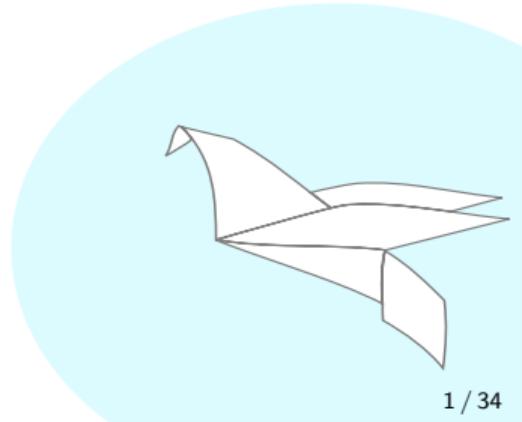


# Sequoia git: Making Signed Commits Matter

Neal H. Walfield <neal@sequoia-pgp.org>  
8F17777118A33DDA9BA48E62AACB3243630052D9

Security Dev Room, FOSDEM 2026

January 31, 2026



# Version Control Systems

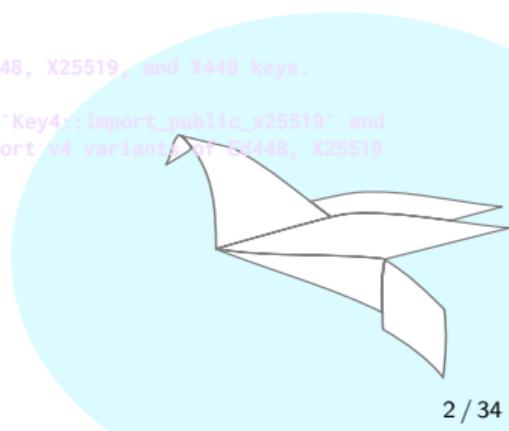
- Version Control Systems track:

- Changes to code
- Authorship
- Other meta-data
- Commit message

```
$ git diff
diff --git a/openpgp/NEWS b/openpgp/NEWS
index 2395a7a59..9cf2dc45b 100644
--- a/openpgp/NEWS
+++ b/openpgp/NEWS
@@ -7,6 +7,17 @@
...
$ git log -1
commit 46b1ccdf72edb4eddc7e73e33f71cf6fd9901dc (HEAD -> main, origin/main)
Author: Neal H. Walfield <neal@sequoia-pgp.org>
Date:   Thu Sep 11 10:22:53 2025 +0200

openpgp: Add support for importing v4 Ed448, X25519, and X448 keys.
```

- Add `Key4::import\_public\_ed448`, `Key4::import\_public\_x25519` and `Key4::import\_public\_x448` to import v4 variants of Ed448, X25519 and X448 keys.



# Version Control Systems

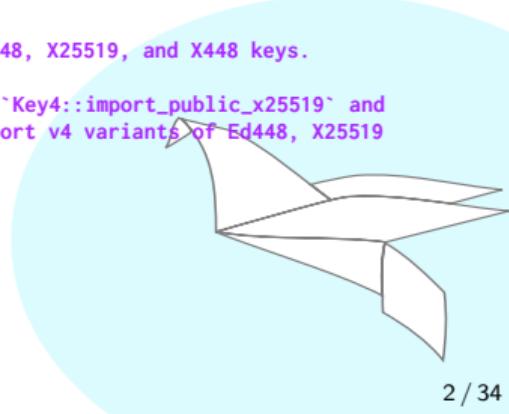
- Version Control Systems track:

- Changes to code ■
- Authorship ■
- Other meta-data ■
- Commit message ■

```
$ git diff
diff --git a/openpgp/NEWS b/openpgp/NEWS
index 2395a7a59..9cf2dc45b 100644
--- a/openpgp/NEWS
+++ b/openpgp/NEWS
@@ -7,6 +7,17 @@
...
$ git log -1
commit 46b1ccdf72edb4eddc7e73e33f71cf6fd9901dc (HEAD -> main, origin/main)
Author: Neal H. Walfield <neal@sequoia-pgp.org>
Date:   Thu Sep 11 10:22:53 2025 +0200
```

openpgp: Add support for importing v4 Ed448, X25519, and X448 keys.

- Add `Key4::import\_public\_ed448`, `Key4::import\_public\_x25519` and `Key4::import\_public\_x448` to import v4 variants of Ed448, X25519 and X448 keys.



# Impersonation

- Author can be faked

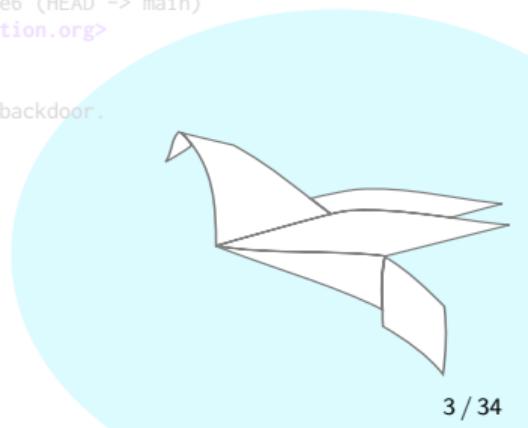
- Meta-data is set by the author
- Including the author's name

```
$ git config user.email torvalds@linux-foundation.org
$ git config user.name 'Linus Torvalds'

$ emacs openpgp/build.rs
$ git add openpgp/build.rs

$ git commit -m 'Improve build. Definitely does not add a backdoor.'
[main ee8f3108d] Improve build. Definitely does not add a backdoor.
 1 file changed, 1 insertion(+)
$ git log -1
commit ee8f3108d4c36fa332a26477b8de5c3f2e3f17e6 (HEAD -> main)
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed Jan 28 13:44:42 2026 +0100

  Improve build. Definitely does not add a backdoor.
```



