STRATOSTORE

Storing sensitive documents “above the clouds”

TECHNOLOGY

Contents

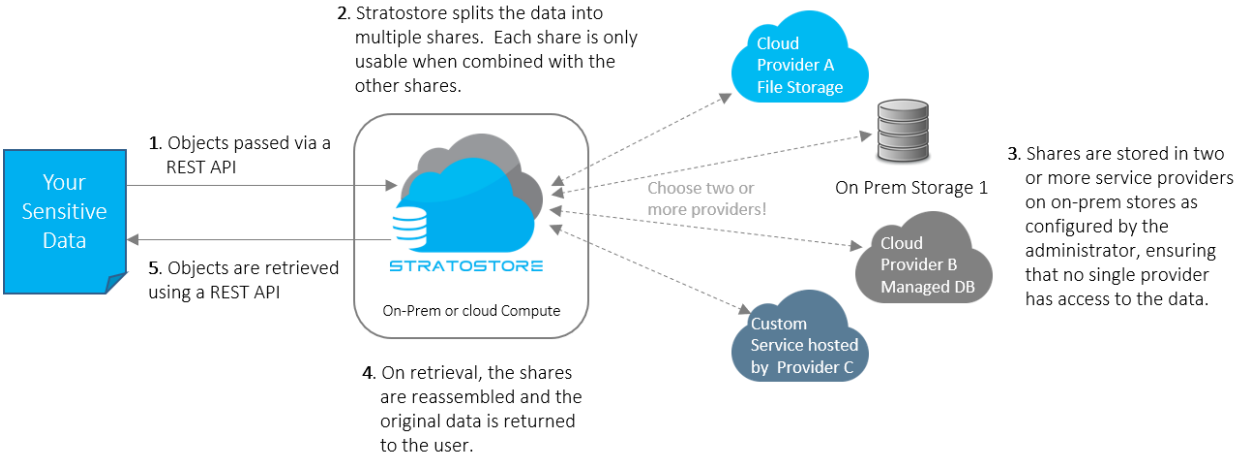
Contents 2
What is Stratostore? 3
What problems does Stratostore solve? 4
Background – the one-time pad. 5
How does Stratostore work? 6
Splitting Data 6
The most basic use case – two shares 6
A visual example. 7
Splitting Data – An example with n shares. 8
Storing Data Shares 9
Configuration Options 9
Red teaming: Trying to Steal Stratostore Data 10
Other Considerations 11
Hardware Random Number Generators vs. Cryptographically Secure Pseudo RNGs (CSPRNGs) 11
To encrypt or not to encrypt Shares? 11
Downsides 12
Contact Us 12

What is Stratostore?

Stratostore is an application that securely splits any file (binary data) into two or more "shares" using a one-time pad approach. Each share is completely meaningless when separated from the other shares.

Stratostore can then write each share to a separate cloud or on-premises database or file system. This allows you to use third-party storage solutions without giving your data to any one provider.

The diagram below shows the flow of data through Stratostore.



Stratostore gets its name because it stores data "above" any one individual cloud. Instead, Stratostore splits the data across multiple clouds. As a result, your data is stored in the cloud, but no single cloud provider has access to your data.

What problems does Stratostore solve?

Stratostore is a storage solution for sensitive data. By splitting your data into shares and storing each share in a distinct location, Stratostore solves the following problems:

- **Eliminates possibility that a cloud administrator can steal your data.** Many corporations are hesitant to move from an on-premises data store to a cloud provider because their data is sensitive, and they can't risk a malicious third-party cloud administrator stealing their data. With Stratostore, each share of data that is stored is meaningless without the other shares of data. So, no single cloud administrator acting alone could compromise your data – he would have to collude with administrators from all other providers storing shares of your data.

- **Minimizes risks associated with misconfiguration of your data stores.** There are frequent reports of data "leaking" from cloud provider storage because administrators have misconfigured the access control permissions and inadvertently exposed data. Further, bugs in cloud provider code could accidentally introduce security vulnerabilities that allow your data to be stolen by a bad actor who is not affiliated with the cloud provider. Since Stratostore puts shares of your data in multiple locations, your data is completely secure as long as one location remains uncompromised.

