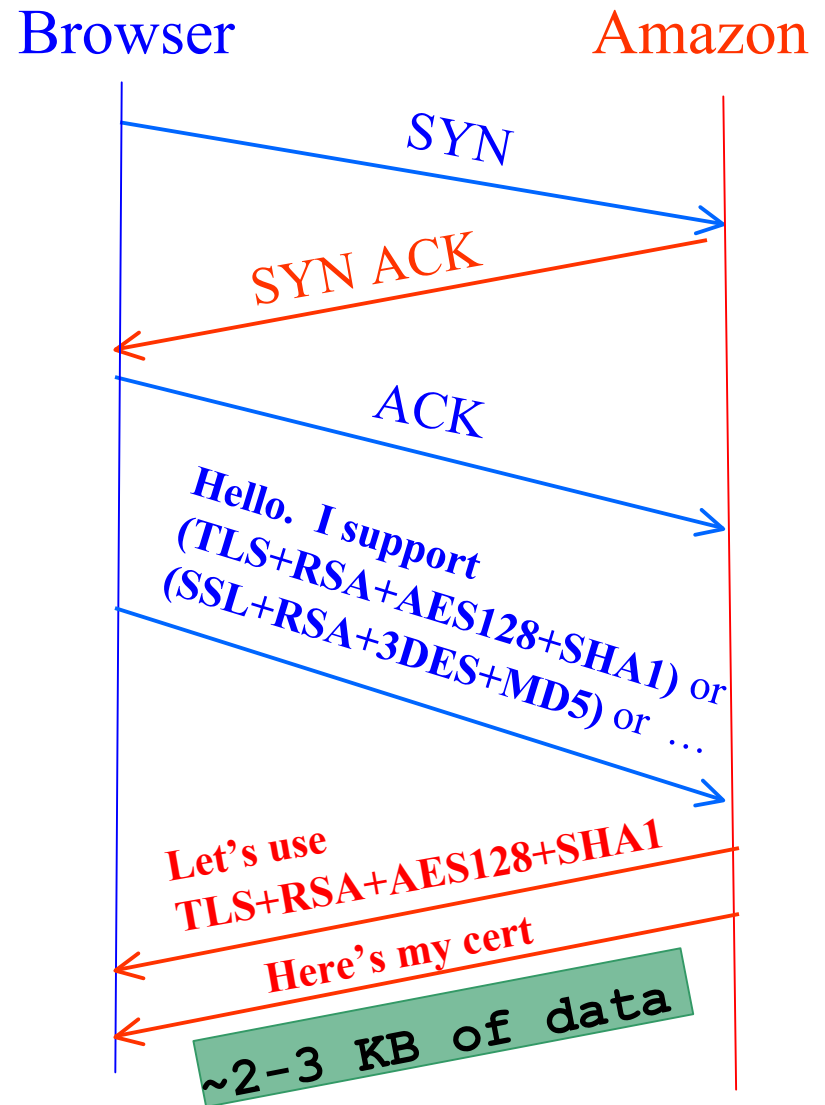



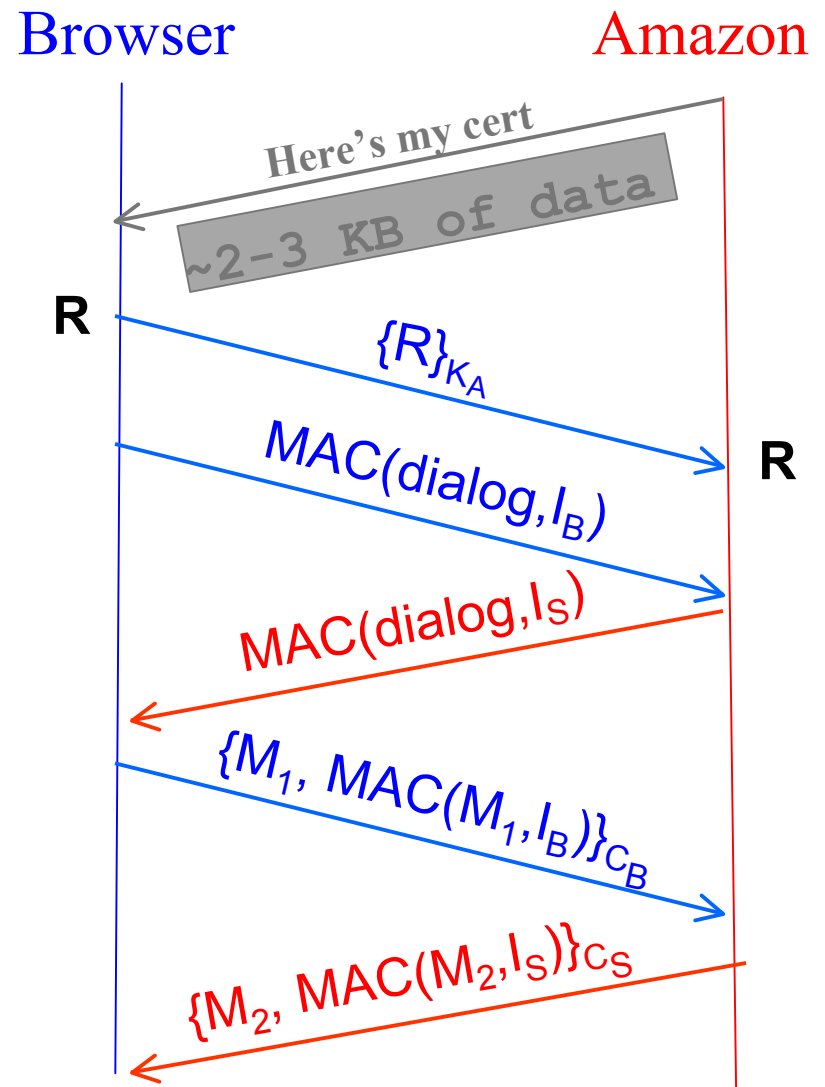
# HTTPS Connection (SSL / TLS)

- Browser (client) connects via TCP to Amazon's **HTTPS** server
- Client sends over list of crypto protocols it supports
- Server picks protocols to use for this session
- Server sends over its certificate
- (all of this is in the clear)
- **Client now validates cert**



# HTTPS Connection (SSL / TLS), con't

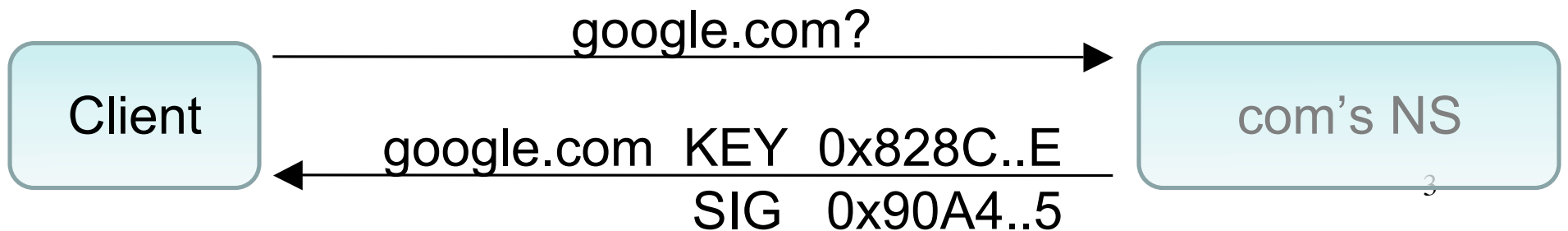
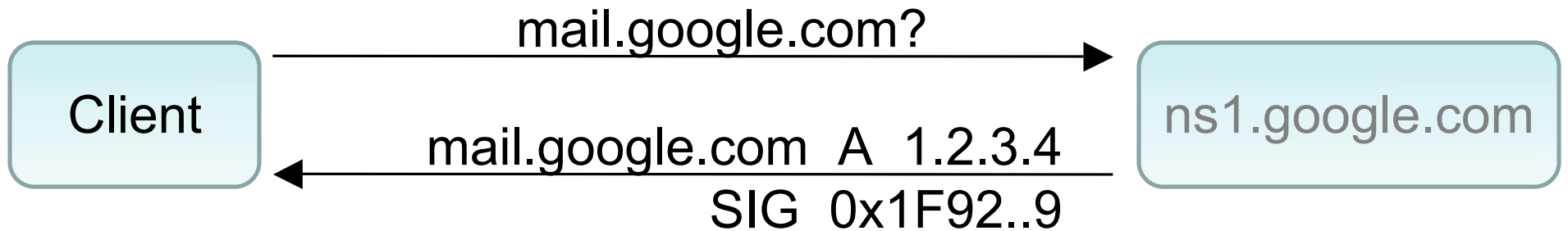
- For RSA, browser constructs a long (2048 bits) random string R
- Browser sends R encrypted using Amazon's public RSA key  $K_A$
- From R browser & server derive pairs of symm. *cipher keys* ( $C_B$ ,  $C_S$ ) and MAC *integrity keys* ( $I_B$ ,  $I_S$ )
  - One pair to use in each direction
- Browser & server exchange MACs computed over entire dialog so far
- If good MAC, Browser displays 
- All subsequent communication encrypted w/ symmetric cipher (e.g., [AES128](#)) cipher keys, MACs
  - Messages also numbered to thwart **replay attacks**