# Impersonation

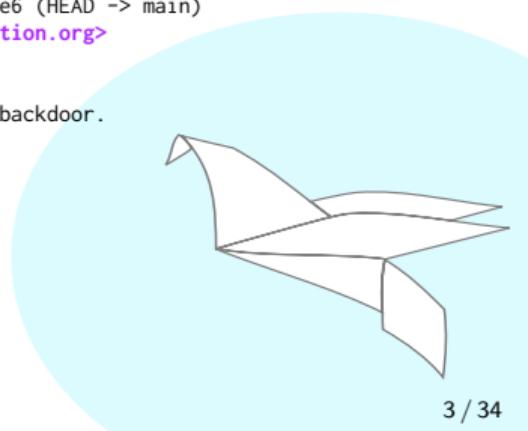
- Author can be faked
  - Meta-data is set by the author
  - Including the author's name

```
$ git config user.email torvalds@linux-foundation.org
$ git config user.name 'Linus Torvalds'

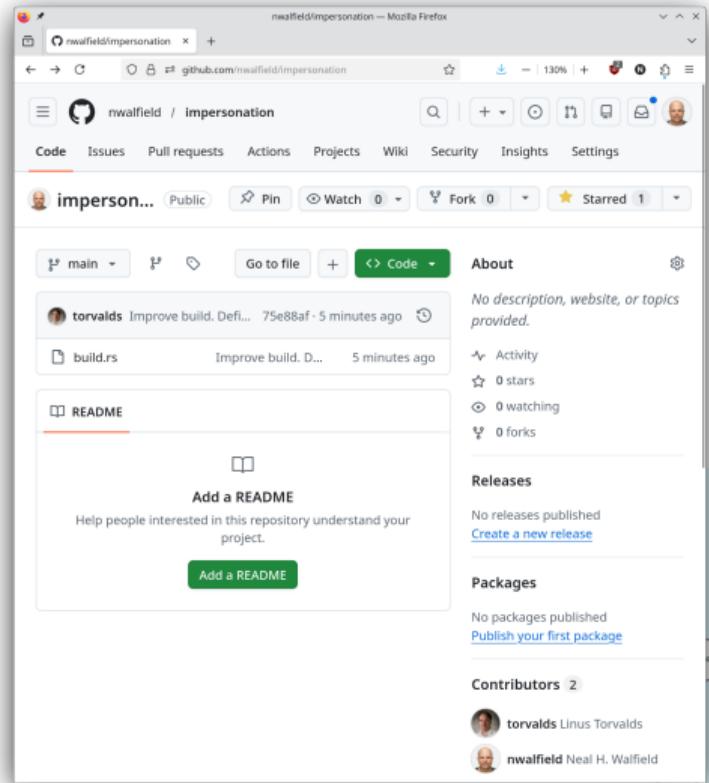
$ emacs openpgp/build.rs
$ git add openpgp/build.rs

$ git commit -m 'Improve build. Definitely does not add a backdoor.'
[main ee8f3108d] Improve build. Definitely does not add a backdoor.
 1 file changed, 1 insertion(+)
$ git log -1
commit ee8f3108d4c36fa332a26477b8de5c3f2e3f17e6 (HEAD -> main)
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed Jan 28 13:44:42 2026 +0100

  Improve build. Definitely does not add a backdoor.
```



- GitHub knows who pushed a commit
  - The person has to be logged in to push
- But the committer and the author are different
  - We don't want to overwrite it
  - We want attribution



# Why care about Impersonation?

- Scenario #1

- Project hosted on a forge
- Forge adds a commit inserting a backdoor
- How do we catch it?



<https://www.flickr.com/photos/31797838@N00/26003405317> by  
Alexandre Dulaunoy, CC BY-SA 2.0.

# Why care about Impersonation?

- Scenario #2
  - Project hosted on forge
  - Account compromised
  - Attacker adds a commit inserting a backdoor
  - How do we prevent it?



<https://www.flickr.com/photos/31797838@N00/26003405317> by Alexandre Dulaunoy, CC BY-SA 2.0.

# Why care about Impersonation?

- Scenario #3
  - Attacker submits low-quality commits
  - Reputation damaged
  - How do we catch it?



<https://www.flickr.com/photos/40702206@N00/367078204> by  
Pierre Tourigny, CC BY 2.0.

# Preventing Impersonations

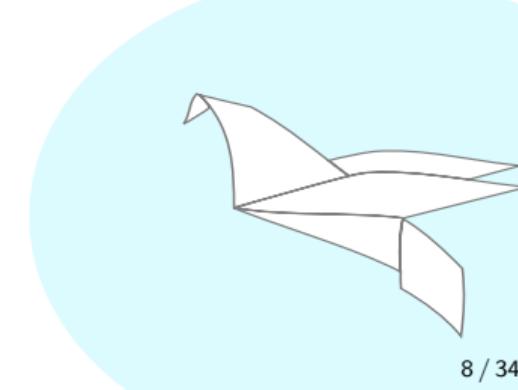
- Digitally sign commits
  - Cryptographic proof! ■

```
$ git log --show-signature
commit 75e88afb252fe350ea23d8d61502ec7f8bd5f747 (HEAD -> main, origin/main)
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed Jan 28 14:07:38 2026 +0100
```

Improve build. Definitely not a backdoor.

```
commit c296a9b13a66eb6acc7ca99d0c112faeac8ec1df
gpg: Signature made Wed Jan 28 14:06:14 2026 +01:00
gpg:                               using EDDSA key 7FAF6ED7238143557BDF7ED26863C9AD5B4D22D3
gpg: Good signature from "Neal H. Walfield (Code Signing Key) <neal@pep.foundation>" [ultimate]
gpg:                               "Neal H. Walfield (Code Signing Key) <neal@sequoia-pgp.org>"
Author: Neal H. Walfield <neal@walfield.org>
Date:   Wed Jan 28 14:06:14 2026 +0100
```

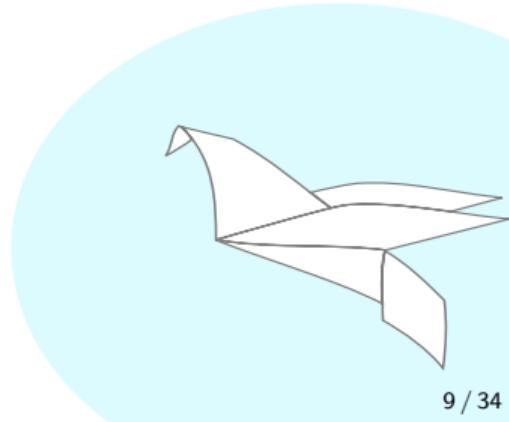
Lots of impressive code.



But...

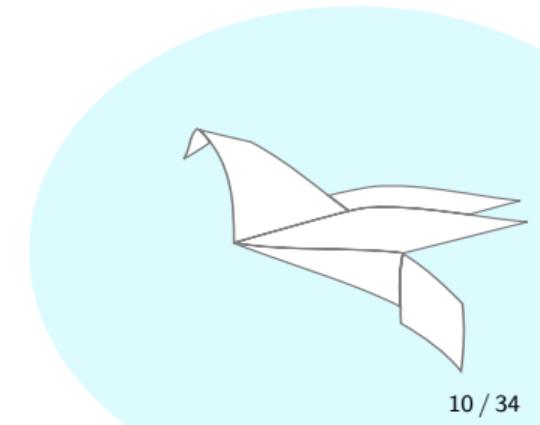
- Nothing stops anyone from making a certificate with a given user ID

```
$ sq key generate --userid 'Linus Torvalds <torvalds@linux-foundation.org>' --own-key
Enter a password to protect the new certificate (press enter to not use a password):
Repeat the password:
- B43099ECCDD9A6E62C7E75510B4ABC607AC2A515
  Linus Torvalds <torvalds@linux-foundation.org>
- certification created
...
```