- **Prevents a malicious corporate insider from stealing your data.** Insider threats are very real. While it can be necessary for your corporate administrators to maintain production access to your data, Stratostore can help you enforce "two-man" rules that would require administrators to collude to steal data. With Stratostore, you can allow one set of administrators to have access to data stored in Cloud Provider A and a second set of administrators can access data stored in Cloud Provider B or on-premises database C. By splitting your administrator access across multiple providers, no individual within your company can steal data without colluding with another administrator, dramatically reducing your insider threat risk.

- **Prevents reliance on symmetric encryption techniques that lack perfect secrecy.** When configured properly, Stratostore's one-time pad approach provides "perfect secrecy." Symmetric encryption schemes, like AES-256, are widely accepted as a secure. However, these encryption techniques are theoretically susceptible to being cracked, especially as quantum computing continues to evolve. Stratostore uses one-time pads, which are information theoretically secure, meaning that the encryption cannot be cracked even in theory. Many cloud providers offer managed encryption key services where they retain your keys for you. While this is very convenient, the thought of turning over encryption keys to third parties can be a concern when dealing with your most sensitive data.

Background – the one-time pad.

To understand how Stratostore works, one must first understand the one-time pad. This section provides background information on the one-time pad.

A one-time pad is an encryption technique that is "information theoretically secure," meaning that it cannot be broken even with unlimited computing power. In a one-time pad, the *plaintext* is encrypted with a *key* (also referred to as a one-time pad) that is at least as long as the message. There are several ways to combine the plaintext with the corresponding bit of the key, but the examples below show the plaintext and key being combined using the "Exclusive Or" (XOR) operation.

A brief explanation of XOR. A bit can have a value of either 0 or 1. When combining two bits A and B, if both bits have the same value, then the result of the XOR operation is 0. If both bits have different values, then the result of the XOR operation is 1. Figure 1 below summarizes all possible outputs of XOR'ing two bits together:

Input A	0	1	0	1
Input B	1	0	0	1
Output (A XOR B)	1	1	0	0

Figure 1 - XOR Truth Table

Using a one-time pad to encrypt plaintext. Let's assume that we have a 10,000-bit piece of plaintext data that we need to encrypt with a one-time pad. First, we must randomly generate a 10,000-bit key. Second, we must XOR the plaintext and the key. The resulting output along with the key are now both "shares" of the data as shown in Figure 2. Each share is worthless without the other share, and both shares will be required to recreate the plaintext.

Plaintext (10,000 bits)	1	0	0	1	0	1	…	0	1	0
Key (randomly generated 10,000 bits, becomes "Share A" of the data)	0	0	1	1	0	1	…	1	1	0
Result (Plaintext XOR Key, becomes "Share B" of the data)	1	0	1	0	0	0	…	1	0	0

Figure 2 - Creating two Shares of Data

Retrieving the plaintext. Sometime later, a user may decide that he would like to combine the two shares to obtain the original data back. The two shares can be combined using the XOR operation to obtain the plaintext as shown in Figure 3.

Share A (Result from above)	1	0	1	0	0	0	…	1	0	0
Share B (Key generated above)	0	0	1	1	0	1	…	1	1	0
Original Plaintext Recovered (Share A XOR Share B)	1	0	0	1	0	1	…	0	1	0

Figure 3 - Combining the Shares to get the Original Plaintext back.

How does Stratostore work?

There are two key elements of Stratostore:

- **Splitting data** into multiple Shares
- **Storing data shares** in multiple cloud providers or on-premises database.

During Stratostore configuration, the administrator can configure the number of shares, the data storage locations, and a few other variables. For more details about available configurations, see the "Setup and Configuration" section.

Splitting Data

Stratostore uses a one-time pad to split data. Two examples are provided below; one in which two Shares are created and another in which n shares are created.

