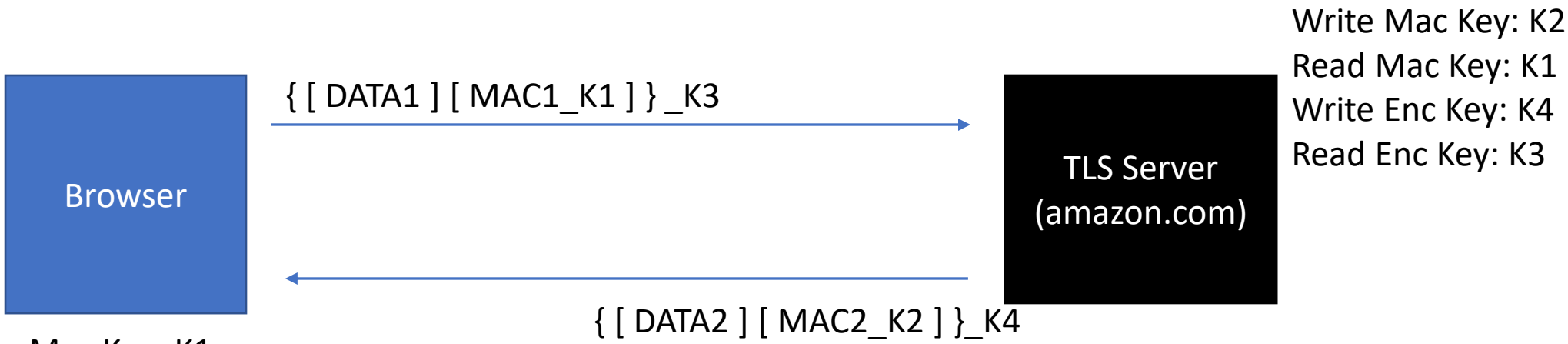
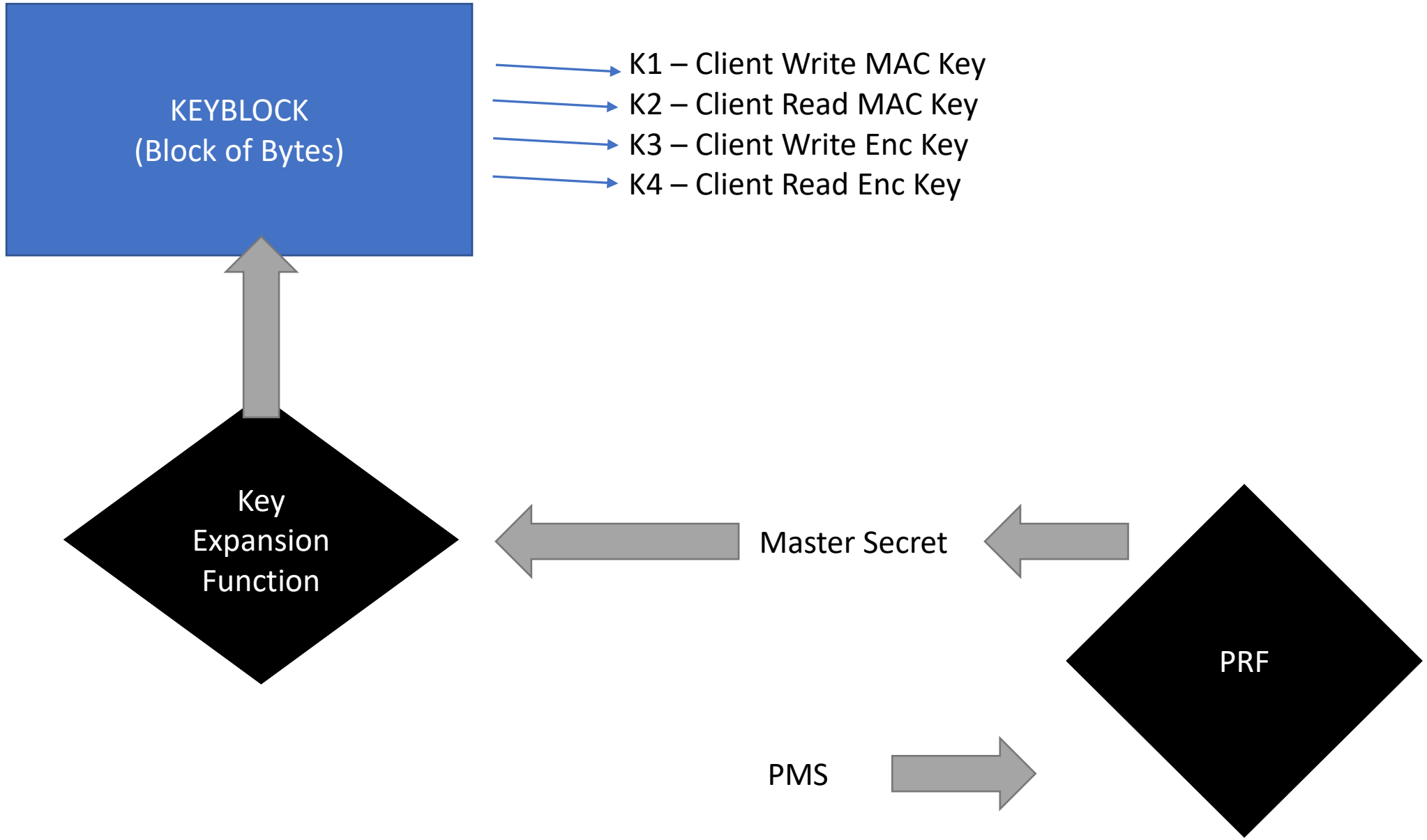
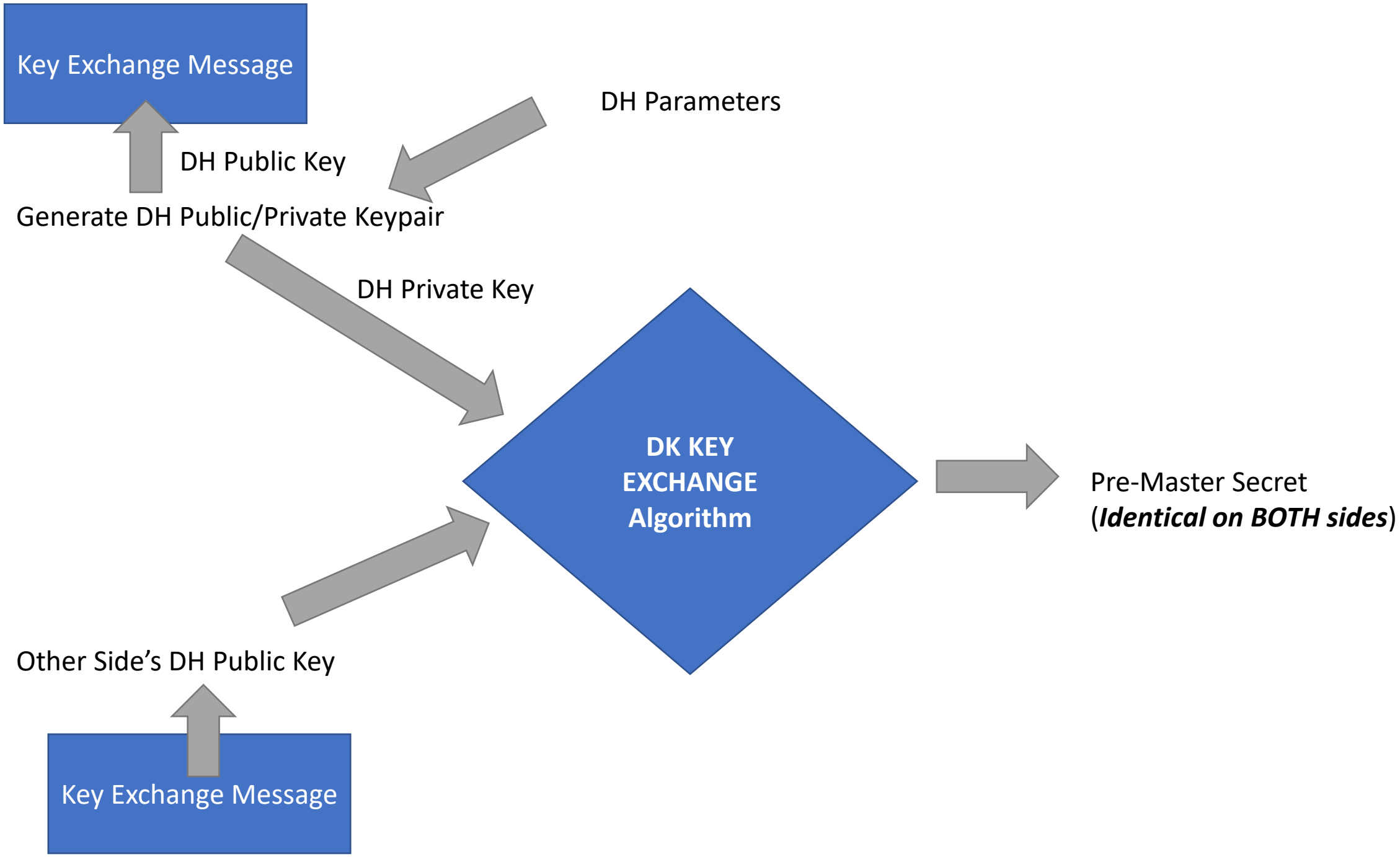


