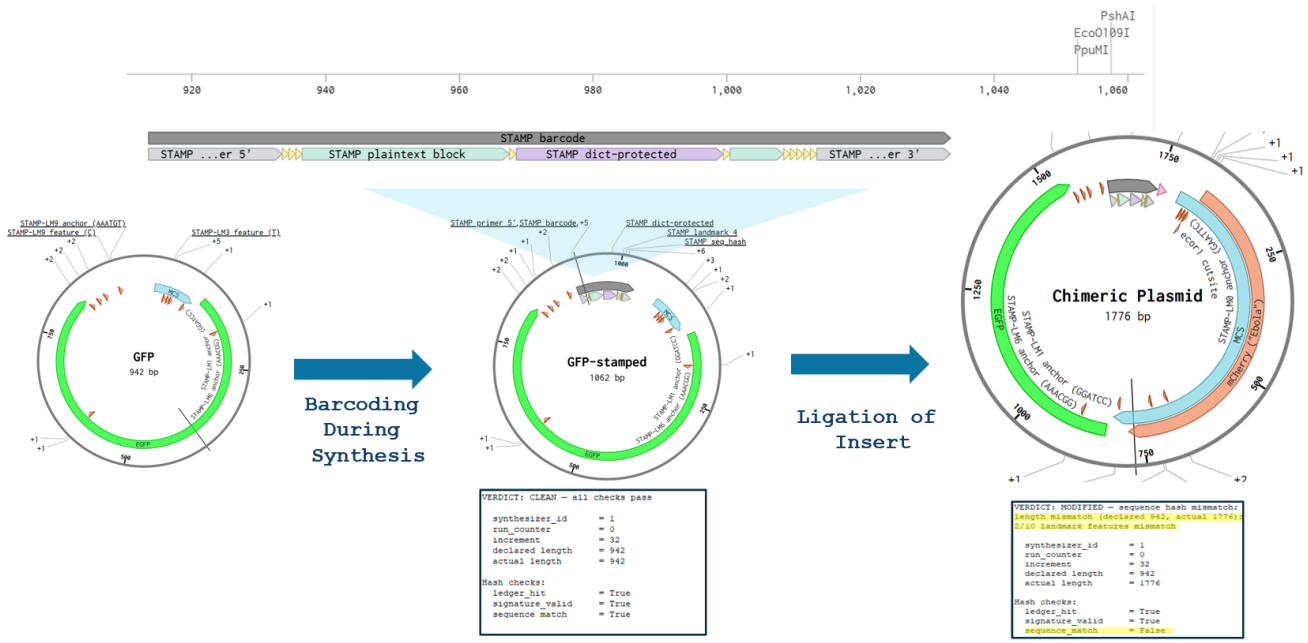

Figure 3. *Case study workflow. Left: original 942 bp GFP plasmid. Center: GFP-stamped construct after barcode insertion (1062 bp; verifier returns CLEAN, all checks pass). Right: chimeric plasmid after attacker ligates dangerous payload into the GFP-stamped backbone via EcoRI (1776 bp; verifier returns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

5.2 Verifier Output

The attackers, unable to easily correct the sequence mismatch, are detected by a compliant sequencer. An investigator arrives and discovers similar recombinant plasmids on used pipette tips, and waste stream of a clandestine laboratory. Without a universal ledger of all plasmids ever constructed, the investigator cannot immediately determine whether these chimeric plasmids were produced by hybridization or by total synthesis on a fully jailbroken synthesizer. However, the barcode gives them immediate access to the public ledger entry for the original synthesis event, which contains the machine ID, run counter, timestamp, original sequence length, and full landmark records. The verifier panel displays:

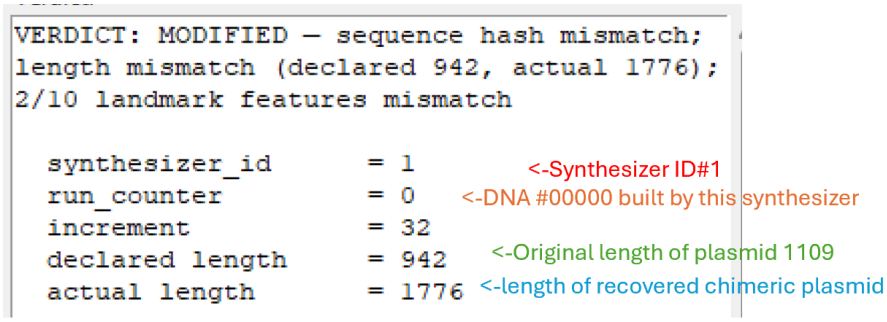

Figure 4. *Verifier output panel for the chimeric plasmid. The barcode reports synthesizer #1, run counter 0, original length 942 bp; the recovered suspect sequence is 1776 bp, and 2/10 landmarks show feature mismatches. Note that 942 + 120 (barcode) + 714 (insert) = 1776 ✓.*