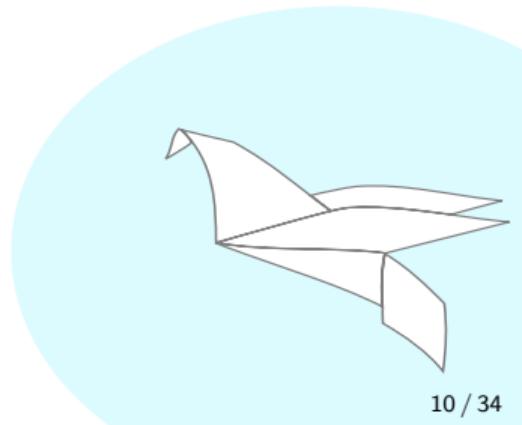
# Need Authentication

- Is B43099ECCDD9A6E62C7E75510B4ABC607AC2A515 really Linus' certificate?
- (No, I just generated it.)



## Need Authentication

- Is B43099ECCDD9A6E62C7E75510B4ABC607AC2A515 really Linus' certificate?
- (No, I just generated it.)



# Authenticating Certificates

- Talk to the developer
- Key signing party
- Central authority
  - Verifying key server: <https://keys.openpgp.org>
  - Linux kernel developer keyring
  - distribution-gpg-keys
  - <https://github.com/user.gpg>
- Linus' certificate is:  
ABAF11C65A2970B130ABE3C479BE3E4300411886



Marta Marczykowska-Górecka from Qubes OS at FOSS Backstage Design, 2025.

# Authenticating Certificates

- Talk to the developer
- Key signing party
- Central authority
  - Verifying key server: <https://keys.openpgp.org>
  - Linux kernel developer keyring
  - distribution-gpg-keys
  - <https://github.com/user.gpg>
- Linus' certificate is:  
ABAF11C65A2970B130ABE3C479BE3E4300411886



Marta Marczykowska-Górecka from Qubes OS at FOSS Backstage Design, 2025.

# Authenticating a Project

- Check that commits are signed

```
$ git log --show-signature
commit 859b3ce7248806e331bd1caa8d07cdc7ecb7f3d6 (HEAD -> main)
gpg: Signature made Wed Jan 28 17:27:14 2026 +01:00
gpg:           using EDDSA key D09FF325B8EFC0D5B4C59BAB92910E29B0D031323
gpg: Good signature from "Linus Torvalds <torvalds@linux-foundation.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:           There is no indication that the signature belongs to the owner.
Primary key fingerprint: ACBB D3A8 19D1 2CCD C50E C25F F5FE 8250 A9BF F91F
Subkey fingerprint: D09F F325 B8EF CD5B 4C59 BAB9 2910 E29B 0D03 1323
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed Jan 28 14:07:38 2026 +0100
```

Improve build. Definitely not a backdoor.

```
commit c296a9b13a66eb6acc7ca99d0c112faeac8ec1df
gpg: Signature made Wed Jan 28 14:06:14 2026 +01:00
gpg:           using EDDSA key 7FAF6ED7238143557BDF7ED26863C9AD5B4D22D3
gpg: Good signature from "Neal H. Walfield (Code Signing Key) <neal@pep.foundation>" [ultimate]
gpg:           "Neal H. Walfield (Code Signing Key) <neal@sequoia-pgp.org>"
Author: Neal H. Walfield <neal@walfield.org>
Date:   Wed Jan 28 14:06:14 2026 +0100
```

Lots of impressive code.



# Authenticating a Project

- Check that commits are signed *by the expected certificates*

```
$ git log --show-signature
commit 859b3ce7248806e331bd1caa8d07cdc7ecb7f3d6 (HEAD -> main)
gpg: Signature made Wed Jan 28 17:27:14 2026 +01:00
gpg:           using EDDSA key D09FF325B8EFC0D5B4C59BAB92910E29B0D031323
gpg: Good signature from "Linus Torvalds <torvalds@linux-foundation.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:           There is no indication that the signature belongs to the owner.
Primary key fingerprint: ACBB D3A8 19D1 2CCD C50E C25F F5FE 8250 A9BF F91F
Subkey fingerprint: D09F F325 B8EF CD5B 4C59 BAB9 2910 E29B 0D03 1323
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed Jan 28 14:07:38 2026 +0100
```

Improve build. Definitely not a backdoor.

```
commit c296a9b13a66eb6acc7ca99d0c112faeac8ec1df
gpg: Signature made Wed Jan 28 14:06:14 2026 +01:00
gpg:           using EDDSA key 7FAF6ED7238143557BDF7ED26863C9AD5B4D22D3
gpg: Good signature from "Neal H. Walfield (Code Signing Key) <neal@pep.foundation>" [ultimate]
gpg:           "Neal H. Walfield (Code Signing Key) <neal@sequoia-pgp.org>"
Author: Neal H. Walfield <neal@walfield.org>
Date:   Wed Jan 28 14:06:14 2026 +0100
```

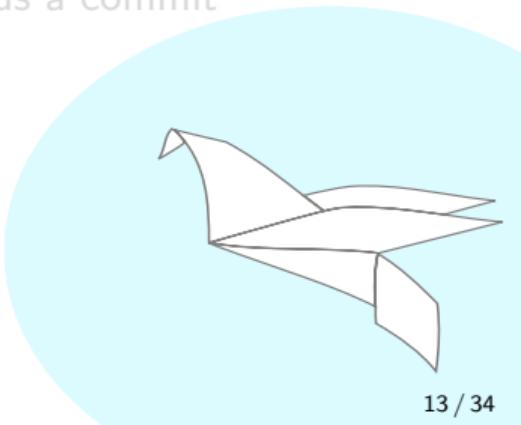
Lots of impressive code.



