TECHNICAL EVIDENCE WHITE PAPER

HTTP6

Payload-Bound Per-Message Authorization

Rosario–Wang

Mathematical guarantees, executable evidence, and technical evaluation

Prepared for NIST technical reviewers evaluating a proposed security architecture

Frank Dylan Rosario

November 9, 2026

Technical review edition 1.0

Author-review submission candidate. No submission, endorsement, or certification is represented.

Abstract

HTTP6 is proposed as an application-channel architecture in which each protected request and response carries a fresh authorization proof bound to its exact payload, executing instance, peer, inherited policy, and one-use event. This paper evaluates the evidence supplied by a runnable Rosario–Wang reference demonstrator. The implementation generates actual calibrated, privately masked membership witnesses and authenticates each witness together with the complete application statement through an independently keyed HMAC. A designated verifier checks the statement, consumes the event, validates the native response, and applies inherited permissions before recording an effect. The supporting mathematics establishes honest correctness, authorized recovery, and perfect passive credential privacy for the specified fresh-mask observation model. Sequential composition preserves that privacy for every supported finite history, including one trillion observations, with an explicit private-randomness budget. A reproduction run completed 69 regression tests and eight request/response exchanges with 18 accepted native proofs and eight local effects. The paper distinguishes these results from native-channel encryption, hardware-backed interaction, distributed state, and broader active-security analysis. Its contribution is an inspectable per-message authorization mechanism and a precise evidence basis for further technical evaluation. [D1–D6; F1]

Keywords: HTTP6; Rosario–Wang; human admission; per-message authorization; payload binding; transcript privacy; hereditary authority; replay resistance; designated verification.

Edition and evidence basis

This white paper adapts the preceding explanation for a technical-review audience while preserving its substantive conclusions, examples, equations, numerical results, and evidence distinctions. Appendix A retains that explanation's wording; citation markers and mathematical delimiters are normalized for publication. An unchanged transcription of the explanation is also included in the source package. Appendix B identifies the executable attachments and reproduction procedure. No NIST acceptance, endorsement, assigned protocol version, or completed production assessment is represented by this edition.

The main text uses four evidence categories: **mathematical result**, for a deduction under stated premises; **reference implementation**, for behavior implemented in the supplied source; **executed observation**, for a retained test result; and **deployment evaluation**, for evidence to obtain from a specified operational implementation. The supplied formalizations use these distinctions explicitly. [F1, §§1–5, 11–15; F2, §6; F3]

Contents

Abstract i

Executive finding 1

1. Contribution, terminology, and review scope 1

2. Executed reference evidence 2

3. A proof bound to the exact payload 3

4. Native correctness, recovery, and credential privacy 4

5. Entropy, finite capacity, and efficiency 5

6. One-use authority and hereditary operation 6

7. Technical evaluation requested 7

8. Conclusion 8

Appendix A. Preserved original explanation 9

Appendix B. Reproduction and artifact index 14

References and source register 16

Executive finding

The reference demonstrator implements a fresh, payload-bound authorization proof for each application request and each application response. The complete payload is included in the authenticated statement, an actual Rosario–Wang masked witness is generated, and both are checked before the operation is permitted. The result is a working proof-per-message authorization model supported by a conditional mathematical proof of credential privacy. [D1, `Peer.respond`, `Runtime.challenge`, `Runtime.submit`; D3, §§A, E–G]

The demonstrated construction combines several independently necessary mechanisms. The native witness protects and verifies the enrolled relationship under its stated model. The HMAC envelope authenticates the complete message context. The consumed-event ledger enforces one-use verification. The inherited-policy gate authorizes the requested effect. The service's response carries a separate native proof and refers to the accepted request. Each mechanism has an explicit role; the credential-privacy theorem is not used as a substitute for message integrity, authorization, or transport confidentiality. [D3, §§D–G]

For review purposes, the central conclusion is affirmative and bounded: **payload-specific proof generation and validation are executable in the supplied model; the complete native HTTP6 channel is a subsequent implementation and evaluation target.** The remaining work is identified through concrete deployment interfaces rather than an undifferentiated assertion about the security of the whole Internet protocol.