The most basic use case – two shares

Stratostore offers a REST interface that allows users to create or update **data**. This data is passed to Stratostore as a byte array.

Stratostore creates a random byte array called **share A** that is the same length as the data. The data and share A are then XOR'd together to create **share B**.

The original data is deleted from memory, and now the data only exists as share A and share B. Share A and Share B are then stored as described in the Storing Data Shares section.

To retrieve the data, one must retrieve both share A and share B from their distinct storage locations. Share A and share B are XOR'd back together by Stratostore, and the original data is returned to the user.

A visual example.

Storage

Suppose that we have the PNG file below that we want to save in Stratostore.

Figure 4 - PNG file to be saved in Stratostore

This file is turned into a byte array and passed into Stratostore. Stratostore creates a random byte array called Share A that is the same length as the PNG file. This random byte array can be visualized as follows:

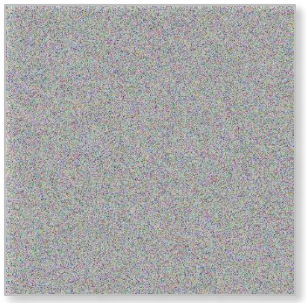


Figure 5 - Share A visualized as an image

Share A, the random byte array shown in Figure 5, is then XOR'd with the original data shown in Figure 4 to create share B. Share B can be visualized as follows:

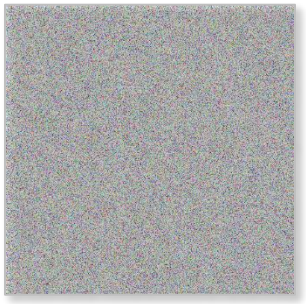


Figure 6 - Share B visualized as an image.

Share A and share B are then stored in individual repositories.

Data Retrieval

Sometime later, a user with the appropriate permissions wants to retrieve the original data from the disparate storage locations. Stratostore retrieves both shares, XORs them back together, and returns them to the user as visualized below:

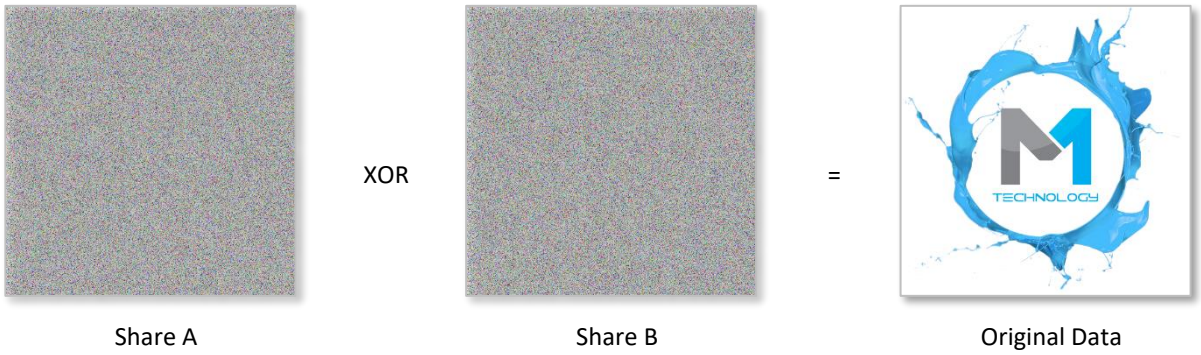


Figure 7 - Visualization of Stratostore combining Share A and Share B to retrieve the Original Data

The example above is an actual example that can be run from the Stratostore unit tests!

Security

Note that share A and share B look like static. They do not reveal any information about the original image. If either share is stolen by an administrator, they simply have an image that looks like static. Because the one-time pad is "information theoretically secure", one part of this file is worthless without the other part. If a human stole share A and XOR combined it with all other possible shares, they would create every possible image of this size, so they can't possibly know which image was the one that was originally saved.

Splitting Data – An example with n shares.

