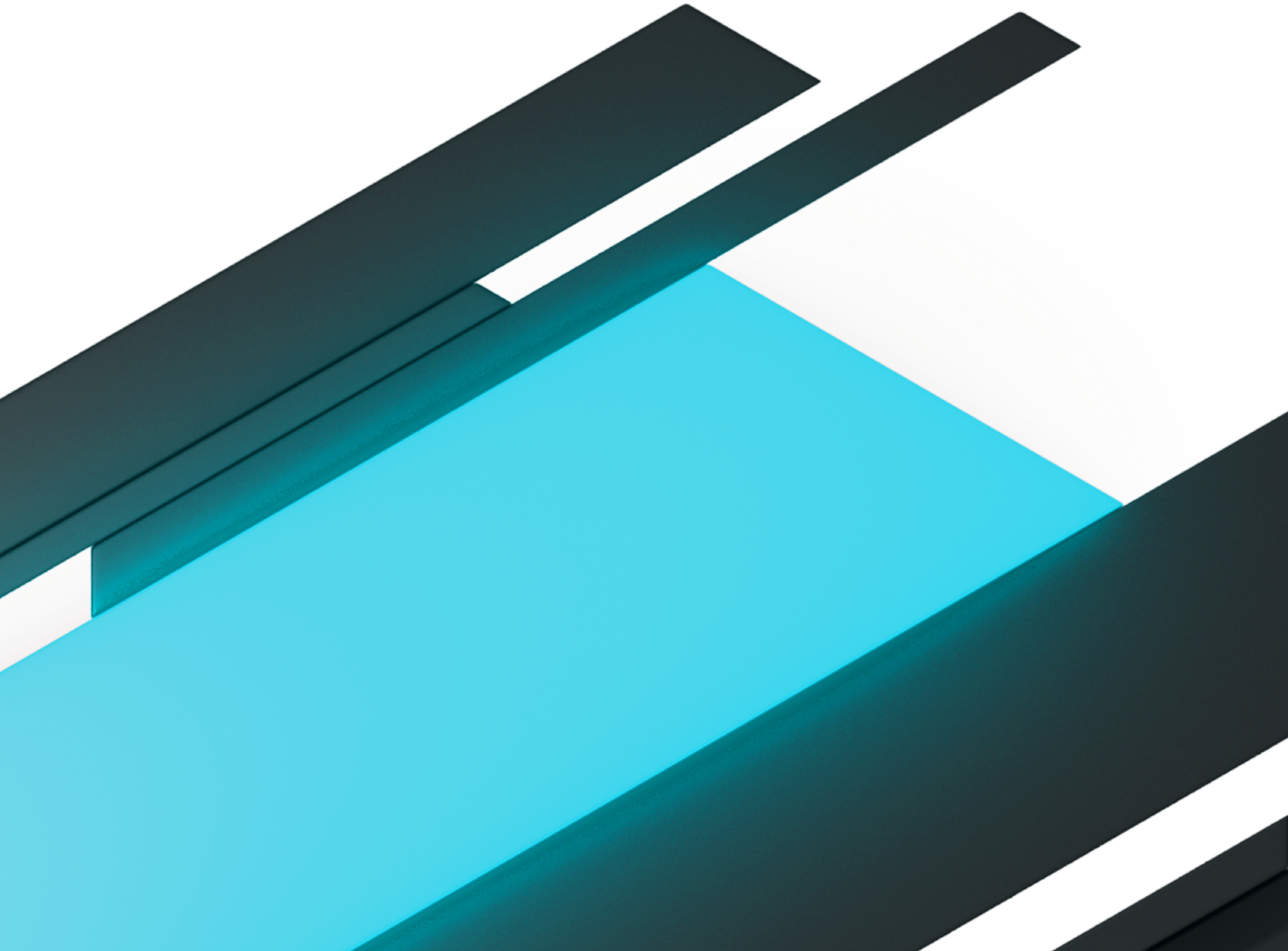
tresorit

WHITE PAPER

www.tresorit.com



tresor [tʀeˈzoːɐ̯]
noun (German)
1. lockable, armoured cabinet

THE CLOUD IS UNTRUSTED

The cloud has huge potential when it comes to storing, sharing and exchanging files, but the security provided by cloud services is questionable. Users, after uploading their files, have no control anymore about the way their data is handled and the location where it is stored. Even worse, users have no means to control access to their data. Considering both corporate and personal data which is often secret and sensitive in nature, one should not blindly entrust it to a cloud storage provider.