1. Contribution, terminology, and review scope

The contribution under evaluation is the composition of a source-defined private-reference mechanism with a new message-binding and authorization envelope. The Human Admission Client is the human-facing interface in the surrounding architecture. Rosario–Wang supplies the finite membership encoding and fresh masking relation. The HTTP6 reference architecture supplies the proposed execution and authority components (Proof Authority, Minting and Delegation Service, Policy Decision Point, and Consumption Ledger). The current executable models a synthetic human admission, an independently credentialed root, a permitted child, and request/response messages under distinct references. It does not copy or derive the human credential into the child. [D3, §G; D4, §2]

The term **per-message proof** in this paper identifies the designated-verifier proof object actually evaluated by the demonstrator. It comprises the application statement, native masked witness, and envelope authenticator. A conventional general proof-of-knowledge claim requires its own statement–witness relation, arbitrary-prover model, extractor, knowledge error, and composition argument. The formal source keeps those stronger classifications separate from honest correctness, passive privacy, and accepted-transcript recovery. [F1, §§11–12]

Likewise, "per message" means per protected application record. It does not mean one proof per byte or automatically one proof per underlying network packet. The current review codec carries bounded canonical JSON records. A large streamed object would require a separately

specified record or chunk format that retains the intended payload and ordering bindings. [D4, §1]

The native membership word and the application payload have different roles. The enrolled string and private dictionary determine the native word. The full payload is included in the surrounding authenticated statement. Consequently, the claim is not that an arbitrary HTTP body is independently authenticated by the bare geometric response. The demonstrated payload binding is supplied by the composed object described in Section 3. [D3, §E]

2. Executed reference evidence

The supplied package records 69 successful regression tests, exhaustive finite privacy checks, a local workflow, and a loopback HTTP carrier experiment. During preparation of this edition, the unchanged package was extracted into a separate build directory, its regression suite was rerun, and an eight-message loopback demonstration and the exact privacy audit were executed again. The newly retained reproduction records are distinguished from the package's original reports and primitive benchmark. These are local reproduction results, not independent laboratory assessment. [D5; D6]

Operation or attempted attack	Observed reference result
Initial synthetic human admission	Accepted
Root authorizes a distinct child	Accepted
Eight application requests	Eight fresh native proofs accepted
Eight corresponding responses	Eight separate fresh native proofs accepted
Replay of an accepted request	Rejected; no additional local effect
Substitution of the destination	Rejected
Authenticated child requests prohibited export	Rejected
Response attempted under the wrong local channel	Rejected
Prepared request submitted after parent revocation	Rejected

The run produced **18 accepted native proofs**: one admission, one minting operation, eight requests, and eight responses. It produced **exactly eight local application effects**. The human participant was synthetic and automated. The carrier was a local HTTP/1.1 listener, and the effect was a local database record. The logical client and verifier remained part of the reference harness; a network exchange does not by itself make their private stores independently isolated. [D6; D4, §§1–2]

The source's finite audit enumerates ideal mask spaces rather than inferring zero information from a sample histogram. Its reported checks include 2,048 binary executions, 294,912 RW-12/4 credential/mask executions at one nonce, 2,048 two-event executions, 56 balanced/separating nonce profiles, and 864 calibrated RW-40 recoveries. These checks support finite implementation correspondence. The general mathematical conclusions retain the proofs and premises in Section 4. [D5; D6; F1]

3. A proof bound to the exact payload

For event e, the implemented message proof object is represented conceptually as

$$\mathscr{P}_e = (x_e, Z_e, \tau_e). \tag{1}$$

The statement x_e identifies the sender, recipient, channel, direction, stream, sequence, policy and lineage, operation, actual payload, nonce, and expiration, together with the declared profile and challenge data. The vector Z_e is the fresh Rosario–Wang masked membership witness. The tag τ_e authenticates the statement and witness together. The actual assigned mask allocation remains private setup material; it is not supplied by an observer who chooses a compatible explanation. [D3, §A]

The independently keyed envelope is

$$\tau_e = \mathrm{HMAC}_{a_v}(\mathrm{Encode}(\texttt{witness-envelope}, x_e, Z_e)) . \tag{2}$$

The key a_v is provisioned independently of the native credential and mask tape. The encoding fixes the bytes being authenticated, and the domain label identifies the envelope's purpose. Because the payload is inside x_e, the payload is inside those bytes. The local participant verifies the challenge and intended context before responding; the verifier compares the submitted statement with the authoritative issued statement and verifies the tag. Altering the payload while retaining the previous proof therefore fails the demonstrated context or authentication checks. HMAC supplies computational message authentication; the native secrecy result has a different premise and purpose. [D1; D3, §E; X1]

For example, an illustrative authorized statement may identify:

```
Sender:       worker-1
Operation:    read
Resource:     /reports/approved
Destination:  reports.example
Payload:      {"report": 7}
Sequence:     42
```

