




End-to-End Encrypted Cloud Storage

Matilda Backendal 
ETH Zurich
Zurich, Switzerland
mbackendal@inf.ethz.ch

Miro Haller 
UC San Diego
La Jolla, USA
mhaller@ucsd.edu

Kenneth G. Paterson 
ETH Zurich
Zurich, Switzerland
kenny.paterson@inf.ethz.ch

Copyright notice. This article is accepted to IEEE Security and Privacy. DOI: 10.1109/MSEC.2024.3352788. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Abstract—End-to-end encryption is rapidly becoming the accepted security goal for personal data. In this article, we examine consumer cloud storage systems, focusing in particular on those systems that attempt to provide end-to-end security for customer data. We survey the security guarantees of current service providers and the issues they face, discuss open research questions, and highlight the challenges that impede the deployment of end-to-end secure cloud storage.

“The cloud”, that is, a cluster of servers accessible from the Internet, has become an indispensable part of modern IT infrastructure. Companies and private individuals alike place their data in the cloud in order to take advantage of features like file sharing, disaster-safe backups, online collaboration, and outsourced computation. The volume of data stored in various cloud services today is already staggering, and it is predicted to increase; in a prognosis by Cybersecurity Ventures [15], 100 zettabytes¹ of data, or roughly half of the world’s total estimated data at the time, is forecast to be stored in the cloud by 2025.

Much research has been done on the implications for security and privacy from this surge of outsourced data, including how cutting edge cryptographic techniques such as private information retrieval (PIR), searchable encryption and fully-homomorphic encryption (FHE) can be used to protect the confidentiality of data while processed or queried. These exciting new techniques push the boundaries of what is possible to achieve with cryptography. For a comprehensive introduction, see e.g. [16].

However, we do not yet have a solid foundation on which to build these novel features; our recent attacks on the largest provider of end-to-end encrypted cloud storage call into question whether the cryptographic community currently has a sufficiently rigorous understanding of cloud storage to ensure even the fundamental properties of confidentiality and integrity. It seems that even these canonical cryptographic goals are challenging in the setting of cloud storage.

In this article, we focus on consumer cloud storage, and in particular on the motivations to secure data in the cloud

with end-to-end encryption (E2EE). We survey the security guarantees of current service providers and the issues they face, discuss open research questions, and highlight the challenges that impede the deployment of end-to-end secure cloud storage.

I. WHY CLOUD STORAGE?

Users voluntarily—but sometimes also unknowingly—store much of their data in the cloud. This happens by choice, for example when using Google Drive to collaborate on a document, or when sharing photos and files over Dropbox, but also more or less transparently, such as when a user’s iPhone automatically creates a backup of photos in iCloud, or when their Windows laptop saves documents in OneDrive by default.

The benefits to the user are usually clear: having your data in the cloud means that they are accessible from multiple devices and locations, protected against device loss or failure, as well as easily shareable with other users. Additionally, the maintenance and security of the data is entrusted to the provider of the cloud storage, alleviating the user from having to host and operate their own storage servers, as well as allowing the storage volume to be adaptively—and cost-efficiently—scaled. The potential drawbacks can be less obvious. In the following, we will explain how the privacy and authenticity of the outsourced data can be at risk, and why this risk is not easily mitigated.

II. CLOUD STORAGE SECURITY

Most service providers, including Google Drive [9], iCloud [17], Dropbox [7] and OneDrive for Business [14], provide so-called encryption at rest of user data. This means that, by default, all files which are uploaded to the cloud are encrypted by the provider before they are stored. This helps to protect the data from external attackers who may compromise the servers or data centers of the service provider. Since the files are stored in encrypted form (using strong, modern cryptography such as AES-256 in a suitable mode), an attacker who only gains access to the ciphertexts, but not to the key used for encryption, cannot decrypt and read the plaintext

¹A zettabyte is 10^{21} bytes, or the equivalent of a trillion gigabytes.

files. Hence, the privacy of user data is ensured against this class of attackers.

While encryption at rest is much better than no encryption, it still leaves the data vulnerable to attacks from stronger adversaries. In particular, since the service provider performs the encryption (and therefore holds the encryption keys), the data are trivially accessible to the provider themselves. This means that, for example, Google can read all documents stored in Google Drive, and Dropbox can view all images stored on their platform.

This may be considered a warranted trade-off. After all, having access to plaintext data allows the cloud provider to offer several services to their users, beyond solely storing their data. For example, a common feature of consumer cloud storage today is live collaboration, which allows owners of a shared file to collaborate and edit it in real time, with all updates being immediately and seamlessly incorporated into the file and displayed to all editors. Another consequence of the cloud having access to plaintext data is that it can perform computation on them. This allows the provider to support functionality like design suggestions for slides or keyword search over outsourced documents. Letting the service provider manage the encryption keys also means that the provider can aid users in recovering their data if they lose access to their accounts, for example by forgetting their password.