The main challenge of cloud storage nowadays is to guarantee integrity, confidentiality and control over all stored data. Users rarely have the possibility to check whether a given cloud provider satisfies these criteria. Furthermore, the quick spread of cloud storage solutions and their user-friendliness can possibly undermine the awareness of users regarding the transmission and storage of their confidential data. Therefore, users should apply security-oriented cloud storage middleware that, on the one hand, keeps the usage of the cloud simple and fast, and, on the other hand, enforces the application of strong cryptographic algorithms and protocols in order to keep private and confidential data from being leaked.

OTHER SOLUTIONS ARE UNSATISFACTORY

Several cloud storage middleware are available with just a couple of clicks on the Internet. Most of them even provide free-of-charge services for home users.

Even though most solutions promise security for your data in the cloud, almost none of them can prevent the cloud storage provider itself peering into your data. This is because these solutions

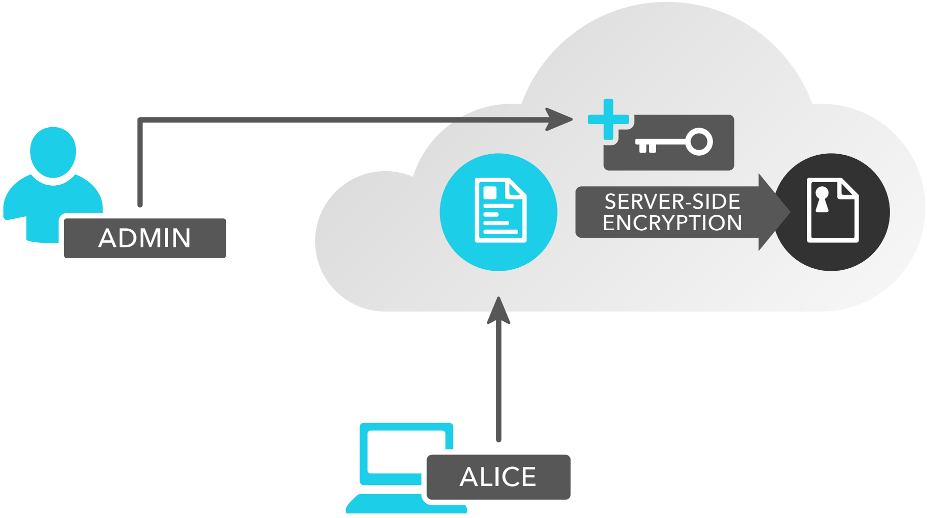


Figure 1 When using traditional cloud storage middleware, documents are visible in plaintext.

apply server-side encryption and the user has no control over how the encryption is done and who has access to the decryption keys. In other words, the user has to trust the cloud storage provider for being honest and not revealing her private data. This trust, however, simply cannot be underpinned [1,2,3].

Without the blind trust in the cloud storage provider, the user could choose to encrypt her files one-by-one and only upload these secure files to the cloud. This works well for things like backup, but only as long as the user does not want to share her files. As the amount of data and the number of people involved in the sharing rises, this approach becomes intractable. The overhead with group management, invitation of new group members, and revocation of expired permissions quickly becomes a burden. This approach is just not flexible enough for collaborative use.

TRESORIT: COMPLETELY SECURE CLOUD COLLABORATION

Tresorit provides a novel approach to secure cloud storage. The software allows you to share files and collaborate with your friends and colleagues with guaranteed cryptographic end-to-end security, without sacrificing the ease of use and performance of unsecure cloud storage services. With Tresorit, you encrypt files on your computer and the only people able to see the content are the ones you expressly give permission to. Contrary to other solutions, no storage provider or network administrator, no unauthorized hacker, not even Tresorit can read your files.

