

# Cryptographic Method and Tooling for Multi-Factor Pseudonymous Attribution Proofs

---

By [Princess Pi](#)

## Table of Contents

1. [Abstract](#)
2. [Usecase](#)
3. [Writeup](#)
  1. [Proving Attribution](#)
4. [Archive Format](#)
5. [Safeguards](#)
6. [Screenscaps](#)
7. [Repo](#)
8. [Poni :3](#)

## Abstract

Spiffy way to sign an archive with an SSH key, plus add a seperate attribution passphrase/message so that users can verify attribution via signed message OR by revealing the passphrase/message, without compromising cryptographic durability.

## Usecase

For when you want to distribute files pseudonymously with two seperate options for proving attribution in whatever way you please.

## Writeup

A new ED25519 SSH key is generated for each round, prompts for an attribution passphrase/message which is hashed along with the inner 7Zip archive via SHA512 and stored in the outer layer of the 7Zip archive.

Internal 7Zip archive is signed with the SSH key, and the signature stored in the outer layer of the 7Zip archive.

SHA512 checksums are generated for all included files and stored in the outer layer of the 7Zip archive.

Included in the archive are bash shell scripts to verify SHA512 matches, SSH key signature match, and archive integrity on the fly, fast and easy. An additional script is used to test an attribution

passphrase/message easily.

The public key included can additionally be used to encrypt messages sent to author.

Final distributable is an optionally encrypted 7Zip archive, containing verification files, scripts, and instructions. Also is an inner 7Zip archive for the contents, where the messages/files/etc are stored.

## Proving Attribution

### 1. Signature Match

Users can match the signature with the provided public key. A shell script in the archive automates this, along with SHA512 matches, and archive integrity check.

Automated Verification: `./verify-everything.sh`

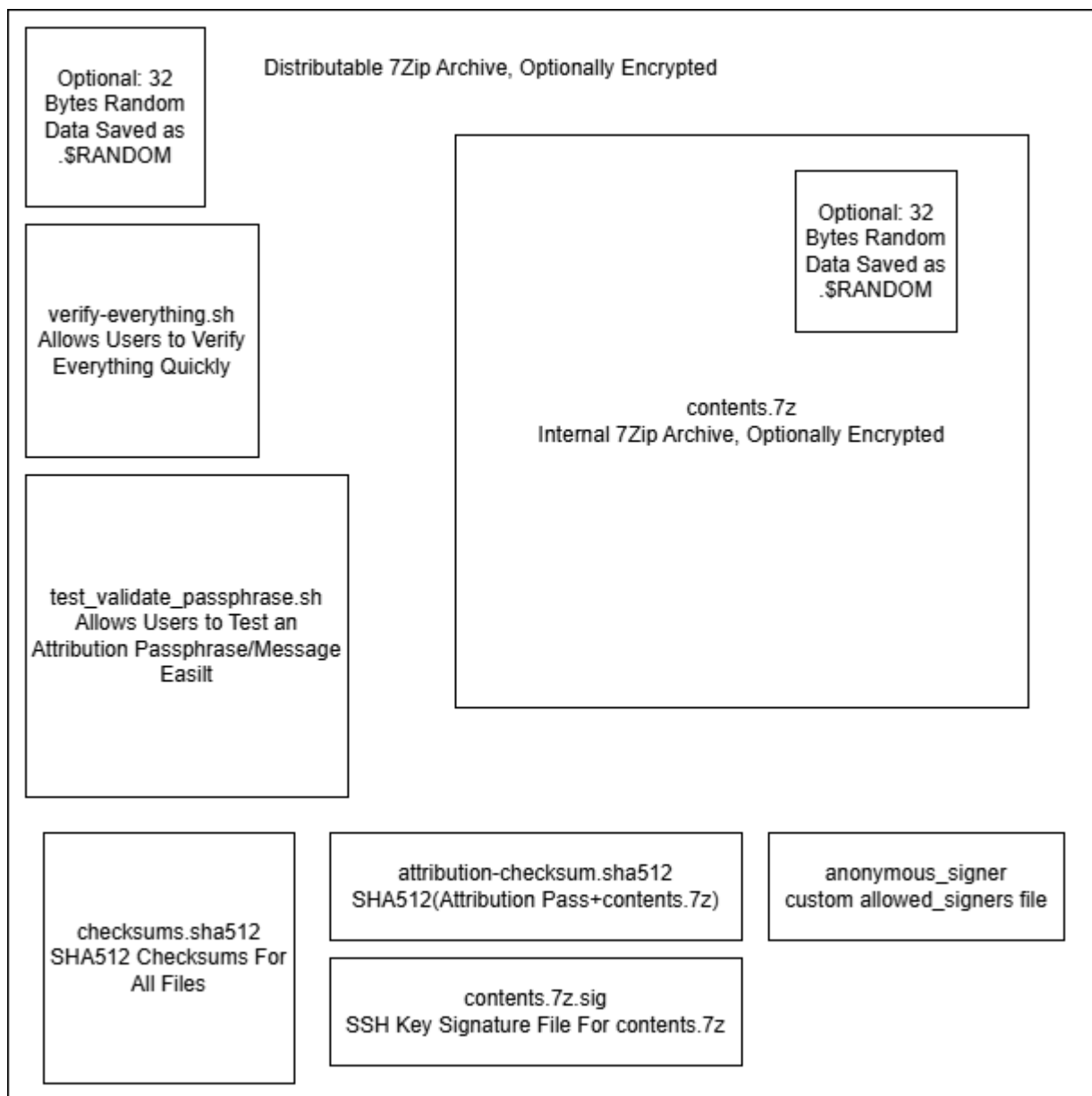
### 2. Attribution Passphrase/Message Reveal