# How to Authenticate a Project

- **Curate a list of contributors and their certificates**
- Track when contributors join and leave project
  - Authenticate old commits of retired contributors
  - Reject new commits of retired contributors
- Complicated much?
  - Maintainer: Need tooling
  - Third-party: Infeasible, but can rely on maintainer

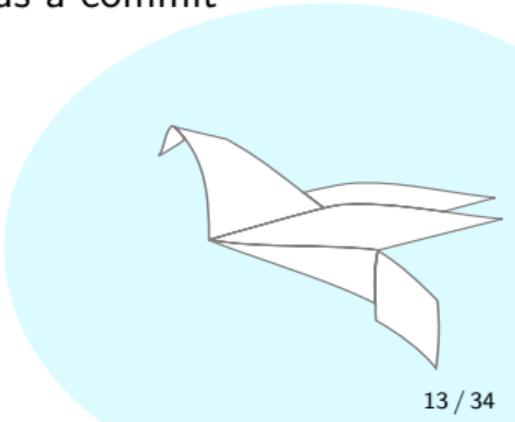
- Alice joins project
- Alice adds a commit
- **Alice leaves project**
- Alice adds a commit
- Now



# How to Authenticate a Project

- Curate a list of contributors and their certificates
- Track when contributors join and leave project
  - Authenticate old commits of retired contributors
  - Reject new commits of retired contributors
- Complicated much?
  - Maintainer: Need tooling
  - Third-party: Infeasible, but can rely on maintainer

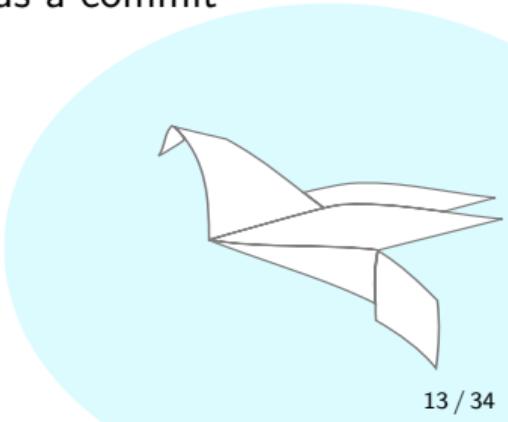
- Alice joins project
- Alice adds a commit
- Alice leaves project
- Alice adds a commit
- Now



# How to Authenticate a Project

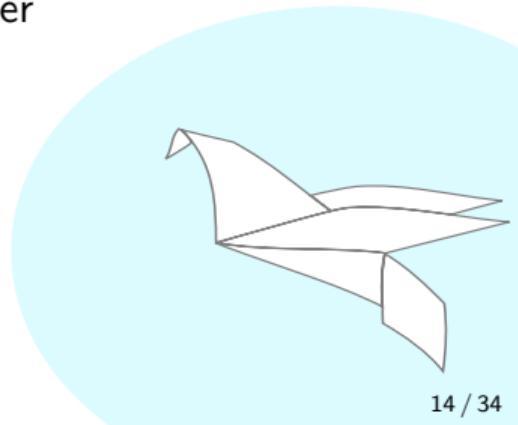
- Curate a list of contributors and their certificates
- Track when contributors join and leave project
  - Authenticate old commits of retired contributors
  - Reject new commits of retired contributors
- Complicated much?
  - Maintainer: Need tooling
  - Third-party: Infeasible, but can rely on maintainer

- Alice joins project
- Alice adds a commit
- Alice leaves project
- Alice adds a commit
- Now



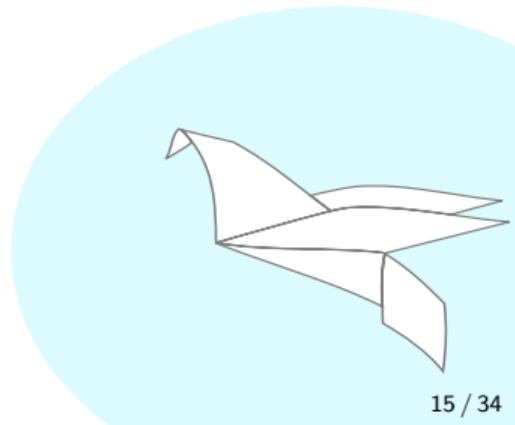
# Why Authenticate a Project?

- Detect unauthorized commits
  - Malicious forge
  - Compromised forge or account
  - Machine in the middle
- Detect when forge or registry gives a project to a new maintainer



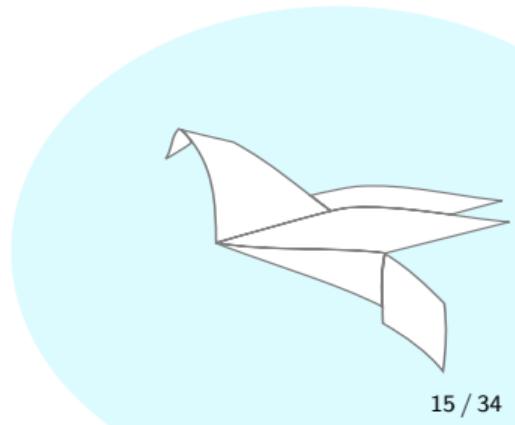
# Shape of Solution

- Clear semantics for signing commits
- Project maintains a signing policy
  - Policy evolves with time
- Third-party uses maintainers' policy to authenticate project
  - Delegates responsibility for signing policy to maintainers
  - Audits project and policy
- Limitations
  - Can't detect social engineering attack



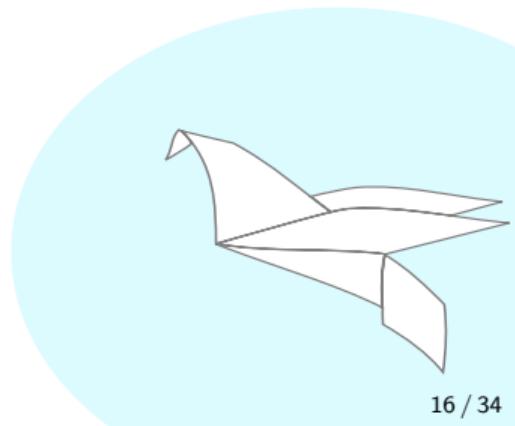
# Shape of Solution

- Clear semantics for signing commits
- Project maintains a signing policy
  - Policy evolves with time
- Third-party uses maintainers' policy to authenticate project
  - Delegates responsibility for signing policy to maintainers
  - Audits project and policy
- Limitations
  - Can't detect social engineering attack



# Goals

- Do not rely on an authority
  - Verification, not attestation
- Work off line



# Sequoia git

- Specification
- Configuration
- Tooling



sequoia-pgp.gitlab.io/sequoia-git/specification/ Table of Contents

Workgroup:	openpgp
Internet-Draft:	<a href="#">draft-nrw-openpgp-supply-chain-security-vcs-00</a>
Published:	20 June 2023
Intended Status:	Informational
Expires:	22 December 2023
Authors:	N. H. Waffield <a href="#">J. Winter</a> <a href="#">Sequoia PGP</a> <a href="#">Sequoia PGP</a>

## Supply Chain Security for Version Control Systems

### Abstract

In a software supply chain attack, an attacker injects malicious code into some software, which they then leverage to compromise systems that depend on that software. A simple example of a supply chain attack is when SourceForge, a once popular open source software forge, injected advertising into the binaries that they delivered on behalf of the projects that they hosted. Software supply chain attacks are different from normal bugs in that the intent of the perpetrator is different: in the former case, bugs are added with the intent to harm, and in the latter they are added inadvertently, or due to negligence.

