An In-Depth Analysis of the Security of the MEGA.io Cloud Storage Service

Executive Summary

The online storage service MEGA has positioned itself in the market as a bastion of privacy, founded on the promise of "zero-knowledge" encryption fully controlled by the user.1 This approach, which claims to make user data technically inaccessible even to the service provider, is its main selling point. However, this report reveals a fundamental contradiction between these marketing claims and the findings of independent research conducted by cryptographers from the Swiss Federal Institute of Technology in Zurich (ETH Zurich) in 2022.

This research, dubbed "Mega-Awry," brought to light critical and fundamental cryptographic flaws in MEGA's architecture.4 These vulnerabilities are not mere bugs, but design flaws that allow a malicious or compromised MEGA server to progressively decrypt user files, and even inject malicious data into their storage space, thus irrefutably breaking the "zero-knowledge" promise.8

In response, MEGA deployed software patches to counter the most direct attack vector. Nevertheless, these measures are considered insufficient by the security community, as they do not address the underlying architectural weaknesses that make the system inherently fragile.7 The complete system redesign recommended by the researchers has not been undertaken.

The verdict of this analysis is therefore nuanced but unequivocal: MEGA's security is fragile and cannot be considered reliable for storing highly sensitive data, especially when the service provider itself is included in the threat model. Its value is considerably diminished for users requiring absolute confidentiality. However, it may remain acceptable for everyday use where the main adversary is not the service provider, but where basic protection against mass surveillance or third-party data breaches is desired.

Section 1: The Architectural Promise: Deconstructing MEGA's "Zero-Knowledge" Encryption Model

1.1 The User-Controlled Encryption (UCE) Framework

MEGA's fundamental value proposition rests on its "zero-knowledge" encryption architecture, also marketed as User-Controlled Encryption (UCE).2 The central claim is that only the user holds the decryption keys to their data. Consequently, no one else, not even MEGA staff, can access, decrypt, or read the files stored on its servers.1 This feature is presented as a key differentiator from major competitors such as Google Drive or Dropbox, where the provider often retains technical access to users' decryption keys, whether for operational reasons or legal compliance.2

To reinforce this promise, MEGA's documentation emphasizes that the user's password is the "master encryption key." The company insists that it has no access to this password and never stores it, which is why it is technically unable to offer a password reset procedure via email. If a user loses their password and their recovery key, access to their data is permanently lost.2 It is on this basis that MEGA asks users to place their trust.

1.2 The Key Hierarchy: From User Password to File Encryption

MEGA's encryption process is structured according to a key hierarchy designed to secure data at multiple levels. This model, although complex in appearance, can be broken down into several logical steps, based on security documentation and researcher analyses.8

1. **Master Key Generation:** It all starts with the user's password. This password is not used directly to encrypt files. Instead, it is used to derive an encryption key which, in turn, encrypts a randomly generated "Master Key" upon account creation.

2. **Encryption of Cryptographic Material:** This master key becomes the linchpin of the account's security. It is used to encrypt all of the user's essential cryptographic material, which includes a 2048-bit RSA key pair (used for secure data sharing with other users), a Curve25519 key pair (for key exchange in the chat function), and an Ed25519 key pair (for digitally signing the other keys).8
3. **Node Keys:** For each individual file or folder uploaded to MEGA, a new, unique symmetric key, called a "node key," is generated. This key is used to encrypt the content of that specific item.8 This approach ensures that the compromise of one file key does not compromise all of the user's data.
4. **Storage on the Server:** The encrypted cryptographic material (the RSA private key, node keys, etc.) is then uploaded and stored on MEGA's servers. This architectural decision is crucial: it allows for multi-device synchronization and account access from any terminal. However, as the vulnerability analysis revealed, this choice to store the user's private key (even in encrypted form) on the server became the system's main point of failure.8

This architecture is a deliberate compromise between security and convenience. To offer a smooth user experience with seamless synchronization and an account recovery option (via the recovery key), MEGA chose to store its users' encrypted private keys on its own servers. A more secure, but significantly less convenient, model would have required the user to manage their private keys entirely and exclusively on their own devices. This design choice, which prioritizes user experience over absolute security, created the very attack surface that was later exploited. The act of centralizing risk by entrusting the custody of encrypted cryptographic secrets to the server made the security of the entire system merely dependent on the honesty and infallibility of MEGA's infrastructure.