Changing the payload to `{"report": 8}` changes the statement requiring authentication. Replacing the operation with `export`, or the destination with `outside.example`, also changes that statement. A fresh valid envelope would still need to satisfy the inherited policy. Authentication of a request and permission to perform it are independently enforced predicates. This example preserves the preceding explanation's illustrative fields; it is not a claimed byte-for-byte capture of the delivered JSON envelope. [D1, `Runtime.submit`]

The response is separately generated and authenticated under the service's own reference and direction-specific context. It names the successful request and is checked for the matching peer and channel. This makes the reverse-direction result part of the demonstrated application relationship rather than treating any returned success label as sufficient evidence. [D1; D3, §G]

4. Native correctness, recovery, and credential privacy

4.1. Native response and honest verification

For each scheduled data coordinate, the source-defined calculation is

$$z_t = \left(\varphi(c_t(k_{r(t)})) + m_t\right) \bmod H. \tag{3}$$

Here, k is the enrolled string, $r(t)$ selects its tested position, c_t identifies the public membership cell, φ privately names that cell, and m_t is a fresh private mask in $\mathbb{Z}_H$. The cell count H is distinct from Shannon entropy notation. The verifier subtracts the actual mask, reverses the dictionary, and checks the resulting membership against its enrolled reference. Substitution into the verification predicate cancels the mask and dictionary, establishing honest correctness. [F1, §§1–3]

Calibration and a separating schedule establish authorized recovery as a separate result. Calibration determines the private naming map; the decoded cell intersections identify each enrolled symbol. The executable's `recover` routine uses the public profile, released word, and actual masks without reading the enrolled reference store. Publicly computable hypothetical masks do not replace those actual masks. The geometric source describes this distinction as an observation fiber containing explanations for many credentials and a private slice selecting the actual target. [F1, §5; F2, §§3–4; D3, §A]

4.2. Exact likelihood and posterior preservation

For every supported credential x, fixed permitted public challenge g, and released q-coordinate word z, independent uniform private masks yield

$$\Pr[Z = z \mid X = x, G = g] = H^{-q}. \tag{4}$$

Once x and g are fixed, the unmasked response is fixed. Exactly one mask vector produces any specified released word, and every vector has probability H^{-q}. Thus the likelihood is the same for every credential. With the corresponding challenge-independence assumptions, observing the specified challenges and masked responses preserves the credential prior. The prior itself need not be uniform. [F1, §§4, 9.1]

This is an information-theoretic statement about the declared observation, not a statement that credential inference is merely computationally expensive. Every function of the permitted record receives the same credential-independent input distribution. The source explicitly distinguishes that result from an enlarged observation containing raw answers, mask material, timing, error feedback, or other correlated signals. Their joint distribution must satisfy the corresponding observation-extension condition before they are included in the same guarantee. [F1, §§4, 9.3; F2, §6]

4.3. A trillion observations under the sequential model

Let B be the declared prior information and $\mathscr{T}_e$ the native public transcript of event e. The supplied sequential theorem establishes

$$\boxed{I(X; \mathscr{T}_1, \dots, \mathscr{T}_N \mid B) = 0} \tag{5}$$

for every supported finite N, when each event's complete conditional likelihood has the source's credential-independent challenge kernel and fresh private-mask law. Multiplying those conditional likelihoods gives an archive likelihood independent of the credential. Bayes' rule then preserves every supported prior. The result includes $N = 10^{12}$ under those same premises; it is not an extrapolation from a small successful benchmark. [F1, §9.2]

The protected variable is the enrolled credential. The theorem does not hide the existence of a transaction, arbitrary readable application content, or every possible activity pattern. The runtime export contains additional metadata and computational tags, and the Python implementation has timing and error behavior. The finite audit fixes its public controls; operational full-view analysis remains a named evaluation task. Possession of public code is included in the modeled background, while possession of actual private masks changes the experiment and enables the documented recovery operation. [D3, §D; D4, §5]

5. Entropy, finite capacity, and efficiency

The exact independent-mask profile has an explicit per-record cost:

$$E_{\text{record}} = q \log_2 H, \qquad E_N = Nq \log_2 H. \tag{6}$$

The variable q counts calibration and data responses. In the implemented RW-12/4 profile, $q = 2\ell + 3$; in RW-40, $q = 3\ell + 3$, where ℓ is the reference length. With four response labels, each independently uniform residue contributes two mask bits. The 24-character RW-40 profile therefore consumes 150 fresh private mask bits for each record. [D3, §C]

Reference profile	Mask bits per record	Mask material for 10^{12} records	Records from 65,536 bits
RW-12/4; 12 characters	54	6.75 TB	1,213
RW-40/4; 24 characters	150	18.75 TB	436