The plasmid was originally made by SynthesizerID #1 during run #0. It was 942 bp + 120 bp barcode = 1062 bp at synthesis time, but is now 1776 bp — implying 714 bp of new sequence has been added. To localize the insertion, the investigator uses the landmark forensic panel:

5.3 Landmark Forensic Reconstruction (with Ledger Access)

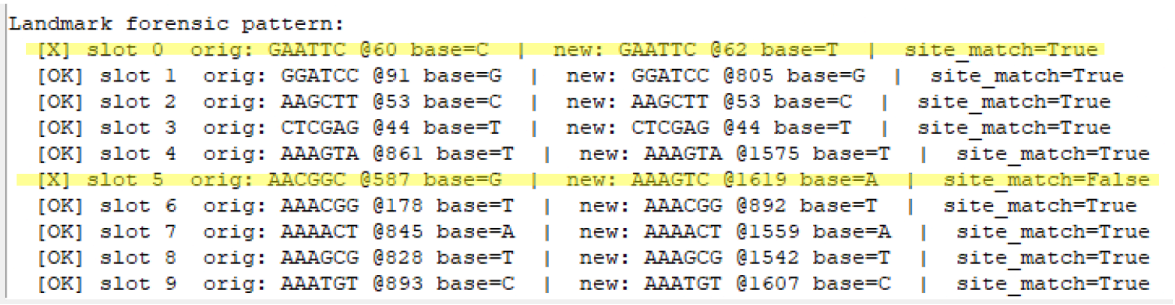

Figure 5. *Landmark forensic readout. Slot 0 (EcoRI, GAATTC) retained its anchor site at a near-original position (60 → 62) but its 5' base changed from C to T, suggesting EcoRI was used to cut and ligate. Slot 5 changed identity entirely (AACGGC → AAAGTC) at a position deep inside the insert region.*

Two of ten landmarks show feature mismatches:

- **Landmark 0 (EcoRI cut site, GAATTC)**. Site identity unchanged at near-original position (60 → 62). The 5' base changed from C to T, consistent with EcoRI being used to cut and ligate at this site.
- **Landmark 5**. Original anchor (AACGGC at position 587) replaced by a higher-priority anchor (AAAGTC at position 1619). A new landmark site was created at position 1619, most parsimoniously by the insertion of a new sequence containing AAAGTC.

To verify and localize the insertion, the investigator examines the displacement pattern across all ten landmarks, sorted by original position:

Landmark #	Original Position (bp)	New Position (bp)	Movement (bp)	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

Figure 6. *Landmark displacement table. Six landmarks (1, 6, 8, 7, 4, 9) all displaced at exactly +714 bp; landmarks2 and 3 unmoved(upstream of cut); landmark 0 shifted +2 bp at the cut site; landmark 5 displaced by +1032 pb because it now anchors to a novel site inside the insert.*

5.4 Investigator's Conclusion

The pattern is unambiguous. A single 714 bp insert was added at approximately position 60, using EcoRI as the 5' cut site, displacing six downstream landmarks by exactly 714 bp. Landmarks 2 and 3, upstream of the cut site, were unaffected. Landmark 5 anchors to a site inside the new insert (offset +1032 from its original position, consistent with a novel match within the inserted material). The 5' base change at landmark 0 from C to T identifies EcoRI as the cut site used. Other common cut sites whose recognition sequences are higher-priority landmarks — HindIII, BamHI, XhoI — were not disrupted, ruling them out as the 5' cloning site. 3’ cut site is unknown but can be narrowed to a short list of probable candidates.