In addition to these user benefits, the service providers themselves have several incentives to retain access to plaintext user data. One of the foremost motivations is data deduplication, i.e., the practice to only store a single copy of identical files uploaded by different users. Microsoft estimates that deduplication saves 30-80% of storage space in various settings [13]. Furthermore, the plaintext data may be a valuable source of information for targeted advertisement or to train machine learning models. Finally, providers may experience external pressure to retain plaintext data, e.g., to aid law enforcement. For instance, in the “San Bernardino case”, Apple (successfully) challenged a court order that required them to write software for the FBI that would compromise iPhone encryption [8].

In conclusion, the big cloud storage providers today apply encryption at rest, thereby achieving confidentiality against snapshot adversaries who gain temporary access to the servers that store user files. However, the standard approach is to perform this encryption server-side. That is, the keys used to encrypt files are generated by, and stored at, the cloud providers, and the encryption takes place on the server. Hence, the providers retain access to user data themselves, for functionality, convenience and for profit.

III. THE CASE FOR E2EE CLOUD STORAGE

As we have seen, encryption at rest is great for securing the confidentiality of data against external adversaries when the service provider is trusted. But what about when this is not the case? There are several reasons that users might want to minimize their trust in the service provider: they might simply want to be conservative and not provide anyone with access to

their data, or they might specifically prefer that their data is not used for things like profiling or algorithm training. Moreover, users at risk may wish to have additional protection for highly sensitive documents against powerful adversaries that collude with or compromise the service provider.

For instance, investigative journalists in an authoritarian regime may fear that cloud providers are compelled to disclose their data to the government, putting the journalists and their sources at risk [18]. And even if the service provider is trusted to protect user data to the best of their ability, and not to collude with the adversary, there is always a chance for security failures or leaks from implementation errors, as well as the risk that the provider is compromised by an external adversary. This is not a theoretical threat; experience shows that both national security agencies and hacker groups target large cloud providers in order to gain access to the data that passes through them [6].

To achieve security in the face of these types of attacks, stronger security measures than encryption at rest on the server are called for. The natural alternative is to switch from server-side to client-side encryption. That is, instead of uploading plaintext files to the cloud, which are then encrypted on the server, users generate their own encryption keys and encrypt the data before they leave their devices. This way, the cloud only sees ciphertexts. Hence, proper client-side encryption provides what is known as end-to-end security, by cryptographically limiting access to the file owner (who has the encryption keys) and people they share the keys with.

End-to-end encryption (E2EE) is quickly becoming the standard security guarantee for data in transit. For example, most Internet traffic today is encrypted E2E between client and server endpoints using protocols like TLS [10]. Furthermore, in secure messaging, popular applications—including WhatsApp, Signal, and iMessage—use E2EE to provide exclusive access to the exchanged messages to the sender and receiver, even in the presence of a malicious or compromised server [12]. It seems reasonable to assume that users should expect similar guarantees for data storage, since personal files can be just as sensitive as private messages and Internet traffic. Hence, the natural next step is to bring E2EE also to cloud storage. However, as we will see, techniques from TLS and secure messaging do not readily translate to cloud storage due to features such as sharing and persistent data access that are unique to this setting.

Despite these challenges, there is significant interest in secure cloud storage from both vendors and users; Apple recently rolled out optional E2EE for some iCloud data [4]², and since about a decade back, there are also dedicated cloud storage services that specifically aim to provide strong privacy guarantees for their users. In the next section, we review two such systems: Mega and Nextcloud.

²The E2E guarantees provided by Apple are limited in order to still allow the features discussed above. For example, they do not apply to shared documents for the sake of supporting live collaboration, and checksums of plaintext data are still shared with Apple servers such that they can perform deduplication.

IV. CASE STUDIES: MEGA AND NEXTCLOUD

A. *Mega*

The seemingly simple change from server-side to client-side encryption turns out to be perilous in practice. Our recent cryptanalysis of Mega, the largest E2EE cloud storage provider with over 300 million users [1], revealed several catastrophic issues with their system. In combination, these practical attacks would allow a malicious cloud provider to break the confidentiality and integrity of user files [5].

The attacks stem from insufficient protection of the outsourced key material, which forms the root of security for the end-to-end encrypted files. Due to the use of an unauthenticated AES mode to encrypt keys, a malicious server could tamper with key ciphertexts, and then decrypt the key material by observing the client's responses during the authentication procedure [5]. Subsequent work improved this attack and showed that it was possible to recover key material after only six login attempts by the user [11].

