Parcela: Split-Key Encryption with 2-of-3 Secret Sharing and Steganographic Shares

B. Iggest Nerd
Independent Researcher

Parcela addresses a simple failure of conventional encryption: the single password. If it is stolen, all is lost. Parcela replaces that lone weakness with two independent requirements: a password-derived key and a quorum of shares. Files are encrypted with AES-256-GCM using Argon2id key derivation, then split into three shares such that any two reconstruct the encrypted blob. We specify formats, show steganographic embedding into valid PNGs, and discuss security under realistic partial-compromise threats. The system is implemented as a cross-platform application with a CLI and a RAM-backed virtual drive.

Keywords: secret sharing, authenticated encryption, Argon2id, AES-GCM, steganography

Introduction

Traditional encryption systems rest on a single secret: the password. It is elegant, but brittle. If the password is compromised, the data is lost; if the file is stolen, attackers can attempt offline guessing. Parcela removes this single point of failure by requiring two independent factors: a password-derived encryption key and physical possession of multiple shares. The aim is not to add exotic cryptography but to eliminate a common mode of defeat with a modest, robust structure that users can actually carry out.

Threat Model

We consider adversaries who can steal the encrypted file, steal a password, or gain access to one share. We assume the attacker does not compromise two independent storage locations at once. The system does not protect against coercion to reveal both password and two shares, nor against re-encoding of steganographic images by third-party services. This is a realistic, not heroic, model: it formalizes the common partial breaches that occur in practice.

System Overview

Parcela operates in four stages: (a) encrypt the file using AES-256-GCM with a key derived via Argon2id, (b) split the encrypted blob into three shares using a 2-of-3 threshold scheme, (c) distribute shares across independent locations or devices, and (d) reconstruct the encrypted blob from any two shares and decrypt with the password. The conceptual workflow is illustrated in Figure 1, and the architecture overview is shown in Figure 2.

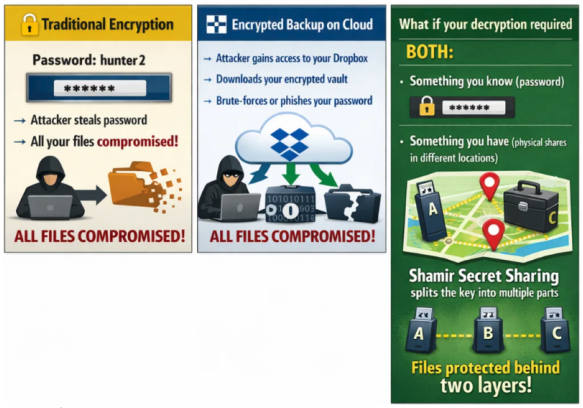

Figure 1

Conceptual workflow for encryption, splitting, distribution, and recovery.

Cryptographic Construction

Key derivation uses Argon2id with a random 32-byte salt, 64 MiB memory, 3 iterations, and 4 lanes, targeting approximately 2 seconds on modern hardware (Argon2 Team, 2015). The resulting 32-byte key is used with AES-256-GCM to encrypt the plaintext (Dworkin, 2007). The output is an authenticated ciphertext and nonce. The cryptography is standard and deliberate: Parcela depends more on composition and distribution than on novel primitives.

The encrypted blob format (v2) is:

PARCELA2 (8 bytes magic)
<salt> (32 bytes)
<nonce> (12 bytes)
<ciphertext>

No conflicts of interest to disclose.

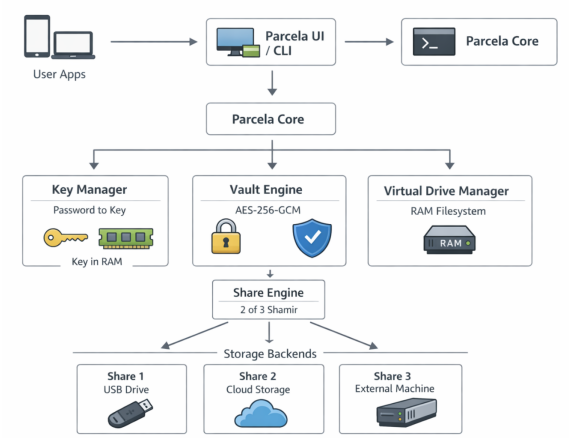

Figure 2

High-level system architecture for Parcela's encryption and share handling pipeline.

