Veil User Manual

Coda Hale

September 6, 2022

Veil is a public-key cryptosystem that provides confidentiality, authenticity, and integrity services for messages of arbitrary sizes and multiple receivers. Veil is implemented as a command line tool `veil`. This document describes its feature set and usage.

Contents

1 Installation

To install Veil, check out this repository and build it yourself:

```
git clone https://github.com/codahale/veil
cargo install
```

Because this is a cryptosystem designed by one person with no formal training and has not been audited, it will never be packaged conveniently. Cryptographic software is primarily used in high-risk environments where strong assurances of correctness, confidentiality, integrity, etc. are required, and `veil` does not provide those assurances. It's more of an art installation than a practical tool.

2 Shell Completion

`veil` can generate its own shell completion scripts for Bash, Elvish, Fish, Powershell, and Zsh:

```
veil complete zsh /usr/local/share/zsh/site-functions/
```

3 Creating A Private Key

To create a private key, use the `private-key` command:

```
veil private-key ./my-private-key
```

You'll be prompted for a passphrase, and `veil` will write the encrypted private key to `./my-private-key`. That's it. There's no user IDs, no key signing, no key servers, no banging on the keyboard to generate entropy.

4 Generating A Public Key

Now that you have a private key, you also have a public key to share with others:

```
$ veil public-key ./my-private-key

TkUWybv8fAvsHPhauPj7edUTVdCHuCFHazA6RjnvwJa
```

You can then give this public key to people, so they can send you encrypted messages.

5 Encrypting A Message

To encrypt a message, you need your private key, the receivers' public keys, and the message:

```
veil encrypt ./my-private-key \
    message.txt message.txt.veil \
    TkUWybv8fAvsHPhauPj7edUTVdCHuCFHazA6RjnvwJa \
    BfksdzSKbmcS2Suav16dmYE2WxifqauPRL6FZpJt1476 \
    --fakes 18 --padding 1234
```

This will create a file `message.txt.veil` which the owners of the two public keys can decrypt if they have your public key. It adds 18 fake receivers, so neither receiver really knows how many people you sent the message to. It also adds 1234 bytes of random padding, so someone monitoring your communications won't know how long the message really is.

6 Decrypting A Message

To decrypt a message, you'll need the key path of the public key the message was encrypted for, the encrypted message, and the sender's public key:

```
veil decrypt ./my-private-key reply.txt.veil reply.txt \
    TkUWybv8fAvsHPhauPj7edUTVdCHuCFHazA6RjnvwJa
```

This will decrypt and verify the message. If successful, you'll know that the owner of the public key encrypted that exact message for you. Otherwise, the message may not have been encrypted for you, it may not have been encrypted by that sender, or the encrypted message may have been tampered with.

7 Signing A Message

To sign a message, you'll just need the message:

```
$ veil sign ./my-private-key announcement.txt

2sXLDBeTwHuECPp7QjWKdLYB3M9oLkju...
```

You can then share `announcement.txt` and the signature and people will be able to verify that the message is from you and has not been modified.

8 Verifying A Message

To verify a signature of a message, you'll need the signer's public key, the message, and the signature:

```
veil verify TkUWybv8fAvsHPhauPj7edUTVdCHuCFHazA6RjnvwJa \
    announcement.txt \
    2sXLDBeTwHuECPp7QjWKdLYB3M9oLkju...
```

If the signature is from the given public key and the message hasn't been altered, `veil` will exit with a status of `0`.

9 Creating Message Digests

To create a digest of a message, you'll just need the message:

```
$ veil digest announcement.txt

5fQPsn8hoaVddFG26cWQ5QFdqxWtUPNaZ9zH2E6LYzFn
```

9.1 Including Metadata

The `digest` command accepts an optional sequence of metadata strings which are included in the calculation of the digest:

```
$ veil digest announcement.txt \
      --metadata 'announcement.txt' \
      --metadata 'made-with-veil'

F8s5aLxQJbGiEhWacUAe4nDCHVSEwycDavYFqe2TyND1
```

9.2 Message Authentication Codes

To create a MAC using a shared key, include the shared key as metadata:

```
$ veil digest announcement.txt \
      --metadata 'our special secret'

9UH6dDyYZ5XrYyqn9DQvuzp1zz9wtiaVfaAPvwyhTZhT
```