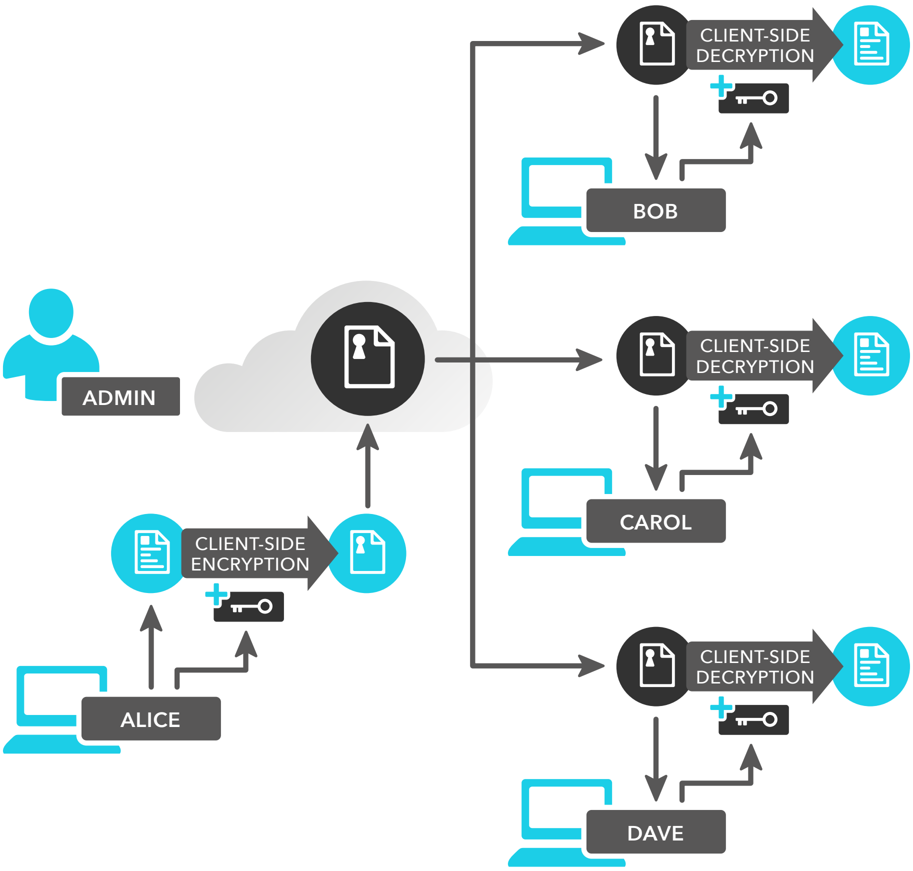


Figure 2 *With Tresorit, encryption and decryption are done at the client side.*

End-to-end Encrypted: Encryption and decryption are done on the client side. No entity is able to recover the data, except for the owner herself and users authorized by the owner. No trust in the cloud storage provider is required. Your data stays as safe as if it was stored securely on your own system.

Shareable: The owner can invite anyone with ease to collaborate and share files with. Only an e-mail address is required to send the invitation. Shared files and folders can be jointly modified, synchronization is performed automatically. Any number of files and directories can be shared among any number of users.

Shared files are encrypted: Files shared between users are still encrypted in the cloud at any time. Security and collaboration go hand in hand.

Everything can be 'tresored': Any directory can be directly turned into an encrypted 'tresor' that is securely stored in the cloud. Similarly, a tresor can be mounted to any convenient location on the hard disk.

Powerful permission handling: Permissions to the data are granted in a hierarchical structure. The owner is on the top of the hierarchy, below her are the managers who are able to add further users. The owner and the managers can grant or revoke editor (i.e., read and write) and reader permissions with just one click. The diversity of the hierarchy can be almost endless, but the owner always stays in full control of it.

Highest level security: Tresorit applies only published, scrutinized, industry-standard security algorithms, every one of which is powerful enough that it exceeds the computing power of currently available hardware. Decrypting them without authorisation is thus mathematically unfeasible – it would take much more than a human lifetime to crack even one 'tresor'. Files uploaded to the cloud are encrypted on the client side with AES-256. Transactions (i.e., file uploads and modifications) are authenticated with RSA-2048 signatures applied on SHA-512 hashes. Encrypted files are uploaded to the cloud using TLS-protected channels. Invitation and key agreement is done with ICE and ITGDH [4]. In Tresorit, nothing is left to chance.

Efficient: Tresorit is efficient in many different ways. Its core is written in C++, allowing for fast and smooth client-side software. Uploading, downloading, and in particular, synchronization with the cloud is multi-threaded, subduing communication time and increasing the achievable bandwidth.

