EmbryoLock

Irreversible Failure as a Deliberate Security Primitive

Version 1.0 — Position & Threat-Model Paper

Status: Public Technical Note

Audience: Security practitioners, system designers, researchers

Abstract

Most security systems are designed around recovery: retries, lockouts, backups, and remediation after failure. This assumption is appropriate for the majority of threat models. However, there exist narrow scenarios where recovery itself becomes a liability—where post-compromise persistence, coercive access, or asymmetric harm makes guaranteed non-disclosure more valuable than availability.

This paper explores irreversible failure—the deliberate destruction of protected data upon authentication failure—as a constrained security primitive rather than a default control. We examine the threat models where such behavior may be rational, the risks and failure modes inherent in its use, and the design constraints required to prevent misuse. EmbryoLock is presented as a small experimental case study illustrating these ideas, not as a general-purpose solution.

1. Problem Framing

Modern endpoint security overwhelmingly optimizes for business continuity and user recovery. Common defensive patterns include:

authentication retries with lockout thresholds

time-based delays and rate limiting

escrowed recovery keys

backup and restore mechanisms

These patterns assume that:

Loss of availability is worse than loss of confidentiality

Compromise can be detected and remediated after the fact

The user remains in control of the endpoint

In many real-world scenarios, these assumptions hold. In others, they fail catastrophically.

2. When Recovery Becomes a Liability

There exist threat environments where the primary failure mode is not accidental data loss, but forced disclosure. Examples include:

coercive access (legal, physical, or social pressure)

post-compromise persistence by a determined adversary

adversaries with unlimited offline time

asymmetric harm (where disclosure causes irreversible damage, but data loss does not)

In these models, availability becomes a secondary concern. The objective is not to preserve data at all costs, but to guarantee that compromise yields nothing.

3. Threat Models Where Destruction May Be Rational

This paper does not argue that destruction-on-failure should be common. It argues only that it may be defensible in tightly scoped cases, such as:

Single-user, local-only data stores

Data whose disclosure produces non-recoverable harm

Environments where compromise is assumed, not prevented

Systems explicitly designed to resist coercion rather than intrusion

Scenarios where the user explicitly accepts irreversible loss

In these contexts, destruction is not a bug—it is the security outcome.

4. Failure Modes and Objections

Any system that destroys data on authentication failure introduces serious risks. These must be acknowledged explicitly.

4.1 False Positives

Legitimate users may lose data due to:

typing errors

memory lapse

environmental factors

software bugs

This risk alone disqualifies such systems from broad adoption.

4.2 Denial-of-Service Attacks

If destruction can be triggered remotely or repeatedly, an attacker can weaponize it to destroy data without gaining access.

4.3 Automation Abuse

Automated triggering at scale (credential stuffing, scripted retries) can render the system unusable.

4.4 Operational Unsuitability

Such designs are incompatible with:

enterprise continuity requirements

shared environments

regulatory recovery mandates

These objections are valid. Any system ignoring them is irresponsible.

5. Design Constraints for Any Viable Implementation

If irreversible failure is to be used at all, it must obey strict constraints:

Hostility to Automation

The system must not tolerate repeated or rapid attempts.

Non-Repeatability

Once triggered, the failure must not be reversible or replayable.

Explicit User Consent

Users must knowingly accept irreversible loss.

Narrow Applicability

The system must clearly state where it should not be used.

Zero Recovery Path

Backdoors, escrow keys, or silent recovery undermine the premise.

Failure to meet these constraints converts destruction from defense into self-harm.

6. EmbryoLock as a Case Study

EmbryoLock is a small local experiment designed to explore these principles.

Its design choices include:

local-only operation

no remote services

no recovery keys

irreversible data destruction after defined authentication failure

EmbryoLock is not presented as a recommended solution. It is an artifact used to reason about the implications of deliberate irreversibility in security design.

7. What This Is Not

To prevent misapplication, it is important to state explicitly what this model does not support:

not consumer-grade security

not enterprise data protection

not ransomware mitigation

not backup replacement

not safe for shared systems

Any attempt to generalize this model beyond narrow threat scenarios is likely harmful.

8. Ethical and Practical Boundaries

The ethical risk of irreversible systems is non-trivial. Designers must consider:

informed consent

power asymmetry

misuse by third parties

false confidence in “absolute” security

Irreversibility should raise the bar for justification, not lower it.

9. Open Questions

This paper intentionally leaves several questions unresolved:

Can intent be reliably distinguished from automation?

What non-repeatable triggers are ethically defensible?

Where should responsibility lie for false destruction?

Are there hybrid models that preserve guarantees without full loss?

These are research questions, not settled conclusions.

10. Conclusion

Irreversible failure is not a general security solution. It is a narrow primitive that trades availability for certainty under extreme threat models. Most systems should reject it outright. A few may rationally accept it.

The purpose of this paper is not to advocate adoption, but to clarify the trade space—so that when such designs appear, they are evaluated with precision rather than reflex.

Status Statement

This document defines EmbryoLock as a conceptual and experimental exploration of irreversible failure in security design. It does not claim novelty, universality, or superiority over established practices.