Verifiable Zero-Knowledge Architecture for Client-Side Encryption

The Cloud's Privacy Deficit

Standard cloud services inherently create a privacy problem. Even with server-side encryption, the provider manages the keys, allowing access to your data. This exposes it to breaches, surveillance, and misuse, building on a foundation of trust that creates inherent risk.

Vulnerability

Provider access to data

Risk

Breaches, surveillance, misuse

Issue

Reliance on provider trust

A Zero-Knowledge Solution

This project proposes a cloud storage system that architecturally removes the need to trust the service provider. All encryption and decryption occur client-side, ensuring the server only stores opaque, unintelligible data blobs, with zero knowledge of your file contents or encryption keys.

Trusted Client (Your Browser)

A simple web interface using the browser's Web Crypto API for all security operations.

Untrusted Server (The Cloud)

A lightweight backend that only stores and retrieves encrypted data, unable to read or interpret it.

The Cryptographic Protocol

The system's security relies on a multi-layered cryptographic process, entirely within the user's browser.

01

Master Key Creation

Your password, combined with a unique salt, creates a strong 256-bit Master Key via PBKDF2, preventing cracking.

02

File Key Generation

Each file gets a new, random File Key for encryption.

03

File Encryption

File contents are encrypted with AES-256-GCM, ensuring confidentiality and integrity.

04

Key Wrapping

The File Key is then encrypted ("wrapped") by your Master Key.

05

Secure Storage

Encrypted file and wrapped File Key are sent to the server, which cannot decrypt them.

Security Evaluation: Key Findings

Testing based on the OWASP Web Security Testing Guide confirmed the system's core security claims.

Zero-Knowledge Verified

Network and database inspection confirmed the server only handles unintelligible ciphertext.

Data Integrity Guaranteed

Malicious alterations on the server caused decryption failure, proving protection against tampering.

Brute-Force Resistance

PBKDF2 adds measurable delay, making offline password cracking computationally expensive.

The Critical Trade-Off

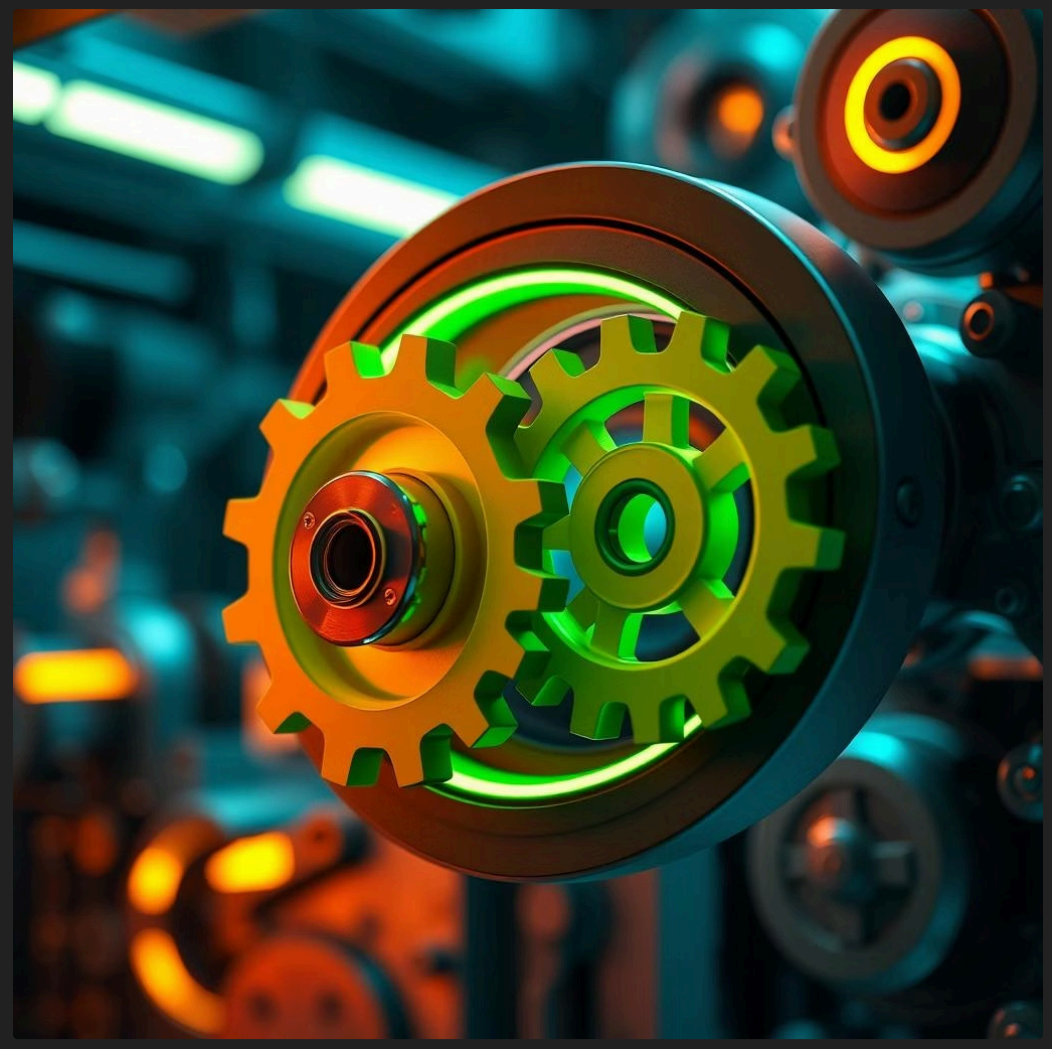
The project demonstrates a practical blueprint for a private-by-design cloud application using standard web technologies. However, its greatest strength is also its greatest usability challenge.

The Challenge: Data Loss Risk

Since the server has zero knowledge of your password or Master Key, it cannot help with password recovery, leading to irreversible data loss if forgotten.

Primary Contribution

This work provides a holistic case study, linking cryptographic theory to a validated implementation, highlighting the conflict between absolute security and user recovery needs.



Future Work: User-Friendly Recovery

The most critical next step is researching user-friendly key recovery mechanisms that don't compromise the zero-knowledge principle.

Social Recovery

Exploring methods where trusted contacts can help restore access without revealing keys.

Decentralized Solutions

Investigating blockchain or distributed ledger technologies for key management.

Usability Research

Conducting studies to balance security with intuitive user experience for recovery.