equinux

VPN Tracker
COMPANY CONNECT

Connection Safe Security Architecture

VPN Tracker 365 Connection Safe – Security Architecture

VPN Tracker 365 Connection Safe has been designed from the ground up to offer highly secure cloud storage for VPN connections. In this paper, we will outline our security architecture and core security principles, so you know exactly how your connections are encrypted and stored. We believe transparency builds trust and security, and look forward to your feedback.

Connection Safe: Core Security Principles

End-to-End Encryption

Your Mac creates all the encryption keys locally. All encryption and decryption takes place locally on your Mac only. No unencrypted key or connection data ever leaves your Mac.

Server Ignorance

Our servers know nothing about your keys or connection data, they only store encrypted data chunks. Even if we ever wanted to decrypt your data, even if somebody was trying to force us to decrypt it, it is simply not possible for us.

Open Cryptographic Standards

We believe the extra scrutiny of open cryptographic standards is the best way to deliver highly-secure encryption. Wherever possible we also prefer open source implementations that have been proofread and tested by many cryptographic experts already, so there are no known implementation mistakes.

Multiple Encryption Layers

To prevent data leaks in scenarios where a currently unbreakable algorithm is broken in the future, we never rely on a single algorithm in the first place.

By mixing at least two algorithms at every cryptographic step, there's always a safety net you can rely on.

How Connection Safe Works

The first time you add a VPN connection to your Connection Safe, VPN Tracker will generate a secure random key, the "Master Key". The Master Key itself is then once encrypted by a key derived from a randomly generated "Recovery Key" and once by key derived from a user chosen password. Finally the connection is encrypted with the Master Key and the encrypted connection together with the encrypted Master Key are both uploaded to our servers.

The next time you sign in on a Mac, VPN Tracker retrieves the encrypted Master Key as well as the encrypted connection data. The user now has to provide the previously chosen password which is used to derive the decryption key required to decrypt the Master Key. Once the Master Key has been decrypted, it is used to decrypt the connection data.

> Note: If users choose to use the same password for Connection Safe and their equinux ID login credentials, the security principle of server ignorance is still preserved: equinux only stores a hashed and salted representation of user credentials and this hashing is a completely different algorithm than the ones used by the Connection Safe architecture.

Encryption Keys

Overview of Keys

Master Key

The Master Key is generated by mixing the output of two random sources together. One of them produces high quality random numbers, the other one is only pseudo-random. Relying on just one random source is always dangerous as it may turn out in the future that the source is more predictable (and thus less random) than assumed and this could weaken the cryptographic strength of the Master Key. When mixing multiple random sources together, as long as just one source provides good random numbers, the result will still be a good random number, no matter how predictable the other sources have been.

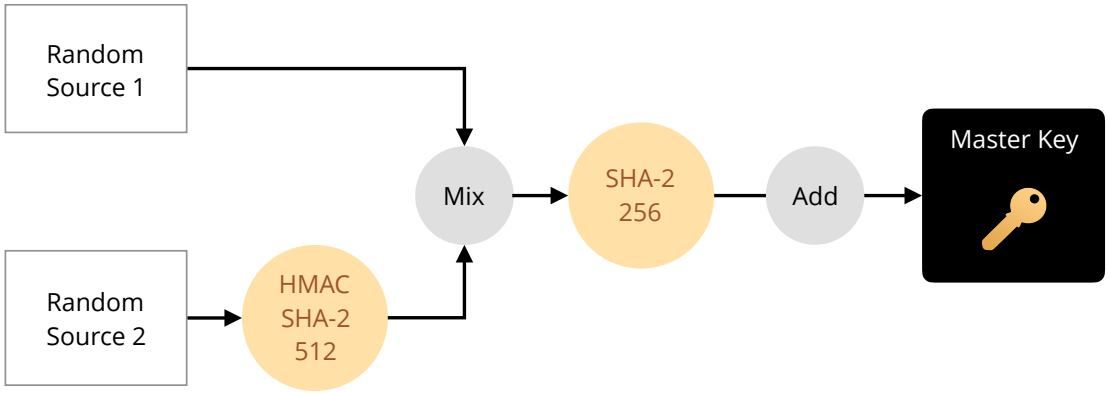


Fig 1. Master Key Generation

Recovery Key

The Recovery Key is a safety net for the case the user has forgotten the chosen password. As long as the user still has the Recovery Key available, it is possible to choose a new user password. It's the counterpart to the "I forgot my password" link you'd find on websites, as such a recovery is not possible for Connection Safe.

