Git Commit Signing with DID's, Part Deux

Title	Git commit signing with DID's – The second signing
Status	COMPLETED
Difficulty	HIGH

Description

In the summer of 2019, Hyperledger ran a mentorship geared towards getting the Git version control tool to understand and use cryptographic credentials in decentralized identity (DID) documents to sign and verify commits. The root cause analysis led the project in the direction of creating a patch set for Git that enabled Git to use any signing tool more easily rather than just its existing support for GnuPG. Work is still ongoing to get those patches landed into Git and out in the wild. This mentorship anticipates the completion of that work and extends the previous work with the construction of a software application that can be called by Git to sign/verify commits using credentials stored in DID docs.

Additional Information

This mentorship is focused on writing an application using the Rust programming language. Please don't apply if you don't already have significant experience programming in any language and are at least willing to put in the time to learn Rust. You will be mentored by experienced Rust developers in the Hyperledger community to help with getting comfortable with Rust.

Learning Objectives

- First and foremost the mentee will learn how to be a positive collaborator and contributor in an active open source project.
- Learn how to work within the Hyperledger open source ecosystem and culture.
- Apply computer science skills to design, develop, and deploy a new cryptographic signing and verification tool that understands DID docs.
- Gain a better understanding of programming in Rust and applying crypto libraries to solve cryptographic problems.

Expected Outcome

- At least a 0.x beta version of a new cryptographic signing tool.
- A report on the design and development process that captures any remaining work.

Relation to Hyperledger

This is an important move forward for self-sovereign identity and Hyperledger projects that support that form of identity. This affects Hyperledger Indy, Hyperledger Aries, and Hyperledger Ursa. It is likely that this project will manifest itself as a Hyperledger lab at first and potentially a Hyperledger project as it matures.

Education Level

The ideal mentee is a university student or a developer with one or two years of experience with a solid background in using cryptography libraries. It is helpful if you already have experience programming with Rust but it isn't required as you will be mentored by experienced Rust programmers.

Skills

- Application design and implementation experience.
- Programming experience in Python, C, C++, or Rust.
- Good communication skills and a willingness to participate in broader open source communities (e.g. email and chat)

Future plans

This signing tool will likely be used as the signing tool for signing/verifying commits in all Hyperledger projects.

