**Summary**

This paper proposes a novel security mechanism called “PreAcher”, specifically designed to counteract ADoS/DDoS attacks that exploit password-based authentication while preserving the confidentiality of user passwords. Compared to previous research, PreAcher’s unique three-party authentication mechanism, which allows CDNs to perform efficient pre-authentication without accessing plaintext passwords, effectively addresses Application-layer Denial-of-Service (ADoS) attacks that arise from the high computational cost of password authentication.

Traditional protection methods like rate limiting, CAPTCHA, and two-factor authentication often require storing plaintext password on CDNs or servers, making them vulnerable to man-in-the-middle attacks or data leaks. PreAcher integrates CDNs into the authentication process through a three-party protocol involving the client, CDN, and origin server. It uses Oblivious Pseudorandom Functions (OPRF) to conceal passwords from the CDN and Locality-Sensitive Hashing (LSH) to enable the CDN to detect likely incorrect passwords without full knowledge of the correct one. The CDN conducts “pre-authentication,” filtering most invalid logins before they reach the server, thus mitigating ADoS while preserving password secrecy.

The design is compatible with current infrastructure, requiring no client/browser or CDN infrastructure changes—only website-side deployment. Evaluations on testbeds and the Internet (Cloudflare) show that PreAcher maintains high throughput under attack, significantly reduces CPU usage compared to alternatives (e.g., DuoHash, SGX-CDN), and adds acceptable latency (42–72 ms in most regions). It also reduces the risk of offline dictionary attacks from a passive CDN attacker, lowering cracking success rates from 8.42% to 0.20%.

Overall, PreAcher offers a deployable, efficient, and secure solution to enhance password-based authentication resilience against both ADoS attacks and password exposure risks.

**Thoughts**

PreAcher shows clear strengths in practical deployment, strong threat modeling, and balanced use of cryptographic techniques. It introduces a pre-authentication stage at the CDN, combining OPRF and LSH to reduce server workload while keeping passwords hidden from third parties. This design elegantly addresses the tension between efficiency and security: the CDN filters out the majority of failed logins, preventing ADoS, while never learning the exact password. Compatibility with current ecosystems is also a major advantage, since websites can deploy PreAcher without requiring modifications to browsers, client devices, or CDN infrastructure.

The evaluation highlights that PreAcher significantly outperforms alternatives. Compared to Baseline, it prevents servers from being overwhelmed by malicious logins. Compared to DuoHash, it achieves far higher throughput and much lower CPU consumption, since DuoHash requires computationally expensive hashing at the CDN. Compared to SGX-CDN, PreAcher avoids reliance on specialized hardware, making it far more deployable.

That said, limitations remain. The system only addresses ADoS attacks targeting password logins; other interfaces (e.g., search queries, data uploads) could still be exploited. The trust model assumes passive CDNs—ignoring scenarios where a CDN might be actively compromised or malicious. Additionally, while the latency overhead (≈40–70 ms) is acceptable for most web applications, it could be problematic for ultra-low-latency services like online gaming or financial trading. Moreover, PreAcher does not address password reuse, weak passwords, or phishing attacks, which remain common vulnerabilities.

In terms of future directions, the pre-authentication concept could be extended beyond password logins. One direction is integrating PreAcher with passwordless authentication methods such as WebAuthn or FIDO2, so that malicious requests can be filtered before invoking expensive cryptographic checks. Another extension is applying the pre-authentication concept to API key and token validation in API gateways, where services face similar risks of ADoS-style abuse. PreAcher could also be enhanced with adaptive security policies—for example, dynamically adjusting LSH parameters or response delays based on traffic anomalies.

Another avenue could be exploring ways to handle active CDN adversaries, perhaps by combining PreAcher with trusted execution environments or multi-party computation could mitigate the risk of an actively compromised CDN. Additionally, exploring multi-factor integration—where the CDN filters passwords while the server enforces secondary checks—may provide stronger defense with minimal usability cost.

﻿