Poster: Exploiting the Two-Phase Gap in the x402 Protocol for Autonomous AI Payments

Yoonseo Hwang
Sungkyunkwan University
qwdfgqw@skku.edu

Hyoung-Kee Choi
Sungkyunkwan University
meosery@skku.edu

***Abstract*—The x402 protocol enables autonomous AI payments via blockchain-based settlement, yet its verification–settlement split introduces a temporal window that can be exploited as a TOCTOU gap. We conduct a black-box evaluation of seven x402-enabled services and find that two are susceptible to parallel "N-way burst" replays. Our findings highlight the need for concurrency-safe consumption tracking and strict idempotency to secure decentralized, per-request monetization against race-condition exploits.**

I. Introduction

The proliferation of LLM-based agentic systems necessitates lightweight, pay-per-request mechanisms for consistent access control and monetization. The x402 protocol addresses this by gating resource access through HTTP-level payment challenges [1]. To reduce end-to-end latency, x402 deployments separate payment processing into a verify phase and a settle phase, which creates a temporal interval between verification and on-chain settlement. We refer to this interval as the Two-Phase Gap. Because x402 settlement is confirmed on a blockchain network, confirmation latency is an inherent part of this separation and thus is central to the security analysis.

When combined with server-side implementation flaws, the Two-Phase Gap compromises the integrity of the payment-provisioning process. This paper formalizes Time-of-Check to Time-of-Use (TOCTOU) replay attacks arising from this architecture and analyzes violations of key security properties through empirical observation of real-world services. We find that some deployments accept concurrent replays of an identical payment proof and return multiple protected responses while the corresponding settlement is confirmed only once on-chain. We conclude with architectural recommendations to ensure robust, payment-coupled resource delivery in autonomous agent-to-service economies. To our knowledge, this is one of the first empirical, black-box studies that characterize replay/TOCTOU exploitability specifically induced by the verification–settlement gap in real-world x402 deployments.

II. Overview of the x402 Protocol

x402 uses HTTP 402 Payment Required as a negotiation signal for protected APIs. Figure 1 summarizes the end-to-end sequence from the initial challenge through verification and settlement. The main actors are a Client, a Resource Server (RS) that gates access to a protected resource, a Facilitator that mediates verification and settlement, and the blockchain network that confirms settlement transactions.

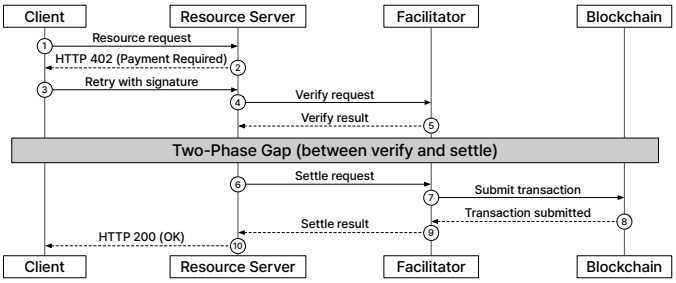


Fig. 1. End-to-end x402 message flow. The figure highlights the Two-Phase Gap between verification and on-chain settlement.

The protocol begins when the Client sends a resource request to the RS, as shown in step 1 of Figure 1. If payment is required, the RS responds with HTTP 402, as shown in step 2. The Client then retries the same request with a signature attached, as shown in step 3. Throughout this paper, we use the term payment proof to denote this cryptographically verifiable authorization artifact that satisfies the server's stated payment requirements. This proof binds payment parameters such as recipient, amount, and validity constraints to the request context.

Upon receiving the signed retry, the RS initiates the verify phase by sending a verify request to the Facilitator, as shown in step 4. The Facilitator checks the cryptographic validity and requirement compliance of the payment proof and returns a verify result to the RS, as shown in step 5. After verification, the protocol proceeds to the settle phase. The RS sends a settle request to the Facilitator, as shown in step 6, and the Facilitator submits the corresponding transaction to the blockchain, as shown in step 7. Once the transaction is confirmed on-chain, as shown in step 8, the Facilitator returns a settle result to the RS, as shown in step 9. Finally, the RS completes the protected request and responds to the Client with HTTP 200 OK, as shown in step 10.

The verify phase occurs in steps 4 and 5, and the settle phase occurs from steps 6 through 9. This separation creates an inherent temporal interval between verification and settlement completion. This Two-Phase Gap is highlighted in Figure 1. The duration of the gap can vary due to external factors such as network congestion and confirmation latency on the blockchain network.