Software supply chain security starts on a developer's machine. By signing a commit or a tag, a developer can assert that they wrote or approved the change. This allows users of a code base to determine whether a version has been approved, and by whom, and then make a policy decision based on that information. For instance, a packager may require that software releases be signed with a particular certificate.

Version control systems such as git have long included support for signed commits and tags. Most developers don't sign their commits, and in the cases where they do, it is usually unclear what the semantics are.

This document describes a set of semantics for signed commits and tags, and a framework to work with them in a version control system, in particular, in a git repository. The framework is designed to be self contained. That is, given a repository, it is possible to add changes, or authenticate a version without consulting any third parties; all of the relevant information is stored in the repository itself.

By publishing this draft we hope to clarify and enrich the semantics of signing in version control system repositories thereby enabling a new tooling ecosystem, which can strengthen software supply chain security.

The RFC Editor will remove this note

### About This Document

This note is to be removed before publishing as an RFC.

<https://sequoia-pgp.gitlab.io/sequoia-git/specification/>

# Signing policy

- Stored in project's git repository
  - Evolves with project
  - Off-line verification
- Users ■
  - Capabilities ■
    - Add a commit
    - Add a tag
    - Create a release
    - Add / remove a user
    - Audit commits
  - OpenPGP certificates ■
- Good list ■
  - Good commits from hard revoked certificates

## openpgp-policy.toml

```
version = 0
commit_goodlist = []

[authorization.neal]
sign_commit = true
keyring = ""

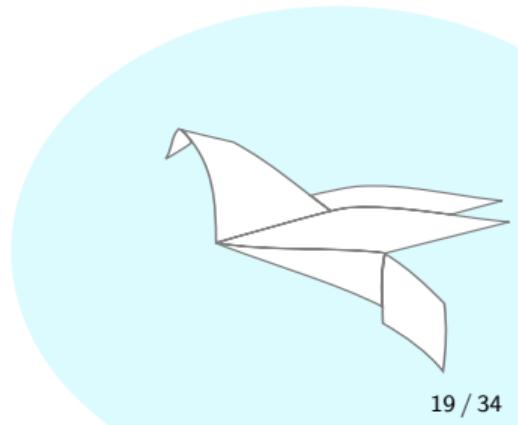
-----BEGIN PGP PUBLIC KEY BLOCK-----
Comment: F717 3B3C 7C68 5CD9 ECC4 191B 74E4 45BA 0E15 C957
Comment: Neal H. Walfield (Code Signing Key) <neal@sequoia-pgp.o
xjMEWhaZ2xYJKwYBBAHaRw8BAQdAing1S6SRXyMb51hMk+mpM4y0Uh0vcGcTyXa+
...
[authorization.neal-offline]
sign_commit = true
sign_tag = true
sign_archive = true
add_user = true
retire_user = true
audit = true
keyring = ""

-----BEGIN PGP PUBLIC KEY BLOCK-----
Comment: 8F17 7771 18A3 3DDA 9BA4 8E62 AACB 3243 6300 52D9
Comment: Neal H. Walfield <neal@sequoia-pgp.org>
xsEhBFUjmukBDqCpmVI7Ve+2xTFSTG+mXMFHml63/Yai2nqxBk9gBfQfRFIjMt74
...
```

# Authorizing a Committer

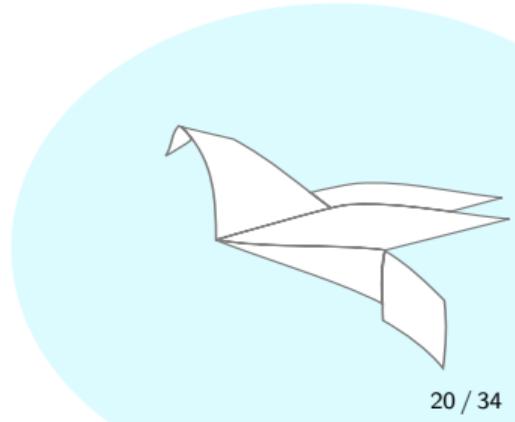
```
$ sq-git policy authorize alice --committer C1A5D7F4CB832012208AA598C4AA55549C5AB20F
```

- User alice was added.
- User alice was granted the right sign-commit.
- User alice: new certificate C1A5D7F4CB832012208AA598C4AA55549C5AB20F.



# Updating Certificates

```
$ sq-git policy sync
Updating F7173B3C7C685CD9ECC4191B74E445BA0E15C957 (neal) from the local certificate store... not found.
Updating F7173B3C7C685CD9ECC4191B74E445BA0E15C957 (neal) from https://keys.openpgp.org... unchanged.
Updating F7173B3C7C685CD9ECC4191B74E445BA0E15C957 (neal) from https://mail-api.proton.me... Cert not found.
Updating F7173B3C7C685CD9ECC4191B74E445BA0E15C957 (neal) from https://keys.mailvelope.com... Cert not found.
Updating F7173B3C7C685CD9ECC4191B74E445BA0E15C957 (neal) from https://keyserver.ubuntu.com... updated.
Updating F7173B3C7C685CD9ECC4191B74E445BA0E15C957 (neal) from https://sks.pod01.fleetstreetops.com... updated.
...
Note: certificates are stripped so not all certificate updates may be relevant.
```



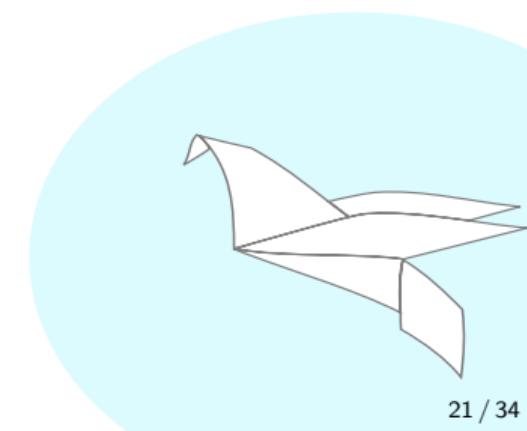
# Examining the Policy

```
$ sq-git policy describe
# OpenPGP policy file for git, version 0

## Commit Goodlist

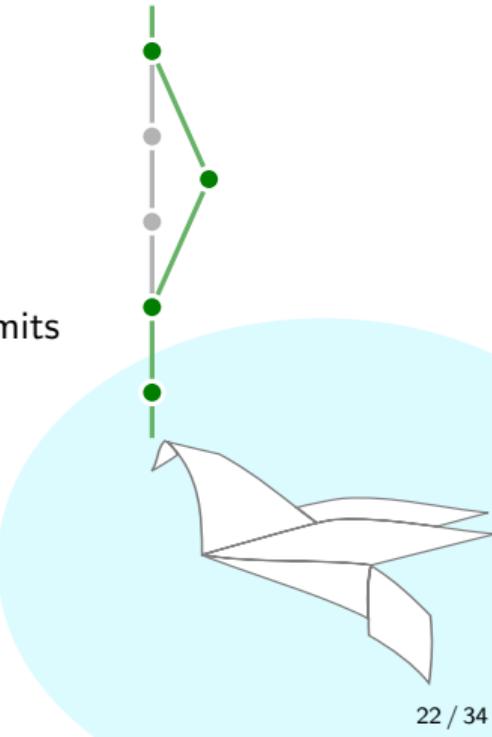
## Authorizations

0. alice
- may sign commits
- has OpenPGP cert: C1A5D7F4CB832012208AA598C4AA55549C5AB20F
1. neal
- may sign commits
- has OpenPGP cert: F7173B3C7C685CD9ECC4191B74E445BA0E15C957
2. neal-offline
- may sign commits
- may sign tags
- may sign archives
- may add users
- may retire users
- may goodlist commits
- has OpenPGP cert: 8F17777118A33DDA9BA48E62AACB3243630052D9
```