Should the originator please, they can release the attribution passphrase/message, which can be tested by a shell script in the archive.

Method: `SHA512(<attribution pass><contents.7z>)`

Automated Check: `./test_validate_passphrase.sh`

## Archive Format Map



## Safeguards

1. All inputted passphrases are verified by match, length cracklib-check, entropy, and HavelBeenPwned.com's API.
  1. HIBP is only sent the first few bytes of the SHA1 hashed passphrase, and full hash is checked against what is returned.
2. 7Zip archives can optionally both have 32 bytes of securely generated random data added to each of them to break signatures.
3. 7Zip archives can optionally be encrypted.
  1. AES-CBC in 256-bit mode with PBKDF2+SHA256 hashing loop.
  2. Encryption configured in 7Zip to encrypt filenames, eliminating filename leaks.
4. Script is written as a schizophrenically safe shell script.
5. Newly used SSH keys and attribution passphrases/messages are stored in an encrypted 7Zip archive for security.

## Screenshots

Inside the distributable 7Zip archive, scripts work to verify signature, integrity, and attribution passphrase/message.

```
Kali-WSL
(princesspi@PrincessPi-Desk)~[~/Downloads/Encrypt-Share-Attribution/archives/out]
$ ./verify-everything.sh
Testing contents.7z integrity... OK!
Checking sha512 checksums... OK!
Checking signature against provided public key... OK!

(princesspi@PrincessPi-Desk)~[~/Downloads/Encrypt-Share-Attribution/archives/out]
$ ./test_validate_passphrase.sh
enter passphrase to test
Economist5-Pretty6-Ruckus7-Carat2-Dad7-Batting9
Attribution With Password Economist5-Pretty6-Ruckus7-Carat2-Dad7-Batting9: OK!

(princesspi@PrincessPi-Desk)~[~/Downloads/Encrypt-Share-Attribution/archives/out]
$
```

Script automates robustly making these archives.

```
Kali-WSL
(princesspi@PrincessPi-Desk)~[~/Downloads/Encrypt-Share-Attribution]
$ ./create-attributable-archive.sh
Setting up environment...
Autoshredding known artifacts...
OK!
Removing previous output directory...
OK!
Rebuilding output directory structure...
OK!
Writing placeholder README files...
OK!
OK!
Copying verification helpers...
OK!
OK!
OK!
OK!
Hardening archives...
OK!
```

## Repo

[@ thecoven.info](https://github.com/PrincessPi/Encrypt-Share-Attribution)

## Poni

Princess Pi is a very pretty pony :3