1.3 The Role of RSA and AES in the Security Chain

MEGA's architecture relies on two standard cryptographic primitives: AES and RSA. AES (Advanced Encryption Standard) is the symmetric encryption algorithm used for the fast and efficient encryption of file contents and node keys.8 RSA is the asymmetric cryptosystem used primarily for the secure sharing of keys between users.11 When a user shares a folder, the key for that folder is encrypted with the recipient's RSA public key, ensuring that only the recipient can decrypt it with their private key.

However, the devil is in the implementation details. To encrypt the cryptographic material (like the RSA private key and the node keys), MEGA used the AES-ECB (Electronic Codebook) mode of operation. This choice is considered a "red flag" by the cryptographic community. ECB is the simplest mode of AES, but it is notoriously insecure for most applications because it encrypts identical plaintext blocks into identical ciphertext blocks, and most importantly, it provides no integrity or authenticity protection.8 This lack of protection is a fundamental weakness that turned out to be the root cause of the discovered vulnerabilities.

1.4 Advertised Security vs. Architectural Reality

At first glance, the description of MEGA's architecture seems robust to a non-specialist. The use of terms like "end-to-end encryption" and "zero-knowledge" creates a powerful but potentially misleading mental model. Users interpret these terms as meaning that MEGA is *technically incapable* of accessing their data under any circumstances.

The reality, however, is that the security model makes such access *procedurally difficult* and depends entirely on MEGA's honesty as the custodian of the encrypted keys. The specific implementation choices—notably storing the user's encrypted private key on the server and using AES-ECB mode—create a fragile system whose security rests on the assumption that the server infrastructure is and will always remain trustworthy. As the next section will demonstrate, this assumption proved to be false, revealing a critical gap between the perception of absolute security conveyed by marketing and the reality of a conditionally secure architecture.

Section 2: Fundamental Flaws: An In-Depth Examination of the "Mega-Awry" Vulnerabilities

In June 2022, a team of researchers from ETH Zurich published a study that dismantled MEGA's facade of impenetrable security. Their analysis revealed a series of five attacks, collectively dubbed "Mega-Awry," which exploit fundamental flaws in the service's cryptographic design. These attacks are not minor theoretical exploits; they allow, under achievable conditions, a total compromise of the confidentiality and integrity of user data.

2.1 The Root Cause: Malleable Encryption and Lack of Integrity Protection

The root cause of all the discovered vulnerabilities lies in a fundamentally flawed cryptographic design choice: the use of the AES-ECB mode of operation for key encryption.8 As mentioned earlier, ECB mode is a "red flag" for cryptographers because it provides no integrity protection. This makes the ciphertext "malleable," a term that describes an attacker's ability to modify a ciphertext and predictably influence the plaintext result after decryption, without ever knowing the encryption key.12

This malleability is the catalyst that allowed the researchers to devise devastating attacks. In the absence of integrity verification (such as those provided by modern authenticated encryption modes like AES-GCM, which the researchers recommended), the MEGA client has no way of knowing if the cryptographic material it receives from the server has been altered in transit.8 It is this open door that was exploited.

2.2 Attack Vector 1: RSA Private Key Recovery

The first and most critical attack allows a malicious server (or MEGA itself) to recover a user's RSA private key.6

- **Mechanism:** The attack exploits the malleability of the ciphertext containing the user's RSA private key, which is stored on MEGA's servers. An attacker controlling the server can systematically alter this ciphertext before sending it to the client during a login attempt. With each login, the way the user's client processes this altered key and responds to the server during the session ID (SID) exchange leaks a tiny amount of information—a single bit—about one of the prime factors of the RSA key.7
- **Impact:** By repeating this process with each user login, the attacker can gradually accumulate enough bits to reconstruct the complete RSA private key. Initially, the ETH Zurich researchers estimated that 512 successful logins were required. However, subsequent research by UCSD researchers significantly improved the attack, reducing this threshold to just six logins.6
- **Consequence:** The recovery of the RSA private key is catastrophic. It allows the attacker to decrypt all data that has been shared with the user, impersonate them to other users, and set the stage for even more serious attacks. This completely shatters the confidentiality model for all shared data.