Because the surviving GFP-overlapping landmarks all moved by exactly the same offset, the investigator concludes with high confidence that GFP was the original payload, that the backbone was largely unmodified outside the insertion, and that EcoRI plus a small set of candidate enzymes (Acc65I, KpnI, SacII, PspOMI, etc.) were used for cloning. Combined with the ledger's record of the synthesis machine, run counter, and timestamp, this constrains the chain of custody to a narrow set of physical possibilities, transforming a molecular trail into a paper one. All conclusions match our blinded ground-truth: a 714 bp restriction-cloning insertion of a payload into the GFP MCS via EcoRI and Acc65I-mediated cuts, with minimal cloning scar.

5.5 Forensic Reconstruction Without Ledger Access

A privacy-preserving deployment may forbid the ledger from storing landmark site or position information, leaving the investigator only the 10 base values physically embedded in the barcode. We replicate the analysis under this constraint (for full reasoning see Appendix 5) which yields the same conclusion as the ledger-based analysis: GFP was the original payload, EcoRI was used at one end, a single ~714 bp insert was placed near landmark 0, and roughly half a dozen MCS enzymes are candidates for the 3' cut. Confidence is lower, but the modification class and approximate location are recovered.

This is a central empirical finding of the case study. The privacy-preserving 10-base landmark fingerprint is sufficient to reconstruct the modification class, approximate location, and probable cloning enzyme used, without any ledger landmark publication. Forensic richness scales with ledger access, but does not collapse without it.

6. Discussion and Limitations

6.1 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 ~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.

6.2 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.

6.3 Privacy and Voluntary Adoption

STAMP's sparse-hash architecture is privacy-preserving by design. The seq_hash reveals nothing useful about sequence content even if cryptographically broken; barcode-stored landmark features encode only 10 bases of structural fingerprint, far below any meaningful sequence-reconstruction threshold. The case study (Section 5.5 and See Appendix 5) demonstrates that this minimal representation is forensically meaningful, providing a deployable privacy-preserving option for sectors where ledger landmark publication is unacceptable. Plaintext metadata provides genuine value to researchers — contamination tracing, reproducibility documentation, automatic chain-of-custody for sequencing core submissions — and we expect this dual-use value to drive voluntary adoption ahead of regulatory requirements.

Limitations

1. **HSM dependence.** STAMP requires a minimal HSM implementation that does not yet ship in benchtop synthesizers. Given the modest hardware requirements and active industry interest, we are comfortable assuming an adequate system will be developed.
2. **Two-synthesizer transplant attack.** As stated in the introduction, 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 is permitted to record 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. When ledger landmark publication is forbidden for privacy reasons, this attack reduces to forging and ligating a barcode sequence, which is technically possible but molecularly difficult, especially for an attacker without direct synthesis access. We argue STAMP's design is close to the theoretical limit of physical-molecule security.
3. **Size limitations.** A ~120 bp tag is non-trivial overhead for very small constructs. We recommend STAMP enforcement only for sequences $\geq$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.

4. **Economic cost.** As has been previously noted in a screening context, control measures including STAMP impose economic costs on laboratories and biotechnology companies, which often operate on narrow margins[10]. We mitigate this cost by keeping the STAMP itself deliberately small, and integrating it with useful metadata features that aid voluntary adoption.
5. **Hash truncation.** STAMP uses 24-bit truncated SHA-256 throughout, below cryptographic standards in isolation. We accept this conservatism for barcode-budget reasons; *in vitro* implementation of a hash-collision attack remains substantially harder than the cryptographic component alone, and a single extra base per hash would meaningfully increase resistance if needed.
6. **Insensitivity to small modifications.** Sparse seq_hash sampling cannot detect point mutations. This is deliberate — small modifications are usually benign and frequent — but means STAMP is not a tool for fine-grained engineering detection. The landmark system is similarly limited to broad characterization of large modifications.

Future Work

Future work could benchmark STAMP against a broader suite of attack types, formalize confidence intervals for barcode-only forensic reconstruction, and integrate ML interpretation of landmark-mismatch patterns. Richer landmark encodings (more landmarks, multi-base context, spatial-order) could likely enhance forensic reconstruction capabilities at minimal cost to base budget. 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 for establishing provenance and forensic documentation of synthesized DNA: a cryptographic triad provides tamper-evident attestation, a content-aware landmark map enables forensic reconstruction of post-synthesis modifications, and plaintext metadata aids both forensic tracing and routine research. STAMP's value is best understood not as a lock but as a cost imposer and evidence generator. In the realistic low-compliance case it provides forensic trails, supply-chain visibility, economic friction, and post-hoc investigative material that meaningfully raise the cost of malicious synthesis. In the high-compliance case it substantially raises the cost of the iterative-development pathway that makes AI-assisted threat design dangerous. The case study demonstrates that even the privacy-preserving variant which relies only on 10 bases of barcode-embedded landmark fingerprint can support meaningful forensic reconstruction.