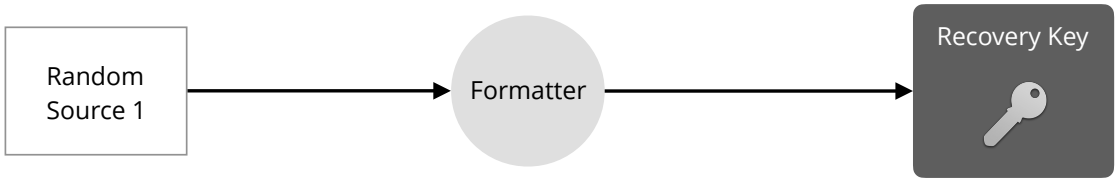


Fig 2. Recovery Key Generation

User Password

The User Password is any password chosen by the user, it protects access to the Master Key and thus also access to the connection. It can be the same password as the equinux ID password of the user or it can be an entirely different password.

Derived Key

Neither the Recovery Key, nor the User Password are ever used directly to encrypt any data, they are only used as input to a key derivation function. The output of this function is the Derived Key. See Key Derivation below.

Local Key Storage

The User Password can be stored in the user's macOS Keychain, or optionally can be entered each time the user accesses their connection vault.

Key Derivation

User chosen passwords are usually not very good cryptographic keys, thus prior to encrypting any data with them, it is always recommended to convert the passwords to keys

using a key derivation function. Such a function also has the advantages that it makes the whole decryption process more expensive to perform and this serves as an additional protection layer against brute force and dictionary attacks because an attacker can test a lot less combinations a minute as if no such function was being used.

Keys are derived using a combination of two key derivation functions:

PBKDF2 and Argon2

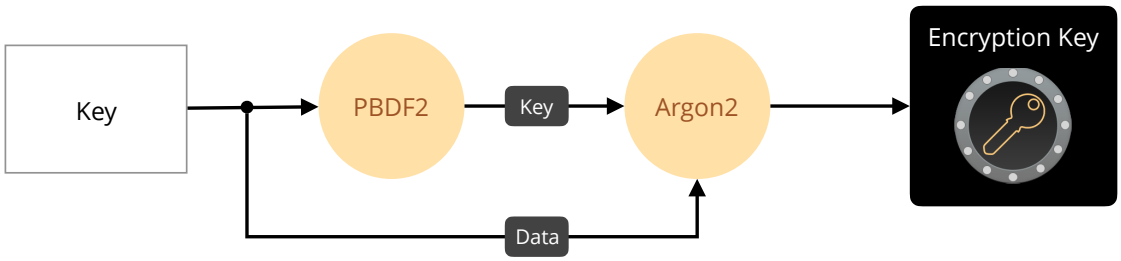


Fig 3. Encryption Key Derivation

PBKDF2 is the recommended key derivation function of the NIST (National Institute of Standards and Technology). It is widely used by server and client software for years and no weakness has ever been found. PBKDF2 has two parameters: a hash function and a number of rounds. The more rounds, the more expensive it is to derive the key and thus the harder an attack will become. We use 25'000 rounds, which is far above the average (as of today, most software uses less than 5'000 rounds). As a hash algorithm we use SHA2-512, which is the strongest algorithm that is known to be secure.

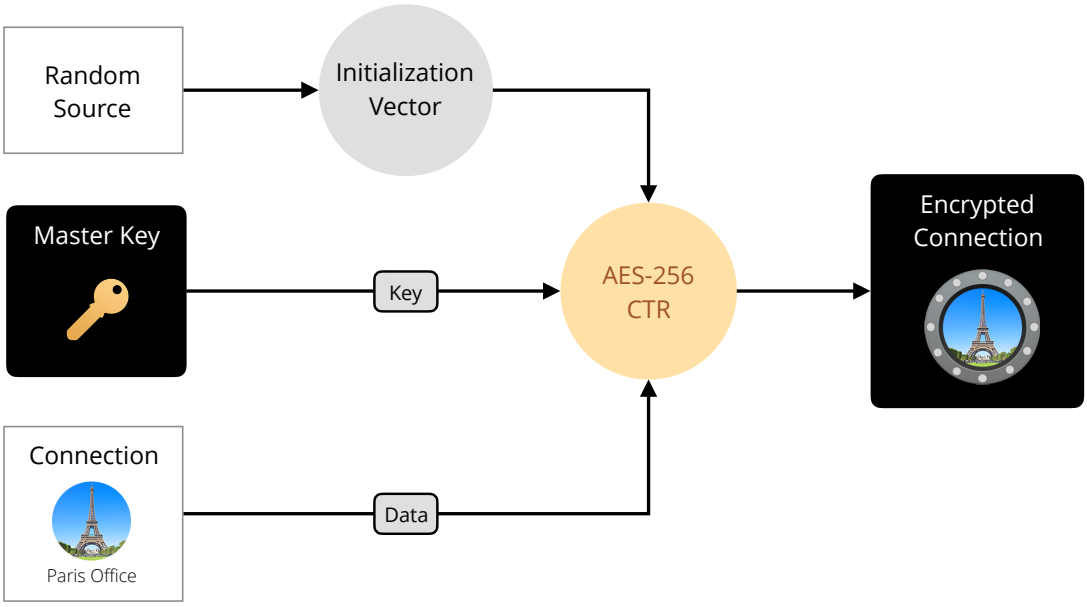
While PBKDF2 is basically proven to be secure, it has one weakness by design: All that is required to break it is a lot of computation power. Assuming a company would produce microchips that only can calculate PBKDF2 keys with SHA2-512 and nothing else, these chips would be relatively inexpensive to produce (they only need a small circuit and almost no memory) and by printing thousands of these onto a single circuit board, this board would be able to calculate PBKDF2 much faster than the fastest general purpose CPUs that are an order-of-magnitude more expensive. And of course nothing stops an attacker to combine hundreds of such boards together. Increasing the number of rounds will

