ZK-ARCHE: Zero-Knowledge Authentication for Resource-Constrained Heterogeneous Environment

Problem Statement and Goals

- Rapid expansion of IoT across smart cities, industrial control systems, and healthcare
- IoT devices are **typically resource-constrained**
- Large-scale deployments make traditional authentication mechanisms inefficient
- Existing **passkey-based** authentication standards are well-suited for human-operated devices
- No equivalent **lightweight and scalable** authentication standard currently exists for IoT
- **Secure and efficient** IoT authentication remains an open research challenge

Approach

- Zero-Knowledge Proofs (ZKPs) have 3 main property

Completeness: honest proof always accepted by verifier

Soundness: a dishonest proof cannot convince verifier

Zero-knowledge: no info about secret beyond its validity

- Devices authenticate **without exposing credentials**
- Inspired by **FIDO2 passkey** security principles
- **Lightweight** design for resource-constrained IoT
- Supports C–Rust **interoperability**, enabling heterogeneous IoT environments

Results

Key Takeaway: Across all tests, the protocol demonstrates strong performance under normal conditions and predictable degradation under heavy traffic, suggesting suitability for real-world IoT deployments where authentication bursts are infrequent but must be handled securely and reliably.

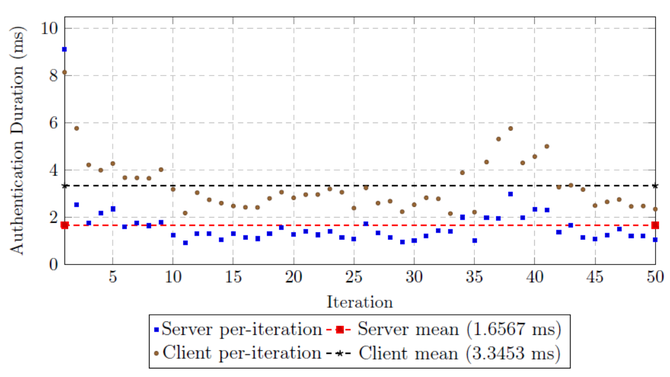


Figure 6.1: Authentication latency for 50 clients one authentication per client. Points show the measured time per client; dashed lines show the mean server and mean client latency.

Test 1: Concurrent Clients (Single Authentication)

With 50 clients each performing a single authentication, server-side latency remained low at ≈1.66 ms on average, while client-side latency averaged ≈3.35 ms. The narrow spread across iterations indicates stable performance and efficient handling of concurrent, non-congested authentication requests.

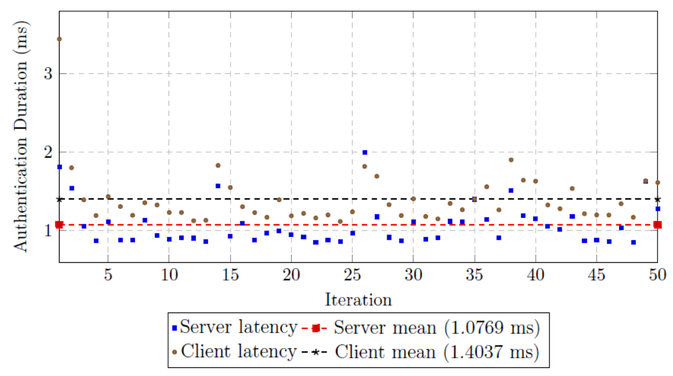


Figure 6.2: Authentication latency for a single client performing 50 sequential authentication attempts against the server. Points show per-iteration durations for both client and server; dashed lines indicate the mean latency for each role.

Test 2: Sequential Authentication (Single Client)

For a single client performing 50 sequential authentications, latency remained highly consistent, with server-side verification averaging ≈1.08 ms and client-side proof generation averaging ≈1.40 ms. No performance drift was observed, confirming negligible per-session overhead and stable repeated authentication behavior.

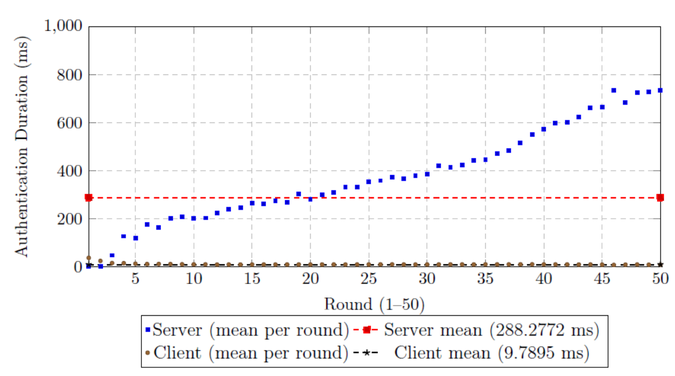


Figure 6.3: Authentication latency under busy traffic (50 clients, 50 authentications each). Client points show the mean per round across clients; server points show mean processing time per round; dashed lines indicate overall means.