TB denotes 10^{12} bytes of continuously packed private mask material. A request and its response use two allocations; the RW-40 pair consumes 300 mask bits. Provisioning, integrity protection, other keys, metadata, and storage overhead are additional resources. The review encoding uses explicit JSON arrays and context, so a packed 150-bit witness is not the complete wire size. The finite tape stops issuance when exhausted. [D4, §6]

The source also establishes that deterministic expansion of a finite secret seed cannot increase its entropy. A pseudorandom construction can define a separate computational profile, while the exact independent-mask theorem retains its specified distribution and private provisioning requirements. The executable uses an operating-system cryptographic random interface for practical demonstration; the exact audit instead enumerates the ideal distribution. These evidence modes are identified separately. [F1, §7; D3, §C]

For the fixed profile tables, native response and verification work is linear in reference length and does not process the entire historical archive. The original 10,000-record benchmark measured 83.7955 microseconds per response generation with allocation and 43.1670 microseconds per native verification, approximately 84 and 43 microseconds respectively. It excluded the network, HMAC envelope, database, hardware, encryption, and private provisioning. This edition retains that benchmark as prior measured evidence and does not relabel it as a new end-to-end performance result. [D7]

6. One-use authority and hereditary operation

The reference gate combines distinct predicates:

$$\begin{aligned} \mathrm{Accept}_e = \mathrm{ContextValid}_e \\ \wedge\, \mathrm{Claim}_e \\ \wedge\, \mathrm{CalibrationValid}_e \\ \wedge\, \mathrm{DataValid}_e \\ \wedge\, \mathrm{PolicyAllows}_e. \end{aligned} \tag{7}$$

The verifier checks the exact context and envelope before an eligible response claims the event. The first eligible complete response atomically changes the event from `ISSUED` to `BURNED` before native verification. A correctly authenticated envelope with wrong native answers also consumes the event and is rejected. At most one worker wins that transition, and subsequent attempts encounter consumed state. The source proves at-most-once verification under the stated durable-state assumptions. [F1, §8; D3, §F]

The local effect table separately enforces a unique sender/operation identifier and compares the canonical payload during reconciliation. A fresh valid status or retry operation can retrieve the prior recorded result without creating another local effect. The demonstrated effect is a database insertion; an external payment, publication, or tool action requires the same explicit relationship to its own execution and reconciliation boundary. [D3, §F]

Inherited permission is evaluated along the registered ancestry:

$$\mathscr{A}_v = \bigcap_{u \in \mathrm{Anc}(v) \cup \{v\}} \mathscr{A}(P_u). \tag{8}$$

Every ancestor restriction and the child's own policy remain necessary. In the demonstration, the approved reader template permits a particular report read, forbids export, and grants no

further minting authority. A policy mutation fails the existing grant check, and current parent revocation is checked before effect creation. The registry uses a MAC addressed to the designated local verifier; independently verifiable cross-organization grants require their specified signature and issuer profile. [D3, §G; D4, §4]

This establishes the demonstrated human-to-machine continuity: one synthetic human ceremony activates a root; the root proves a permitted minting operation; the child and service then generate their own fresh native proofs. The human does not solve a new ceremony for every automated message. Privacy, provenance, permission, and one-use state remain complementary mechanisms rather than one result inferred from another.

7. Technical evaluation requested

The requested review concerns a proposed architecture and its current evidence. The existing HTTP6 control crosswalk identifies relevant relationships to replay resistance, context binding, service authentication, least privilege, and resource enforcement. These are author-proposed mappings reproduced from the supplied source, not an official NIST profile or a determination that all requirements of a control have been satisfied. [M1, §§2–5, 8, 12–14, 21]

Review subject	Existing evidence	Next implementation evidence
Payload and peer binding	Complete statement coverage; substitution and wrong-channel tests	Independent endpoints and trusted establishment of channel and intended-peer facts
Credential privacy	Fresh-mask theorem; exact finite distributions; resource accounting	Assessed private distribution, provisioning, and complete observation analysis
Replay and effect control	Atomic local event consumption; local effect uniqueness	Failure, rollback, and distributed-state tests at the actual resource boundary
Hereditary permission	Approved child template; ancestry checks; revocation and mutation tests	Production issuer, registry, resource adapters, and independently appraised instance identity
Hardware-bound phishing/relay resistance	Logical instance/channel rejection interfaces	Real device evidence, protected confirmation, and trusted input/output relay experiments
Confidential application traffic	Explicit payload-authentication envelope	Encrypted records and authenticated native-channel setup, with their own review

