January 23, 2019

Opacity Storage Whitepaper v0.1

Secure. Anonymous. Opaque.

Introduction

Few cloud storage providers offer a service that is convenient, private, and highly secure. Despite the frequent use of cloud storage among private users and enterprises of all sizes, there have been a plethora of instances that represent how unsecured cloud infrastructure can cause critical data breaches.

In October 2017, misconfigured S3 Buckets used by a prominent software company led to the exposure of crucial infrastructure data, including 40,000 passwords, decryption keys, certificates, as well as administrator login credentials. Additionally, in June 2017, unsecured S3 Buckets led to the breach of sensitive US voter information and subsequent marketing research related to potential voter behavior in future elections that affected 198 million American citizens. Recent studies suggest that 7% of all AWS S3 Buckets are publicly-accessible and as many as 35% are unencrypted. Yahoo, Facebook, Equifax, and many other 'trusted' names have had data breaches affecting hundreds of millions of their users around the world.

Although cloud storage offers a convenient alternative to offline storage devices, it is evident that cloud providers are not necessarily the most secure option available to consumers. Not only are they susceptible to identity theft, financial theft and even

potential for physical harm, but the average data breach cost businesses $3.62 million in a 2017 study. Businesses are vulnerable to litigation, to customer and employee data loss, to placing their Intellectual Property at risk, and to a potential impact to reputation. This situation is not ideal and unsustainable.

Opacity Storage provides a highly secure storage solution that will significantly decrease the probability of cloud-related data breaches. This whitepaper describes the Opacity Storage infrastructure and the security mechanisms embedded within Opacity's architecture, as well as the current and future ways in which users can interact with the Opacity Storage interface.

January 23, 2019

Infrastructure

Two Ledgers, One Storage Platform

In the software's current iteration, Opacity Storage implements the use of both the Ethereum blockchain and the Tangle. The Ethereum Blockchain is used as a valid payment structure, using Opacity's native utility token, OPQ. OPQ is based on ERC-20 architecture, and one OPQ currently represents 64GB of storage on the Opacity storage interface. OPQ is presently available on several exchanges and is the commodity that users spend to store data on the Opacity Storage platform (how users interact with Opacity's storage interface using OPQ will be detailed in the user experience section of this paper). OPQ is also the incentive mechanism within the network for brokers to perform the work of file attachment as described later in the document.

The Opacity platform uses its own instance of the Tangle, a Directed Acyclic Graph (i.e., a blockless distributed ledger), for uploading and securing user-uploaded data. Each submitted transaction on the Tangle performs Proof-of-Work for two prior transactions, ultimately confirming them. These two transactions are referred to as the branch and trunk. Each transaction has a payload capacity which is used to retain the data that is uploaded by a user. Transactions are propagated throughout a network of Nodes that have mutually peered with each other, while each Node maintains a redundant copy of the transactions. This leads to a significant redundancy of data

copies which heavily mitigates the risk of data-loss while not relying on a centralized hosting provider.

Data is stored on the Tangle in ~1 KB parts within the transaction payload. A SHA256 hash is first converted to a SHA384 hash and becomes the referenced basis for storing and retrieving data on the Tangle. When the SHA384 hash has been selected to represent data, it is converted into its trinary form to represent the recipient address of the transaction. To retrieve the data from the Tangle, the hash is again converted into its trinary form to produce the recipient address, and then all of the transactions under the address are recovered. The transaction with the oldest issuance timestamp contains the payload data that represents the selected hash.

Initial File Upload and Encryption

When a user wants to upload a file via Opacity, the data is split into parts and encrypted locally in the browser. This isolation ensures that a malicious actor cannot retrieve the user's data, as it can only be accessed with the corresponding encryption key, the *Opacity Handle*.

The first eight characters of the Opacity Handle represents the name of the file. These eight characters are copied from the first eight characters of the name of the file uploaded. If the file name is less than eight characters in length, a salt of eight characters is added to the file name, and the first eight characters of the newly-generated file name are used in the Opacity Handle.

January 23, 2019

The Primordial Hash is a 64 character long SHA256 hash of random input that is generated from within the user's browser with as much entropy as possible.
The last eight characters of the Handle is the cryptographic salt that differentiates the Primordial Hash from the encryption key. The salt is used to protect the data further in case the Primordial Hash is found because of a future weakness in a hash function or a rainbow table attack on the Genesis Hash. Therefore, the entire 80-character long Handle is the whole encryption key used to encrypt and decrypt the split parts of the data. The Primordial Hash initiates a sequence of SHA256 hashes (from which we derive sha384 hashes which we then convert to their trinary form) that represents split pieces of the data. Each SHA256 hash is a hash of the previous SHA256 hash in the data map. The data is broken down into ~1 KB parts. Then, each piece is individually encrypted with the entire Handle as the key. Each section is sequentially represented by a Hash iteration (Genesis, N1, N2, etc.), and is eventually submitted as a Tangle transaction each by two Broker Nodes.