2.3 Attack Vector 2: Plaintext Recovery and Decryption of User Files

Once the RSA private key is compromised, the attacker can move on to the next phase: decrypting any of the user's files, even those that have never been shared.8

- **Mechanism:** The attacker uses the same malleable login process. This time, instead of simply altering the RSA key, they can "cut and paste" ciphertext blocks corresponding to file keys (the node keys) into the data stream where the client expects to receive the RSA key. The unsuspecting client decrypts these file keys using the user's master key and sends back information that allows the attacker to reconstruct the plaintext file key.12
- **Impact:** This technique allows an attacker to recover one file key per login attempt.12
- **Consequence:** This is the ultimate violation of the "zero-knowledge" promise. An attacker controlling MEGA's servers can, over time, acquire the keys to all of a user's files and thus access their entire stored content.

2.4 Attack Vectors 3 & 4: Integrity Violation and Malicious File Injection ("Framing" Attacks)

The attacks are not limited to a passive breach of confidentiality. The researchers demonstrated that an attacker could also move to an active phase by injecting malicious files into a user's account.8

- **Mechanism:** By understanding MEGA's file encryption structure and having the ability to decrypt and re-encrypt keys, a malicious server can forge an entirely new file (e.g., the `hacker-cat.png` file used in the proof-of-concept) along with its corresponding encrypted node key. It can then insert this file and its key into the list of files that the server sends to the user when they log in.8
- **Impact:** From the user's perspective, the malicious file appears in their cloud storage just like any other file. Because it is accompanied by a validly encrypted node key (albeit forged by the attacker), it appears perfectly authentic and decrypts correctly on the user's machine when opened.8

- **Consequence:** This attack breaks data integrity. A user can no longer be certain that the files in their storage space are the ones they originally uploaded. An attacker could replace legitimate documents with altered versions containing malware or disinformation, without the user being able to detect it.

2.5 Attack Vector 5: The GaP-Bleichenbacher Padding Oracle Attack

The last identified attack is a more complex variant of a classic cryptographic attack, which targets the key exchange mechanism of MEGA's old chat function.7

- **Mechanism:** This is a "padding oracle" attack, where an attacker exploits the way a server responds to malformed RSA messages (with incorrect padding) to progressively deduce the content. Although described as more "costly" in terms of computation, it remains a practical attack vector.7
- **Consequence:** It provides another method for a malicious server to decrypt user data (in this case, chat keys), further reinforcing the finding of systemic fragility in MEGA's cryptographic architecture.

Table 1: MEGA's Security Claims vs. Independent Research Findings

To summarize the impact of these discoveries, the following table directly contrasts MEGA's marketing promises with the technical reality revealed by the researchers.

MEGA's Claim	Reality ("Mega-Awry" Discovery)	Implication for User Security
"No one has access to your password but you — not even us." 2	A malicious server can exploit a user's repeated logins to systematically reconstruct their RSA private key. 8	**Total Loss of Confidentiality for Shared Data.** MEGA (or an attacker controlling its servers) can decrypt all files and folders shared with the user.
"With MEGA, you control the encryption. You hold the keys..." 2	A malicious server can trick the client into decrypting arbitrary data, including individual, unshared file keys. 8	**Total Loss of Confidentiality for All Data.** MEGA can gain access to and decrypt *all* files in a user's cloud storage, directly contradicting the "zero-knowledge" principle.
User files are authentic and can only be modified by the user.	A malicious server can forge and inject arbitrary files into a user's cloud storage, which appear perfectly authentic to the user. 8	**Total Loss of Data Integrity.** Users can no longer be certain that the files in their storage are the ones they originally uploaded. Malicious documents or spyware could be inserted without detection.