For technical reviewers, the first question is whether the implementation realizes the formal source's observation and private-resource assumptions. The second is whether every consequential application effect crosses the current authorization gate. The third is whether trusted

context is genuinely established by the operational endpoints rather than supplied as unverified strings. These questions preserve the original explanation's separation of mathematical proof, executable behavior, and deployment evidence. [F1, §§9, 12–14; D3, §§D–I]

The next implementation milestone is two separately running endpoints with authenticated private-material provisioning, protected key use, encrypted application records, and durable state tested across failures. A hardware-bound profile then exercises real attestation and the trusted interaction path against live relay attempts. The current wrong-channel test establishes the logical rejection rule; hardware evidence establishes whether a deployed component reliably supplies its premises. [D4, §§4, 7; F1, §12.4]

The proposal should be evaluated under its stated unit of protection: the complete application record. The current model is not a native encrypted transport or a production-ready replacement for HTTPS. Its existing positive result is more specific and reproducible: a native witness is generated for each request and response, bound to the exact statement, and checked together with present permission and one-use state. Completion of the named deployment interfaces extends that result through a separately assessable implementation boundary.

8. Conclusion

HTTP6's proposed per-message authorization mechanism is executable. Each application request and response in the reference model carries a fresh Rosario–Wang witness bound to the exact payload, execution context, inherited authority, and one-use event. The reported and reproduced local tests establish the specified behavior for the supplied source and experiment. [D1–D6]

The supplied mathematics establishes zero additional credential information for the specified fresh-mask public-view model, including every supported finite history of one trillion events. That result preserves the actual credential prior and requires its fresh private resources. Computational HMAC security protects payload authenticity; the ledger enforces one use; policy determines permitted effects; protected transport supplies the separate confidentiality objective. [F1, §§4, 7–9; D3, §§C–F]

The contribution for technical evaluation is therefore a clear combination of native private-reference proofs and message-specific authorization, together with the artifacts needed to inspect its present claims. The source, exact finite checks, evidence records, explicit entropy budget, and deployment program let reviewers assess each proposition at the boundary that actually supports it.

Appendix A. Preserved original explanation

The following is the original explanation to the author, preserved in its original order and wording. Publication reference labels replace chat citation markers, and mathematical delimiters are normalized to `$...$` and `$$...$$`. First-person statements refer to the original explanatory response; this edition's separately retained reproduction is identified in Section 2 and Appendix B. The source transcription in `original_explanation_verbatim.md` retains the original citation tokens.

Yes—the demonstrator implements a fresh, payload-bound authorization proof for each application request and each application response. That mechanism works in the supplied reference implementation. The code includes the complete payload in the authenticated statement, produces an actual Rosario–Wang masked witness, and checks both before permitting the operation. [D1] [D1]

What we have established is a working proof-per-message authorization model, supported by a conditional mathematical proof of credential privacy. That is narrower than proving that the complete HTTP6 replacement for HTTPS is already implemented and secure.

1. What I just verified

I reran the complete test suite: **all 69 tests passed**. I also ran another eight-message demonstration through the local HTTP carrier:

Operation or attempted attack	Freshly observed result
Initial synthetic human admission	Accepted
Root authorizes a distinct child	Accepted
Eight application requests	Eight fresh proofs accepted
Eight corresponding responses	Eight separate fresh proofs accepted
Replay of an accepted request	Rejected; no additional effect
Substitution of the destination	Rejected
Authenticated child requests prohibited export	Rejected
Response attempted under the wrong local channel	Rejected
Prepared request submitted after parent revocation	Rejected

That produced **18 accepted native proofs**: one admission, one minting operation, eight requests, and eight responses. It produced **exactly eight local application effects**.

These are actual reference-program results, not just expected behaviors written into the paper.

2. Yes: the proof covers the message payload

The implementation does **not** merely attach a login proof to an otherwise unprotected message.

Conceptually, each message contains:

$$\boxed{\text{Message proof object} = (x_e, Z_e, \tau_e)}$$

Here:

- x_e is the complete statement: sender, recipient, channel, direction, sequence, policy, operation, **actual payload**, nonce, and expiration.
- Z_e is the fresh Rosario–Wang masked membership witness.
- τ_e authenticates the statement and witness together.

The implementation calculates:

$$\tau_e = \mathrm{HMAC}_{a_v}(\mathrm{Encode}(\texttt{witness-envelope}, x_e, Z_e)).$$

The payload is inside x_e, so it is inside the authenticated bytes. The verifier also compares the submitted statement with the exact statement stored when that event was issued. **Changing the payload while keeping the old proof fails those checks.** [D3] [D1]

For example, suppose the authorized message says:

```
Sender:       worker-1
Operation:    read
Resource:     /reports/approved
Destination:  reports.example
Payload:      {"report": 7}
Sequence:     42
```

The existing proof cannot simply be moved to:

```
Payload:      {"report": 8}
```

or:

```
Operation:    export
Destination: outside.example
```

Those are different statements requiring their own valid authorization. Even a freshly authenticated request must still pass the inherited policy. The program checks that separately. [D1]

“Per message” means per protected application record—not automatically one proof per byte or per underlying network packet. A large streamed object would need a specified record/chunk format. The current attachment demonstrates complete bounded JSON records. [D4]

3. What the Rosario–Wang mathematics proves

The native response calculation is:

$$z_t = \left(\varphi(c_t(k_{r(t)})) + m_t\right) \bmod H.$$

The verifier knows the actual private masks and the enrolled reference. Subtracting the masks and reversing the private dictionary recovers the membership answers. Honest answers therefore pass by direct algebraic cancellation. Calibration and a separating schedule also permit authorized recovery of the enrolled string and dictionary. Those are mathematical results in the supplied formalization. [F1] [F1]

For the outside observer, the decisive result is:

$$\Pr[Z = z \mid X = x, G = g] = H^{-q}.$$

Every possible credential produces the same probability for each released word, because exactly one equally likely private mask vector explains that word for each candidate credential.

Therefore, observing the specified challenges and masked witnesses does not improve the observer's estimate of the original credential. It is not merely "too difficult to calculate." Under the stated masking and observation assumptions, the additional information is mathematically zero. [F1]

The Python distribution audit checks finite instances of that property exactly. The general result comes from the proof, rather than from extrapolating a successful test.

4. Does that continue after trillions of observations?

Yes, the supplied sequential privacy theorem covers any supported finite number of honest observations—including one trillion—when every event receives the required fresh private masks and preserves the specified observation boundary.

Its conclusion is:

$$\boxed{I(X; \mathscr{T}_1, \dots, \mathscr{T}_N \mid B) = 0.}$$

The proof multiplies credential-independent conditional likelihoods across the history. Consequently, the complete archive remains credential-independent. Increasing N does not, by itself, create leakage. [F1]

That statement protects the credential. It does not mean that every component of the message system has information-theoretic security.

Property	What supports it here
Privacy of the enrolled credential in the specified released transcript	Rosario–Wang's fresh-mask theorem
Integrity and authentication of the exact payload	The independent HMAC and context checks
Rejection of a second use	The atomic consumed-event ledger
Permission to perform the operation	The inherited-policy gate
Confidentiality of application content	A transport/encryption component still to be integrated

The attachment expressly makes those distinctions. HMAC is a keyed message-authentication mechanism; its security rests on cryptographic assumptions, rather than the fresh-mask secrecy theorem. [D3] [X1]

The fresh-entropy requirement is real

For the demonstrated 24-character RW-40 machine profile, each record consumes **150 fresh private mask bits**. Thus one trillion records require **18.75 TB of continuously packed masking material**, excluding other overhead. A request and response consume two allocations. A 65,536-bit tape supplies **436 complete records** under that profile. [D4]

The script stops when its tape runs out.

Your formalization explicitly proves that deterministic expansion of a fixed seed cannot create additional entropy. A short seed can support a separately analyzed **computational** implementation; it does not supply unlimited independent masks for this exact theorem. [F1]

5. Which HTTP6 claims are already demonstrated?

The demonstrated application mechanism is:

> An admitted child produces a fresh native witness for a particular message; the authenticated envelope binds that witness to the exact payload and context; the verifier checks current authority and consumes the event before allowing the protected effect.

The response performs the corresponding operation in the opposite direction and names the accepted request. The child uses its own reference, and the service uses its own reference. **The human does not have to solve another ceremony for every automatic message.** [D3]

This establishes an executable basis for your intended architecture. It also establishes that the mechanism need not search an expanding history to generate the next native proof: for the fixed tables, the native work is linear in the reference length. Full handling additionally processes the payload, ancestry, and durable state. [D3]

The earlier benchmark measured approximately **84 microseconds for native response generation with allocation** and **43 microseconds for native verification**. Those measurements excluded the network, HMAC envelope, database, hardware, encryption, and private provisioning. They demonstrate inexpensive local arithmetic—not completed Internet-scale throughput.

