No more Copies! Re-Designing TCPLS Multipath

Elkoulak, Hosam

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

Citation for pulished version (HARVARD):
Elkoulak, H 2024, 'No more Copies! Re-Designing TCPLS Multipath', 2024 Cyberwal in Galaxia, Redu, Belgium, 2/12/24 - 6/12/24.

No more Copies! Re-Designing TCPLS Multipath

Hosam Elkoulak

[1]University of Namur

Background

What is TCPLS?

TCPLS is an intertwined TLS1.3/TCP design offering transport features like:

- Multipath.
- Connection migration.
- Fail-over.

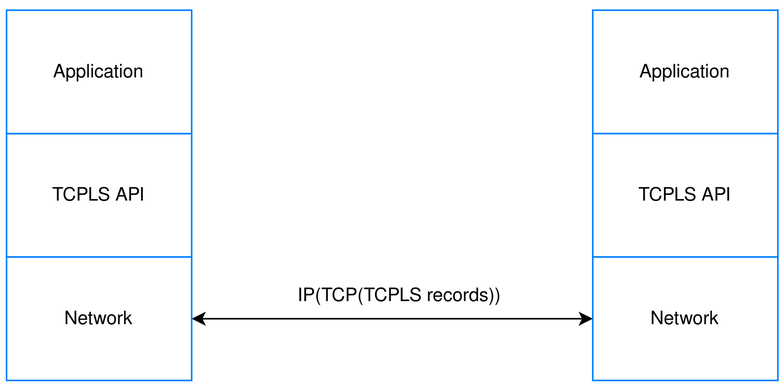

Figure 1. The TCPLS network stack.

Reverso[1]

Two key principles may be applied to any encrypted transport protocol:

- **Principle 1.** The field order within all encrypted control and data chunks is reversed. For a chunk with type (u8), foo (u16), and bar (u64), the order becomes bar (u64), foo (u16), type (u8).
- **Principle 2.** Only a single chunk of data is allowed within a record followed be arbitrary number of control chunks.

Contiguous Zero-Copy

Applying the principles of reverso leads to a simplified receive code path. Data received can be decrypted directly into the application buffers avoiding unnecessary copying.

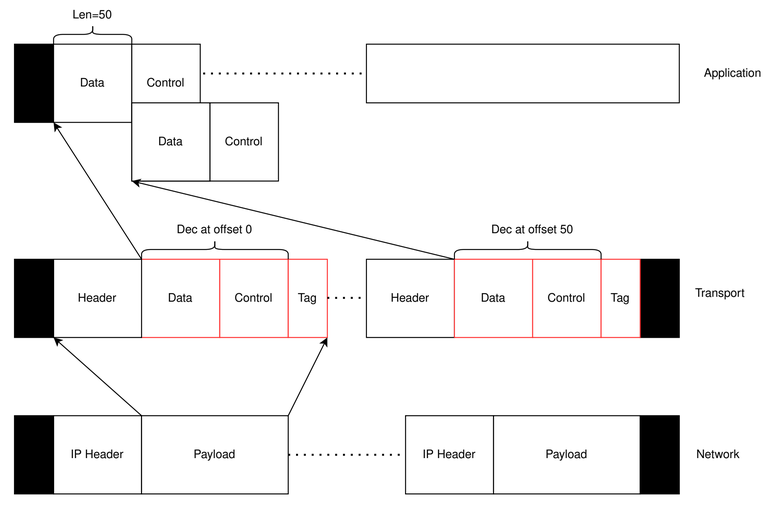

Figure 2. contiguous decryption in appliction buffers

Drawbacks vs. Solution

Previous Design Drawbacks

The sender's inability to identify the stream to which the received record belongs has resulted in two design drawbacks:

Stream Multiplexing

Sender has to send the TCPLS' frame **Stream Change** before switching between streams which consumes more bandwidth and requires extra processing for encryption and decryption.