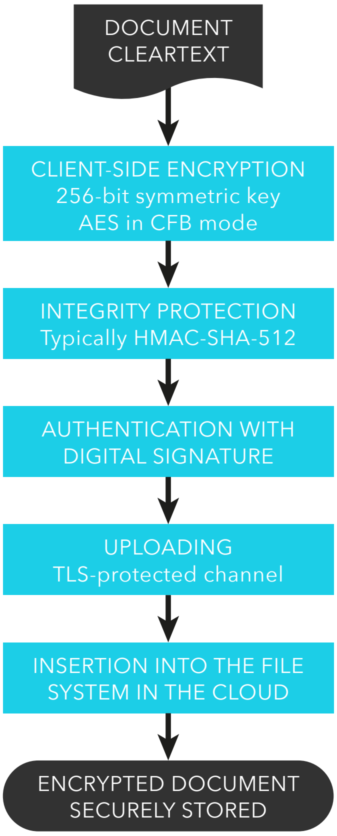
Cloud independent: Tresorit is independent of cloud providers. We plan to base the software on more than one cloud platform, making it possible to store a tresor on the cloud of the user's choice, or even to distribute it among several clouds.

360° security: Local drive encryption and powerful device management are being developed for the software, which will make it possible to protect data in Tresorit even when someone gains physical access to the device holding the documents. This can be ensured by encrypting files on users' local drives and enabling remote wiping of documents from devices.

TECHNOLOGY INSIGHT

Uploading and storage: If the user wants to upload a file to a 'tresor', which is an encrypted and integrity protected storage, it is encrypted and uploaded to the cloud by the client-side Tresorit application.The encryption is performed with a fresh 256-bit symmetric key chosen by the client-side application. The encryption algorithm Tresorit applies is AES in CFB mode. The integrity of the files is protected with HMAC, typically HMAC-SHA-512.

Following this, a TLS tunnel is established between the client machine and the cloud. TLS is used to further protect the data against eavesdropping and tampering while being uploaded. Each upload is authenticated with the digital signature of the uploader.


DOCUMENT CLEARTEXT
CLIENT-SIDE ENCRYPTION 256-bit symmetric key AES in CFB mode
INTEGRITY PROTECTION Typically HMAC-SHA-512
AUTHENTICATION WITH DIGITAL SIGNATURE
UPLOADING TLS-protected channel
INSERTION INTO THE FILE SYSTEM IN THE CLOUD
ENCRYPTED DOCUMENT SECURELY STORED

Finally, once the encrypted file is uploaded, it has to be added to the existing directory structure in the cloud. This remote directory structure is the exact copy of the client-side directory structure and consists of directories and files. All the files and directories are encrypted and a directory containing files and directories also holds the related symmetric keys. This results in a layered hierarchy of directories, files and keys of arbitrary depth and complexity. The top of the hierarchy is the root directory. A file can be inserted into the directory structure in the cloud only if one knows the key the root directory is encrypted with.

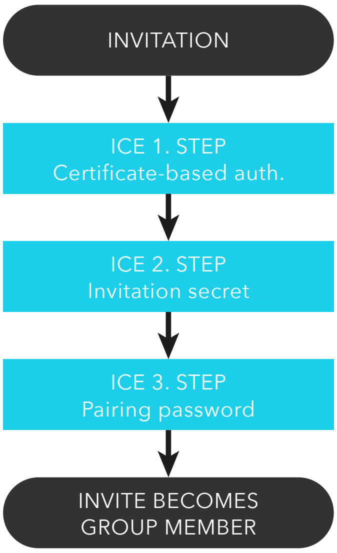
The key that decrypts the root directory and thus gives access to all the files and directories in a tresor's directory structure is provided by Tresorit's Agreement Module. The Agreement Module can be RSA-based or TGDH-based.

An RSA-based Agreement Module contains a set of pre-master secrets, one for each user who is sharing the files, encrypted with the user's public RSA key. Furthermore, the Agreement Module also stores the RSA public-key certificates of the users. When a user wants to get access to the shared files, he or she has to provide a private key to the Agreement Module. The latter decrypts the pre-master with the provided private key and calculates the symmetric key of the root directory. The calculation is done by applying HMAC to the user certificates as inputs and with the pre-master secret as the key. The TGDH-based Agreement Module works similarly to the RSA-based Agreement Module. The only difference is that the former does not store encrypted pre-master secrets, but the Diffie-Hellman certificates of the user. The key decrypting the root directory in case of a TGDH-based Agreement Module is calculated using ITGDH.