We would be remiss not to mention the ethical gray zone generated by cryptographically tagging genetic material, especially in an ecosystem of universally enforced compliance. This has deeply uncomfortable implications for the tracking and copyrighting of genetic constructs and engineered organisms. We have built the technical primitive for biological provenance tracking — whether that

is a forensic tool, a compliance infrastructure, or the substrate for something more concerning depends entirely on policy decisions we explicitly are not making here.

Code and Data

- **Github repository**: https://github.com/nal060/hashtag
- **Walkthrough video**: https://www.loom.com/share/cdab0544c6774450a7fd2dc46eb6a6ab
- **Interactive Streamlit demo**: https://stamped.streamlit.app

References

1. Anguzu S. Preventing Biosecurity Risks Posed by Next-Gen Benchtop DNA Synthesizers. Health Economics and Management Review. 2025;6(1):126–143. https://armgpublishing.com/journals/hem/volume-6-issue-1/article-9/
2. Langenkamp M. Securing Benchtop DNA Synthesizers. *Institute for Progress*, December 10, 2024. https://ifp.org/securing-benchtop-dna-synthesizers/
3. Oxford Nanopore Technologies. Sequencing devices and platforms. https://nanoporetech.com/products/sequence
4. Heider D, Barnekow A. DNA watermarks: A proof of concept. BMC Molecular Biology. 2008;9(1):40.
5. Lee SH. DWT-based coding DNA watermarking for DNA copyright protection. Information Sciences. 2014;273:263–286.
6. Gretton D, Esvelt KM. Exact-match search with functional variant prediction enables automated DNA screening. bioRxiv 2024.03.20.585782
7. Hamad S, Elhadad A, Khalifa A. DNA Watermarking Using Codon Postfix Technique. IEEE/ACM Trans Comput Biol Bioinform. 2018 Sep-Oct;15(5):1605-1610. doi: 10.1109/TCBB.2017.2754496. Epub 2017 Sep 20. PMID: 28945600.
8. International Gene Synthesis Consortium (IGSC). *Harmonized Screening Protocol, Version 3.0*. [Internet]. [place unknown]: IGSC; 2024 Sep 3 [cited 2026 Apr 26]. Available from: https://genesynthesisconsortium.org/wp-content/uploads/IGSC-Harmonized-Screening-Protocol-v3.0-1.pdf
9. U.S. Department of Health and Human Services, Administration for Strategic Preparedness and Response (ASPR). *Screening Framework Guidance for Providers and Users of Synthetic Nucleic Acids* [Internet]. Washington, DC: HHS; 2023 Oct [cited 2026 Apr 26]. Available from: https://aspr.hhs.gov/S3/Documents/syndna-guidance.pdf
10. Diggans J, Leproust E. Next Steps for Access to Safe, Secure DNA Synthesis. Front Bioeng Biotechnol. 2019 Apr 24;7:86. doi: 10.3389/fbioe.2019.00086. PMID: 31069221; PMCID: PMC6491669
11. Carter SR, Yassif J, Isaac C. *Benchtop DNA Synthesis Devices: Capabilities, Biosecurity Implications, and Governance*. Washington, DC: Nuclear Threat Initiative (NTI); 2023 May. Available from: https://www.nti.org/wp-content/uploads/2023/05/NTIBIO_Benchtop-DNA-Report_FINAL.pdf
12. National Academies of Sciences, Engineering, and Medicine; Committee on Assessing and Navigating Biosecurity Concerns and Benefits of Artificial Intelligence Use in the Life Sciences. *The Age of AI in the Life Sciences: Benefits and Biosecurity Considerations*. Washington (DC): National Academies Press (US); 2025 Apr 23. Available from: https://pubmed.ncbi.nlm.nih.gov/40392968/. doi: 10.17226/28868
13. Brent R, McKelvey G Jr. *Contemporary Foundation AI Models Increase Biological Weapons Risk*. Santa Monica, CA: RAND Corporation; 2025. (RAND Perspectives; PE-A3853-1). Available from: https://www.rand.org/pubs/perspectives/PEA3853-1.html.
14. Wheeler N. Responsible AI in biotechnology: balancing discovery, innovation and biosecurity risks. *Front Bioeng Biotechnol*. 2025 Feb 5;13:1537471. doi: 10.3389/fbioe.2025.1537471. PMC118358
15. Adam L, McArthur GH. Substitution attacks: a catalyst to reframe the DNA manufacturing cyberbiosecurity landscape in the age of benchtop synthesizers. *Appl Biosaf*. 2024 Sep 18;29(3):172–180. doi: 10.1089/apb.2023.0035. PMC11447128.

