Filen

# 보안 백서

2021년 8월

# 콘텐츠

# 1 소개

## 1.1 파일이란 무엇인가요?

Filen은 완전한 제로 지식 엔드투엔드 암호화 클라우드 스토리지 및 커뮤니케이션 플랫폼입니다.
제로 지식 종단 간 암호화는 중개자가 없음을 의미합니다. 파일엔(또는 파일엔의 직원)은 사용자의 암호화 키에 액세스할 수 없으므로 플랫폼에서 전송되거나 저장된 데이터는 온전히 사용자의 손에 남아 있습니다.

## 1.2 엔드투엔드 암호화 및 지식 제로

대부분의 다른 클라우드 스토리지 플랫폼과 달리, Filen에서는 사용자만 플랫폼에 저장된 데이터에 액세스할 수 있습니다. 처음부터 Filen은 사용자가 제어하는 엔드투엔드 암호화를 중심으로 설계되었습니다. 데이터는 플랫폼으로 전송되기 전에 사용자 컴퓨터에서 암호화됩니다. 암호화 키는 사용자만 보유하며, Filen조차도 암호화 키에 액세스할 수 없습니다. 사용자가 데이터를 공유하고자 하는 경우, 데이터를 전송하기 전에 수신자의 공개 키로 필요한 암호화 키를 암호화합니다. 이 프로세스는 100% 데이터 소유권과 개인정보 보호를 보장합니다.

## 1.3 투명성

필요한 모든 암호화 작업은 사용자 컴퓨터에서 직접 수행됩니다. Filen은 애플리케이션의 전체 최신 소스 코드를 공개하여 구현의 투명성을 제공합니다. 소스 코드 링크는 메인 홈페이지에서 확인할 수 있습니다.

## 1.4 개인 정보 보호

제로 지식 엔드투엔드 암호화를 사용하는 Filen은 설계부터 개인정보 보호 기능을 제공합니다. 대부분의 다른 클라우드 스토리지 및 통신 플랫폼과 달리, Filen의 사용자 제어 암호화는 사용자만 플랫폼에 저장된 데이터에 액세스할 수 있도록 합니다(암호화 키를

자발적으로 공유하는 경우 제외). 폴더 이름, 파일 메타데이터(예: 마임 유형, 크기 등)도 암호화됩니다. Filen은 정책별 개인정보 보호가 적용되는 사용자의 이메일 및 IP 주소와 같은 다양한 기타 거래 메타데이터를 저장합니다. Filen의 개인정보 처리방침은 메인 홈 페이지에서 확인할 수 있습니다.

## 1.5 취약점 관리

Filen은 항상 보안을 유지하기 위해 이전에 알려지지 않은 취약점이나 버그를 신고하는 모든 분께 보상을 제공합니다. 취약점을 발견하면 지원팀에 문의하시면 최대한 빠른 시일 내에 개발팀에서 연락을 드릴 것입니다.

## 1.6 중복성

Filen은 인프라를 직접 운영하며 타사 클라우드 제공업체에 의존하지 않습니다. 모든 하드웨어는 독일에 위치한 안전한 시설에서 호스팅됩니다. 미국에는 어떠한 데이터도 저장되지 않습니다.

Filen에는 다음과 같은 다양한 범주의 인프라가 있습니다:

1. 여러 노드와 로드 밸런서로 구성된 웹 클러스터로 고가용성 보장
2. API 클러스터
3. 여러 데이터센터에 복제된 여러 노드로 구성된 데이터베이스 클러스터
4. 6+3 삭제 코딩을 사용하는 스토리지 클러스터를 여러 데이터센터로 분할하여 데이터 일관성과 가용성을 보장합니다. 고장난 하드 드라이브는 몇 분 안에 핫스왑할 수 있습니다.
5. 실시간 메시지 브로커 서비스
6. Filen의 속도 테스트 노드 및 자체 호스팅 분석 서비스와 같은 기타 서비스

## 1.7 규정 준수

Filen은 규제 요건을 최고 수준으로 준수하기 위해 운영됩니다. Filen의 서비스는 독일 법률의 적용을 받습니다. Filen은 플랫폼에 저장된 데이터를 볼 수는 없지만, 저작권이 있는 콘텐츠가 신고되면 이를 삭제합니다. Filen 사용자는 플랫폼을 통해 공개 링크의 형

태로 데이터를 공유할 수 있습니다. 이러한 공개 링크는 사용자가 파일/폴더에 대해 생성하도록 선택할 때 URL 해시에 필요한 암호 해독 키가 추가됩니다. Filen은 남용 신고 또는 통지를 받으면 모든 등록된 사용자가 동의한 서비스 약관에 따라 요청 유형에 따라 문제가 되는 파일(폴더 포함)을 즉시 제거하거나 해당 파일에 대한 액세스를 비활성화합니다.

