**FACTOM**

**COMMUNITY**

**DOCUMENT FACTOMIZING**

**DOC 211**

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION | DATE | CHANGED BY | CHANGES |
| 0.1 | 2018-09-11 | Samuel Vanderwaal | Initial document adapted from Brian Deery’s previous instructions found [here](https://drive.google.com/open?id=1cJv1qt8trm5EyZPDs2DXLtfMb2j9skGFjOniWowmqQE) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Introduction

This document describes the process for factomizing community and governance documents. The goal of factomizing documents is to provide an auditable and verifiable trail for anyone to validate that the document they are looking at is the correct version and has not be altered since it was ratified and factomized.

1. Process

# Overview

The process for factomizing a document is a two-part process: we hash the document using SHA256 and then we sign it with a PGP key. This creates a digital fingerprint of the document but also validates it with a trusted signature ensuring that people can’t submit false hashes to the same Factom chain and attempt to fool people regarding the validity of a community or governance document.

The following instructions assume a Unix system (Linux or MacOS or a Windows bash-compatible terminal) and basic familiarity with the bash command line.

# Instructions

## Uploading to an Existing Chain

1. Save a checkpoint of the google doc and make a PDF
   1. In the Google Drive folder, go to File > Version history > Name current version
   2. Give it a name and include the version number. E.g.: “V1.0 - factomized”
   3. Download PDF version: File > Download as > PDF Document
2. Get the SHA256 hash of the document
   1. With the downloaded local copy of the document run this command in your terminal:

|  |
| --- |
| sha256sum NAME\_OF\_DOCUMENT.pdf > output-file-name.txt |

The output file name does not matter as it will not be part of the Factom entry. An example command is shown below:

|  |
| --- |
| sha256sum 'Minutes - Guide Meeting #20 - 2018-09-03.pdf' > meeting-20.txt |

1. Sign the created text document with a trusted PGP key.
   1. Current documents are signed by Samuel Vanderwaal with his PGP key: 7BC3E3CA1B50025F0E995D7290B92F0A97AD7089 which is posted publicly in multiple places.
   2. To sign the document run this command, using the same example document from above:

|  |
| --- |
| gpg --clearsign meeting-20.txt |

* 1. This creates a document entitled “meeting-20.txt.asc”.
  2. Open both .txt and .txt.asc documents to verify their contents. The first one will have the document hash and document name an the second will contain the PGP signed contents of the first document.

1. Add the signed hash to the Governance Tracking Chain
   1. Run the following command to upload the contents of the .txt.asc file as a new entry to the Governance Tracking Chain

|  |
| --- |
| cat meeting-20.txt.asc | factom-cli addentry -c 957477d3dd8564cb15fcbed377b50f3ada7ef52c8ac7dc6be25155121f96cda6 EC2N8fmeQQ5GJhM8Bd6xp2rsrfJLJVNBi4fHWM8ybTNYRg5Ei4rh |

replacing the Entry Credit address with one you control.

1. Upload the PDF to the archive folder on Google Drive
   1. Link to drive: <https://drive.google.com/drive/folders/1WmTBN17epeTuPh2FRMR67jlzY5VVDVmJ>
2. Add entry metadata and document hash to the appropriate table in [Doc 201](https://docs.google.com/document/d/1gFguzGSbQ1l3xmE_nx3hq_hqd49KEvsOpGdty98Pyqo/edit?usp=sharing).
   1. Once the entry has been included into a Factom block copy the entry hash, chain ID and document hash to the appropriate places in Doc 201.
   2. Fill out the rest of the table data.

## Creating a New Chain

1. To create a new chain run the following command replacing CHAIN\_NAME with an appropriate name and the entry credit address with one you control.

|  |
| --- |
| echo "This chain contains hashes and metadata about the Factom governance documents" | factom-cli addchain -n "CHAIN\_NAME" EC2... |

1. Save the created ChainID and include it in Doc 002 when adding entries.

## Viewing and Validating Entries

1. To view all entries on the governance document chain:

|  |
| --- |
| factom-cli get allentries 957477d3dd8564cb15fcbed377b50f3ada7ef52c8ac7dc6be25155121f96cda6 9d4c0a030165363b214ac5f50ac13b84aa5db6b24d056bf775c61fa4fbb526d9 |

1. To view a specific entry:

|  |
| --- |
| factom-cli get entry ENTRY\_HASH |

1. To validate the signature of the entry, import [Samuel’s public PGP key](https://pgp.mit.edu/pks/lookup?op=get&search=0x90B92F0A97AD7089) or from [here](https://factomize.com/forums/threads/samuel-vanderwaals-pgp-key.747/#post-4198).
2. Download the entry to a document:

|  |
| --- |
| factom-cli get entry bf30a4fb38a5a04897803f4bf3cd9450de0e45168a9ce1d0da5a615737ac4a9c > entry.txt.asc |

1. Verify with gpg:

|  |
| --- |
| gpg --verify entry.txt.asc |

Example output from `gpg --verify` command.

|  |
| --- |
| gpg: Signature made Tue 11 Sep 2018 08:44:56 AM AKDT gpg: using RSA key 7BC3E3CA1B50025F0E995D7290B92F0A97AD7089 gpg: Good signature from "Samuel Vanderwaal <samuel.vanderwaal@gmail.com>" |

1. Compare the hash of the file in the signed entry to manually running sha256sum against a downloaded version of the PDF.