# Authenticating Commits

- A commit is authenticated if:
  - The policy of at least one parent says the commit is acceptable
  - (Policy determines types of changes that are allowed.)
- A range of commits is authenticated if:
  - There is a path from one commit to another along which all commits are authenticated



# Authenticating Commits: Example

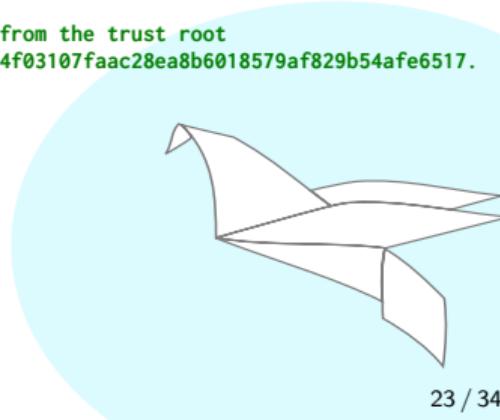
```
$ git log --graph
* commit 34f03107faac28ea8b6018579af829b54afe6517 (HEAD -> master)
| Author: Alice <alice@example.org>
| Date:   Fri Jan 30 15:53:31 2026 +0100
|
|   Add feature.

* commit a7a40e007990e84ce1794a5b4e62bb38b81e443d
| Author: Neal H. Walfield <neal@sequoia-pgp.org>
| Date:   Fri Jan 30 15:46:48 2026 +0100
|
|   Make alice a committer

* commit 61cd610f0c53cbd73e1733c10edaa081ab356c96
| Author: Neal H. Walfield <neal@sequoia-pgp.org>
| Date:   Fri Jan 30 15:46:15 2026 +0100
|
|   Make neal (coding signing cert) a committer

* commit aab50d699240a3d022fc11464ac458b5fbb7bb22
| Author: Neal H. Walfield <neal@sequoia-pgp.org>
| Date:   Fri Jan 30 15:47:35 2026 +0100
|
|   Make neal a project maintainer
```

```
$ sq-git log --trust-root aab50d699240a3d022fc11464ac458b5fbb7bb22
a7a40e007990e84ce1794a5b4e62bb38b81e443d..34f03107faac28ea8b6018579af829b54afe6517:
  Signer: alice [C1A5D7F4CB832012208AA598C4AA55549C5AB20F]
  Add feature.
61cd610f0c53cbd73e1733c10edaa081ab356c96..a7a40e007990e84ce1794a5b4e62bb38b81e443d:
  Signer: neal-offline [8F17777118A33DDA9BA48E62AACB3243630052D9]
  Make alice a committer
aab50d699240a3d022fc11464ac458b5fbb7bb22..61cd610f0c53cbd73e1733c10edaa081ab356c96:
  Signer: neal-offline [8F17777118A33DDA9BA48E62AACB3243630052D9]
  Make neal (coding signing cert) a committer
Verified that there is an authenticated path from the trust root
aab50d699240a3d022fc11464ac458b5fbb7bb22 to 34f03107faac28ea8b6018579af829b54afe6517.
$ echo $?
0
```



# Not Authenticating a Commit

```
$ git log --show-signature -n 2
commit d709fccaaeb4c45b33080c07ceb1eb68eaca2aeb (HEAD -> master)
gpg: Signature made Fri Jan 30 16:27:28 2026 +01:00
gpg:           using EDDSA key 1F660C60C700B41CA8AE351FD44ECFCFB9093212
gpg: Good signature from "Bob" [full]
gpg:                               "<bob@example.org>"
Author: Bob <bob@example.org>
Date:   Fri Jan 30 16:27:28 2026 +0100
```

Improve build system.

```
commit 34f03107faac28ea8b6018579af829b54afe6517
gpg: Signature made Fri Jan 30 15:53:31 2026 +01:00
gpg:           using EDDSA key 1742E6737DAE88C2227FD1B05A78679CBCA65B79
gpg: Good signature from "Alice" [full]
gpg:                               "<alice@example.org>"
Author: Alice <alice@example.org>
Date:   Fri Jan 30 15:53:31 2026 +0100
```

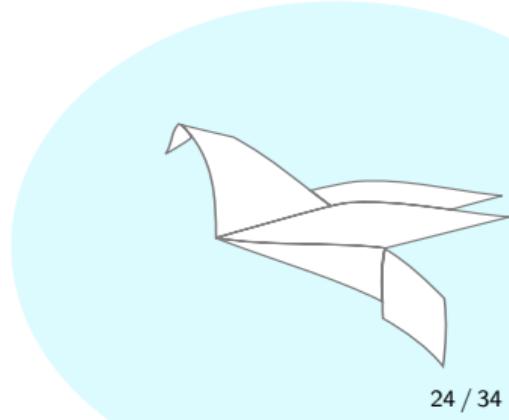
Add feature.