# 2 등록 및 로그인

## 2.1 등록 절차

계정을 만들 때 사용자는 이메일 주소와 비밀번호를 입력해야 합니다. Filen은 비밀번호를 직접 저장하지 않습니다. 비밀번호 처리를 위해 Filen은 확립된 PBKDF2 표준을 사용합니다. 이 표준은 최신, 최고 또는 최첨단은 아니지만, 잘 알려져 있고 특히 WebCrypto API에서 광범위한 언어를 지원하며 기본 속도로 중요한 성능을 발휘합니다.

비밀번호 해시는 다음과 같이 계산됩니다:

소금 = 암호화 방식으로 안전하게 생성된 임의의 256 문자 값
해시 = SHA-512
반복 = 200.000
비트 길이 = 512

파생 키 = PBKDF2(비밀번호, 소금, 반복, 해시, 비트 길이)

이 작업을 수행하면 512비트 키가 생성되며, 이 키는 16진수로 변환됩니다.

그런 다음 파생된 키는 왼쪽에서 오른쪽으로 균등하게 분할됩니다. 첫 번째 절반은 사용자 마스터 키로, 두 번째 절반은 SHA-512를 사용하여 다시 해시한 다음 인증 키로 작동합니다.

그러면 이 데이터(사용자 이메일 주소, 솔트 및 해시된 인증 키)가 API로 전송되고 계정이 생성됩니다. 사용자의 이메일 주소로 전송된 이메일을 확인한 후 로그인이 가능합니다. Filen은 해시되지 않은 인증 키를 저장하지 않습니다. 해시된 각 인증 키는 데이터 유출 시 해시 공격 통과를 방지하기 위해 서버 측에서 Argon2를 사용하여 다시 해시됩니다.

## 2.2 로그인 프로세스

로그인 절차는 다음과 같습니다:

1. 사용자는 클라이언트 인터페이스에 이메일 주소와 비밀번호를 입력해야 합니다.
2. 이메일 주소가 API로 전송됩니다.
3. 이메일 주소가 포함된 레코드가 데이터베이스에 존재하면 API는 사용자의 솔트로 응답합니다.
4. 데이터베이스에서 이메일 주소를 찾을 수 없는 경우, API는 무작위로 생성된 솔트로 응답하여 이메일 주소 무차별 대입 공격을 방지합니다.
5. 이제 클라이언트는 등록 프로세스에서 설명한 대로 사용자의 마스터 키와 해시된 인증 키를 계산할 수 있습니다.
6. 해시된 인증키가 계산되면 클라이언트는 이메일 주소와 해시된 인증키를 API로 전송하고, 인증에 성공하면 사용자의 API 키로 응답합니다.
7. 사용자의 첫 로그인인 경우 클라이언트는 마스터 키를 암호화하여 API로 전송합니다. 또한 사용자의 마스터 키(AES-GCM 256비트)를 사용하여 암호화한 RSA-OAEP 키쌍(4096비트 모듈러스, SHA-512 해시)이 생성되어 API로 전송됩니다. Filen은 암호화되지 않은 키를 저장하지 않습니다. 암호화된 키는 사용자가 사용할 수 있는 다른 클라이언트에서 데이터를 해독할 수 있는지 확인하는 데 사용됩니다. 예를 들어 비밀번호가 변경되는 경우 Filen은 새 비밀번호에서 파생된 새 마스터 키를 이전 마스터 키에 추가합니다. Filen에서는 이를 마스터 키 체인이라고 부르며, 이를 통해 비밀번호를 쉽게 변경할 수 있습니다. 이 프로세스는 계정 복구 섹션에 자세히 설명되어 있습니다.

## 2.3 계정 복구

사용자 비밀번호는 모든 클라이언트 측 암호화의 근간이 되므로 비밀번호를 잊어버리면 사용자의 데이터를 해독할 수 없게 되어 파일엔이나 사용자 모두 사용자 계정에 저장된 데이터를 복구할 수 없게 됩니다.

사용자가 다른 디바이스에서 로그인한 상태라면 비밀번호를 변경할 수 있습니다. 비밀번호를 변경하면 등록 절차에 설명된 대로 새 마스터 키가 생성됩니다. 그런 다음 이 마

스터 키가 기존 마스터 키에 추가되고 암호화되어 API로 전송됩니다. Filen은 암호화되지 않은 키를 저장하지 않습니다. Filen은 이 마스터 키 체인화를 호출합니다. 이 프로세스를 통해 이전 마스터 키로 암호화된 데이터의 암호 해독이 가능하며, 새 데이터는 새 암호화 키를 사용하여 암호화됩니다.

