Dead Drop Data Drive

Or how I learned to stop worrying and love encryption

Disclaimer

Any opinions presented in this talk by the presenter do not in any way represent an official endorsement of these opinions by Freeside Technology Spaces, Inc., nor is intended to reflect the views of Freeside and its membership.

Freeside does not encourage or promote any illegal activity, including but not limited to mass surveillance by a totalitarian state, state persecution of whistleblowers, or warrantless and abusive search and seizure of information.

“No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.”

Article 12, The Universal Declaration of Human Rights

This Is A Thing

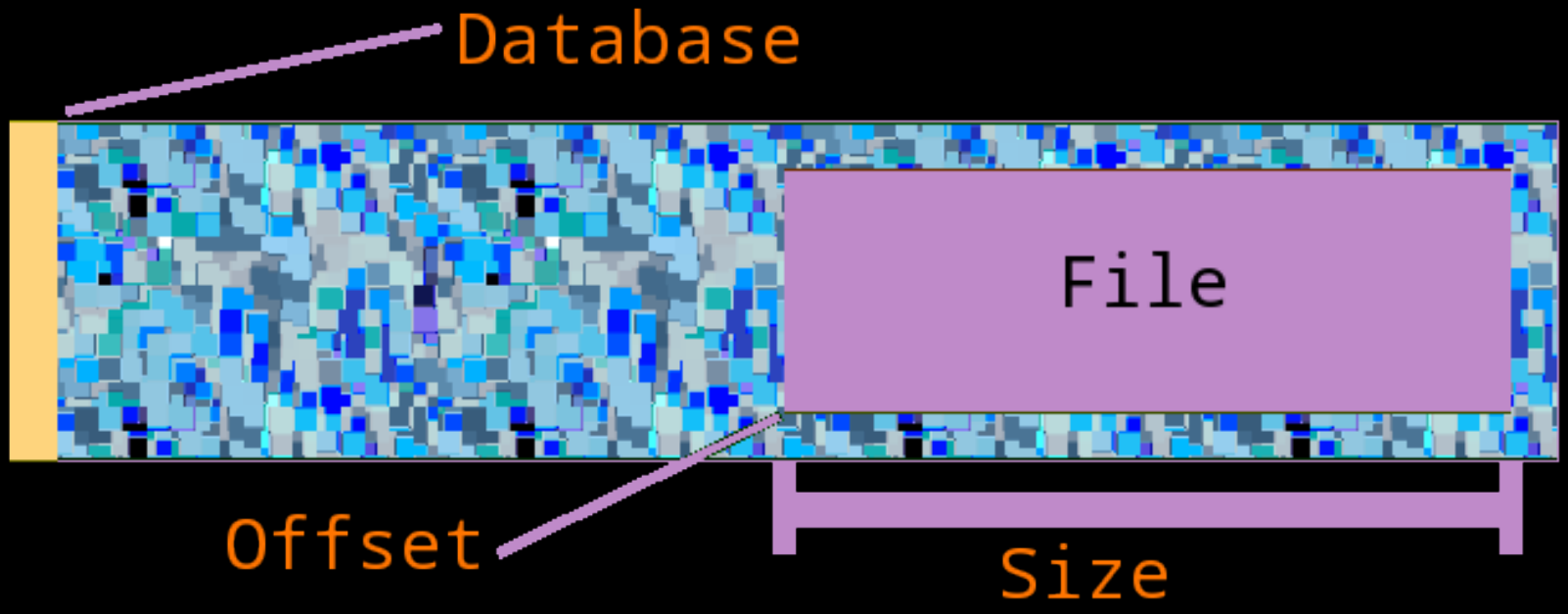
Overview

- A *dead drop* is an encrypted public drive that can be used to securely exchange data with others
  - **Asynchronous**: Store data today, pick it up tomorrow
  - **Offline**: Never connected to the Internet
  - **Highly Secure**: Only possible to decrypt a *portion* of drive with *correct* recipient's private key
- High Bandwidth, High Latency

Upload

- Encrypt the data using your private key, and your recipient's public key
- Upload the encrypted file to the drive using the program
- The program stores two pieces of information in a database:
  - File **size**
  - **Offset** (or hash, guid, locator) - this is some value that can be used to locate the file's position in the drive

Conceptual Diagram



Download

- Have the sender provide you with the **offset** value through some other channel (ex. text message, encrypted email, or sheet of paper you eat)
- Run the program to download, providing the **offset** value
- The program returns **size** of raw data located at the **offset**
- Decrypt the file with your private key

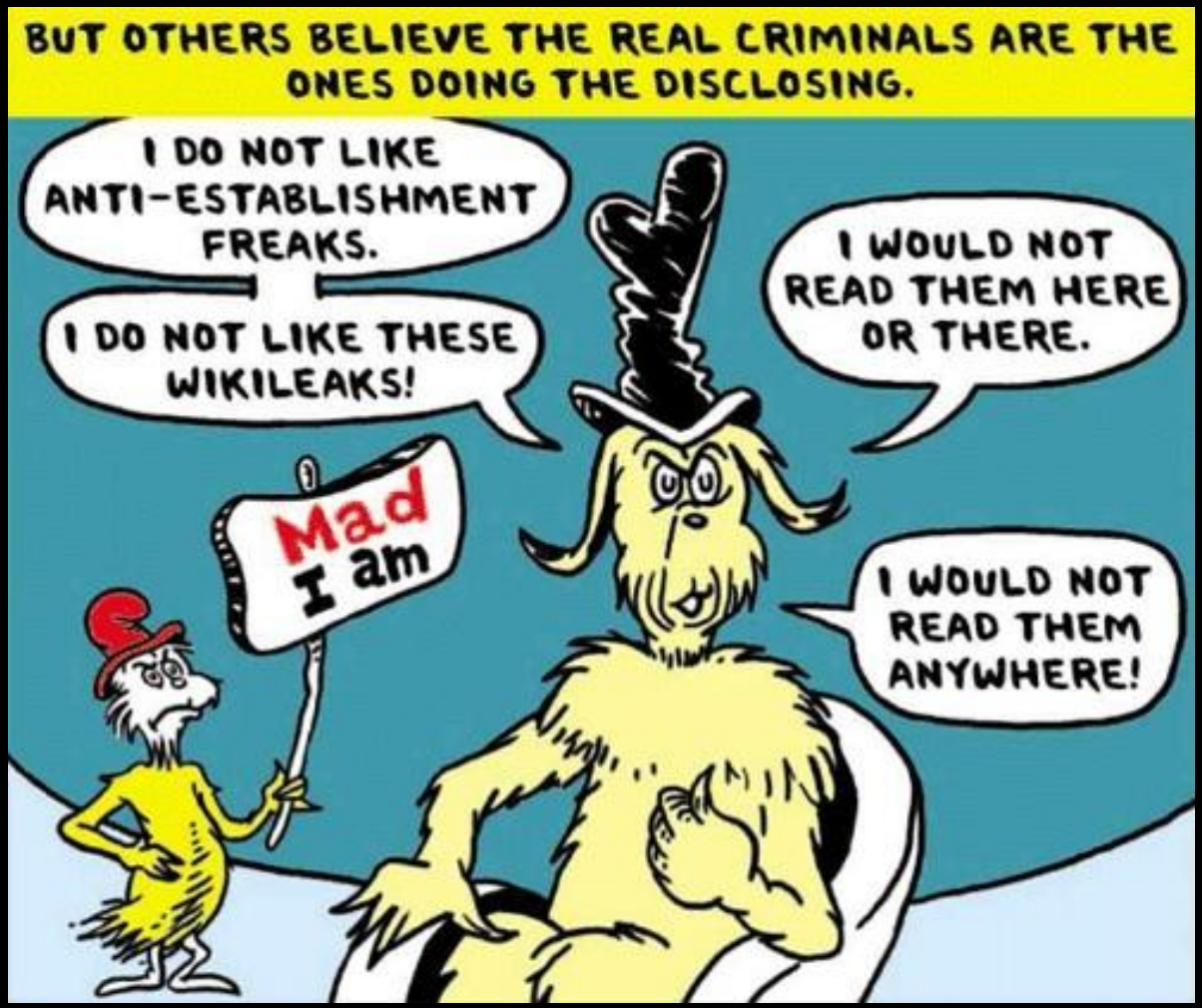
Analysis

If only the encrypted drive is compromised (stolen), an attacker must brute force query the database with sequential **offset** values in order to retrieve files.

Because the whole drive is encrypted, it is unknown where files start and end without the database.

If the database is compromised, then an attacker has all the encrypted files, but cannot identify neither the sender nor the recipient of each file.

Now You Can Download Anything



Open Questions

- Design
  - Database stored on drive itself, or stored online?
  - Does database need to exist at all (how important is preserving file integrity across multiple uses)?
  - Encrypt files for the user during upload?
- Attacks
  - Compromised files?
  - Binary diff after each use?
  - Direct surveillance?
  - Intercept of offset?
  - Compromised hardware?

Contribute!

- Donate a drive
  - 1TB+ desired
- Donate an enclosure
  - USB and e-SATA interfaces would be awesome!
- Contribute code to the git repository:
  - https://github.com/freesideatlanta/deaddrop
- Contribute ideas at your local CryptoParty!

The Fourth Amendment

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.