Section 3: Crisis Management: Evaluating MEGA's Mitigation Measures and the Path Forward

Faced with the publication of such fundamental vulnerabilities, MEGA's response was highly anticipated. The company reacted by releasing software updates and communicating about the measures taken. However, a thorough analysis of this response reveals a significant gap between the applied patches and the recommendations of security experts.

3.1 Analysis of Deployed Patches and Software Updates

In a blog post published in June 2022, MEGA officially acknowledged the ETH Zurich report and announced the release of software updates to fix the "critical vulnerability."9 The main patch aimed to neutralize the first and most effective attack: the recovery of the RSA private key. The update strengthened how the MEGA client handles the encrypted RSA key received from the server, making it insensitive to the manipulations that allowed the information leak.9

MEGA also later confirmed that this patch was effective against the improved attack (requiring only 6 logins) discovered by the UCSD researchers, thus reassuring users that this specific attack vector was indeed closed.9

3.2 The Divergence: A Quick Fix vs. a Necessary Overhaul

Herein lies the core of the disagreement between MEGA and the security community. MEGA's communication presented the situation as a flaw that was identified and fixed, implying a return to normal.9 In contrast, the researchers and other experts stressed that the patches, while necessary, were largely insufficient because they did not address the fundamental architectural problems. The root causes, such as the use of malleable encryption (AES-ECB) and the lack of key separation, remain unchanged.4

Professor Kenneth Paterson, one of the study's authors, publicly expressed his disappointment that MEGA had not committed to a complete redesign of its architecture, calling the remaining cryptography "rather brittle."7 The researchers' recommendation was clear: a complete system redesign using modern and robust primitives, such as AES-GCM authenticated encryption and more secure authentication protocols (augmented PAKE).8

MEGA's response can be interpreted as a crisis management action focused on public relations rather than a security-driven technical overhaul. By fixing the most spectacular and easily explained vulnerability, the company was able to publicly declare that "the problem is solved." However, it avoided the much more costly and complex task of redesigning its cryptographic architecture, a process that could have required re-encrypting petabytes of user data and potentially breaking backward compatibility.9 This choice suggests a business calculation where the cost and complexity of a complete redesign were deemed greater than the perceived risk of future attacks exploiting the remaining architectural weaknesses.

3.3 The Bug Bounty Program as a Security Measure

MEGA has long had a bug bounty program, offering up to €10,000 per reported flaw.14 As part of this program, the company paid a "significant reward" to the ETH Zurich researchers, which shows a degree of good faith in respecting responsible disclosure practices.9

The program explicitly invites reports of anything that could break MEGA's cryptographic security model, while excluding certain scenarios like social engineering or attacks requiring an already compromised client.15 However, while a bug bounty program is a sound security practice, it also highlights a reliance on external researchers to find flaws in a system that lacks formal, independent audits. A bug bounty is a reactive measure, whereas a proactive security audit is a preventive one. The fact that such fundamental design flaws persisted for nearly a decade before being discovered by academics suggests that the bug bounty program functioned as a substitute for, rather than a complement to, the rigorous validation that a formal audit would have provided.

3.4 Persistent Architectural Risks and Unresolved Concerns

The current state of MEGA's security is therefore ambiguous. The most direct and effective attack vector for recovering the RSA private key has been fixed. However, the underlying architecture, described as "brittle," remains. The use of legacy code and cryptographic "shortcuts" dating back nearly a decade is a risk acknowledged by MEGA itself.9

Furthermore, the fact that a user changing their password does not trigger a re-encryption of existing keys or files is a concern. This means that if an account had been compromised *before* the patch was applied, it could theoretically remain vulnerable in some respects.8 This is a subtle but important point that underscores that the patches are not retroactive for damage potentially already done.

Section 4: The Trust Equation: Transparency, Audits, and Operational History

Trust in a secure storage service rests not only on its cryptographic architecture but also on its transparency, verification practices, and history. On these points, MEGA's evaluation reveals significant strengths and weaknesses.