Secret Sharing

The encrypted blob is split with a 2-of-3 threshold scheme. Any two shares can reconstruct the original blob; one share alone reveals no information (Shamir, 1979). Shares are stored as either binary files or embedded within PNG images. This is a simple application of a well-known method, chosen for its robustness under partial compromise.

The share file format is:

PSHARE01 (8 bytes magic)
<index> (1 byte: 1, 2, or 3)
<total> (1 byte: 3)
<threshold> (1 byte: 2)
<length> (4 bytes, big-endian)
<payload>

Steganographic Image Shares

To improve concealment during storage and transport, Parcela embeds share data in custom PNG chunks. The images remain valid PNG files and can be opened by standard image viewers. The data survives normal file copying but is lost if the image is re-encoded by social platforms or editors. Example steganographic shares are shown in Figure 3.

Virtual Drive

Parcela includes a RAM-backed virtual drive that exposes decrypted files to the operating system without writing plaintext to disk. On Windows, it uses Projected File System (ProjFS); on macOS and Linux, a tmpfs-backed directory in `/tmp` is used. Files exist only in memory and are cleared on shutdown. This keeps ordinary workflows intact while limiting disk exposure.

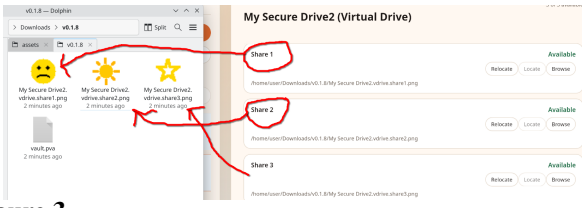

Figure 3

Sample PNG image shares containing embedded secret-sharing payloads.

Security Analysis

Password compromise alone is insufficient because reconstruction requires two shares. Share compromise alone yields no information about the encrypted blob or plaintext. File theft alone is ineffective due to Argon2id hardening and AES-GCM authentication. The design thus resists common offline attacks and improves resilience against partial breaches.

Residual risks include loss of shares, improper handling of steganographic images, and coercion attacks. Users must distribute shares across independent locations to preserve the threat model. The central claim is modest but useful: the system behaves well under the kinds of incomplete failures that occur in the real world.

Implementation Notes

Parcela is implemented in Rust with a cross-platform GUI and a CLI. The CLI supports splitting and combining shares, and can output either PNG-embedded or legacy binary shares for compatibility. The implementation focuses on dependable defaults and minimal operator error.

Use Cases

Parcela targets scenarios where a single key is too risky: estate planning, password backups, sensitive personal records, and high-value corporate secrets. Distribution of shares across physical and organizational boundaries is the primary security benefit.

Limitations

Parcela does not protect against an adversary who obtains the password and two shares. It also does not protect against malware on the local machine at the time of decryption. Steganographic shares must not be altered by services that re-encode images. These limitations are explicit and operational, not theoretical footnotes.

Conclusion

Parcela combines password-based encryption with threshold secret sharing to remove single points of failure. By requiring both a password and a quorum of shares, it offers a practical approach to long-term secure storage that is robust under partial compromise rather than perfect under ideal conditions.

References

[1] Argon2 Team. (2015). *Argon2: The memory-hard function for password hashing and other applications*. `https://argon2.online/argon2-specs.pdf`

[2] Dworkin, M. (2007). *Recommendation for block cipher modes of operation: Galois/Counter mode (GCM) and GMAC* (NIST Special Publication 800-38D). National Institute of Standards and Technology.

[3] Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612-613.

[4] Parcela project documentation and source code.