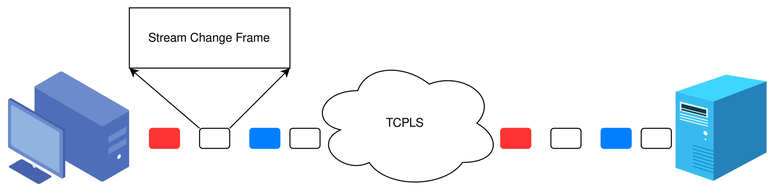

Figure 3. send stream change frame before swithing streams

Extra copy along receive path

Decryption happens in place and the result is copied in the application receive buffer.

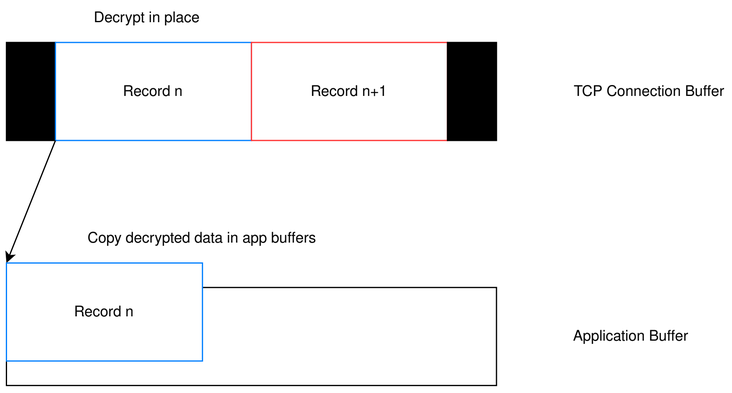

Figure 4. Additional copy is required to deliver data in app buffers

Solution

TCPLS Protected Header

TCPLS header consists of 4 bytes encoding the offset where to decrypt in app buffers and 4 bytes encoding the stream ID.

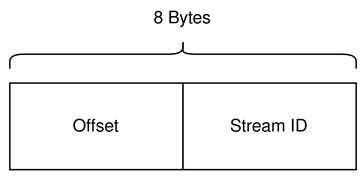

Figure 5. TCPLS header

As a result two improvements were achieved:

- Remove the need for **Stream Change** frames, as the TCPLS header already contains this information, saving bandwidth and eliminating extra encryption/decryption operations for the frame.
- TCPLS header is secured separately. This makes it possible to decrypt directly in app buffers avoiding the extra copy operation. In addition to that the order of the records can be checked before decrypting.

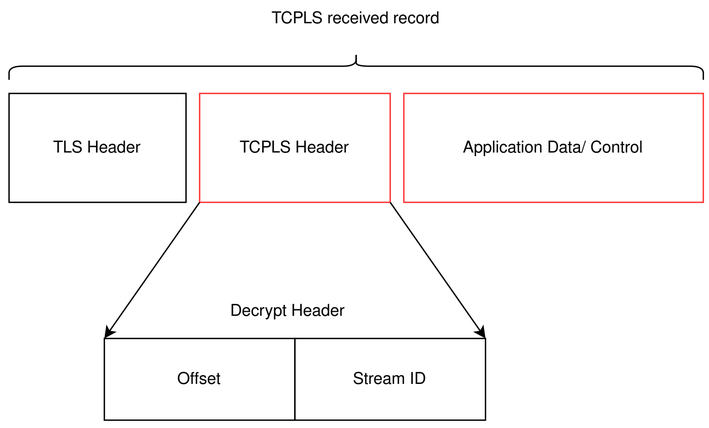

Figure 6. Avoid copying by decrypting only records in the right order

References

[1] Florentin Rochet. Improving encrypted transport protocol designs: Deep dive on the quic case, 2024.

[2] Florentin Rochet, Emery Assogba, Maxime Piraux, Korian Edeline, Benoit Donnet, and Olivier Bonaventure. Tcpls: Modern transport services with tcp and tls. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 45–59, 2021.

UNIVERSITÉ DE NAMUR

Cyberwal by | digital wallonia .be