



Technology Whitepaper

Updated February 2008

The STOREFolder Backup system has been designed to be a secure, reliable, efficient, scalable, modular, and portable data backup solution. An overview of each of the following components will be given: data encryption, revision management, pass phrase recovery, client/server protocol, data repository, and account management.

Executive Summary

Each file is encrypted using the AES-256 standard using a user-provided pass phrase as the key. The encrypted information is sent across an authenticated and secure (SSL/TLS) Internet connection to the server, where it is stored encrypted on the RAID-6 array.

Data Encryption

The information is encrypted by the client program using the AES-256 symmetric encryption algorithm. The 256-bit encryption key (as well as another 256-bit HMAC key) is generated by running the PBKDF2 algorithm on the pass phrase as described in RFC 2898.

Each file is divided into 2K blocks. Each block is encrypted using AES-256 in CTR mode (each block uses a different crypto-random nonce), and an HMAC (using SHA-256) is appended (to guarantee integrity upon restore). The data is fully encrypted before it ever enters the network. Encrypting the data on the client is more secure, and it makes the server more efficient and scalable.

Filenames are currently not encrypted. Encryption of file and directory names is a planned future enhancement, and can be done by upgrading the client software – no changes need to be made by the server.

Revision Management

The use of encryption precludes the possibility of performing file delta calculation at the server. Performing the delta calculation at the client also increases server scalability and efficiency.

As each block is stored on the server the client stores a 64-bit CRC (two 32-bit CRCs generated from different polynomials) associated with that block. During the next backup if a file has been changed (its modified date/time stamp has changed or its size has changed) then the

client will compare the new 64-bit CRC with the stored CRC. If the CRC has changed the client will upload the new encrypted block. Otherwise the client will tell the server that block has not changed.

The CRC block fingerprints and other data are protected by transactions such that if the backup of a file fails, all 64-bit CRCs and other information are rolled back to the consistent state.

Pass Phrase Recovery

The pass phrase is central to the system's security. A strong pass phrase is vital for sufficient encryption strength. However, a strong pass phrase can be hard to remember, and all data would be worthless without the exact pass phrase. Thus, a secure pass phrase recovery system has been implemented.

When a pass phrase is created or changed the user answers several security questions. The answers to the questions generate an encryption key, which is used to encrypt the pass phrase. Combined with public-key cryptography, we have created a secure system that requires the cooperation of both the customer that created the pass phrase and STOREFolder to recover the pass phrase. When the pass phrase is finally recovered, only the customer has the ability to view the recovered pass phrase.

Client/Server Protocol

Each end user is given an account username and password (which can be changed). The client connects to the server via a TLS (SSL) connection (providing confidentiality and integrity), and requires the server's certificate to be issued by the STOREFolder Repository root certificate authority (to prevent spoofed servers from stealing login credentials).

The client authenticates to the server with its username and password. At this point the server may redirect the user to a different server and port.

This allows the customer to use the storage server in different locations, as may be required by the customer's

disaster recovery plan. The end user is not aware of this complexity and never needs to change anything.

The communication protocol itself is an endian-independent, flexible protocol designed to support changes without breaking backwards compatibility.

The protocol also allows for precise control of bandwidth usage, allowing the user to specify a maximum cap on the bandwidth usage during business and off hours.

Data Repository

All data is stored on top of industry proven system components. Our storage backend employs modern, proven filesystems that have the latest technology for scalability and reliability, such as journaling, multi-terabyte partition support, 64-bit file sizes, and millions of files per directory, offering both horizontal and vertical scalability. We use additional technology that adds an additional layer of redundancy above and beyond RAID-6 to offer additional protection against silent data corruption.

The current version of the file always stores the complete file (encrypted). Historical versions store data blocks that differ from the next (more recent) version. Thus, to restore the 5th version of the file the software applies the deltas from the previous 4 versions and then applies the 5th delta. This is done as the file is uploaded to the server and is very efficient. This method also makes uploading new versions efficient, as all previous historical versions need not be changed.

When the client detects that a file has been deleted it notifies the server during the next backup. The server annotates the filename with the deleted date/time and moves it to the deleted data area, where it will be kept until the user specifies (by default, 365 days). The client program will enumerate and destroy old deleted data once a week. The user is able to restore deleted data at any time using the visual restore wizard.

An end user can use the file manager to destroy data. When data is destroyed it is moved to a parallel repository designed to hold the “destroyed” data. Destroyed data is held for an additional 30 days, in case the destruction of data was unintentional. This is an additional safeguard against data loss.

All actions in the repository are transactional so that the system is always in a consistent state. At no point is the data on the disk inconsistent, even if the power fails (at either end), the network connection breaks, the software crashes, etc. At the end of a file update the software will ensure that all changes are permanently committed to disk, so that if the server indicates the file has been successfully updated, then no future event (even power failure, OS crash, etc.) will cause that file update be lost.

We have built a unique system to efficiently track the amount of disk usage in use by a folder and all of its subfolders. At any time the client can discover how much data a folder and all of its folders are using by using the disk usage explorer tool. In typical systems, this information has to be recreated on the fly when requested by the software, and when there are millions of files this can take a very long time. Our solution has been engineered to provide instant answers, facilitating management and the billing process.

The server backend also supports the ability to replicate an account’s data across multiple data centers, allowing the needs of all types of customers to be satisfied.

Account Management

All account information is stored in a central database for easy reporting, instant provisioning, and account management. This database also contains billing information and a detailed audit trail.

The web portal is built using proven open source technologies, which are operated under secure conditions protected by multilayer security.

First-class Infrastructure

Our infrastructure is located within secure data centers that meet or exceed the following specifications:

- 99.999% reliability for core systems
- Parallel, redundant UPS Systems, generators, cooling systems
- Feeds from multiple power grids
- Fire suppression and early smoke detection
- 5 Gig-E Internet backbone links with redundant BGP routing
- 24/7 security with video surveillance and recording, biometric scanning, card key protected access, and physically locked cabinets.