16. Au KF, et al. Nanopore sequencing technology, bioinformatics and applications. *Nat Biotechnol.* 2021;39:1348–1365. doi: 10.1038/s41587-021-01108-x.
17. Institute for Progress. *Securing Benchtop DNA Synthesizers*. Washington, DC: IFP; 2024 Dec. Available from: https://ifp.org/securing-benchtop-dna-synthesizers/
18. Stillman C, Bravo JE, Boucher C, Rampazzi S. Toward security-aware portable sequencing. *Nat Commun.* 2025;16:9829. doi: 10.1038/s41467-025-66024-z. PMC12603292.

Appendix

Appendix 1: Detailed Technical Specification

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 ~100 common cutters of length $\geq$6 bp. The encoder tries increments 0..255 and accepts the first that produces a constraint-clean barcode.

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 (~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.

Landmark Map
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 ~2048 canonical 6-mers. These are split evenly into 10 mutually exclusive priority lists of ~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.

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

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.

Appendix 2: 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 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:

$$\begin{aligned} &sig \text{ validates correctly} && \rightarrow \text{same in both cases} \\ &H(seq) \neq H(X) && \rightarrow \text{same in both cases} \\ &ledger\_entry \text{ points to } X && \rightarrow \text{same in both cases} \\ &seq \text{ and } seq' \text{ are both } \neq X && \rightarrow \text{same in both cases} \end{aligned}$$

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

Corollary 1.1. *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.*

Appendix 3: 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.

Appendix 4: 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 (See Appendix 2) 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.

STAMP VERIFICATION TRUTH TABLE

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

Figure 7 : STAMP verification truth table mapping verifier output patterns to attack classes. Pass/fail check is not the verdict—specific pattern of passes and failures is diagnostic.

Appendix 5: 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 (Fig 7, 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

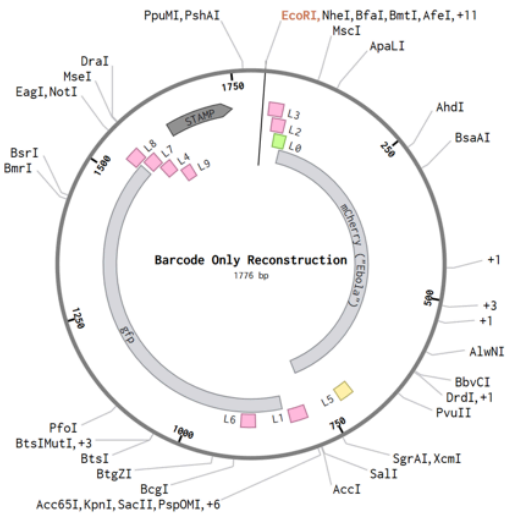

Figure 8. *Barcode-only reconstruction. Left: landmark table sorted by current position; only landmark 0 (EcoRI cut site) and landmark 5 (novel site, 5' base changed) deviate from baseline. Right: chimeric plasmid map. Pink rectangles = unchanged landmarks; green = EcoRI cut site; yellow = novel landmark in insert; orange = displaced. The single sufficient gap to fit a 750 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.

Appendix 6: Limitations and Dual-Use Considerations

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 ~200 GB/year at the current schema. Tractable for transparency-log infrastructure but non-trivial for global federation.

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 5.5), 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 (See Appendix 4), 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.

Responsible disclosure
We did not discover deployment vulnerabilities in existing systems during this work. The closest item: STAMP's impossibility-theorem gap (See Appendix 2) 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.

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.

Suggested future improvements
- Real cryptographic primitives (Ed25519 + transparency log) replacing the demo mocks.
- Formal characterization of barcode-only forensic reconstruction confidence (See Appendix 5).
- 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.

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.