$$H((K_a \oplus opad) \| H((K_a \oplus ipad) \| M))$$

64 bytes 64 bytes

sha-1 - input block for the compression function of the hash function: 64 bytes (512 bits)

output: 20 bytes

For all hash functions H used in TLS the application of H itself uses an encoding step named Merkle-Damgård strengthening: the message M is concatenated with 8-byte length and at least 1 byte of padding to align the block to multiplicity of 64 bytes. At least 9 bytes are appended to M.

So if $|M| \leq 55$ bytes then the inner hash needs 2 applications of sha-1 compression function. and outer hash still need 2 more evaluations.

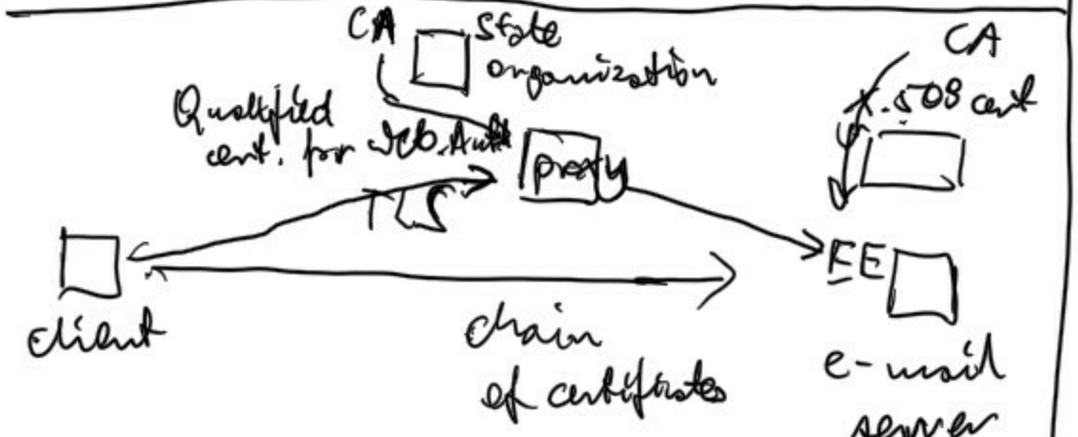
bytes

if $|M| > 55$ ✓ left

$|M| \leq 64 + 55$ bytes

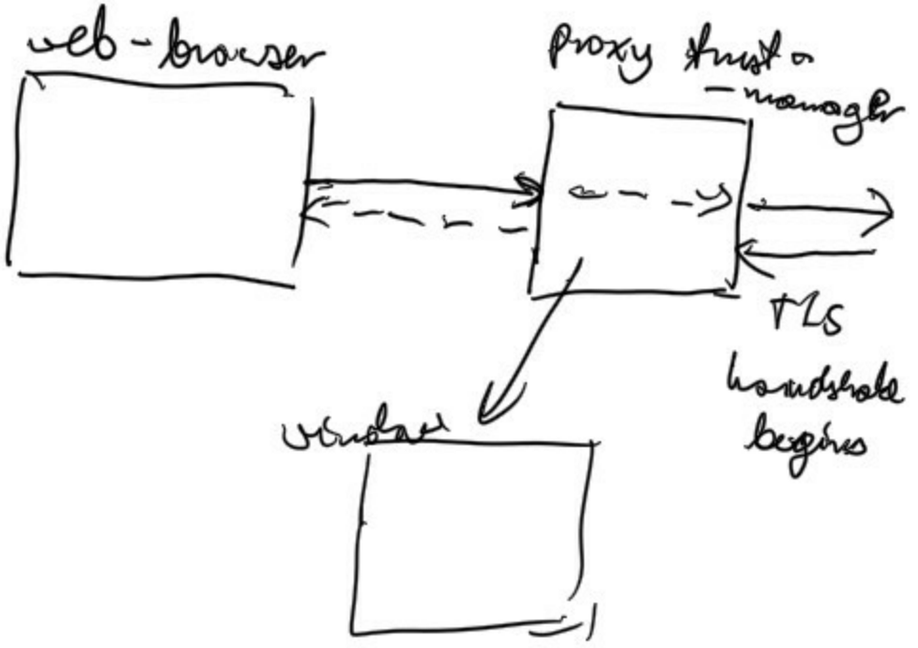
then the inner hash needs 3

evaluations of the compression function. The outer hash still needs 2 evaluations, so we have at least 5 evaluations.



