1. [Page 4](javascript:%20void(0);)
2. TLS 1.3

**TLS 1.3**

TLS 1.3 made several improvements over TLS 1.2. That includes dropping support for insecure cryptographic parameters and thereby reducing complexity. Furthermore, improvements to the handshake were made to allow for faster session establishment.

**Cipher Suites and Cryptography**

Several cryptographic improvements have been made with the new version TLS 1.3. These enhancements include the removal of older, less secure cryptographic techniques and the addition of newer, more secure techniques. TLS 1.3 also includes improved key exchange algorithms and support for post-quantum cryptography. In particular, TLS 1.3 only supports key exchange algorithms that support PFS.

For instance, a TLS 1.3 cipher suite looks like this:

TLS\_AES\_128\_GCM\_SHA256

It is significantly shorter than TLS 1.2 cipher suites since it only specifies the encryption algorithm and mode as well as the hash function used for the HMAC algorithm. TLS 1.3 cipher suites do not specify the method used for server authentication and the key exchange algorithm.

**Handshake**

Several changes were introduced in the TLS 1.3 handshake process. Some messages have been redesigned for efficiency, while other messages have been eliminated completely to reduce the latency and overhead of the handshake which enables a faster connection establishment.

Just like in TLS 1.2, the TLS 1.3 handshake begins with the ClientHello message. However, in TLS1.3 this message contains the client's key share in addition to the supported cipher suites. This eliminates the need for the ClientKeyExchange message later on in the handshake. This key share is contained in an extension that is sent with the ClientHello message.

The server responds with the ServerHello message that confirms the key agreement protocol and specifies the chosen cipher suite, just like in TLS 1.2. This message also contains the server's key share. A fresh key share is always transmitted here to guarantee PFS. This replaces the need for the ServerKeyExchange message in TLS1.2, which was required when PFS cipher suites were used. The server's certificate is also contained within the ServerHello message.

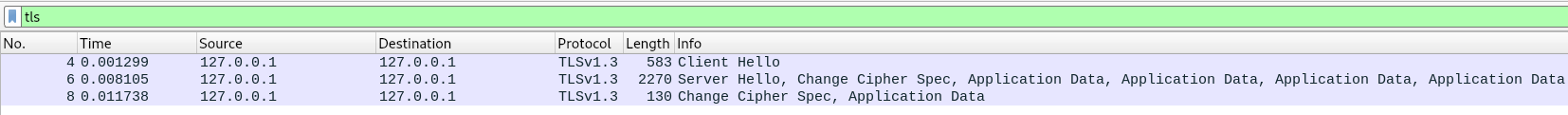
The handshake concludes with a ServerFinished and ClientFinished message.

**Note:** All messages after the ServerHello are already encrypted. Therefore, the TLS 1.3 handshake is significantly shorter than the TLS 1.2 handshake.



**Analyzing a TLS 1.3 Handshake in Wireshark**

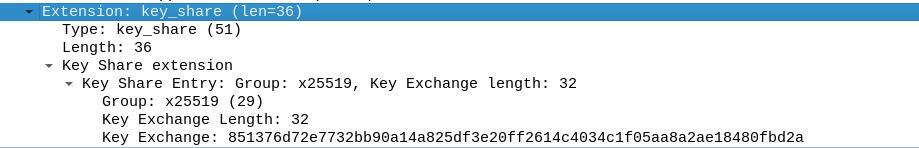
When looking at a TLS 1.3 handshake in Wireshark, the differences to a TLS 1.2 handshake become apparent. In particular, we can see that there are no Certificate and ClientKeyExchange messages since they have been removed:



We can find the client's key share in the key\_share extension in the ClientHello message. In this case, the client chooses two different shares for different groups. That is because the group is chosen by the server in the ServerHello message, which the client has not received yet. Therefore, the client transmits multiple shares to increase the chance of agreement on a group with the server:



The server's key share can be inspected in the key\_share extension in the ServerHello message. The server chooses a group and only transmits its key share for that group:



From this point on, all transmitted data is encrypted, as indicated by the EncryptedApplicationData tag in Wireshark.