The lack of key separation in Mega's system then enabled the adversary to decrypt one file per login. Moreover, because of issues with the authentication method for file encryption, the malicious server was also able to insert arbitrary new files into a user's cloud storage, violating integrity. To make matters worse, while mitigating these attacks Mega introduced an error oracle that also compromised confidentiality in new ways [3].

B. *Nextcloud*

Another approach to E2EE is offered by Nextcloud; a company which provides open-source software that lets individuals and businesses create and host their own cloud storage platforms. Nextcloud's system is used by more than 20 million customers, including several European governments, universities, and organizations such as Amnesty International.

In Nextcloud, E2EE is applied at a folder level, and public key cryptography is used to (indirectly) encrypt the metadata and file keys of the files in the folder. Unfortunately, the encryption mode used (RSA-OAEP) lacks authenticity. This means that a malicious server can generate valid key ciphertexts by simply picking keys and encrypting them under the public key of a user. Hence, an adversary controlling the server can replace encrypted metadata keys with encryptions of keys that it knows. When the victim user fetches the folder data from the server the next time, it will fetch and decrypt the malicious keys and subsequently use them to (re-)encrypt the files in the folder, which are then accessible to the server. That is, the lack of authenticity of public key encryption leads to a complete security break. Moreover, because of implementation issues such as IV reuse in the file encryption protocol, a malicious server could additionally compromise the confidentiality and integrity of files in E2EE folders in several other ways [2].

C. *Takeaways*

These attacks on Mega and Nextcloud show that designing and implementing secure E2EE cloud storage is prone to errors. The interactions between different parts of the system

are complex, and both confidentiality and integrity must be ensured against a very strong adversary. Furthermore, it is difficult to mitigate vulnerabilities in a running system due to the scale and the need to maintain backwards compatibility. (For example, even under the most optimistic assumptions Mega would have needed more than half a year to re-encrypt all user data with a more secure encryption mode [5].)

The challenges faced by Mega and Nextcloud are not unique to these providers. Secure consumer cloud storage has received relatively little attention from the cryptographic community; in particular, there has been no coordinated effort to develop a protocol with provable security guarantees, meaning that providers are left to implement their own ad-hoc systems. On top of that, achieving end-to-end security for cloud storage is not trivial. Beyond the complexity induced by the strong threat model, which requires careful system design and the use of strong, modern cryptographic primitives, there are several (inherent) obstacles on the path to E2EE cloud storage.

V. CHALLENGES OF E2EE CLOUD STORAGE

We have already touched upon some of the challenges of designing and implementing E2E encrypted cloud storage. For instance, with client-side encryption, users could permanently lose access to their account if they forget their password, since the server cannot assist in recovering their encryption keys. This problem is related to an inherent cryptographic challenge with end-to-end encryption: key management.

The challenge of key management stems in part from the fact that humans are bad at managing cryptographic keys, necessitating the need for human-memorable secrets like passwords, and in part from functionality requirements such as multi-device access; even though one user device picks the keys for end-to-end encryption, cloud storage applications require all devices of that user to be able to decrypt files, meaning that the keys need to be available wherever the user chooses to access their data. Often, the human user is the only trusted channel that can be assumed between their devices. Hence, they either need to manually port the keys from one device to the next, or—more realistically—the keys need to be exchanged (in encrypted form) over the untrusted cloud provider, who may actively tamper with exchanged messages.

Related applications such as secure messaging rely on ephemeral key material to simplify the key management problem and achieve strong guarantees like forward security and post-compromise security. However, in cloud storage, users expect that their files remain accessible indefinitely. This persistency requirement makes it difficult to use techniques from messaging that rely on short-lived keys. Furthermore, even expensive operations like key rotation and re-encryption are insufficient to achieve properties like forward security against a malicious server, since access to the files needs to be preserved with the new keys.

Another challenge particular to cloud storage is that of file sharing. This feature introduces interaction between users that is not present in other cryptographic applications with encryption at rest. This further increases the key management

problem, since shared files need to be accessible not only on all devices of one user, but also to all users with whom they are shared. Unlike for devices of a single user, the human can no longer be used as a trusted channel for transferring secrets between end devices.

For this reason, and more, sharing turns out to be one of the most difficult challenges of E2E-encrypted cloud storage. In fact, the attacks on both Mega and on Nextcloud were possible due to issues with the sharing protocols. In Mega, the sharer picks an arbitrary file key which the client of the recipient automatically re-encrypts with their own key material. This unintentionally exposed an encryption oracle to the malicious server, which enabled some of the devastating attacks on confidentiality in that system. In Nextcloud, the server could compel clients to re-encrypt file keys with a server-controlled key thanks to a feature that was implemented for the sake of the sharing protocol.