With the symmetric key provided by the Agreement Module, the user can decrypt the root directory and access the files and folders, along with the corresponding symmetric keys, of the directory structure. By navigating in the same way through the directories as one would do on her local machine, authorized users can add, delete and modify files and directories with ease.

Invitation: In case the user wants to collaborate with others using files in the cloud, he or she has to share the uploaded files. This is done via invitation with the help of the ICE protocol.

Users can be invited with ICE through e-mail using the following 3-step authentication process:



STEP 1 CERTIFICATE-BASED AUTHENTICATION

Users have their own X.509 certificate which contains their personal data, usually a name and an e-mail address. Users authenticate themselves with the strong, RSA-2048 private key in the certificate.

STEP 2 INVITATION SECRET

When invitation is sent to the user, a 256-bit random secret is sent along with it. During the handshake, the invitee proves to the inviter that she knows the secret value by a challenge-response protocol. This is like sending a card to the invitee, and asking for showing the invitation card at the door.

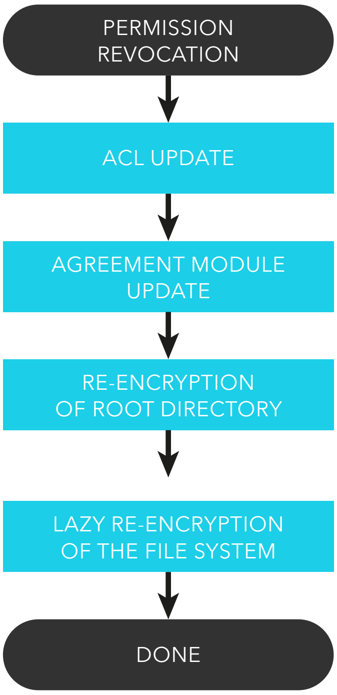
STEP 3 PAIRING PASSWORD (OPTIONAL)

A pairing password has to be agreed on between the inviter and the invitee in advance using an out-of-band channel (e.g., mobile phone, instant messaging, etc.). This option provides even more additional security.

With the execution of ICE, inviter and invitee learn each other's public-key certificates and establish a long-term high-entropy symmetric key. The public-key certificate of the invitee will be added to the list of partners. The long-term symmetric key can be applied later for other invitations.

The ICE protocol is designed to work in a semi-trusted environment. It does not rely solely on the trustworthiness of the certificate authority issuing the users' certificates, neither does it rely on the secrecy of e-mail communication, an out-of-band channel, or on the strength of the pairing password. Tresorit asserts that an adversary cannot tamper with all the above pillars at the same time.
After the successful execution of ICE, the invitee is added to the group sharing the given tresor. The invitee is now able to access the directory structure by querying the Agreement Module with her private key.

Permission revocation: The permission of a user is her ability to view and modify the content of the tresor. When it becomes necessary to revoke such a permission, like when an employee leaves the company, it is required to remove his or her permissions in order to avoid later, possibly malicious influence on the tresor. With Tresorit, permission revocation is efficient and secure.


PERMISSION REVOCATION
ACL UPDATE
AGREEMENT MODULE UPDATE
RE-ENCRYPTION OF ROOT DIRECTORY
LAZY RE-ENCRYPTION OF THE FILE SYSTEM
DONE

STEP 1 PERMISSION REMOVAL

First of all, the removed user's permissions are deleted from the cloud's ACL. This step alone would be enough to keep the removed user from accessing the files in the tresor, but would require trust in the cloud.

STEP 2 CERTIFICATE DELETION AND ROOT RE-ENCRYPTION

We do not want to require such trust, so Tresorit deletes the removed user's certificate from the Agreement Module and re-encrypts the root directory with a new symmetric key generated by the user who is performing the removal. The removed user will not have access to the root directory anymore because of his or her inability to retrieve the new encryption key.