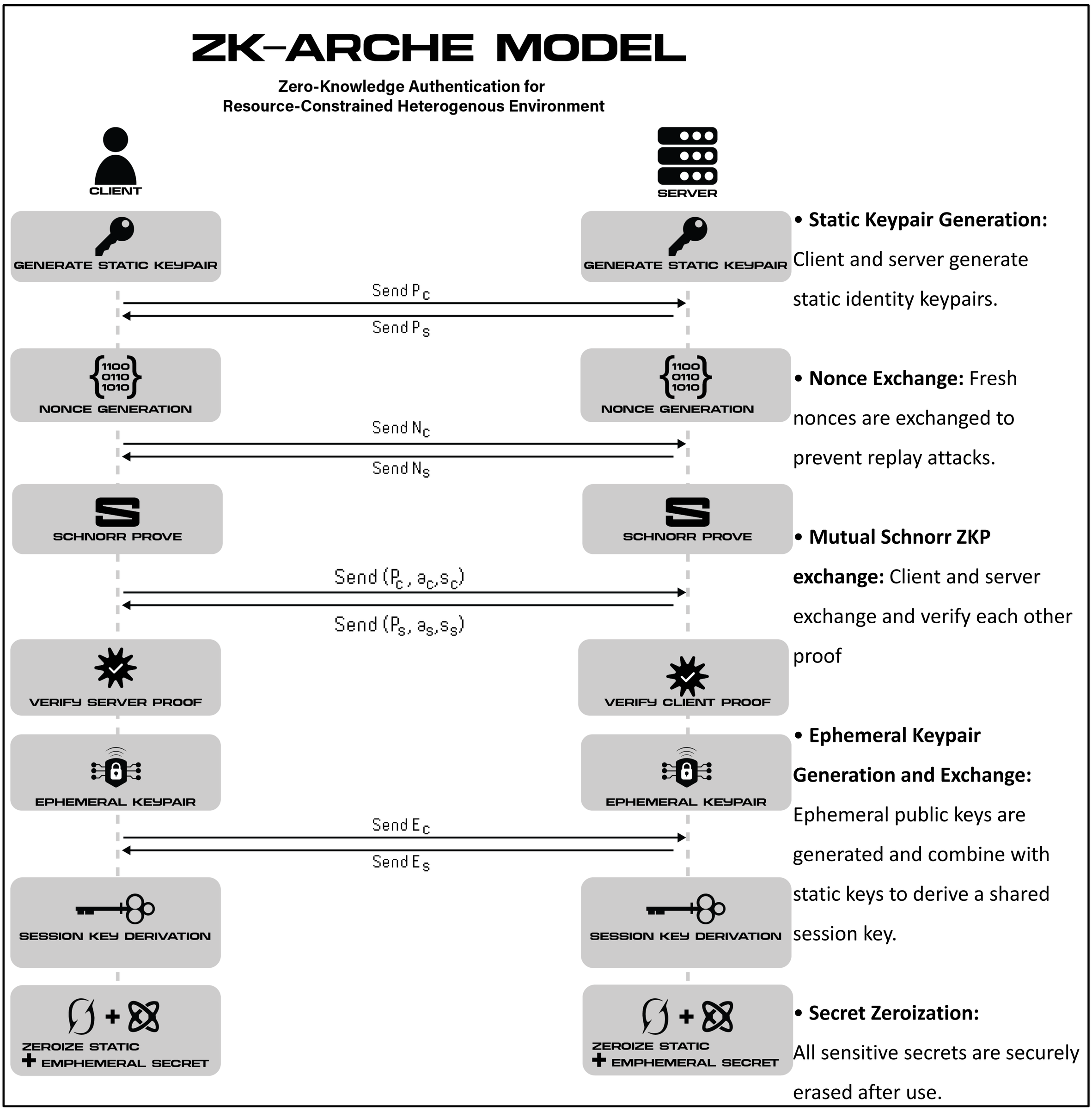
Test 3: High-Load Scenario (Busy Traffic)

Under heavy load conditions (50 clients each performing 50 authentications) average server-side latency increased significantly to ≈288 ms on average due to queuing and contention, while client-side latency rose moderately to ≈9.79 ms. Despite increased server load, client overhead remained low, indicating that server-side processing is the primary bottleneck under sustained high concurrency.

Khang Tran
Project Advisor: Dr. Geethapriya Thamilarasu
University of Washington Bothell

ZK-ARCHE: Zero-Knowledge Authentication for Resource-Constrained Heterogeneous Environment

Abstract



- **Static Keypair Generation:** Client and server generate static identity keypairs.
- **Nonce Exchange:** Fresh nonces are exchanged to prevent replay attacks.
- **Mutual Schnorr ZKP exchange:** Client and server exchange and verify each other proof
- **Ephemeral Keypair Generation and Exchange:** Ephemeral public keys are generated and combine with static keys to derive a shared session key.
- **Secret Zeroization:** All sensitive secrets are securely erased after use.

Khang Tran
Project Advisor: Dr. Geethapriya Thamilarasu
University of Washington Bothell