## Ordinary DNS:



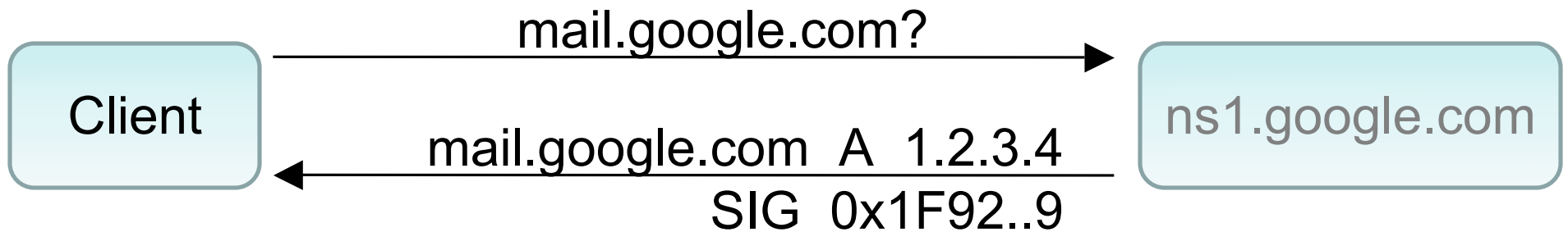
## DNSSEC:



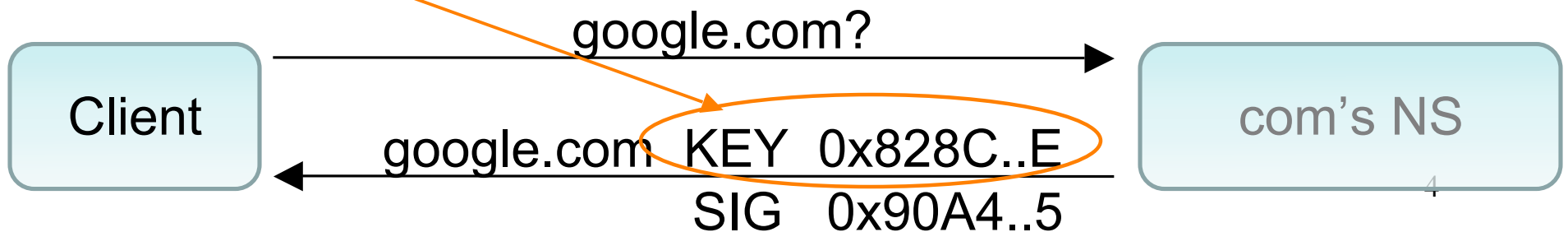
## Ordinary DNS:



## DNSSEC:



This key ...



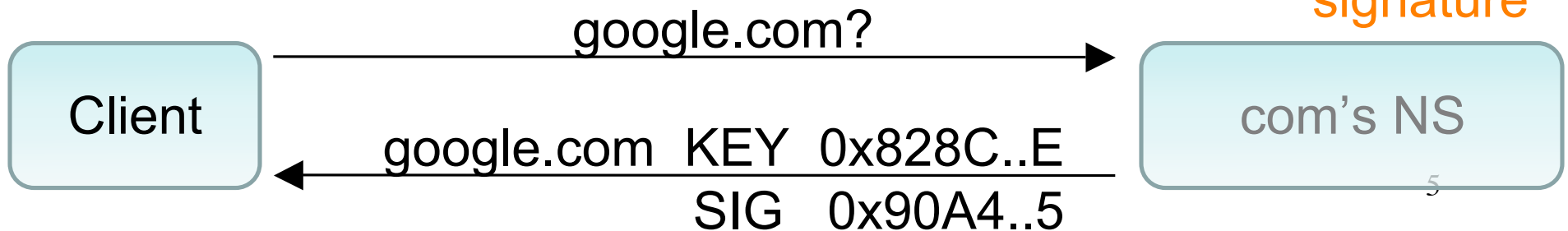
## Ordinary DNS:



## DNSSEC:



... validates this signature



## Ordinary DNS:



## DNSSEC:

