horcrux

Geo-Locked File Splitter

horcrux

- A utility that splits files into multiple pieces AND…
  - Distributes the pieces across peer nodes
  - Encrypts individual pieces such that they can be only accessed if the owning node is at a particular location and/or it is a certain time of the day
- Uses an overlay network to connect peers with each other
  - In the POC implementation, AWS is used

Splitting files

- Calculate Hash of the file contents
- Divide the file into N pieces and store the filename, hash and other information into each piece
- Encrypt few or all of the Pieces (more on this later)
- Distribute N-1 pieces among other available nodes.
- Store 1 piece locally

Joining files

- A node where join is requested takes the filename as a key and searches the local pieces for a match. If no match found, the search terminates.
- If a match is found, the requesting nodes broadcasts a get request for the hash stored in the local piece
- A receiving peer, takes the hash and looks for the hash among its local pieces. If a match is found, it is decrypted using a known criteria stored in the piece (or shared out of band) and sent to requester
- The decrypted pieces are received and assembled. A hash is calculated and compared to the one stored locally. If they match file is created and returned to the user.

Encrypting

- When a piece is being created, it can be encrypted using the GPS co-ordinates. A piece can also be encrypted using a time of day.
  - For e.g. A piece p1 can be encrypted using GPS co-ordinates of San Francisco and also with time 2300-0300 hours.
  - Likewise another piece p2 can be encrypted using GPS co-ordinates of Boise, Idaho
  - And a third piece using time 0100-0500 hours
- This means a file can be joined only if p1's node is in San Francisco between 11pm-3am, p2's is in Boise and p3's is online during 1am and 5 am

Decrypting

- When a node receives a hash that has a matching local piece it decrypts the piece using GPS data and current time as a key
  - The decryption keys used follow an approximations, for e.g a piece tied to GPS location: -122.419416, 37.774929 would have been encrypted with -122,37 as the key
  - And the decryption engine would try the current Longitude and Latitude as the encryption key with certain approximations.