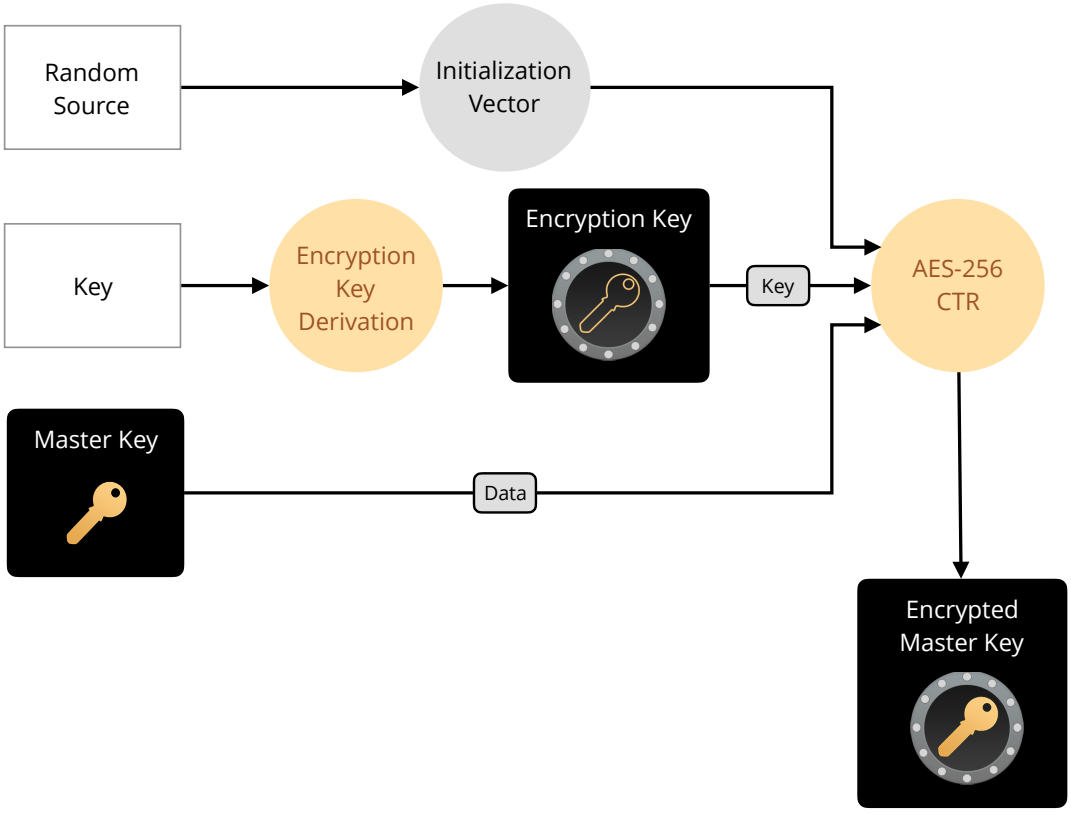
still help in that case but it's not possible to increase the number endlessly as that also slows down the encryption for the user.

Argon2 is a relatively new key derivation function. It is the winner of the Password Hashing Competition in 2015. It is defined by 4 parameters: The variant (Argon2d, Argon2i, Argon2id), a number of rounds, the amount of memory required, and the number of lanes. Just like PBKDF2, more rounds make they key harder to compute but additionally to that a certain amount of memory is required (and memory is expensive!) as well as a certain degree for parallelism. This makes it much more expensive to produce dedicated hardware that can be used to break Argon2. So Argon2 is by far the better key derivation function but it's also relatively new and thus it may contain weaknesses that just haven't been discovered yet, so only relying on Argon2 would be an unnecessary security risk.

Argon2d has been designed to hold up against massive brute force attacks and is usually used on servers, while Argon2i has been designed to withstand side-channel-attacks and is usually used on clients. We use Argon2id, which is a hybrid scheme of both variants that tries to combine the advantages of the other two schemes. We use 1'000 rounds, we require 1024 kiB of memory and use two lanes (so a CPU requires at least two cores to calculate the key efficiently).

Connection Encryption





Connection Safe for Teams

Connection Safe has been designed to be suitable for multi-user scenarios, where an administrator may require issue user-specific keys, in order to be able to revoke one at any time, without revoking the entire connection.

Currently multi-user deployment is not yet supported by my.vpntracker.com, but it is planned for a future release. Please contact us if you would like access to the Connection Safe for Teams beta.