## 2.4 2단계 인증 인증

Filen은 시간 기반 일회용 비밀번호(TOTP)를 사용하여 2단계 인증(2FA)을 구현했습니다. 공유 비밀 생성에는 32개의 무작위 바이트가 사용되며, 이는 Base 32로 변환되어 사용자에게 QR코드와 일반 Base 32 문자열로 표시됩니다. 활성화 시 사용자에게 복구 키가 표시되며, 다음과 같은 경우 지원팀에 연락하여 계정을 복구하는 데 사용할 수 있습니다.

2FA 장치 분실.

# 3 클라우드 드라이브 암호화 및 암호 해독

## 3.1 데이터 업로드 암호화

각 파일에는 고유한 암호화 키가 있습니다. 파일 데이터, 이름, 메타데이터, 폴더 이름이 암호화됩니다.

파일 암호화를 위해 암호학적으로 안전한 256비트 무작위 키가 생성됩니다. 폴더 이름은 사용자의 마스터 키를 사용하여 암호화됩니다.

각 파일은 1MB 청크로 분할되고, AES-GCM 256비트(데이터 무결성을 보장하기 위해 인증됨)를 사용하여 암호화된 다음 API에 업로드됩니다.

그런 다음 파일 암호화 키를 포함한 파일 메타데이터는 사용자의 마스터 키를 사용하여 암호화되어 API로 전송됩니다.

| | |
|---|---|
| | 파일= AES-GCM(chunkData, fileKey) |
| 파일 메타데이터 | AES-GCM(fileMetadata,userMasterKey) 폴 |
| 더 이름 | AES-GCM(folderName, userMasterKey) |

## 3.2 썸네일 세대

클라이언트는 암호화된 데이터를 다운로드하고, 암호를 해독하고, 결과 이미지의 크기를 조정하고, 압축한 다음 클라이언트의 로컬 저장소에 저장하여 재사용할 수 있도록 이미지 썸네일을 생성합니다. 비디오 미리보기는 대역폭과 로컬 저장 공간을 절약하기 위해 전체 비디오를 다운로드하지 않고 처음 몇 개의 청크만 다운로드한다는 점을 제외하면 거의 동일한 방식으로 생성됩니다.

## 3.3 다운로드

클라이언트는 암호화된 청크를 다운로드하고 해독한 다음 해독된 청크를 올바른 순서로 사용자의 로컬 파일 시스템 또는 브라우저 다운로드 디렉터리로 스트리밍하여 파일을 다운로드합니다.

폴더 다운로드는 모든 파일이 클라이언트 쪽에서 압축된다는 점을 제외하면 동일한 방식으로 작동합니다. 이렇게 하면 사용자가 많은 개별 파일을 다운로드하지 않아도 됩니다. 브라우저에는 메모리 제한이 있으므로 폴더 다운로드는 크기가 제한됩니다. 사용자는 Filen의 데스크톱 클라이언트를 다운로드하여 무제한 크기의 폴더를 다운로드할 수 있습니다.

# 4 협업

## 4.1 다른 Filen 사용자와 데이터 공유

사용자가 다른 파일 사용자와 파일 또는 폴더를 공유하기로 선택하면 수신자 RSA-OAEP 공개키를 사용하여 모든 메타데이터를 암호화합니다.

| | |
|---|---|
| 파일 메타데이터 | RSA-OAEP(fileMedata+fileKey,recipientPublicKey) 폴 |
| 더 메타데이터 | RSA-OAEP(folderName, recipientPublicKey) |

그러면 모든 데이터가 API로 전송되어 수신자가 사용할 수 있게 됩니다. 그러면 수신자는 RSA-OAEP 개인 키를 사용하여 모든 데이터를 해독하고 공유 데이터에 정상적으로 액세스할 수 있습니다.

## 4.2 공개 링크

사용자가 공개 링크를 사용하여 파일 또는 폴더를 다른 파일 사용자와 공유하기로 선택하면 클라이언트는 암호학적으로 안전한 256비트 키를 새로 생성합니다. 이 키는 모든 메타데이터를 암호화하는 데 사용됩니다. 이 키는 공개 링크의 URL 해시에 추가됩니다. 키는 또한 사용자의 마스터 키를 사용하여 암호화되고 나중에 사용할 수 있도록 저장됩니다.

| | |
|---|---|
| 파일 메타데이터 | AES-GCM(fileMetadata+fileKey,newEncryptionKey) 폴 |
| 더 메타데이터 | AES-GCM(folderName, newEncryptionKey) |

공개 링크는 만료되거나 비밀번호로 보호되도록 구성할 수 있습니다. 공개 링크의 비밀번호 해시는 PBKDF2(비밀번호, salt, 반복, 해시, bitLength)를 사용하여 원래 비밀번호에서 파생되며, 여기서 salt는 임의의 256비트 문자열, 반복은 200.000으로 설정, 해시는 SHA-512로 설정, bitLength는 512로 설정합니다.