Hyperledger Mentorship Project Presentation

November 2020
Universal Cryptographic Signing for Git

Introduction

- **Name**: Jimit Bhalavat
- **Location**: Fort Collins, CO, USA
- **University**: Colorado State University, CO, USA
- **Mentor**: David Huseby
- **Hyperledger Project**: Git Commit Signing
Universal Cryptographic Signing for Git

› Project Description:

› Add to Git the ability to support any signing/verification tools besides GPG
› Create the ability to support signed patches submitted by email
› Simplify and unify config for signing/verification
› Normalize command line switches
› Maintain backwards compatibility
Universal Cryptographic Signing for Git

› Project Approach/Objectives:

› Abstract away signing/verifying tool specifics so that Git can use any tool, not just GPG
› Remove GPG specific code, config, and naming
› Reorganize and normalize config settings to support multiple tools
› Make signatures stored in Git objects
› Preserve backward compatibility
› Deprecate, but maintain support for old config and command line switches; with warnings
Universal Cryptographic Signing for Git

Project Deliverables:

- Goals of the project
- Currently how signing and verification works
- Approach of the project, options going forward

Deliverable 2: Technical Design Proposal:
- Detailed description of what we were trying to achieve this year
- Layout of the Universal Cryptographic Signing

Deliverable 3: Universal Cryptographic Signing for Git:
- In-depth description of Protocol-based approach for Git Signing and Verification
Universal Cryptographic Signing for Git

› **Project Execution & Accomplishments:**

› Finished writing a Technical Design Proposal that will help the project to continue. The Technical Design Document describes the approach by Ibrahim (2019 Hyperledger Mentee) and why the patches were rejected in 2019. The document also describes the approach that we took this year and where the project is headed. We were able to form an alliance with OpenSSF Developer ID WG and Google.
› We were unable to land patches into Git.
› The fact that we were able to figure out the best approach moving forward and making Git universally cryptographic signing in upcoming months.
› In the beginning of the project, I found that Git signing is complex and requires effort to go through the code and understand what the code is actually accomplishing.
Universal Cryptographic Signing for Git

Options going forward:

- **Standard Pipe-Fork Interface and Config Based** (not preferred)
  - Complicated config for specifying how each and every tool is to be called.
  - This solution would have to deal with all of the legacy complication of signing tools.
- **Protocol-based Approach Inspired by GPGme** (i.e. Assuan Protocol) **(preferred)**
  - Cleanly separates Git from tool-specific details and code.
  - Makes `.gitconfig` for each tool simple.
  - Stores verification options - and optionally the pubkey - inside of the Git object.
  - Allows for “git format-patch” and “git am” to support signed email patches.
  - Externalizes signature creation and verification.
Universal Cryptographic Signing for Git

Why Protocol-based approach?:

› Cleanly separates Git form the details of external signing and verification tools.
› Configuration settings for each tool is much simpler.
› Eliminates the need for signature format specific parsing code to be present inside of Git.
› Allows for more Git subcommands to utilize external signing and verification tools.
› With a protocol design, Git is separated from the underlying communication mechanism.
› Support for external cryptographic hardware, network attached hardware security modules (HSMs) and other enterprise identity tools such as LDAP.
Universal Cryptographic Signing for Git

Project Output or Results:

- Technical Design Proposal (Deprecated in lieu of the protocol design)
- Technical Design Document (Deprecated in lieu of the protocol design)
- Universal Cryptographic Signing for Git
- Git Forked Repo (branch: gpg-project)
Universal Cryptographic Signing for Git

Insights Gained:

Contributing to an open-source project has been a new experience and I gained some tremendous insights about how real-world projects work.

The advice I would like to pass on to other mentees is that make sure you have good communication with the mentors. They are always there to help you and their main goal is to see you successful.

Experience with git and how Git signing works would have helped me learn the process better.
THANK YOU!