III. N-way Burst Replay Attack

A. Attack Intuition

If a service delivers the protected resource based on a verify result without enforcing strict single-use semantics, the Two-Phase Gap can become a TOCTOU window [2]. An attacker can exploit this window by acquiring a valid payment proof through legitimate procedures and then replaying the proof in a synchronized burst of concurrent requests. The goal is to trigger multiple protected responses from a single authorization before the service records consumption or before settlement finalization closes the window.

B. Exploit Conditions

The attack becomes exploitable if the server meets the following conditions. First, the server does not atomically record consumption of a proof and does not reject subsequent uses of the same proof under concurrency. Second, the server treats replayed requests as independent and valid during the interval between verification and settlement confirmation. Third, the protected operation is not idempotent, so each accepted replay produces an additional deliverable such as extra content, tokens, credits, or access.

C. Threat Model

The attacker possesses ordinary client privileges and can follow the protocol to obtain a valid payment proof. The attacker can issue multiple concurrent HTTP requests and control their timing to induce race conditions. We exclude signature forgery, key theft, and attacks on the underlying blockchain consensus from scope. Our focus is on vulnerabilities that arise from the asynchronous separation between verification and settlement confirmation in the two-phase architecture.

IV. Experimental Methodology and Evaluation

We evaluated seven publicly accessible x402-enabled services using a black-box methodology and without source-code access. For each service, we first triggered the HTTP 402 challenge to obtain payment requirements and then followed the normal protocol flow to acquire a single valid payment proof in the form of a signed authorization.

Each service was tested by replaying an identical payment proof in a synchronized burst of concurrent requests. In each burst, we issued N parallel requests that reused the same proof, and we set N to 10 for all experiments. We measured two outcomes per burst. First, we counted S, the number of requests that returned a protected response with HTTP 200 within the burst, where an HTTP 200 indicates that the service delivered the protected resource. Second, we checked whether settlement occurred more than once for the same proof by inspecting publicly observable on-chain evidence, including transaction identifiers and nonce progression when available. Throughout this section, we report the burst success ratio as S/N, which denotes the fraction of successful protected responses among the N concurrent replay attempts.

Table 1 summarizes outcomes across all seven services. Five services consistently suppressed replay effects, meaning that only one request in a burst succeeded and the remaining requests were rejected or otherwise failed to produce additional delivery. In contrast, two services exhibited repeated delivery under proof reuse during the interval between verification and settlement confirmation. We observed multiple HTTP 200 responses while settlement was confirmed only once on-chain.

TABLE I
Burst replay outcomes across seven x402 services. The number of transactions recorded on-chain was 1 in all bursts.

Service	Service type	S/N	Vuln
E***	On-chain/token/DeFi data API	2~10/10	✓
H***	Research service	1/10	✗
MI**	Web extraction and scraping API	1/10	✗
MO**	x402 reference/demo site	4~5/10	✓
N***	SNS service	1/10	✗
S***	Micropayment service	1/10	✗
Z***	Web extraction and scraping API	1/10	✗

For the first vulnerable service, which exposes a payload-independent protected endpoint, we observed S equal to 4 in one burst and S equal to 5 in another. For the second vulnerable service, we observed S ranging from 2 to 10 across four bursts under identical payloads. When we issued requests that reused the same proof but carried ten different payloads within the burst, we observed S values ranging from 6 to 10.

V. Suggested Solutions

To mitigate these risks, we propose the following recommendations:

- **Atomic single-use consumption**: Atomically mark each proof as consumed at verify time.
- **Strict idempotency**: Treat identical proofs as the same request outcome and reject replays [3].
- **Confirmation-aware provisioning**: Prefer deliver-after-confirmation. If not feasible, add controls that prevent cumulative effects under settlement disruptions.

VI. Conclusion

This paper shows that the Two-Phase Gap in x402 is the temporal interval between verification and on-chain settlement confirmation, and that it creates a tangible TOCTOU vulnerability surface. We show that when server-side state management does not enforce single-use semantics, replaying the same payment proof via an N-way burst can induce unauthorized multiple deliveries.

References

[1] x402, "x402: An Open Standard for Internet-Native Payments," Whitepaper. [Online]. Available: https://www.x402.org/x402-whitepaper.pdf. Accessed: Jan. 20, 2026.

[2] MITRE, "CWE-367: Time-of-check Time-of-use (TOCTOU) Race Condition." [Online]. Available: https://cwe.mitre.org/data/definitions/367.html. Accessed: Jan. 22, 2026.

[3] G. Ramalingam and K. Vaswani, "Fault Tolerance via Idempotence," in *Proc. of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '13)*, Rome, Italy, 2013, pp. 249–262, doi:10.1145/2429069.2429100.

Poster: Exploiting the Two-Phase Gap in the x402 Protocol for Autonomous AI Payments

Yoonseo Hwang, Hyoung-Kee Choi | Sungkyunkwan University | {qwdfgqw, meosery}@skku.edu

Key Takeaway

In a black-box study of seven public x402 services, we find two endpoints where one proof triggers multiple protected deliveries while only one settlement transaction confirms on-chain.

Introduction

Motivation

- AI increasingly depends on paid tools and APIs, making per-request access control essential
- x402 enables this model using an HTTP 402 challenge and a signed payment proof on retry

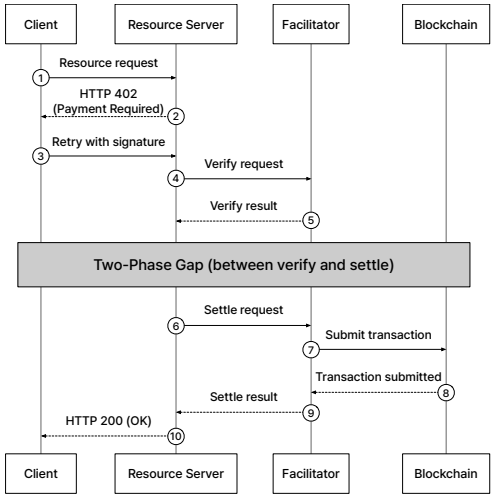
Design Rationale

- To reduce user-perceived latency, x402 verify first and settle later on-chain

Research Goal

- We test whether this two-phase split opens a **TOCTOU** window under burst replays

x402 Two-Phase Design



Two-Phase Gap

Verification completes before **settlement** finality, creating a temporal interval whose length depends on confirmation latency.

Two-Phase Flow

- RS verifies the payment proof via the Facilitator (fast, off-chain)
- The Facilitator settles on-chain later; finality is variable

Why it matters

If RS delivers after verification, this gap can become a **TOCTOU** window when requests are processed concurrently.

Latency Trade-off

- Decouples user-perceived latency from blockchain finality (verify now, settle async)
- Makes replay safety a concurrency problem: single-use consumption must be atomic

Attack Overview

Exploit Logic

- Race against the server's **consumed state** by replaying a valid signature under concurrency
- Trigger multiple protected deliveries from a single authorization artifact

Vulnerability Conditions

- Deliver before settlement finality
- Missing **atomic consumption tracking** (marking proof as "spent")
- Non-idempotent delivery logic amplifies impact under replays

N-Way Burst Methodology

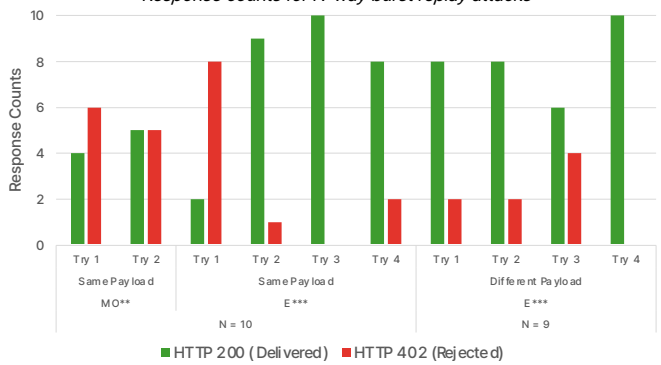
- Synchronized burst of $N = 10$ concurrent requests using the same proof
- Threat model: ordinary client; no key theft or signature forgery
- Success: multiple HTTP 200 deliveries while only one on-chain settlement confirms

Evaluation

N-way Burst Attack Success Across 7 Services

Service	Service Type	Observed HTTP 200 / N	Vuln
E***	On-chain/token/DeFi data API	2~10/10	O
H***	Research service	1/10	X
MI**	Web extraction and scraping API	1/10	X
MO**	x402 reference/demo site	4~5/10	O
N***	SNS service	1/10	X
S***	Micropayment service	1/10	X
Z***	Web extraction and scraping API	1/10	X

Response counts for N-way burst replay attacks



Methodology Notes

- Secure baseline: Each burst replays the same payment proof N times concurrently; a secure service should return 1×HTTP 200 and (N−1)×HTTP 402.
- On-chain settlement multiplicity was checked via publicly observable transaction IDs / nonce progression
- Only a single on-chain settlement transaction was confirmed for each burst of requests

Conclusions

Finding

The verification–settlement split in two-phase x402 deployments creates a practical **TOCTOU** surface under concurrency when single-use semantics are not enforced.

Impact

Delivery amplification allows one proof to be consumed multiple times, undermining per-request monetization even though settlement finalizes once.

Suggested Solutions

- **Atomic single-use** (concurrency-safe check-and-set)
- **Strict idempotency enforcement** (same proof → same outcome)
- **Confirmation-aware provisioning** (or compensating controls)