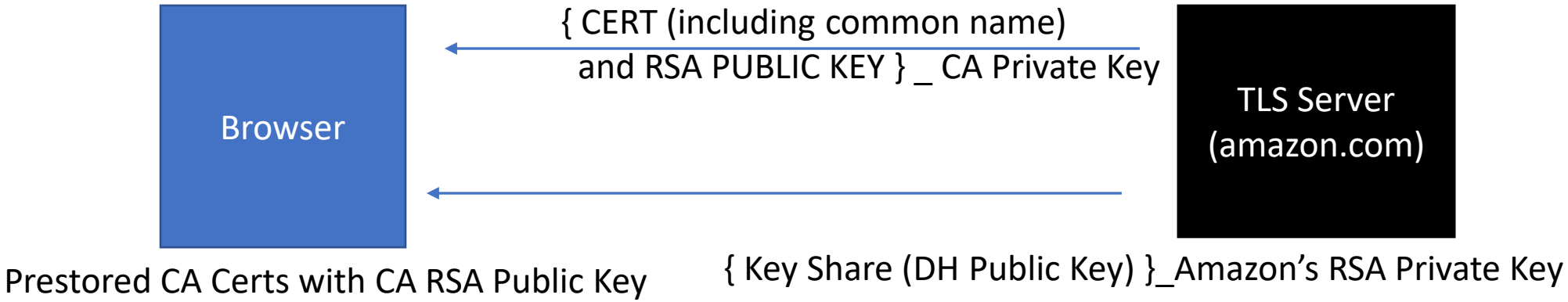
How do we know that the data is:

- From the correct party (either Amazon, or browser)
- Unmodified
- Confidential (to other parties)


KEYBLOCK
(Block of Bytes)
K1 – Client Write MAC Key
K2 – Client Read MAC Key
K3 – Client Write Enc Key
K4 – Client Read Enc Key
Key
Expansion
Function
Master Secret
PRF
PMS


Key Exchange Message
DH Parameters
DH Public Key
Generate DH Public/Private Keypair
DH Private Key
DK KEY EXCHANGE Algorithm
Pre-Master Secret
(*Identical on BOTH sides*)
Other Side's DH Public Key
Key Exchange Message

How does the Client **VERIFY** the DH Public Key from the Server?



Browser ***VERIFIES*** Amazon's cert by:

1. Checking that common name = domain name in URL Bar
2. Verify signature on Cert with CA RSA public key from local storage

Browser ***VERIFIES*** that the DH Public Key is ***SIGNED*** by using the RSA public key of the authentic party from the certificate.

Stealing CA Private Key vs Amazon Private Key

- IF an attacker has Amazon's cert (which is public)…
  - Still cannot impersonate Amazon because it requires Amazon's private key
  - NOTE: can't deduce Amazon's private key from their public key
- IF an attacker steals a CA private key or inserts CA into browser
  - Can generate a NEW public/private keypair
  - Can generate a NEW certificate based off (fake, just generated) public key
  - Can SIGN fake certificate with fake public key using CA Private key