6. What remains to establish the full HTTP6 claim?

The present program is a **local designated-verifier model**. Its private provisioning places material into logical participant objects; its hardware identities are modeled attributes; its application payloads remain readable. It does not yet implement independently secured client/server processes, a native encrypted handshake, or a real trusted-screen ceremony. These are explicitly identified in the attachment, rather than silently counted as tested capabilities. [D4] [D4] [D4]

The next implementation milestone is therefore **two separately running endpoints**, with authenticated private-material provisioning, protected key use, encrypted application records, and durable state tested across failures. A hardware-bound profile must then exercise the real device attestation and trusted input/output path against live relay attempts.

The existing wrong-channel test demonstrates the **logical rejection rule** needed for a relayed-screen defense. It does not yet demonstrate that real hardware reliably supplies that rule's trusted local facts. The source itself treats live forwarding, active authentication, and complete observation handling as additional protocol obligations. [D3] [F1]

The conclusion

A proof per message payload: yes, implemented and exercised in the reference model.

No additional credential information from trillions of observations: mathematically established for the specified fresh-mask, complete-observation model.

An entirely information-theoretic, production-ready replacement for HTTPS: not established by this attachment. Its payload authentication is presently computational, its payload encryption remains to be integrated, and its deployment trust boundaries still need implementation evidence.

The strongest accurate statement is:

> **HTTP6's proposed per-message authorization mechanism is executable: each request and response carries a fresh Rosario–Wang witness bound to its exact payload, execution context, inherited authority, and one-use event. The supplied mathematics establishes passive credential privacy under fresh private masking; the demonstrator establishes the tested local authorization behavior.** [D3] [D3]

Appendix B. Reproduction and artifact index

B.1. Reproduction procedure

Extract the unchanged companion archive and enter its `HTTP6_RosarioWang_Demonstrator` directory. The script uses Python 3.10 or later and the standard library. Create new output directories for each demonstration rather than overwriting an earlier record. The following commands reproduce the relevant workflows; the benchmark command is optional and produces a new measurement rather than replacing the historical figure. [D4]

```
python3 -m unittest -v
python3 http6_rosario_wang_demo.py demo \
  --messages 8 --loopback --out review_loopback
python3 http6_rosario_wang_demo.py privacy \
  --out review_privacy.json
python3 http6_rosario_wang_demo.py capacity \
  --observations 1000000000000
python3 http6_rosario_wang_demo.py benchmark --records 10000
```

The reproduction for this edition used Python 3.13.5 on Linux x86_64. The suite completed 69 tests successfully. The loopback result recorded one synthetic admission, one child mint, eight requests, eight responses, 18 accepted native proofs, and eight local resource effects. The privacy audit returned exact equal-likelihood and posterior-preservation results for its enumerated finite cases. A trillion-event workload was not executed, and the benchmark in Section 5 remains the original recorded measurement. [D6; D7]

B.2. Artifact inventory

Package location	Purpose
`HTTP6_Per_Message_Proof_NIST_Review_White_Paper.pdf`	Typeset technical-review paper
`HTTP6_Per_Message_Proof_NIST_Review_White_Paper.md`	Editable source with LaTeX mathematics
`typesetting/`	Standalone LaTeX, build script, and publication styling
`original_explanation_verbatim.md`	Preserved explanation before audience editing
`support/HTTP6_RosarioWang_Python_Demonstrator.zip`	Unchanged executable review attachment, tests, schemas, and original evidence
`evidence/`	This edition’s regression log, loopback outputs, exact privacy audit, and reproduction receipt
`sources/`	Supplied formalizations, review appendix, source code, and mapping sources used by the paper
`manifest.json`	SHA-256 integrity inventory of packaged artifacts
`VALIDATION.txt`	Document, preservation, reproduction, and PDF checks

The unchanged demonstrator archive provides the complete test and source package. The separately copied `http6_rosario_wang_demo.py` is checked against the archive's corresponding script. Original source documents are preserved without editing. Generated observer files contain public records and synthetic application data, not enrolled private words, mask tapes, or authentication keys. The demonstration database is retained as local effect evidence, with its exact scope described in the README. [D4]

B.3. Claim-to-evidence concordance

Original explanation	Formal presentation	Supporting evidence
What was verified	Section 2	Regression log and local workflow report [D5, D6]
Proof covers the payload	Section 3	Complete statement and envelope code [D1, D3]
Mathematical correctness and privacy	Section 4	Native and sequential proofs [F1, F2]
Trillion-observation consequence and resource cost	Sections 4.3 and 5	Sequential theorem and capacity accounting [F1, D3, D4]
Demonstrated HTTP6 claims and primitive efficiency	Sections 5 and 6	Source, workflow, and historical benchmark [D1, D6, D7]
Full implementation milestone	Section 7	Explicit implementation boundaries and active-security model [D3, D4, F1]
Final scoped conclusion	Section 8	Combined source and evidence record

References and source register

The source register identifies supplied materials and their specific evidentiary roles. Mathematical claims use their stated hypotheses. Artifact test records describe the reference experiment. The existing NIST crosswalk supplies author-proposed relationships, not new normative requirements or program-specific submission instructions. No new external standard requirement is introduced by this editorial conversion.