4.1 The Illusion of Open Source: Why Visibility Isn't Verification

MEGA highlights the publication of the source code for its client applications (web, desktop, mobile) as proof of its commitment to transparency.2 This allows, in theory, anyone to examine the code to ensure it does what it claims to do.

However, this transparency has a critical limitation: it only applies to the client-side code. The server-side code, which is at the heart of the "Mega-Awry" attacks and manages the infrastructure, remains proprietary and closed.18 Without the ability to verify the server's behavior, the transparency of the client code loses much of its value in protecting against a malicious provider. The client is forced to trust what the server sends it, which was precisely the starting point of the attacks.

4.2 The Missing Piece: The Lack of Independent Security Audits

Despite its claims of high security, there are no public, independent, third-party security audits of MEGA's cryptographic architecture or infrastructure.18 This is a major gap and a red flag for any service that aims to be a leader in security. It is important not to confuse mega.io with mega.ai, another company that does display SOC 2 and ISO 27001 certifications.22

An independent audit would provide the rigorous verification that simply publishing the client code cannot offer. The absence of such audits means that for nearly a decade, the system's fundamental flaws were neither identified by internal processes nor discovered by an external audit, but had to await the initiative of a university research team.

4.3 A History of Controversy and Abuse

MEGA's reputation is also shaped by its history. The service was launched by Kim Dotcom as the successor to Megaupload, a site shut down by authorities for massive copyright infringement.14 Although Dotcom has since left the company and even advised against using it following alleged takeovers, this controversial origin continues to mark the brand's image.14

Beyond its origins, MEGA has experienced operational security incidents. In 2018, its official Chrome browser extension was compromised and modified to steal cryptocurrencies and website credentials, demonstrating a flaw in its software supply chain security.19

More recently, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) identified Mega.io as a tool used by operators of the Phobos ransomware for exfiltrating stolen data.23 This illustrates how the very features that attract privacy-conscious users (strong encryption, perceived anonymity) also make it a tool of choice for cybercriminals.24 A service that promises that even its operator cannot see user data is the perfect exfiltration solution for ransomware groups. This puts MEGA in a delicate position, forced to combat abuse, as evidenced by its transparency reports on the removal of illegal content 25, while trying to preserve its promise of confidentiality.

4.4 Domain Reputation and Infrastructure Analysis

Operationally, MEGA uses several domain names: mega.nz remains the main portal for logged-in users, while mega.io has been used since 2021 for company and product pages, a decision motivated by better international SEO.14 Traffic analysis confirms a significant global user base.28

An undeniable positive point of MEGA's infrastructure is the location of its data centers. Data is stored in Europe or in countries deemed adequate by the GDPR, such as New Zealand, with an explicit policy of not storing user files in the United States.21 From a jurisdictional privacy perspective, this is a major advantage for users concerned about U.S. surveillance laws like the CLOUD Act.

Section 5: The Legal and Privacy Framework: Examining Policies and Compliance

The security assessment of a cloud service cannot be complete without a careful examination of its legal framework and privacy policies. These documents define the rules of the game regarding the collection, use, and protection of user data.

5.1 Data Residency, Jurisdictional Risk, and GDPR

As mentioned previously, MEGA's data residency strategy is a major asset. By hosting data in secure data centers in Europe and in countries recognized for their adequate level of protection, such as New Zealand, MEGA positions itself favorably from a privacy standpoint.21 The deliberate avoidance of U.S. data centers is a concrete measure to protect users from the reach of American surveillance agencies.21

Furthermore, MEGA has committed to applying the protections and rights conferred by the European Union's General Data Protection Regulation (GDPR) to all its users worldwide, not just those residing in the EU.29 This is a positive step that strengthens its image as a privacy-conscious company.

