Technical Overview

In this technical overview we will go deeper into certain technical aspects. If you want to learn more about how Whisply (https://whisp.ly) works, the following information will be of your interest.

Please first have a look at the following table with important terms you need to know:

Important terms:

Term	Description
Unlocking Key	An AES key used to decrypt the encrypted file key, which is stored in the files themselves. The information on the Whisply server, the PIN/password and the Whisply link is required to calculate the Unlocking Key.
Shareable ID	An ID (12 characters long) uniquely identifying one file or a block of files. This value is sent with the Whisply link and is stored on the Whisply server.
Link Key Part	A secret information necessary to decrypt the file (13 characters long). It is sent with the Whisply link in a way that the Link Key Part is not sent to the Whisply server when the receiver clicks on it.
Password Key	A key generated from your PIN or password. This key is never stored anywhere.
Password Hash	A one-way hash used to authorize the receiver of the file to download the file specific information stored on the Whisply server.

How Whisply encrypts and decrypts files

Whisply uses the same file encryption format as Boxcryptor (currently without filename encryption) - see https://www.boxcryptor.com/en/technical-overview#anc01 for details. Instead of encrypting the file key with a user's public key, though, the file key is encrypted with an Unlocking Key. This Unlocking Key is generated from information stored on the Whisply-Server, PIN or password and information attached to the generated Whisply link.

How the sharing process works

Example: Alice wants to share a file with Bob. Then the following steps are executed:

1. Alice creates a random Link Key Part, a random Shareable ID, a random File Key Part (256 bits), a random Shareable Key (256 bits), and a random salt (1024 bytes)
2. Alice calculates an Unlocking Key from the Link Key Part and the File Key Part (with HKDF-SHA-512 and the salt, which is internally split in two for the HKDF-parameters "salt" and "info")

3. Alice encrypts her file with the Unlocking Key and uploads it to her preferred cloud storage provider - she retrieves a download link for the uploaded, encrypted file.
4. Alice generates or chooses a PIN or password and calculates a Password Hash and a Password Key from the PIN or password
5. Alice encrypts the File Key Part with the Shareable Key and encrypts the Shareable Key with the Password Key
6. The ID, salt, encrypted File Key Part, File Download Link, Password Hash and encrypted Shareable Key are sent to the Whisply server
7. The ID and Link Key Part are sent to Bob with the Whisply link. PIN or password are sent separately, e.g. via SMS

To download and decrypt the file, Bob executes the following steps:

1. Bob calculates the Password Hash and requests the information stored on the Whisply server for the ID he received with the Whisply link
2. Bob calculates the Password Key and decrypts the Shareable Key
3. Bob decrypts the File Key Part with the Shareable Key
4. Now Bob has all necessary information to calculate the Unlocking Key
5. Bob downloads the encrypted file from the File Download Link and decrypts the file with the Unlocking Key

How the sharing process of several files works

The process for several files is equivalent to the one for sharing one file. The only difference is that every file gets its own random File Key Part as well as its own random salt.

How the Password Hash and Password Key are calculated

To calculate the Password Hash, Whisply uses PBKDF2-SHA512 with 10.000 iterations and a static string linked with the ID as salt. The resulting hash is 512 bit long.

To calculate the Password AES-Key, Whisply uses PBKDF2-SHA512 with 10.000 iterations and a different, static string linked with the ID as salt.

How the “link only” encryption without a PIN works

In this case, an empty string is used as password

Why a short PIN is still secure

A short PIN can be a problem if the attacker has the possibility to try several possible PINs quickly after one another (brute-forcing). But since the PIN (or better the hash of the PIN) is used to authorize the user to download the necessary information to continue the decryption, Whisply can control how many trials an attacker gets. We force the attacker to wait for a while after

entering a wrong PIN - and this waiting time increases by the factor 2 every time a wrong PIN is sent (exponential back-off). That way, an attacker can only try very few possible PINs and a short PIN is considered secure enough. This is equivalent to your banking PIN: If you enter your short PIN too often, the card will get blocked. There's a more detailed explanation about which security level is best suited for your purposes in the Security level section.

How Whisply is zero-knowledge

Whisply is a zero-knowledge provider because Whisply as a provider has never enough information to reconstruct the key used to encrypt your files. Whisply has no access to the key part stored in the Whisply link which is required to calculate the file key - in none of the security levels. Additionally, all information necessary to create the file key which is stored on our server is encrypted with the PIN or password, and a good password prevents us to decrypt those information (which would still not be enough to calculate the file key!).

The zero-knowledge character is ensured by performing all cryptographic operations in the browser and on your own device. Information sent to the Whisply server is either not secret, encrypted, or missing crucial parts necessary to use them. The PIN or password is never sent to the Whisply server.

Which data is stored on the Whisply server

The following information is stored on the Whisply server for each uploaded file

Information	Description
Shareable ID	A random, non-secret, unique ID that identifies the upload. Needed to specify which file has to be downloaded and to calculate a unique password hash and password key
Random salt for ever file uploaded	A non-secret, random string that is used during the calculation of the Unlocking Key
Download link for every shared file	A link that can be used to download the encrypted file from the cloud storage provider
Encrypted File Key Part	A key encrypted with a random, so-called Shareable Key. This key is necessary to calculate the Unlocking Key
Password / PIN hash	A one-way hash of the PIN or password, used to verify if a user requesting the information on the server is authorized
Shareable Key	A key encrypted with the user's PIN or password, necessary to decrypt the encrypted File Key Part

Which cryptographic libraries are used in Whisply

In order to perform the actual "low level" encryption and random number generation, Whisply relies on established and proven third-party libraries.

- The WebCrypto API is used where available
- The Stanford Javascript Crypto Library is used as fallback when WebCrypto is not available. (https://github.com/bitwiseshiftleft/sjcl)
- The HKDF-algorithm is implemented by ourselves, but uses the primitives from the WebCrypto API or the Standford Javascript Crypto Library internally.

How the security levels are defined

Whisply offers three different security levels - depending on your security requirements and depending on how you want to share your files. In the following table we assume that the Whisply link is shared via mail and the PIN or password (if used) is shared via SMS.

Security level	Description	Security
Link	Whisply only generates one link	▪ We cannot decrypt your files ▪ Your cloud provider cannot decrypt your files Only the receiver of the link can decrypt the files. The key includes all information necessary to decrypt the file.
Link and PIN	Whisply generates a link and a 4 digit PIN	▪ We cannot decrypt your files ▪ Your cloud storage provider cannot decrypt your files ▪ Your mail provider cannot decrypt your files ▪ Your SMS service cannot decrypt your files ▪ Someone who hacked both our server and your SMS service cannot decrypt your files Your cloud storage provider, your SMS provider and we do not have the information necessary to decrypt the file which is included in the Whisply link. Your mail provider does not have the PIN required to request the additional information from our server, which is needed to decrypt the file

Link and password	Whisply generates a link and allows you to choose your own password	▪ We cannot decrypt your files ▪ Your cloud storage provider cannot decrypt your files ▪ Your mail provider cannot decrypt your files ▪ Your SMS service cannot decrypt your files ▪ Someone who hacked both our server and your SMS service cannot decrypt your files ▪ Someone who hacked both our server and your mail provider cannot decrypt your files The first five points are equivalent to the "link and PIN" security level. The password is required to decrypt the encrypted Shareable Key located on our server. This Shareable Key is required to decrypt your files. If only a PIN is used and someone gets access to both our server and your mails, he can try possible (short) PINs without us being able to stop him. If a strong password is used, he still cannot guess the password and therefore cannot decrypt the files.