Broker Nodes

In the Opacity storage infrastructure, Broker nodes are responsible for performing the proof-of-work necessary to attach data to the Tangle. Additionally, Broker nodes are responsible for verifying that a user has paid via the Ethereum blockchain. Once the user’s file has been broken into chunks and encrypted locally in the browser, the Broker nodes attach the encrypted chunks to the Tangle. The

proof-of-work is split between two Broker nodes; however, the broker nodes are capable of initiating AWS Lambda functions (serverless functions) to offset some of the processing required for the proof-of-work if necessary. As proof-of-work for each Tangle transaction containing the encrypted data is completed, the chunks are stored on the nodes of Opacity's private Tangle.

File Retrieval

When a file is retrieved from Opacity using a Opacity Handle, the client derives the genesis hash from the handle, converts it to a Tangle address, then retrieves metadata chunks from Opacity's private Tangle and decrypts it with the handle. Through decrypting the metadata, the client discovers the name of the file, the extension, and how many chunks are in the file. The client uses the genesis hash and the information on how many chunks are in the file to rebuild the datamap. Each hash of the data map is converted to a Tangle address which is the location of a chunk on the private Tangle. The client asks the private tangle for all of those chunks, and decrypts each of them and reassembles them into the file. Opacity has written a custom IXI module that is utilized during this retrieval process to increase the efficiency of file retrieval by transferring only the payload, not the whole transaction, and allows batching these requests together.

January 23, 2019

Pegging the Opacity Token

The Opacity token (OPQ) represents 64GB of storage space as defined in the smart contract. This quantity is not yet fixed in order to allow for adjustments that may be required to fine-tune the network capabilities as the system grows. Over time, the intention is to provide a stabilizing reference for the utility provided by the Opacity token, such that the storage amount received adjusts to storage market supply and demand. The mechanism to properly adjust this value will need to be defined in the future and the 64GB amount is subject to change until such time the peg becomes fixed.

January 23, 2019

User Interface

Users can interact with Opacity Storage through the web interface at storage.opacity.io. The web interface provides a quick and user-friendly way of uploading and retrieving a file using Opacity Storage. The web interface has several stages of interaction depending on the user's choice of action.

When a user needs to upload a file, they are prompted with the standard upload form (as of January 2019, all files must be 250MB or less). This form will allow the user to choose a file from their device to upload via Opacity Storage, and will present them with the price of OPQ necessary to upload the selected file. In its current form, the upload interface has the number of years of file retention fixed to one and the broker nodes are currently chosen for the user. However, these settings may change in the future.

After confirming the upload cost and file to be uploaded, the user is presented with an Ethereum address where the appropriate OPQ amount must be sent. The broker node confirms the payment and the user is presented with a new screen displaying a progress bar of the file as it is broken into individual chunks, encrypted and sent to the broker. After the broker receives the fully encrypted file, it will begin to attach each chunk to the tangle, and the user will receive the Opacity Handle when this process begins. This enables the user to bookmark the URL at this point and navigate away from the page without interrupting the upload. Once the handle is obtained, the

user, or anyone else with the handle, can access the file from the storage interface once the attachment has been fully completed.

The file retrieval action of the storage interface is straightforward. When a user chooses to retrieve a file, they are prompted with a form asking for their handle. After the handle is submitted and verified to be legitimate, the corresponding file begins to download via the user's browser.

Alternatively, files can be linked and viewed directly in the browser with the experimental alpha.opacity.io interface. If enabled, a Handle history is stored within the browser, as well.

Opacity Telegram Robot

In addition to the two browser based interfaces, users can also interact with the Opacity Telegram bot. The Opacity Robot is easily accessible through the Telegram application which can be installed on mobile or desktop. It provides a convenient, although less private upload and download capability directly within the application. The user can initiate a direct conversation to the bot using @opacityrobot and by sending it a file. The robot will start the session and request payment from the user by providing the ETH address for payment, along with a QR code for convenience. After the payment is received, the bot follows the same process as the web interface and provides the user the Opacity Handle after file chunks are uploaded to the brokers. It proceeds to

provide a status of the attachment of chunks to the Tangle and returns the completed status to the user.

January 23, 2019

Scaling for the Future

Further improvements to the storage interface are planned in order to promote widespread adoption.