STEP 3 CONTENT RE-ENCRYPTION

Finally, the content in the shared tresor has to be re-encrypted, otherwise the removed user might be able to access the files and directories directly by bypassing the hierarchical tree structure. Re-encrypting the whole tresor with every group change, however, would be extremely costly in terms of computation. Therefore, Tresorit applies the principle of lazy re-encryption.

With lazy re-encryption, documents only get re-encrypted when they change. This means that the re-encryption of the root directory sets a so called 'dirtiness' flag at each and every file and directory that has not been re-encrypted yet since the last group change. Once a user with permission modifies a dirty document, he or she has to re-encrypt it with a new key. Trivially, the old key for the changed document has to be replaced with the new key in the directory that stores the document and the latter has to be re-encrypted with a new key, and so on, while reaching a directory at a hierarchical level that is not dirty. Once a file or directory is re-encrypted with a new key, its dirtiness flag is deleted and the next user accessing can already use the recently established key.

As the re-encryption of tresor content is achieved with lazy re-encryption, permissions can be efficiently revoked with only one re-encryption (applied to the root directory), one deletion (applied to the entries specific to the removed user in the Agreement Module), and one change in the ACL, making an otherwise extremely demanding operation efficient and fast.

CASE STUDY

Imagine a consulting company, TreeSpin Consulting, would like to outsource the overwhelming effort of storing and maintaining its file system. Uploading everything to the cloud would be the natural way to go, but as security is of paramount importance in this case, the cloud storage provider has to be chosen with great care. Unencrypted documents must not leave the company under any circumstance.

TreeSpin Consulting has more than 5000 clients from all around the world and 300 employees working in 12 departments. Clients often need advisory services that require multiple departments within the firm to work closely together. Sharing data and documents between collaborating departments can be cumbersome, especially when several employees are involved from a number of departments. The same burden appears on the client company's side that has a department dedicated to working with the input received from TreeSpin Consulting and also provides feedback to the consultants. Furthermore, employees involved in the actual consulting project might leave TreeSpin Consulting and the client company during the project's timeframe, while new employees might be hired at either side and assigned to the on-going project.

Such a dynamically changing user group requires a flexible storage solution that is accessible by both parties. Collaborating employees, however, should have different access rights to shared documents according to their role in the project and level of hierarchy in the corresponding company. If TreeSpin Consulting has no capacity or desire to set up its own accessible and dependable storage environment (which would involve reconfiguring its own dedicated IT department as well), it would decide to store everything in the cloud. But how can TreeSpin Consulting make sure that their data is well-protected but accessible for collaboration?

Trivially, TreeSpin Consulting can by no means rely on any cloud provider to protect its data. That data is just too sensitive and valuable to hand it over to any third party without first securing it, and legal issues could easily arise from using unprotected cloud environments. This requires the company to take care of data protection on its own. The files have to be accessible even if encrypted by the personnel according to the company's department structure. This puts emphasis on the need of a flexible permission handling system. Finally, but most importantly, sharing of the data must not compromise its security.

Tresorit would be the perfect choice for TreeSpin Consulting.

Tresorit provides end-to-end encryption, only encrypted data gets transferred through its TLS-protected channels. Users, i.e., employees, can be arbitrarily organized into a hierarchy following the departmental structure of the company, while files can be shared among employees and groups of employees independently of the file size and the directory structure. Tresorit handles permissions and permission changes efficiently, i.e., when a new employee has to get access to shared files or when an employee leaves the project and her permission have to be revoked.

With Tresorit, the TreeSpin Consulting would be able to make use of the cloud's advantages while remain in full control of its valuable outsourced data and without compromising the highest level of security. All with just a few clicks.