During setup, the administrator can configure the number of shares that are generated when splitting data.

As **data** is passed to Stratostore as a byte array, Stratostore creates **n-1 random shares** of the same length as the data, where n is the number of shares requested. Shares 1 through n-1 are all XOR'd with the data to create **share n**. The data is purged from memory and shares 1 through n are the stored in n distinct locations.

Upon retrieval, all shares are XOR'd together to retrieve the original data.

Storing Data Shares

Stratostore connects to multiple data storage locations and more options are continuously being added. Examples of supported data sources include Amazon S3 and Mongo DB. Alternatively, Stratostore can be configured to POST data to an IP address where it can be stored or further processed by a custom service. The full list of supported storage options can be viewed on the Stratostore Github page.

When choosing data storage options, it is important to consider two things:

1. **Choose different providers so that no single provider has access to all your data.** For example, if you choose to store your data in Cloud Provider A's file storage solution, don't choose a second store that is a managed by that same Cloud Provider (i.e. a managed database also hosted by Cloud Provider A). Instead, choose a different provider like Cloud Provider B or store the data in an on-premises database.
2. **Ensure that your own administrators do not have administrator access to both storage solutions.** Administrators can either by given access to Cloud Provider A or Cloud Provider B, but no single administrator should ever have both. Similarly, if data is stored in one cloud provider and an on-premises database, the administrators for the cloud should be different than those managing your on-premises solution.

Configuration Options

Prior to running Stratostore, the following configurations must be set.

Setting	Description
Number of Data Stores	The number of stores that the shares will be distributed to. A minimum of 2 are required.
Individual Data Store Configuration	Configuration per store, like store name, IP address, associated keys, etc.
Source of Random	Specify that either a true random number generator (hardware) or a cryptographically secure pseudo-random number generator (like Java's SecureRandom) are to be used.

Figure 8 Stratostore Configurations

Red teaming: Trying to Steal Stratostore Data

Figure 9 Malicious administrators cannot access both Shares of data, so administrators must collude to steal data. below shows where shares of data (1 and 2) are stored in the Stratostore environment (Cloud A or Cloud B).