5.2 Analysis of the Terms of Service and Privacy Policy

A detailed review of MEGA's legal documents reveals important nuances that users should understand.30

- Privacy Policy 30: The most crucial point is to understand what MEGA collects in unencrypted form. Although the *content* of the files is protected by "zero-knowledge" encryption, a significant amount of metadata is not. MEGA collects and retains account data (email address, IP address history, browser and operating system type), usage data (file sizes, parent-child relationships between files, contacts' email addresses, chat metadata like start and end times), and financial transaction records. This metadata collection, while necessary for the service's operation, strongly nuances the promise of absolute privacy.
- Terms of Service 31: These terms reveal a multi-layered privacy model. While the main cloud storage service is presented as being fully user-controlled encrypted (UCE), other products in the range, such as MEGA S4 object storage, are *not*. The terms clearly state that MEGA can access the content of files stored on S4 and provide them to authorities if required by law. This is a critical distinction for users of the MEGA ecosystem, who might mistakenly assume that the encryption guarantee applies to all of the brand's products.

5.3 Data Collection: What MEGA Knows and Keeps

By synthesizing the information from the privacy policy, a clear picture can be drawn of what MEGA knows about its users, even without being able to read their files. MEGA knows: who you are (via your email), when and from where you connect (IP address logs), with whom you communicate (contact lists and chat participants), and how your data is structured (file sizes, folder tree).30

For many threat models, particularly those involving state surveillance, this metadata can be as revealing as the content itself. A government agency with a court order could, from this metadata, reconstruct a contact network, track a user's activities, and infer the nature of their data, even without ever decrypting it. The "zero-knowledge" promise is thus betrayed by the richness of the collected metadata.

Regarding retention, MEGA keeps personal data as long as the account is active. After an account is terminated, the data may be retained for up to 12 months to allow for possible reactivation, after which it is anonymized.30

Section 6: Final Verdict and Strategic Recommendations

6.1 Synthesis of Evidence: Re-evaluating the Value of MEGA's Security

The analysis conducted in this report paints a complex picture of MEGA's security. The initial promise of an impenetrable, user-controlled storage system was found to be based on a fragile cryptographic architecture. The "Mega-Awry" vulnerabilities were not mere bugs, but symptoms of fundamental design flaws that persisted for nearly a decade.

The company's response, though swift, consisted of applying a band-aid to the most visible wound by fixing the most direct exploit, while leaving the underlying architecture vulnerable. This choice to prioritize business continuity over a complete security overhaul is telling. Furthermore, the persistent lack of independent security audits is a critical trust gap, turning the open-source code transparency argument into a form of security theater.

Therefore, the "value" of MEGA's security is highly conditional. It depends entirely on the user's threat model: who is the potential adversary, and what is the sensitivity of the data to be protected?

6.2 Risk Profiles and Usage Recommendations

Based on this analysis, clear recommendations can be formulated for different types of users.

- **For Casual Users (Non-sensitive data):** For storing family photos, non-critical documents, or sharing large files, where the main concern is to avoid the data mining practiced by services like Google, MEGA is likely "good enough." The patched vulnerabilities make a sophisticated server-side attack less likely against a low-value target. Its generous free offering remains a strong selling point.1
- **For Privacy-Conscious Individuals (Personal but sensitive data):** For archiving financial documents, personal journals, or sensitive communications, MEGA represents a risky choice. The main threat here is the possibility that the provider itself (or a government compelling it) becomes an adversary. The "Mega-Awry" findings prove that this is a viable threat vector.
  - **Recommendation:** Do not use MEGA for this purpose, unless applying an additional layer of client-side encryption *before* uploading. Using a trusted, audited third-party tool like Cryptomator or Veracrypt turns MEGA into a simple "dumb" storage pipe, thus neutralizing the risk associated with its flawed cryptography.
- **For Professional and Business Use (Highly sensitive data):** For journalists, activists, lawyers, or companies handling trade secrets or client data, MEGA is **strongly discouraged**. The demonstrated architectural flaws, the lack of independent audits, and the potential for data integrity attacks (file injection) make it an unacceptable risk in a professional context. The possibility of a malicious server compromising both the confidentiality and integrity of data is a deal-breaker.