[D0] Preceding explanatory response, *What the demonstrator proves and whether HTTP6 supports a proof per message payload.* Preserved in Appendix A and in `original_explanation_verbatim.md`. Editorial source, not an independent assessment.

[D1] `http6_rosario_wang_demo.py`. Reference demonstrator version `http6-rw-review-1.0`. In particular, `Peer.respond`, `Runtime.challenge`, `Runtime.submit`, `verify_native`, `recover`, `PrivateTape`, and `Ledger`. Supplied executable source, preserved unchanged.

[D2] `HTTP6_RosarioWang_Python_Demonstrator.zip`. Supplied runnable attachment containing the source, 69 regression tests, message schema, source map, exact finite audit, public records, and original measurement files. Archive retained unchanged in `support/`.

[D3] Rosario, Frank Dylan. *HTTP6: Executable Rosario–Wang Proof and Observer Demonstration.* Reference demonstration 1.0, November 9, 2026. Supplied `HTTP6_RosarioWang_Executable_Appendix.md`, particularly §§A–I and Equations A1–A10.

[D4] *HTTP6 / Rosario–Wang: executable review attachment.* Supplied `HTTP6_RosarioWang_Demo_README.md`; corresponding `README.md` in [D2]. Reproduction, profile definitions, function map, resource budget, observer model, and deployment boundaries.

[D5] `HTTP6_RosarioWang_Demo_Validation.txt`. Original attachment-validation record. Reports 55 artifact checks, 69 tests, finite distribution counts, local workflow, and a three-request original loopback run. This original three-request carrier record is distinct from the eight-request reproduction in [D6].

[D6] *Technical review edition reproduction record.* New execution of the unchanged [D2] source during preparation of this paper. Files: `evidence/unittest.txt`, `evidence/loopback/report.json`, `evidence/privacy.json`, and `evidence/reproduction_receipt.json`. Records 69 passing tests and the eight-request/eight-response loopback result. The receipt identifies versions and hashes. Local reproduction is not independent laboratory review.

[D7] `results/benchmark_10000.json` in [D2]. Original 10,000-record primitive-core benchmark: 83.7955048 microseconds for response generation with allocation; 43.1669605 microseconds for verification. Excludes network, TLS/AEAD, HMAC envelope, SQLite, hardware, and private provisioning. Preserved as `evidence/original_benchmark_10000.json`.

[F1] Rosario–Wang. *Consolidated construction, proofs, worked examples, nonce entropy, burn state, and proof-of-knowledge requirements.* Consolidated technical document, September 16, 2026. Supplied `Rosario_Wang_Consolidated_Formalization.md`. Principal locations: §§3–5 (native correctness, privacy, recovery), §7 (entropy), §8 (event burn), §9 (sequential and complete-view privacy), and §§11–14 (knowledge, active security, and review obligations).

[F2] *A Unified Geometry of Private Navigation and Observable Projection.* Version 1.0, September 13, 2026. Supplied `Unified-Geometric-Formalization.md`, particularly §§3–6. Establishes finite fiber/slice and conditional full-mask privacy relationships under explicit hypotheses.

[F3] *Appendix B · Lean 4 formalization registry.* Supplied `B-lean-formalization-registry(1).md`. Cross-references source declarations and distinguishes source correspondence, finite enumeration, and kernel validation of a specified checkout. No new Lean run is asserted by this paper.

[M1] Rosario, Frank Dylan. *HTTP6 and NIST: Human-Originated Identity, Hereditary Authority, and Security Control Enforcement.* Expanded Edition 2.0, November 9, 2026. Supplied `HTTP6_NIST_Requirements_and_Control_Mapping.md`, particularly §§2–5, 8, 12–14, and 21. Author-proposed architecture mapping and assessment-evidence framework.

[X1] Krawczyk, H., M. Bellare, and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication.* RFC 2104, February 1997. External reference retained from the original explanation for the role of HMAC; not a validation of this composition. https://www.rfc-editor.org/info/rfc2104/