What This Is

This is a fork of OpenSSL to enable QUIC. In addition to the website, the official source distribution is at https://github.com/openssl/openssl. The OpenSSL `README` can be found at README-OpenSSL.md

This fork adds APIs that can be used by QUIC implementations for connection handshakes. Quoting the IETF Working group charter, QUIC is a "UDP-based, stream-multiplexing, encrypted transport protocol." If you don't need QUIC, you should use the official OpenSSL distributions.

The APIs here are used by Microsoft's MsQuic and Google's Chromium QUIC

We are not in competition with OpenSSL project. We informed them of our plans to fork the code before we went public. We do not speak for the OpenSSL project, and can only point to a blog post and openssl-project email that provides their view of QUIC support.

As stated in their blog post, the OpenSSL team is focused on their 3.0 release (released 2021-09-07), and does not intend to add QUIC functionality to 1.1.x. There is a community need for a QUIC-capable TLS library. This fork is intended as stopgap solution to enable higher level frameworks and runtimes to use QUIC with the proven and reliable TLS functionality from OpenSSL. This fork will be maintained until OpenSSL officially provides reasonable support for QUIC implementations.

This fork can be considered a supported version of OpenSSL PR 8797. We will endeavor to track OpenSSL releases within a day or so, and there is an item below about how we'll follow their tagging.

On to the questions and answers.

What about branches?

We don't want to conflict with OpenSSL branch names. Our current plan is to append `+quic`. Release tags are likely to be the QUIC branch with `-releaseX` appended. For example, the OpenSSL tag `openssl-3.0.0` would have a branch named `openssl-3.0.0+quic` and a release tag of `openssl-3.0.0+quic-release1`.

How are you keeping current with OpenSSL?

(In other words, "What about rebasing?")

Our plan is to always rebase on top of an upstream release tag. In particular:

- The changes for QUIC will always be at the tip of the branch -- you will know what is from the original OpenSSL and what is for QUIC.
- New versions are quickly created once upstream creates a new tag.
- The use of git commands (such as `cherry`) can be used to ensure that all changes have moved forward with minimal or no changes. You will be able to see "QUIC: Add X" on all branches and the commit itself will be nearly identical on all branches, and any changes to that can be easily identified.

What about library names?

Library names will be the same, but will use a different version number. The version numbers for the current OpenSSL libraries are `1.1` (for the 1.1.0 and 1.1.1 branches) and `3` (for the 3.0 branch). We will be prefixing `81` (ASCII for 'Q') to the version numbers to generate a unique version number.

- `libcrypto.so.81.3` vs `libcrypto.so.3`
- `libcrypto.so.81.1.1` vs `libcrypto.so.1.1`

- `libssl.so.81.3` vs `libssl.so.3`
- `libssl.so.81.1.1` vs `libssl.so.1.1`

The SONAME of these libraries are all different, guaranteeing the correct library will be used.

...and the executable?

We currently do not have any plans to change the name, mainly because we haven't made any changes there. If you see a need, please open an issue.

The `openssl version` command will report that it is `+quic` enabled.

...and FIPS?

We are not doing anything with FIPS. This is actually good news: you should be able to load the OpenSSL 3.0 FIPS module into an application built against this fork and everything should Just Work™.

How can I contribute?

We want any code here to be acceptable to OpenSSL. This means that all contributors must have signed the appropriate contributor license agreements. We will not ask for copies of any paperwork, you just need to tell us that you've done so (and we might verify with OpenSSL). We are only interested in making it easier and better for at least the mentioned QUIC implementations to use a variant of OpenSSL. If you have a pull request that changes the TLS protocol, or adds assembly support for a new CPU, or otherwise is not specific to enabling QUIC, please contribute that to OpenSSL. This fork is intended to be a clean extension to OpenSSL, with the deltas being specific to QUIC.

Who are you?

This is a collaborative effort between Akamai and Microsoft. We welcome anyone to contribute!