Git signing with DIDs

Title	Git signing with DIDs
Status	PROJECT COMPLETED
Difficulty	LOW

Description

Hyperledger projects require their contributors to sign Git commits to attest to the Developer Certificate Origin (DCO). New developers have provided feedback that this process can be confusing and represents a barrier to contributing to the project.

Hyperledger Indy provides an identity system that can be used by developers to issue verifiable credentials attesting to the DCO. The Sovrin network is a public instance of Indy that would allow third parties to verify Indy credentials that use the network. Using Indy to sign Git commits would not only improve the contributor experience, but would also serve as a great real-world application of the Indy technology.

Additional Information

https://developercertificate.org/

https://github.com/hyperledger/indy-sdk/

https://github.com/infominer33/awesome-decentralized-id/tree/master/indy-sovrin-evernym

Learning Objectives

Successful completion of this project will involve learning skills such as:

- User experience research
- Solution design
- Software engineering
- Collaborating with on open source community
- Agile methodology
- Source control tools
- Digital identity solutions and verifiable credentials
- Standardization process for emerging protocols
- Cryptographic tools and primitives
- Technical writing and documentation
- Presentation and training

Expected Outcome

- A tool is contributed to the Indy SDK that allows developers to sign Git commits with the Hyperledger DCO using a credential issued on the Sovrin ledger.
- The tool has both unit and integration tests.
- The Indy SDK Getting Started Guide is updated to walk developers through the process of using the tool to obtain a Sovrin credential and use it to sign a contribution to the project.

Relation to Hyperledger

Immediately useful for Hyperledger Indy, but could make it easier to contribute to all Hyperledger projects.

Education Level

Enrolled in an undergraduate program in a technical subject.

Skills

Basic programming experience is required. It is also desirable to have familiarity with Git, Rust, and Python.

Future plans

As a result of this project, the intern will be familiar with Hyperledger Indy SDK and able to contribute in a variety of ways including improved documentation, testing, and feature development. A good follow on feature would be to integrate Indy credentials with GPG and SSH by exposing ed25519 keys to those tools.

Preferred Hours and Length of Internship

Either full time or part time is acceptable.

Mentor(s) Names and Contact Info

Richard Esplin, richard.esplin@evernym.com, esplinr, Evernym

Dave Huseby, dhuseby@linuxfoundation.org, dhuseby, Hyperledger

Mentee

Ibrahim El Rhezzali

Project Plan

Git integration with external signing tools

Currently git supports signing / verifying commits and tags using GPG only. The goal of this project is to make the git signing interface compatible with external signing tools and with DIDs (Distributed Identities) using programs such as bettersign for example.

This project will be the continuation of the work already done by David Huseby on the subject. His previous work is here:

- https://github.com/dhuseby/did-git-spec
- https://github.com/dhuseby/did-git-impl

This project's working fork:

- https://github.com/ibrahimel/did-git-impl

The main sections of the project are updating the user configuration and the command handling when a signing or verifying operation occur. The actions needed in each section can be listed below and will be evolving as the project evolves.

UPDATE: The request for proposal has been sent to the git mailing list and can be tracked here:

https://public-inbox.org/git/CACi-FhDeAZecXSM36zroty6kpf2BCWLS=0R+dUwuB96LqFKuTA@mail.gmail.com/T/#u

The commits that will be submitted as a patch can be found here:

https://github.com/ibrahimel/did-git-impl/commits/did-git-impl-signing-patch

After receiving feedback from the git community, we opted for a config based approch and a tool-agnostic signing interface. Below is the explained approach sent to the mailing list:

https://public-inbox.org/git/R3X1WzWH0sgOh85GuUmXwsTC6CPKysi4TRzN_BPecDVGr__ET2-mitZ2DZA0_bpKkzLRtnTtoomIWxZtL52_1XkihYBVBAuWMpSdwoboixY=@pm.me/T/#u

The user configuration would define the signing tool and other parameters as keys, identity and keychain. It would look like this:

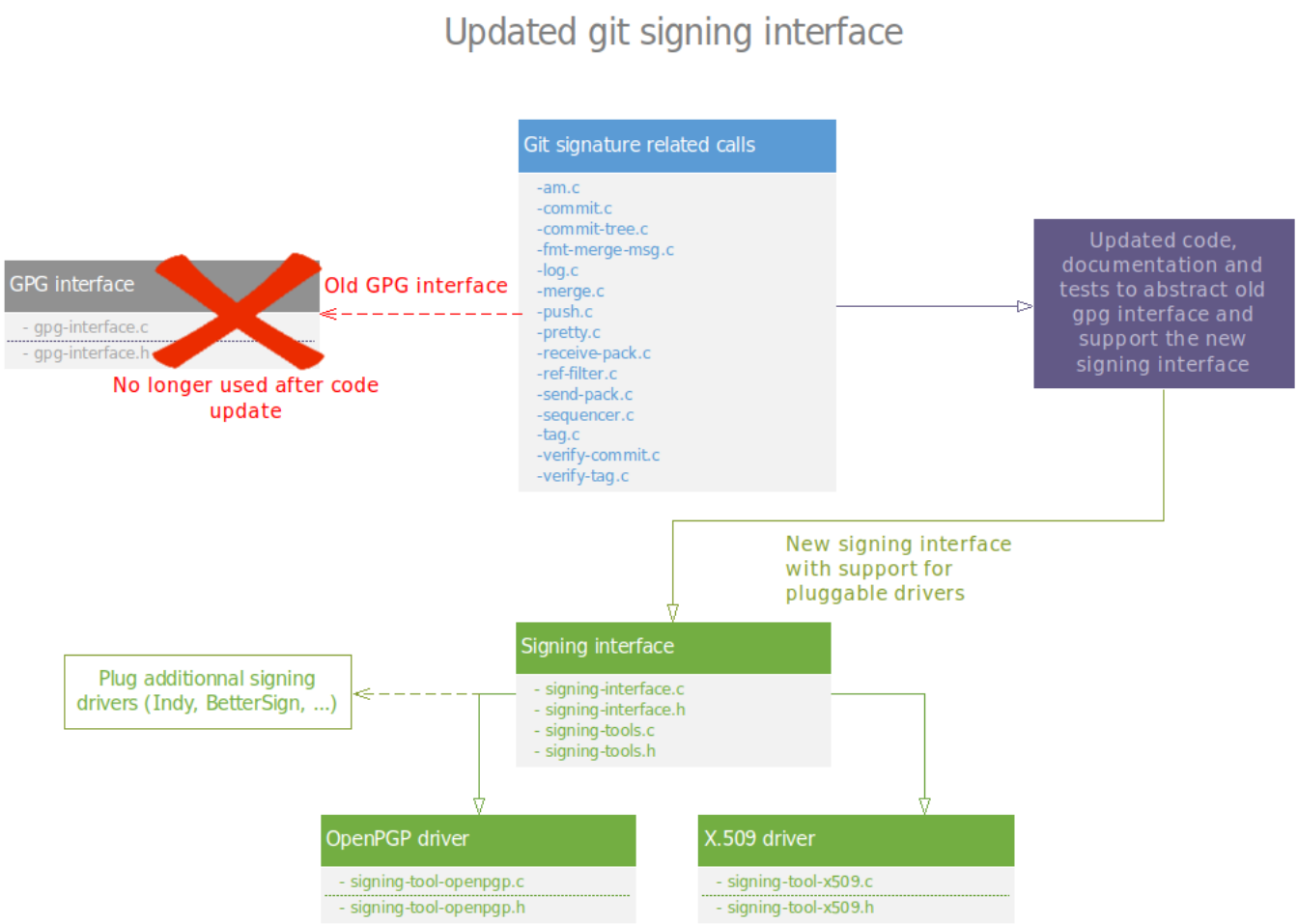
```
[signing]
  format = openpgp

[signing "openpgp"]
  program = /usr/bin/gpg
        keyring = "--keyring pubring.kbx --no-default-keyring"
        identity = "--local-user \"Jane Committer <jane@hackers.com>\""
        sign = "$program --sign --status-fd=2 --detach-sign --ascii"
        verify = $program --verify --status-fd=2"

[signing "openpgp.signature"]
        regex = "^-----BEGIN PGP SIGNATURE-----$[^-]*^-----END PGP SIGNATURE-----$"
        multiline = true
```

