Preliminary

version 0.0.1
3 April 2018
Douglas A. Bebber (dab@i2pmail.org)
https://github.com/dougbebber/Monero-Crypto-Lock

Introduction

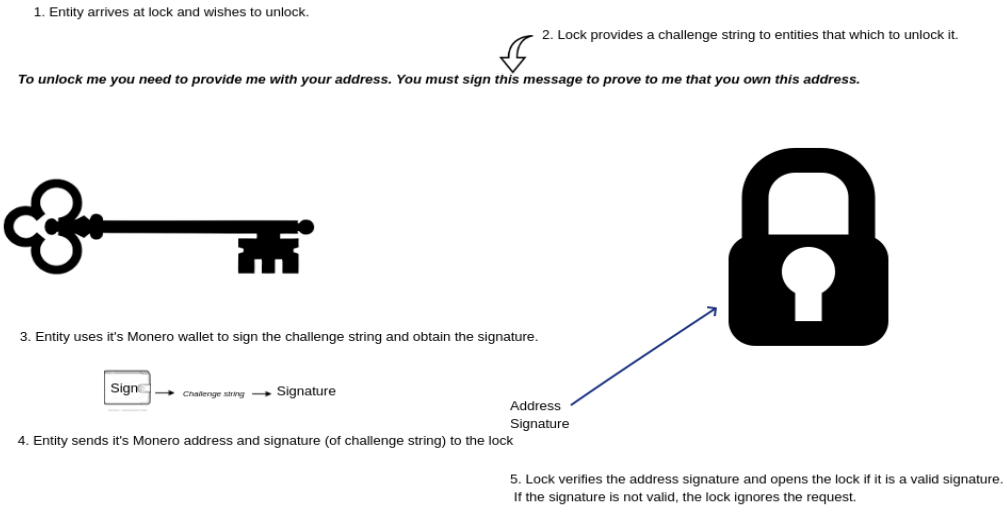
Monero Crypto-Lock is a point-to-point approach to public-key authentication using the Monero protocol. Monero Crypto-Lock enables secure and anonymous authentication utilizing Monero wallet commands.

How it Works

An entity requests secure authentication providing a Monero wallet address. The lock protecting access provides the requester with a challenge string that must be signed by the requester (in order to prove that the requester owns the Monero wallet address). The requester digitally signs the provided challenge string using it's Monero wallet address. The requester provides the lock with it's Monero address and the signature for verification.

The lock, uses it's Monero wallet to verify that the requester has a valid signature for the challenge string on behalf of the provided Monero wallet address. If the signature is "good" then the requester has proven that it owns the Monero wallet address and is granted access.

If the signature is not valid, the lock ignores the request.This is illustrated in the graphic below.



Dependencies
You need a Monero wallet, and two programs which are part of the Monero software distribution, *monerod*, and *monero-wallet-rpc*.

As of this writing, the monero-v0.11.1.0 distribution is being used.

It is recommend that you create a new wallet for use only with Monero Crypto-Lock. Don't use it to hold value (XMR) just for use with Monero Crypto-Lock related work. So create a new wallet.
If your new to Monero visit: https://getmonero.org/get-started/using/

Run the *monerod* (daemon) using the --offline parameter on the command line:

```
./monerod --offline
```

This will start the daemon, however, it will not attempt to synchronize with the Monero bockchain.

Then run the *monero-wallet-rpc* program using parameters similar to:

```
./monero-wallet-rpc --rpc-bind-port 18082 --disable-rpc-login --wallet-file test
```

You will need specify the name of your own wallet file (where the wallet named *test* is specified above).

You should now be ready to use the Monero Crypto-Lock.

Protocol Design Pattern

The protocol works like this:

1. Someone, or some-thing makes a request for authentication identifying themselves using a cryptocurrency address.
2. The entity responsible for authenticating such requests (the gatekeeper), submits a challenge (data string) to the requester.
3. The requester digitally signs the challenge (data string) with the private-key of the cryptocurrency address and submits the cryptocurrency address and digital signature to the gatekeeper.
4. The gatekeeper then verifies that the digital signature submitted is a valid signature of the cryptocurrency address for the challenge (data string). If it is valid, then the requester is authenticated. If it is not valid, authentication is denied.

Use Cases

One important use case is the elimination of user names and passwords for authentication to digital resources. Using Monero Crypto-Lock, Monero public addresses are the subjects being authenticated. Humans, as well as machines, can possess Monero public addresses that can be used for authentication to digital resources

Monero Crypto-Lock has many use cases beyond authentication. A few examples are summarized below:

- **Authorization** - Once authenticated, a specific Monero public address can be authorized for specified capabilities.
- **Encrypting & Decrypting Files** - Monero Crypto-Lock can be used with encryption technology such as gpg to encrypt and decrypt individual files. For example, Monero Crypto-Lock could be a user interface to file encryption where gpg is used to encrypt/decrypt individual files using the Monero public address as the gpg passphrase. An example of such an approach will be made available in the github repository.

- **Digital Lock-Box** - Monero Crypto-Lock can be used to lock and unlock secure, encrypted file system containers that can hold code and data providing a means to maintain tamper-proof content.
- **Software Secure Computing Environment** - Building on the Digital Lock Box use case, the digital lock-box would represent a *secure zone* where the contents (code and data) can be securely maintained as tamper-proof. Monero Crypto-Lock can be used to launch executable code, such that only authenticated, and authorized Monero public addresses are able to execute programs. Multiple executable programs can be run from the lock-box secure zone and interact with each other, as well as with programs external to the lock-box secure zone (non secure zone). Such digital lock-boxes could be remotely deployed over the network to support various use cases. Many examples of this pattern will be made available in the github repository.
- **Physical Security** - Monero Crypto-Lock could be used as a digital lock securing physical items such as doors. For example, an embedded SoC running Monero Crypto-Lock could be physically interfaced with hardware to lock/unlock a door.

And much more.

Many use case demonstration program source code and documentation will be made available in the github repository.