- Bob adds a commit
- Bob is not authorized

```
$ sq-git log --trust-root 34f03107faac28ea8b6018579af829b54afe6517
34f03107faac28ea8b6018579af829b54afe6517..
d709fccaaeb4c45b33080c07ceb1eb68eaca2aeb:
Error: Key `1F660C60C700B41CA8AE351FD44ECFCFB9093212` missing
Improve build system.
Error: Could not verify commits 34f03107faac28ea8b6018579af829b54afe6517..
d709fccaaeb4c45b33080c07ceb1eb68eaca2aeb
```

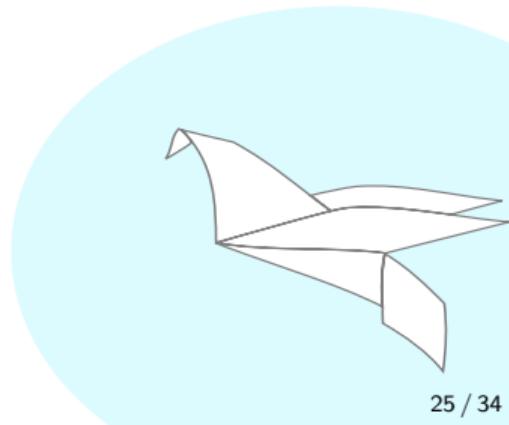
Caused by:

```
 0: While verifying commit d709fccaaeb4c45b33080c07ceb1eb68eaca2aeb
  1: Key `1F660C60C700B41CA8AE351FD44ECFCFB9093212` missing
$ echo $?
1
```



# Merging Bob's Commit

- Strategies
  - Committer re-signs Bob's commit
    - Preserves linear history
  - Committer merges Bob's commit
    - Preserves Bob's signature
    - Allowed by single path rule



# Merging Bob's Commit

```
$ git merge 34f03107faac28ea8b6018579af829b54afe6517 d709fccaaeb4c45b33080c07ceb1eb68eaca2aeb --no-ff
```

```
$ git log --graph 34f03107faac28ea8b6018579af829b54afe6517^..
```

```
* commit 2fccc9948ad094c56c551c4d8550d4ced6f6751 (HEAD -> master)
```

```
  Merge: 34f0310 d709fcc
```

```
  Author: Alice <alice@example.org>
```

```
  Date: Fri Jan 30 16:52:46 2026 +0100
```

```
  Merge Bob's work.
```

```
* commit d709fccaaeb4c45b33080c07ceb1eb68eaca2aeb
```

```
  Author: Bob <bob@example.org>
```

```
  Date: Fri Jan 30 16:27:28 2026 +0100
```

```
  Improve build system.
```

```
* commit 34f03107faac28ea8b6018579af829b54afe6517
```

```
  Author: Alice <alice@example.org>
```

```
  Date: Fri Jan 30 15:53:31 2026 +0100
```

```
  Add feature.
```

```
$ sq-git log --trust-root a7a40e007990e84ce1794a5b4e62bb38b81e443d
```

```
34f03107faac28ea8b6018579af829b54afe6517..2fccc9948ad094c56c551c4d8550d4ce
```

```
  Signer: alice [C1A5D7F4CB832012208AA598C4AA55549C5AB20F]
```

```
  Merge Bob's work.
```

```
d709fccaaeb4c45b33080c07ceb1eb68eaca2aeb..2fccc9948ad094c56c551c4d8550d4ce
```

```
  Cached positive verification
```

```
34f03107faac28ea8b6018579af829b54afe6517..d709fccaaeb4c45b33080c07ceb1eb68e
```

```
  Error: Key `1F660C60C700B41CA8AE351FD44ECFCFB9093212` missing
```

```
  Improve build system.
```

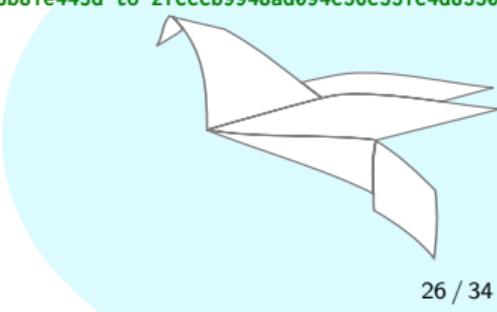
```
a7a40e007990e84ce1794a5b4e62bb38b81e443d..34f03107faac28ea8b6018579af829b54
```

```
  Signer: alice [C1A5D7F4CB832012208AA598C4AA55549C5AB20F]
```

```
  Add feature.
```

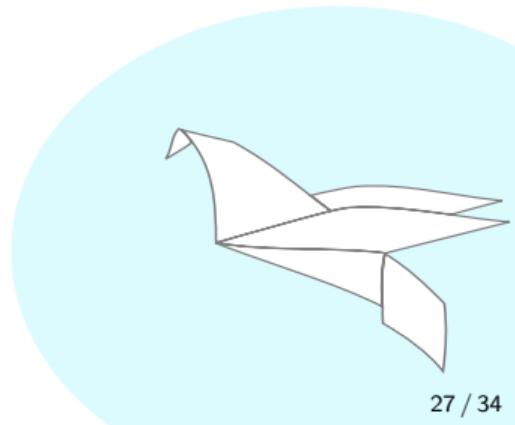
```
Verified that there is an authenticated path from the trust root
```

```
a7a40e007990e84ce1794a5b4e62bb38b81e443d to 2fccc9948ad094c56c551c4d8550d4
```



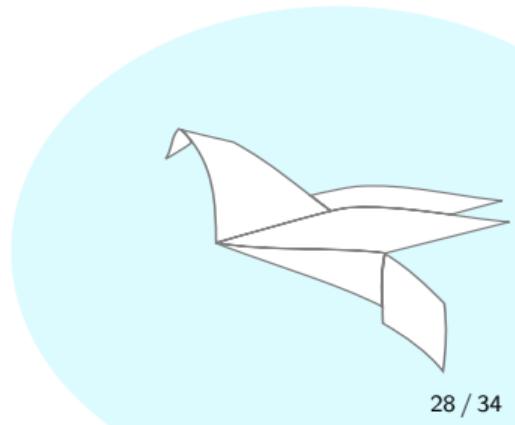
## More Details

- Updating own expired certificate: yes
- Recover from a hard revoked key: yes, commit good listing
- Verifying tags and tarballs: yes, but incomplete
- See man pages and documentation



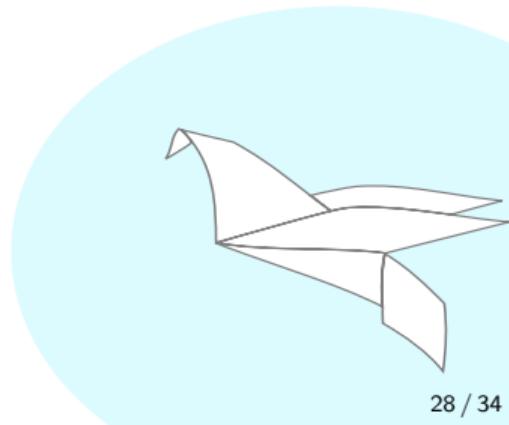
# Forking a Project

- New maintainer not added to the signing policy?
  - sq-git will not consider the first commit authenticated
- These are the semantics we want!
- To use a fork, users must opt in by setting a new trust root