REFERENCES

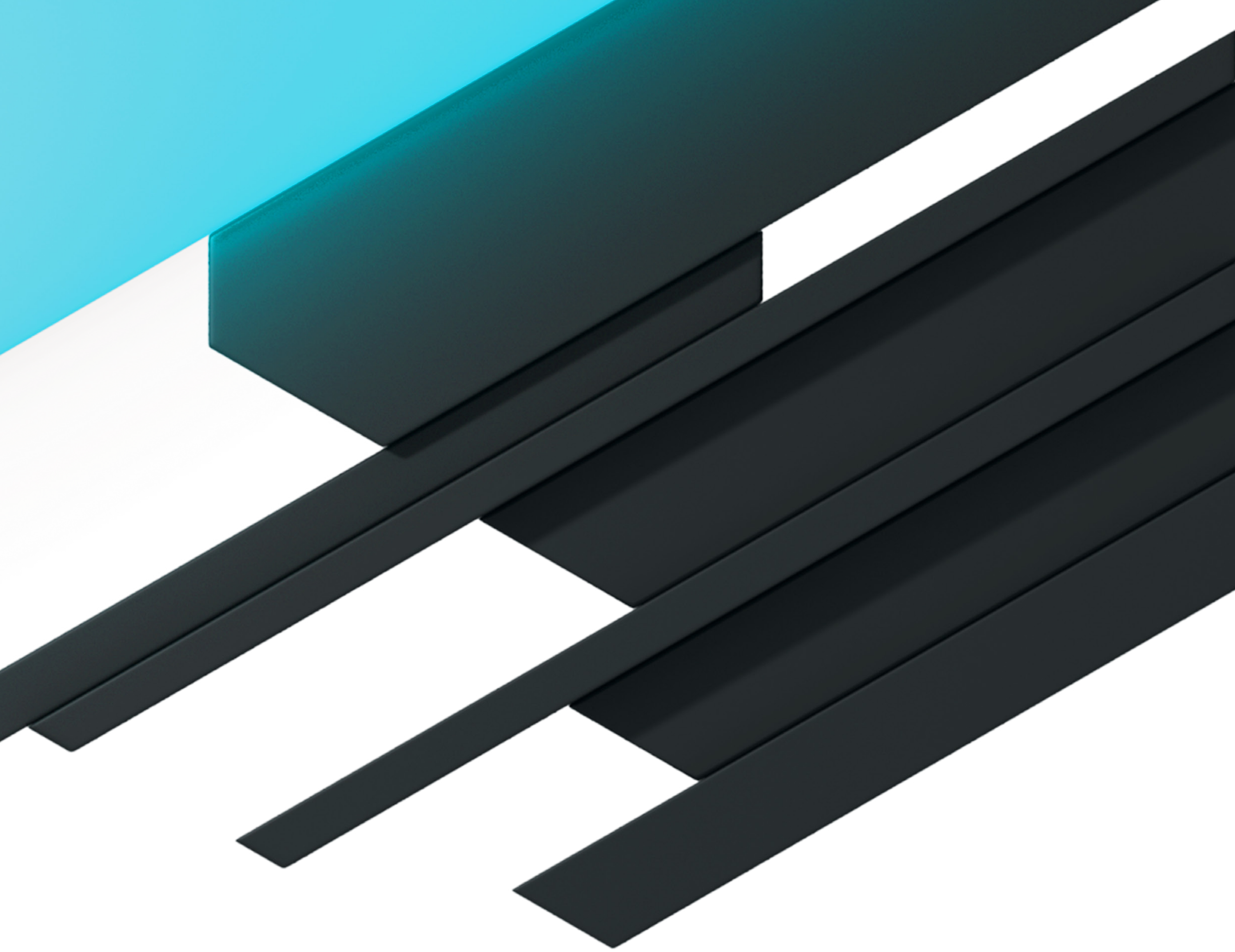
[1] Cloud Security Alliance, "Top Threat to Cloud Computing V1.0," March 2010. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf

[2] "Microsoft accuses former employee of cloud data theft", February 2011. http://www.zdnet.com/microsoft-accuses-former-employee-of-cloud-data-theft-4010021691/

[3] J. Mutch, "How to Steal Data from the Cloud - An Easy Guide for IT Admins", Vol. 1, Issue 7, 2010, Cloudbook Journal, http://www.cloudbook.net/resources/stories/how-to-steal-data-from-the-cloud

[4] I. Lám, Sz. Szebeni, L. Buttyán, "Invitation-oriented TGDH: Key Management for Dynamic Groups in an Asynchronous Communication Model", International Workshop on Security in Cloud Computing (CloudSec), Pittsburgh, PA, September 2012

[5] I. Lám, Sz. Szebeni, L. Buttyán, "Tresorium: Cryptographic File System for Dynamic Groups over Untrusted Cloud Storage", International Workshop on Security in Cloud Computing (CloudSec), Pittsburgh, PA, September 2012



tresorit

www.tresorit.com