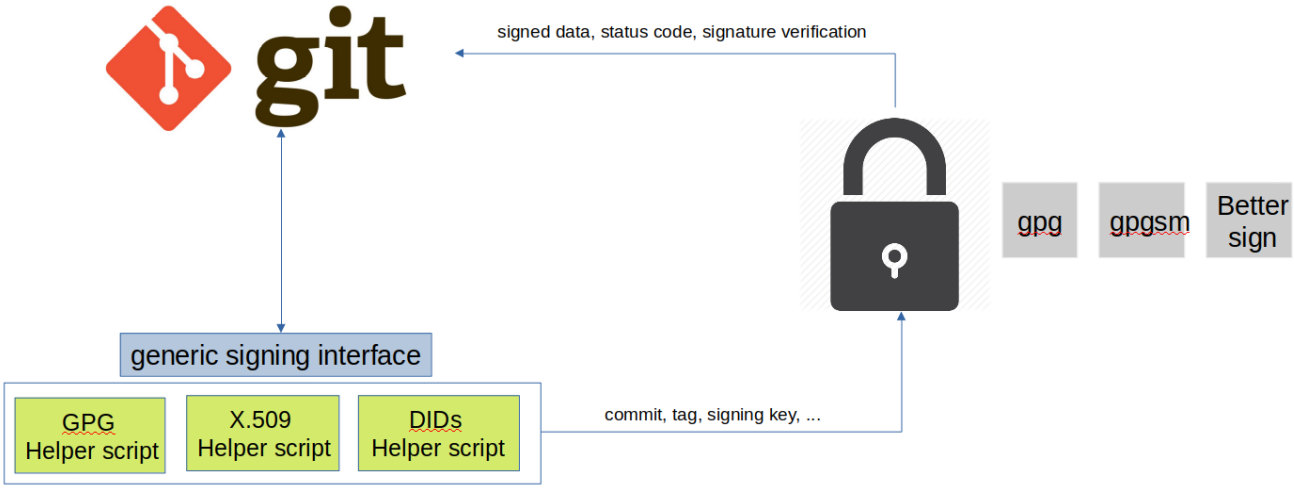
And the same goes for the command line:

```
git commit \
  --sign
        --signing-format=openpgp \
        --signing-openpgp-program="/usr/bin/gpg" \
        --signing-openpgp-keyring="--keyring pubring.kbx --no-default-keyring" \
        --signing-openpgp-identity="--local-user \"Jane Committer <jane@hackers.com>\"" \
        --signing-openpgp-sign="--sign --status-fd=2 --detach-sign --ascii"
```

As for tool that don't work natively with the signing interface, a support for helper bash or python scripts will be added.

Below in an illustrative model of the expected outcome:





Milestones

- [x] Submit project proposal (June 18 2019)
- [x] Review previous work (June 25)
- [x] RFC with the technical design document sent to git mailing list (July 15 2019)
- [x] Receive feedback (August 31)
- [x] RFC patchbomb to the mailing list (September 30 2019)
- [x] Second RFC to the mailing list (October 22 2019)
- [ ] Reveive Feedback (November 10 2019)
- [ ] Update code to support the config based approach (November 10 2019)
- [ ] Implement a Python Indy signing tool for DIDs (November 15 2019)
- [ ] Refine implementation and follow through to landing (November 15 2019)

Deliverables

- [x] Git documentation
- [x] Git signing plugin
- [x] Project description and model
- [ ] Helper signing script template
- [ ] Stretch deliverable: Indy signing tool
- [x] Project presentation
- [ ] Recorded presentation

Detailed Steps

Update documentation

- [ ] Git man page
- [x] Other documentation

Update user configuration handling

- [x] Change existing configuration keys to a more normalized new organization (`commit.gpgSign` to `commit.sign`, `gpg.program` to `signing.openpgp.program`, `gpg.<format>.program` to `signing.<format>.program`)
  - [x] `Debug the git_signing_config and the openpgp_config/x509_config functions and verify that all unit tests pass.`
  - [x] `Add new unit tests to verify that all deprecated aliases work as expected and produce good warnings.`
  - [x] `Add new unit tests to verify that all new configuration keys work.`

- [ ] Implement a config based signing interface

Add Helper scripts template

- [ ] Bash template
- [ ] Python template

Testing

- [x] Integration testing
- [ ] Manual testing git with new functionalities and signing programs

Back-burner Tasks

These are secondary tasks to do while waiting for feedback or assistance, or finished early:

- [x] Learn Rust
- [x] Learn about Indy SDK, Indy CLI, and Verifiable Credentials
- [ ] Signing through the Indy CLI
- [ ] Contribute to BetterSign