Cosign Sign and verify artifacts

CONCEPTS

> Sign by digest (image@sha256:...), not by tag: a tag can move to a different image after you sign.

OIDC login Google, GitHub, ...
↓ identity token
Fulcio short-lived cert
↓ sign with ephemeral key
Signature stored in registry
↓ log entry
Rekor transparency log
Keyless signing

> Keyless verify needs both an expected identity and an expected OIDC issuer; without them anyone with any Sigstore identity would pass.

KEYS

`cosign generate-key-pair`
Create cosign.key and cosign.pub

`--key awskms:// / gcpkms:// / azurekms:// / hashivault://`
Key in a KMS (URI schemes)

SIGN

`cosign sign --key cosign.key <image>@<digest>`
Sign with a local key

`cosign sign <image>@<digest>`
Keyless sign (OIDC login)

`cosign sign --yes <image>@<digest>`
Skip confirmation prompts (CI)

VERIFY

`cosign verify --key cosign.pub <image>`
Verify with a public key

`cosign verify --certificate-identity <id> --certificate-oidc-issuer <url> <image>`
Verify a keyless signature

ATTESTATIONS

`cosign attest --key cosign.key --predicate <file> <image>@<digest>`
Attach a signed predicate

`cosign attest --type <type> --predicate <file> <image>@<digest>`
Set the predicate type

`cosign verify-attestation --key cosign.pub <image>`
Verify attestations

`cosign verify-attestation --type <type> --policy <file> --key cosign.pub <image>`
Check predicate against a policy

BLOBS

`cosign sign-blob --bundle <bundle> <file>`
Keyless sign a file into a bundle

`cosign verify-blob --bundle <bundle> --certificate-identity <id> --certificate-oidc-issuer <url> <file>`
Verify a keyless file signature

`cosign sign-blob --key cosign.key <file>`
Sign a file with a key

`cosign verify-blob --key cosign.pub --bundle <bundle> <file>`
Verify a file with a public key

INSPECT AND MANAGE

`cosign tree <image>`
Show signatures, SBOMs, attestations

`cosign version`
Print the cosign version

ENVIRONMENT

`COSIGN_PASSWORD=<pw>`
Key password without a prompt

`SIGSTORE_ID_TOKEN=<token>`
Supply an OIDC token, skip login