# Forking a Project

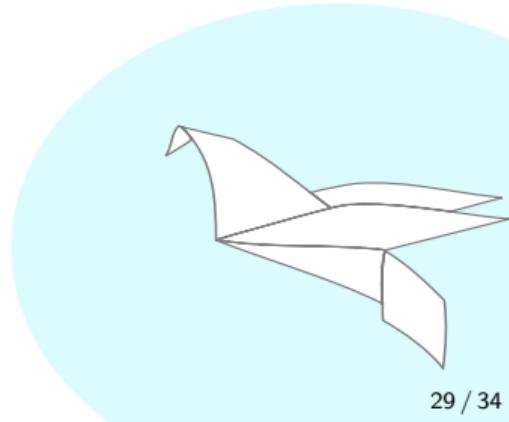
- New maintainer not added to the signing policy?
  - sq-git will not consider the first commit authenticated
- These are the semantics we want!
- To use a fork, users must opt in by setting a new trust root



# Integrating sq-git into Your Git Repository

- Insert the following line into hooks/update

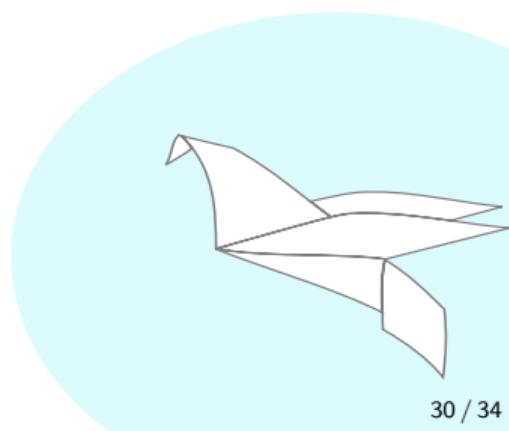
```
sq-git update-hook --trust-root=<COMMIT> "$@"
```



# Integrating sq-git on GitLab

- Add to your `.gitlab-ci.yml`:

```
authenticate-commits:
  stage: test
  image: registry.gitlab.com/sequoia-pgp/sequoia-git:latest
  before_script: []
  script:
    - sq-git policy describe
    - /usr/sbin/gitlab.sh # Script baked-in to image
  after_script: []
  rules:
    - if: '$CI_COMMIT_BRANCH != $CI_DEFAULT_BRANCH'
```



# Integrating sq-git on GitHub

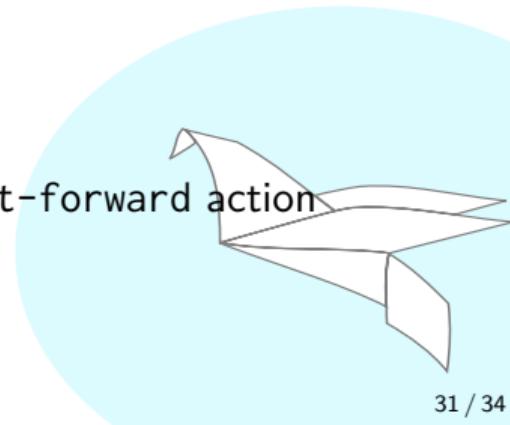
- Add `.github/workflows/authenticate-commits.yml`:

```
name: authenticate-commits
on:
  pull_request:
    types: [opened, reopened, synchronize]
jobs:
  authenticate-commits:
    runs-on: ubuntu-latest

    permissions:
      contents: read
      pull-requests: write
      issues: write

    steps:
      - name: Authenticating commits
        uses: sequoia-pgp/authenticate-commits@v1
```

- GitHub destroys signatures on fast-forward, so need to use fast-forward action



# authenticate-commits on GitHub

Release 1.10.0 by nwalfie: x +

github.com/rpm-software-management/rpm-sequoia/pull/102

Merged Release 1.10.0 #102  
github-actions merged 3 commits into `main` from `staging` on Nov 25, 2025

github-actions bot commented on Nov 25, 2025

```
* eb3cd2bfaf42a8806cba5bd060c7458654625c94 Release 1.10.0
| - Authorized by Neal H. Walfield <neal@pep.foundation> [F7173B3C7C685CD9ECC4191B74E445BA0E
* b94763b6503a1fff9ad0e85ddb0c609b57bc14747 Update Cargo.lock.
| - Authorized by Neal H. Walfield <neal@pep.foundation> [F7173B3C7C685CD9ECC4191B74E445BA0E
* 828a28bbfaacf35b4c1590bc0c72d734ca24505 meta: Use cargo's MSRV-aware resolver, if available
| - Authorized by Neal H. Walfield <neal@pep.foundation> [F7173B3C7C685CD9ECC4191B74E445BA0E
* 73dfb54e8d790ceac54053c836ddd72cd63d38aa salt: Fix typo in error message
- Trust root.
```

The pull request's base ( [73dfb54](#) ) authenticates the pull request's head ( [eb3cd2b](#) ).

nwalfie commented on Nov 25, 2025

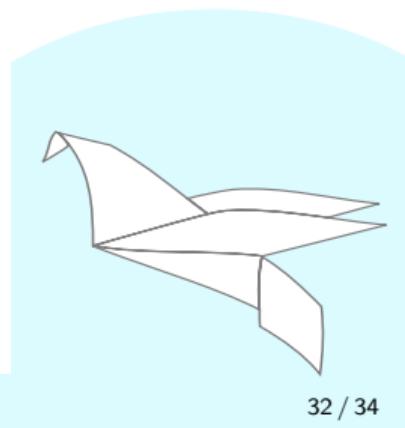
Collaborator Author ...

/fast-forward

github-actions bot merged commit [eb3cd2b](#) into `main` on Nov 25, 2025

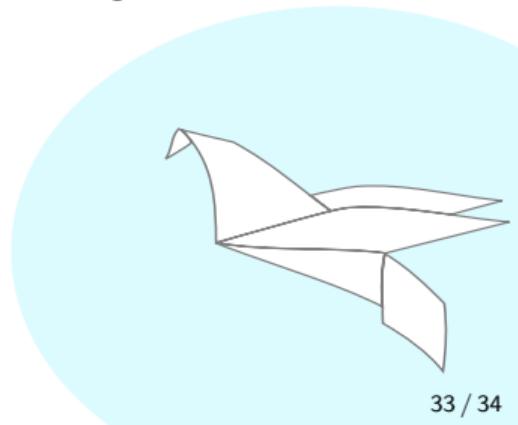
View details

6 checks passed



## Related Work

- *Building a Secure Software Supply Chain with GNU Guix* by Ludovic Courtès
- *Arista: Commit Signing with Git at Enterprise Scale*
- List of commit signing solutions
  - <https://gitlab.torproject.org/tpo/tpa/team/-/wikis/service/gitlab#git-repository-integrity-solutions>



# Summary

- Signing commits is useful
  - Protects against impersonation
  - Detects unauthorized commits
  - No need to trust forge
- sq-git helps
  - Clear semantics
  - Tooling for managing signing policies
  - Tooling for checking commits against a policy

# Questions?