Finally, previously simple operations such as keyword search and deduplication become challenging cryptographic problems once data is encrypted. The E2EE versions of these two particular features are called encrypted search and convergent encryption, respectively. Both of these involve a difficult-to-navigate tradeoff between inherent leakage and security, as they need to relax standard security notions to provide more functionality.

In conclusion, there are multiple challenges on the path to secure and private cloud storage. Due to the lack of a rigorous understanding of the aforementioned challenges, deployed E2EE cloud storage protocols have made insecure design choices, leading to practical attacks. Clearly, something is missing to allow cloud storage providers to deploy secure E2EE.

VI. THOUGHTS TOWARDS ACHIEVING SECURE E2EE CLOUD STORAGE

We believe that achieving secure and efficient E2EE for cloud storage will take a joint effort from cryptographers, vendors, and implementers. First of all, the security goals and corresponding challenges need to be thoroughly understood. This will involve the formalization of security guarantees that an end-to-end-encrypted cloud storage system should provide, as well as research into how cryptography can be used to achieve these security notions, while still providing the functionality that users and vendors expect. This, in turn, requires input from providers and implementers to ensure that the solutions are efficient and fulfill practical requirements.

Ideally, we would like to see a standardization effort to design a well-analyzed and practical E2EE protocol, which can finally bring trustworthy privacy and integrity guarantees to consumer cloud storage.

ACKNOWLEDGMENT

The authors would like to thank the editors of the Security and Privacy Magazine for their feedback in the editing process of this article.

REFERENCES

- [1] About Us – Encrypted Cloud Storage – MEGA. <https://mega.io/about>. Visited on March 18, 2024.
- [2] Martin Albrecht, Matilda Backendal, Daniele Coppola, and Kenneth G Paterson. Share with care: Breaking E2EE in Nextcloud. In *Euro S&P 2024*. 2024.
- [3] Martin R. Albrecht, Miro Haller, Lenka Mareková, and Kenneth G. Paterson. Caveat implementor! Key recovery attacks on MEGA. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 190–218. Springer, Heidelberg, April 2023.
- [4] Apple. Advanced data protection for iCloud. <https://support.apple.com/guide/security/advanced-data-protection-for-icloud-sec973254c5f/web>, 2022. Visited on October 24, 2023.
- [5] Matilda Backendal, Miro Haller, and Kenneth G. Paterson. MEGA: Malleable encryption goes awry. In *2023 IEEE Symposium on Security and Privacy*, pages 146–163. IEEE Computer Society Press, May 2023.
- [6] Ashkan Soltani Barton Gellman. NSA infiltrates links to Yahoo, Google data centers worldwide, Snowden documents say, October 2013.
- [7] Dropbox Help Center. How Dropbox keeps your files secure – Dropbox help. <https://help.dropbox.com/security/how-security-works>. Visited on October 10, 2023.
- [8] EPIC: Electronic Privacy Information Center. Apple v. FBI. <https://epic.org/documents/apple-v-fbi-2/>. Visited on October 21, 2023.
- [9] Google Cloud. Default encryption at rest | documentation. <https://cloud.google.com/docs/security/encryption/default-encryption>. Visited on October 10, 2023.
- [10] Cloudflare. Cloudflare Radar: Adoption and usage. <https://radar.cloudflare.com/adoption-and-usage>. Visited on October 24, 2023.
- [11] Nadia Heninger and Keegan Ryan. The hidden number problem with small unknown multipliers: Cryptanalyzing MEGA in six queries and other applications. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 147–176. Springer, Heidelberg, May 2023.
- [12] Alfred Menezes and Douglas Stebila. End-to-End security: When do we have it? *IEEE Security & Privacy*, 19(4):60–64, 2021.
- [13] Microsoft. Data deduplication overview. <https://learn.microsoft.com/en-us/windows-server/storage/data-deduplication/overview>. Visited on October 10, 2023.
- [14] Microsoft. Whitepaper: Security for OneDrive for business. <https://www.microsoft.com/en-us/download/details.aspx?id=53884&culture=en-us&country=US>. Visited on October 10, 2023.
- [15] Steve Morgan. The 2020 data attack surface report – whitepaper. <https://cybersecurityventures.com/wp-content/uploads/2020/12/ArcserveDataReport2020.pdf>, <https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/>, June 2020. Visited on October 10, 2023.
- [16] Nigel Smart. Computing on encrypted data. *IEEE Security & Privacy*, 21(4):94–98, 2023.
- [17] Apple Support. iCloud data security overview. <https://support.apple.com/en-us/HT202303>. Visited on October 10, 2023.
- [18] Mike Isaac Vinod Sreeharsha. Brazil arrests Facebook executive in WhatsApp data access case, March 2016.