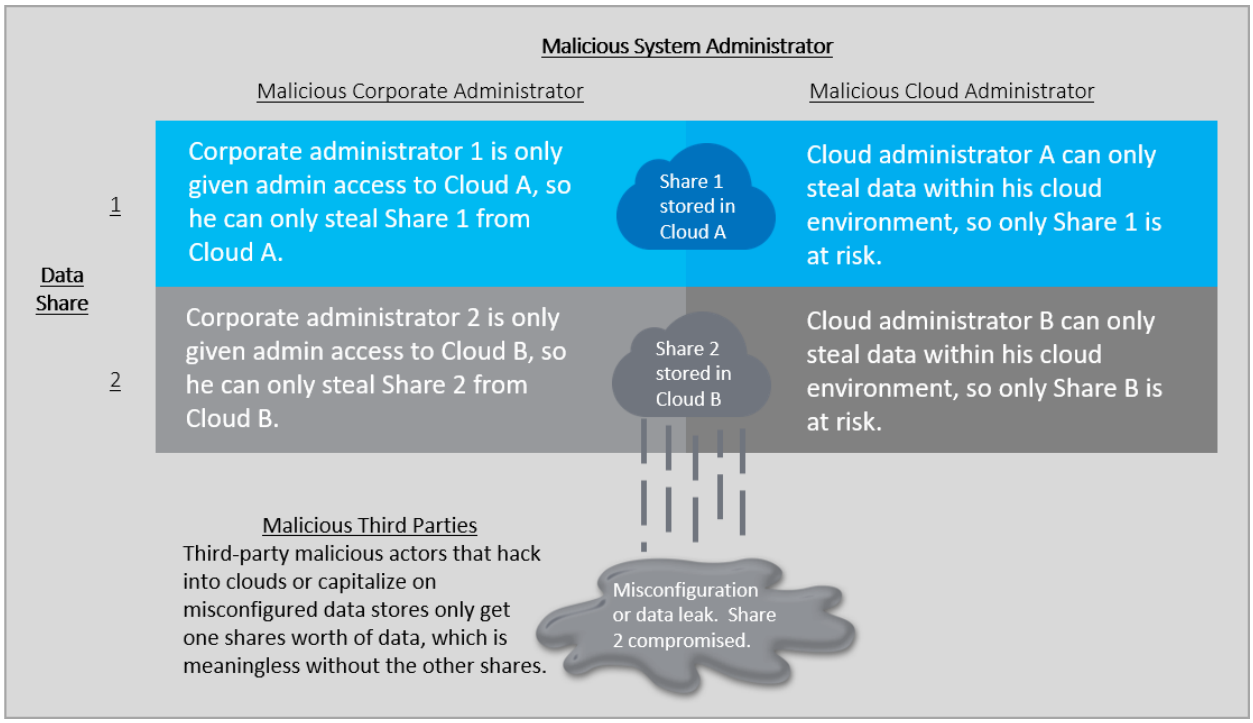


Figure 9 Malicious administrators cannot access both Shares of data, so administrators must collude to steal data.

If configured properly using at least two corporate administrators and two different cloud on on-premises stores, then Stratostore can prevent any single individual from having access to your data. Since no one individual has access to all of the data, it forces two administrators to collude to steal data, which dramatically decreases your risk. Further, since Cloud Administrator A and Cloud Administrator B to not know of each other's existence, their ability to collude is very low.

Finally, Figure 9 Malicious administrators cannot access both Shares of data, so administrators must collude to steal data. shows that data that "leaks" from a misconfigured or hacked cloud provider results in only one share being lost. Since the data is split across multiple shares, the impact of an individual data spill is significantly less than if the actual data or all shares were compromised.

Other Considerations

Hardware Random Number Generators vs. Cryptographically Secure Pseudo RNGs (CSPRNGs)

Stratostore can accept an input of random numbers (e.g. from hardware) or use a Cryptographically Secure Pseudo-Random Number Generators (CSPRNG) to generate random numbers. The one-time pad is only information theoretically secure if true random numbers are used. There are differing opinions as to whether CSPRNGs are as secure as hardware-based random numbers. In practice, most implementations will be happy using pseudo-random numbers to create byte arrays, possibly using available tools to increase the amount of entropy on a machine. It is up to the end user to choose the right source of random to meet their requirements.

To encrypt or not to encrypt Shares?

If a true random number generator is used, there should theoretically be no reason to encrypt your shares prior to storage. That said, Stratostore has not gone through the FIPS 140-2 validation program, so it may not meet the stringent requirements of some customers. There is no harm in encrypting your shares prior to storage, so it is recommended that you do so.

For users that typically don't trust keys generated by a cloud provider, perhaps the Stratostore approach gives you enough piece of mind to use customer managed keys in a cloud provider rather than developing your own per-document encryption scheme.

Downsides

The following are items to consider when choosing whether to use Stratostore to protect your data:

- **Slight decrease in availability.** All data stores must be available for Stratostore to work. By choosing two data stores with high availability, you must multiply the availability of both stores to determine the system's availability. For example, if you choose two stores each with 99.99% availability, your system availability will only decrease by 0.01% to 99.98%. This is still very high!
- **Storage costs increase linearly with the number of shares.** Each share is the same size as the original data. So, your storage costs will be double if you have two shares, triple with three shares, etc. Fortunately, storage costs for files are generally inexpensive, so this is not a huge concern.
- **Slight decrease in durability.** If any shares of data are lost, your data is completely unrecoverable. Some file stores offer 11 9's of durability (i.e. 99.999999999%). By choosing two very durable sources, you are less likely to lose data.

Contact Us

Stratostore is an open source product managed by M1 Technology, LLC. Please reach out to stratostore@m1technology.com for help implementing Stratostore to protect your sensitive data.