

0.1 Claims

1. A computer-implemented system for producing falsifiable process evidence, comprising:
 - (a) a capture component configured to observe and record process state for a digital artifact;
 - (b) a time-locking component configured to bind successive checkpoints to a non-back-dateable timeline;
 - (c) a device-binding component configured to associate checkpoints with a device-specific signal or attestation; and
 - (d) an append-only evidence log configured to store checkpoints such that modification or deletion of prior checkpoints is detectable, wherein each checkpoint is cryptographically linked to a prior checkpoint and to a capture environment declaration.
2. The system of claim 1, wherein the time-locking component comprises a verifiable delay function.
3. The system of claim 1, wherein the time-locking component comprises sequential hashing, chained hash delays, or a trusted monotonic counter.
4. The system of claim 1, wherein the append-only evidence log comprises a Merkle Mountain Range.
5. The system of claim 1, wherein the append-only evidence log comprises a hash-chained log with periodic authenticated roots.
6. The system of claim 1, wherein the device-binding component comprises a physical unclonable function, a trusted platform module, a secure enclave, or a device-unique key store.
7. The system of claim 1, wherein the device-binding component includes behavioral binding derived from real-time interaction timing.
8. The system of claim 7, wherein the behavioral binding comprises cryptographic jitter values derived from a session secret and a document state hash, and wherein the jitter values form a chained proof of process activity.
9. The system of claim 1, wherein each checkpoint is signed with a scoped session key and the scoped session key is ratcheted forward and destroyed after use.
10. The system of claim 1, wherein the capture component records at least one of: document hashes, file metadata, interaction timing, or environment measurements.
11. The system of claim 1, further comprising an export component configured to generate a portable evidence packet including at least one of: an inclusion proof, a time-lock proof, a device attestation, or a verification transcript.
12. The system of claim 1, wherein the capture environment declaration includes at least one of: operating system version, kernel version, secure boot status, virtualization indicator, or executable hash, and explicitly records unavailable fields.
13. The system of claim 1, further comprising an external anchoring component configured

- to bind a checkpoint root to a public time source.
14. A computer-implemented method for producing falsifiable process evidence, comprising:
 - (a) recording a capture environment declaration at session start;
 - (b) capturing process state and producing a checkpoint representing the process state;
 - (c) applying a time-locking function to the checkpoint to enforce a minimum elapsed time or non-back-dateable ordering;
 - (d) applying device binding to the checkpoint using a device-specific signal or attestation;
 - (e) appending the checkpoint to an append-only authenticated log; and
 - (f) exporting a portable evidence packet with verification data.
 15. The method of claim 14, wherein the time-locking function is implemented using a verifiable delay function, sequential hashing, or a trusted monotonic counter.
 16. The method of claim 14, wherein device binding includes behavioral binding derived from interaction timing and cryptographic commitment to a document state hash.
 17. The method of claim 14, wherein each checkpoint is signed with a scoped session key and key lifecycle metadata is recorded and bound to the evidence packet.
 18. The method of claim 14, further comprising anchoring a checkpoint root to a public ledger or a trusted timestamp authority.
 19. The method of claim 14, wherein the evidence packet includes a verification transcript or evidence class indicator.
 20. A non-transitory computer-readable medium storing instructions that, when executed by one or more processors, cause the processors to perform the method of claim 14.