6.3 Best Practices to Mitigate Risks When Using MEGA

For any user who chooses to continue using MEGA, a series of best practices should be rigorously applied to minimize risks:

1. **Use a strong, unique password:** This is the most fundamental measure, as this password is the basis of the entire encryption chain.3
2. **Enable two-factor authentication (2FA):** This protects the account against credential theft and credential stuffing attacks.1
3. **Keep the Recovery Key in a safe place:** Losing it means permanently losing access to the account and data.2
4. **Encrypt sensitive data yourself beforehand:** This is the most effective mitigation measure. Use trusted client-side encryption software like Cryptomator before uploading files to MEGA.
5. **Be wary of sharing links:** Do not blindly trust shared files, as the integrity attack remains a theoretical possibility.
6. **Regularly check session history:** Monitor for any unrecognized logins to detect a potential account compromise.33

Table 2: Summary of "Mega-Awry" Vulnerabilities and their Mitigation Status

The following table summarizes the vulnerabilities, their impact, and, most importantly, the current status of their remediation, highlighting the architectural risks that justify the cautious recommendations of this report.

Vulnerability Name	Description	Potential Impact	Mitigation Status and Residual Risk	
1. RSA Key Recovery	A malicious server alters the encrypted RSA key to leak information over 6+ logins.	Decryption of all shared data; user impersonation.	**Fixed.** The specific client-side vulnerability that allowed the information leak has been patched.9	
2. Plaintext Recovery	A malicious server uses the recovered RSA key to trick the client into decrypting file keys.	Decryption of *all* user files, shared or not.	**Indirectly Mitigated.** This attack depended on the prior recovery of the RSA key. With Attack 1 fixed, this specific vector is blocked. **Residual Risk:** The underlying mechanism (malleable ECB encryption) remains.	

Vulnerability Name	Description	Potential Impact	Mitigation Status and Residual Risk	
3. "Framing" Attack	A malicious server forges and injects malicious files into the user's storage.	Total loss of data integrity; user could download and execute malware disguised as a legitimate file.	**Partially Mitigated.** MEGA claims to have mitigated this attack.13	**Residual Risk:** Researchers argue that the fundamental lack of integrity protection in the file format makes such attacks plausible until a complete redesign.
4. Integrity Attack	A variant of the "Framing" attack, also exploiting the lack of integrity controls.	Total loss of data integrity.	**Partially Mitigated.** Same status as the "Framing" attack. **Residual Risk:** The same as above. The main architectural flaw persists.	
5. GaP-Bleichenbacher	A padding oracle attack against the key exchange mechanism of the old chat.	Decryption of chat keys.	**Not Addressed (according to initial reports).** MEGA prioritized fixing more critical flaws. **Residual Risk:** Demonstrates that several parts of the cryptographic system were built with "brittle" or non-standard implementations.	

Citation Sources

1. MEGA Cloud Storage: Create a Free Account, accessed September 21, 2025, https://mega.io/storage
2. How MEGA Protects Your Privacy and Data, accessed September 21, 2025, https://mega.io/security
3. MEGA Review (MEGA.IO) [Secure Cloud Storage] - The Latest Backup Software Reviews | BestBackupReviews.com, accessed September 21, 2025, https://www.bestbackupreviews.com/reviews/mega-review/
4. Le chiffrement du Cloud MEGA mis à mal par des chercheurs - Clubic, accessed September 21, 2025, https://www.clubic.com/disque-dur-memoire/stockage-en-ligne/actualite-428061-le-chiffrement-du-cloud-mega-mis-a-mal-par-des-chercheurs.html
5. Is MEGA private and secure? : r/PrivacyGuides - Reddit, accessed September 21, 2025, https://www.reddit.com/r/PrivacyGuides/comments/sffoi2/is_mega_private_and_secure/
6. MEGA Security Flaw: A Full Cloud Storage Security Review in 2025 - Cloudwards, accessed September 21, 2025, https://www.cloudwards.net/mega-security-flaw/
7. Mega's unbreakable encryption proves to be anything but - The Register, accessed September 21, 2025, https://www.theregister.com/2022/06/22/megas_encryption_broken/
8. MEGA: Malleable Encryption Goes Awry, accessed September 21, 2025, https://mega-awry.io/
9. MEGA Security Update June 2022, accessed September 21, 2025, https://blog.mega.io/mega-security-update
10. MEGA: Protect your Online Privacy, accessed September 21, 2025, https://mega.io/
11. How does Mega's encryption work for sharing? - Stack Overflow, accessed September 21, 2025, https://stackoverflow.com/questions/18346054/how-does-megas-encryption-work-for-sharing
12. MEGA: Malleable Encryption Goes Awry - YouTube, accessed September 21, 2025, https://www.youtube.com/watch?v=nJQ-DbntAUs
13. Critical Flaws in MEGA Cloud Storage Let Attacker Decrypt User Data - GBHackers, accessed September 21, 2025, https://gbhackers.com/critical-flaws-mega-cloud-storage/