Preferred Hours and Length of Internship

Full-time or part-time

Mentor(s) Names and Contact Info

David Huseby, dhuseby@linuxfoundation.org, dhuseby on chat.hyperledger.org

Mentee Name and Contact Info

Jimit Bhalavat, jimitbhalavat144@gmail.com

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

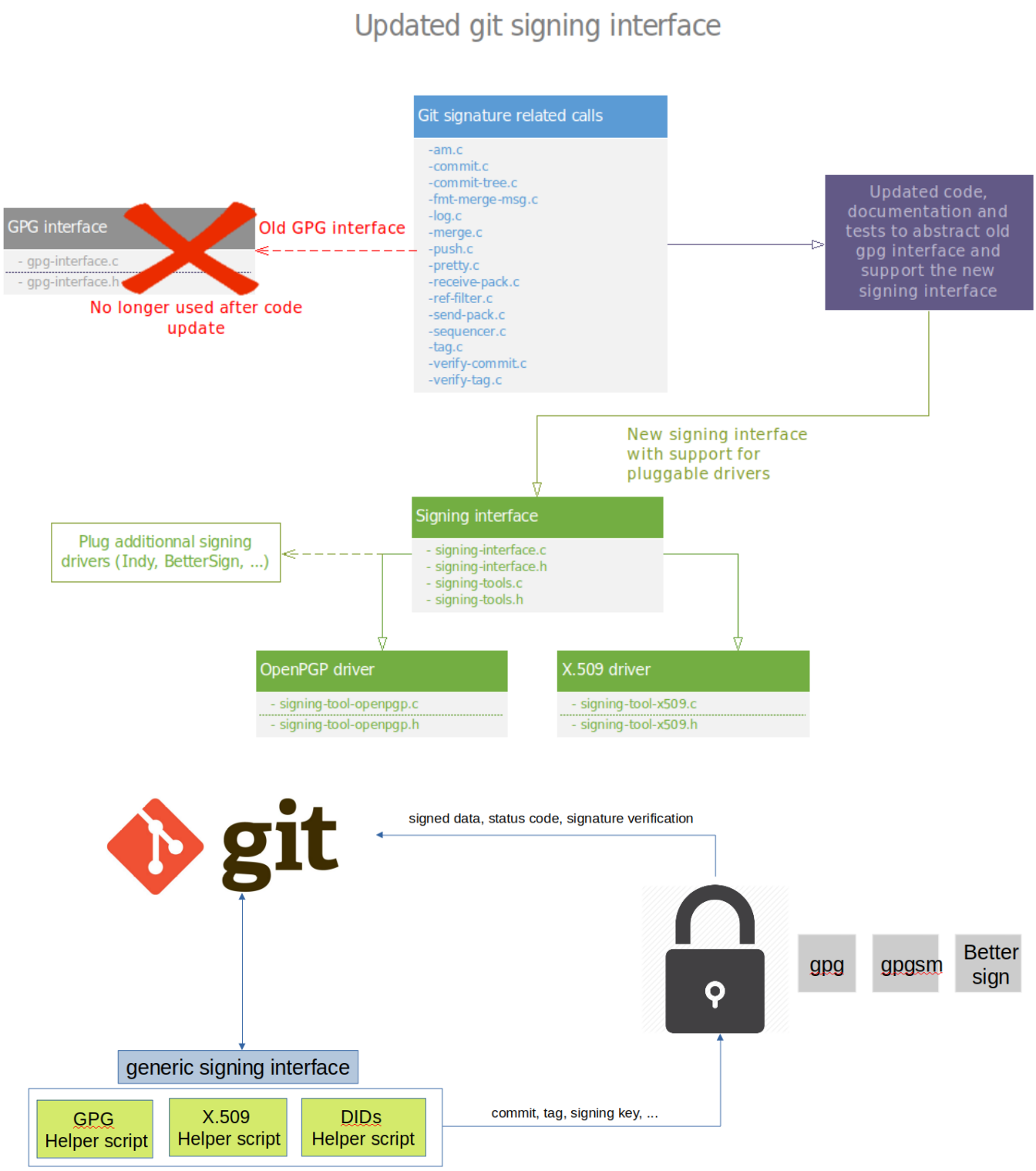
The main sections of the project are updating the user configuration and the command handling when signing or verifying operation occurs. The actions needed in each section can be listed below and will be evolving as the project evolves.

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

NEW: Collaborate with Google FIDO Team to land the patches successfully. Attempt to reconnect with the community.

Below is an illustrative model of the expected outcome:



Milestones

- [x] Submit project proposal (June 15, 2020)
- [x] Review 2019 patches and make changes to the new branch accordingly
- [x] This will enable Git to use any signing tool more easily rather than just its existing support for GnuPG.
- [x] Create a Technical Design Proposal
- [x] Edit the Design Proposal with the design for the future
- [x]

- [x] Create a Technical Design Document (deprecated in lieu of the protocol design)
- [x] Edit the Technical Design Document with the flow of code through git for how signing works currently
- [x] Create a Technical Design Document (i.e Assuan Protocol)
- [x] Create a Universal Cryptographic Signing for Git presentation
- [ ] Update code to support the protocol based approach

Detailed Steps

Testing

- [ ] 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:

- [ ] Learn Rust
- [x] Continue developing proficiency in C and C++

Project Results

Deliverables

- [x] Technical Design Document (Deprecated in lieu of the protocol design)
- [x] Technical Design Proposal (Deprecated in lieu of the protocol design)
- [x] Technical Design Document (Assuan Protocol)
- [x] Universal Cryptographic Signing for Git

Final Report

Lightening Talk Recording


Git Commit Sign..., Part Deux.mp4