Certificate Transparency framework

Let us consider a technical trick when the web-browser is split into its components:



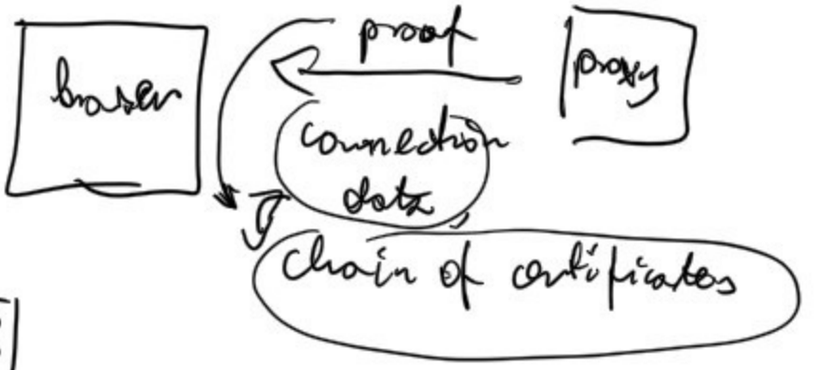
Maybe the web-browser should do the handshake to check if the connection is really end-to-end and then to display the name(s) of subject(s) in the certificate chain.

In TLS 1.3 the chain of certificates goes encrypted.

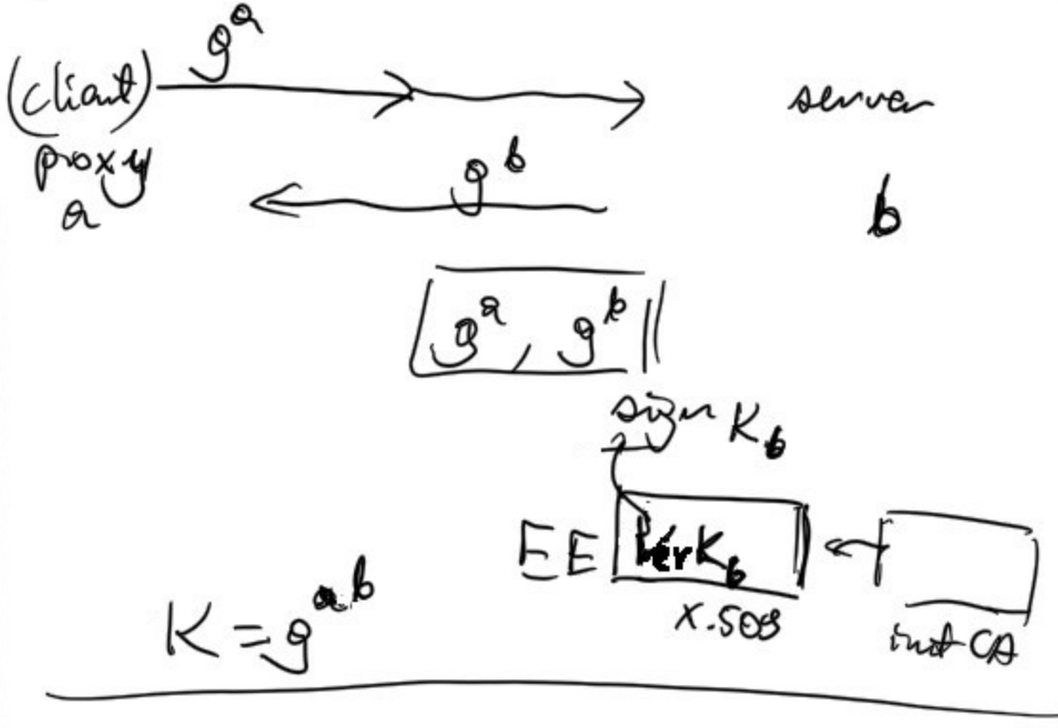
So in TLS 1.3 the proxy must be involved into the TLS handshake.

In case of TLS 1.3 only DH and ECDH are allowed as the methods of establishing the master key for the connection.

On the other hand we would like to leave the web-browser a possibility of verifying if the connection is really genuine, and to bind the connection to the chain of certificates the browser sees.



1.3



$\{g^a$

transcript of the handshake

$\text{sign}_{K_b}(\_\_)$, the chain of certificates