Objective

To address secure naming delivery with SDS, the original issue is raised on envoy#4763 to allow deliver trust CA as a separate resources from CertificateValidationContext. This is mainly for upstream TLS context.

Secure naming is a 1:N mapping from service (cluster) to service accounts (subject alt name).

Current Implementation

- CVC contains verify_subject_alt_name from CDS
- Trust CA is delivered locally and when pilot agent does hot restart if it changes

Available Options

1. CertificateValidationContext from NodeAgent (SDS)

Citadel and NodeAgent figure out the secure naming mapping and deliver along with Trust CA via SDS

Pros

- No code change required in Envoy
- Secure naming change doesn't drain cluster/listener
- SDS as single source of truth for authentication

Cons

- Substantial amount of work in Citadel/NodeAgent to figure out service to service account mapping
- A lot of duplicate work with Pilot

2. CertificateValidationContext from Pilot (CDS/LDS)

Pilot to figure out the secure naming mapping and deliver along with Trust CA via SDS

Pros

- No code change required in Envoy
- Easier than 1 for pilot to get root CA vs citadel to get secure naming mapping via service registry

Cons

- Pilot need to fetch root CA to deliver with secure naming.

3. TrustCA secret type

Proposed in PR#4730, add a new secret type as Trust CA.

Pros

- Less code change to Istio control plane implementation, Node agent to deliver trust CA

Cons

- Nested xDS resource dependency (need to avoid)
- Complex config provider with dynamic trust CA

4. Merge semantics

Use protobuf to merge CertificateValidationContext delivered from LDS/CDS with the one from SDS.

Pros

- Less code change to Istio control plane implementation, Node agent to deliver CVC with trust CA

Cons

- Hard to figure out what is the actual CertificateValidationContext used in particular Listener/Cluster,
- Priority is confusing: perhaps SDS override LDS/CDS? Reject if a field exist in both?
- Code change involves secret manager with inline and dynamic providers

5. Delivery SAN verification via EDS (pilot)

Use EDS to deliver SAN verification field by allowing specifying some validation context per endpoint, CertificateValidationContext delivered from SDS.

Pros

- Strong consistency when a service account (pod) is added/removed from a service
- Stronger mapping between endpoint and SAN
- Secure naming change doesn't drain cluster/listener

Cons

- Larger scope of Envoy API change and substantial code changes to support.
- Medium code change to pilot.

Recommendation

5 to be long term solution:

- secure naming discovery (service name to service account mapping) and endpoints discovery (service name to ip mapping) naturally fits together.

In short term, figure out how easy 2 is, 3 or 4 is acceptable if 2 is too large work.