Expand Core Capabilities

Opacity currently allows only a single file to be uploaded at a time via the web interface. Opacity's capabilities must expand to allow for the upload and download management of many files simultaneously. Multi-file upload, as well as a larger file size limit (currently set at 250Mb per file), will provide an improved user experience that will be comparable to popular storage providers like Dropbox or Google Drive.

Also, users will be able to pay upfront for larger buckets of storage in a variety of sizes, and not limited to the current per-file payment style of the web interface.

Metamask Integration

The payment invoice can be sent directly to Metamask for approval, making it easier to make payment directly within the browser.

January 23, 2019

Served Content

Opacity will provide the ability to be able to host and serve content such as videos and webpages in order to make stored content easily accessible, publicly or privately.

Native Desktop and Mobile Applications

Expand platform to desktop and mobile by providing native applications. These applications will allow users to interact with the Opacity storage platform, and manage their files straight from their preferred device, without the need to visit the Opacity website.

API

An S3-compliant application programming interface (API) will expose the complete functionality of Opacity's secure storage solution for developers to build applications using Opacity as the backbone for their secure storage requirements. Being S3-compliant will also allow enterprises to seamlessly switch from their current cloud storage solutions to Opacity using their existing codebase.

January 23, 2019

Decentralization

The security of the Opacity network will not be complete until it becomes fully decentralized. Enabling public nodes to support the ecosystem and be rewarded with OPQ tokens as payment will secure the network for the future.

January 23, 2019

Inherited Limitations

Several limitations involved in Opacity's architecture were inherited from the project from which Opacity was forked: the Oyster Protocol. Although future-oriented, the Oyster Protocol created some considerable limitations. The Opacity team is exploring all options to address these known issues for future iterations of the software.

The Tangle and Bottlenecked File Uploads

Due to the nature of the Tangle's underlying software, only an extremely small portion of data can be stored in each transaction on the Tangle. As files grow in size and as many users attempt to upload many large files, the number of transactions required to upload these files to the Tangle increases significantly, leading to longer upload times and bottlenecks on the client side.

Moreover, the Tangle itself comes with a myriad of other issues that create performance concerns, including the proof-of-work obligation defined by the MWM, data expiration, and the time associated with excessive transaction querying involved in piecing a retrieved file back together.

Opacity will need to build solutions or seek alternatives in order to overcome these performance challenges.

January 23, 2019

Immutable Files

Due to Opacity's use of the Tangle as a storage medium and breaking each file into a large number of chunks, it becomes complicated to create solutions for users to edit their stored files, which is a common feature many people have come to expect from online storage providers.

Web Node Adoption Uncertainty

The web nodes were an important part of the Oyster Protocol, the project that Opacity was forked from. Handing off work to the web nodes, and broadcasting those file chunks to a vast network of Tangle nodes, would allow the broker nodes to overcome some of the inefficiency inherent within the system. However, if too few website owners used the web node script, the broker nodes would not have sufficient access to this source of labour.

Also, web nodes had several underlying real world issues that made it difficult to execute effectively on the web. Websites with high bounce rates and/or short session durations would hog up genesis hashes only to touch a very small portion of the file's Tangle transactions. In addition, relying on IOTA meant that web nodes had to depend on WebGL2 to perform proof of work, which is currently not yet supported on a significant number of mobile browsers. This dependency severely limited the future of

the web nodes, since mobile web browsing is trending to overtake most of the web traffic market share.

Streaming Challenges

Some users expect streaming video and audio from storage platforms. In Opacity's system, the client must check for treasure in random locations when downloading. This creates a challenge with streaming due to this randomness and the need for deterministic byte locations in order to stream the data.

Subscription Payments Not Integrated

Although it is possible to eventually create smart wrappers to allow users to have a subscription option for storage on Opacity's network, this capability isn't an integrated component of the protocol and the smart wrappers to make this possible would be very complex.

January 23, 2019

Conclusion

With major data breaches related to cloud storage still plaguing the global IT industry, there is extensive evidence supporting the need for a more secure and trustless storage solution. Opacity aims to provide this much-needed service in the form of an encrypted, decentralized storage solution. By separating uploaded user data into encrypted chunks, and supplying a uniquely-generated Opacity handle as the sole mechanism of file retrieval, Opacity offers a unique storage solution that is both secure and zero-knowledge. Furthermore, with plans to expand Opacity's web storage interface into desktop and mobile applications, and by providing a robust API, Opacity will fit the secure storage needs of both consumers and enterprises. By using Opacity for their storage needs, users can rest assured knowing that their data is secure and resilient to the far-too-common data breaches that occur in the cloud storage sector.