14. Mega (website) - Wikipedia, accessed September 21, 2025, https://en.wikipedia.org/wiki/Mega_(service)
15. Bug Bounty: Report Vulnerabilities - MEGA, accessed September 21, 2025, https://mega.io/bug-bounty
16. A curiosity in MEGA's 2022 security whitepaper : r/Bitwarden - Reddit, accessed September 21, 2025, https://www.reddit.com/r/Bitwarden/comments/1asdncm/a_curiosity_in_megas_2022_security_whitepaper/
17. AI flagged some serious crypto concerns about Mega - should I be concerned? - Reddit, accessed September 21, 2025, https://www.reddit.com/r/MEGA/comments/1lhjkq9/ai_flagged_some_serious_crypto_concerns_about/
18. audit - Does MEGA cloud service say ALL the truth? - Information Security Stack Exchange, accessed September 21, 2025, https://security.stackexchange.com/questions/95563/does-mega-cloud-service-say-all-the-truth
19. Mega (service) - Wikipedia, accessed September 21, 2025, https://en.wikipedia.org/wiki/Mega_(service)
20. MEGA Transparency Report 2021, accessed September 21, 2025, https://blog.mega.io/mega-transparency-report-2021
21. MEGA Cloud Storage Review (2025 Test Results) - CyberInsider, accessed September 21, 2025, https://cyberinsider.com/cloud-storage/reviews/mega/
22. Security and Compliance - Safeguarding Data With Highest Priority - Mega AI, accessed September 21, 2025, https://www.mega.ai/security-and-compliance
23. #StopRansomware: Phobos Ransomware | CISA, accessed September 21, 2025, https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-060a
24. An Encounter with Ransomeware-as-a-Service: MEGAsync Analysis - Recon InfoSec, accessed September 21, 2025, https://blog.reconinfosec.com/megasync-analysis
25. MEGA Transparency Report, accessed September 21, 2025, https://mega.io/transparency
26. Quels sont les domaines utilisé par MEGA à titre officiel, accessed September 21, 2025, https://help.mega.io/fr/security/data-protection/official-domains
27. Additional Domain for MEGA: mega.io - Reddit, accessed September 21, 2025, https://www.reddit.com/r/MEGA/comments/l08d0s/additional_domain_for_mega_megaio/
28. mega.io Website Traffic, Ranking, & Analysis [August 2025] - Semrush, accessed September 21, 2025, https://www.semrush.com/website/mega.io/overview/
29. Avis MEGA.nz (2025) : stockage cloud sécurisé et confidentiel - Clubic, accessed September 21, 2025, https://www.clubic.com/stockage-en-ligne/avis-352094-mega-nz.html
30. Privacy Policy - MEGA, accessed September 21, 2025, https://mega.io/privacy
31. Terms of Service - MEGA, accessed September 21, 2025, https://mega.io/terms
32. MEGA Review 2025: Is it More Secure than Google Drive, OneDrive, and other Cloud Services, accessed September 21, 2025, https://www.experte.com/cloud-storage/mega
33. How do I secure my MEGA account after it has been compromised?, accessed September 21, 2025, https://help.mega.io/security/data-protection/account-compromised
34. How can I check MEGA login sessions and log them out remotely?, accessed September 21, 2025, https://help.mega.io/security/data-protection/login-sessions