TLS to the K8s.Service/Backend

Author: bowei@google.com, robertjscott@google.com, <add>
Shared to community

Some notes on describing TLS properties to the K8s Backend from a Gateway API *Route reference.

There are a number of things here that the different user personas can describe and we should be careful about:

1. **Who** gets to describe it and how this ability is scoped/granted/revoked?
2. **What** a given configuration means for a conformant implementation. Can the implementation pick and choose which things are meaningful based on the desired security properties? For example, would conformance to an API result in a user being able to downgrade the security of the infrastructure?

What people want to describe:

- **Protocol-only**: Gateway infrastructure should use TLS when talking to the Backend. Use whatever default authentication policy and configuration for the communication. This is what is currently possible with Ingress + Service.AppProtocol = "tls". This doesn't give any knobs for configuration of Authn or TLS protocol properties.
- **Authn**: Gateway should only use this specific authentication / trusted root or expect this set of identities when communicating with this Service.
- **Protocol-specific configuration**: like protocol-only, but more specific. Use this set of ciphers, versions.
- **Complex Authn e.g. mTLS**: like Authn, but instead of static configuration (e.g. configuring a CA.crt secret), the definition of a valid peer is part of a larger system of identity. The canonical example is support for strong secure identities using mTLS.

More broadly:

- We should consider this as a pattern that should be reusable to decorate/configure options of an upstream protocol to a Service **in general**. It would be less desirable to not be able to port this pattern sideways to other protocols outside of TLS. We don't need to create a blob that must meet all use cases, but we should try to reuse the reasoning/design.