TLS VISIBILITY

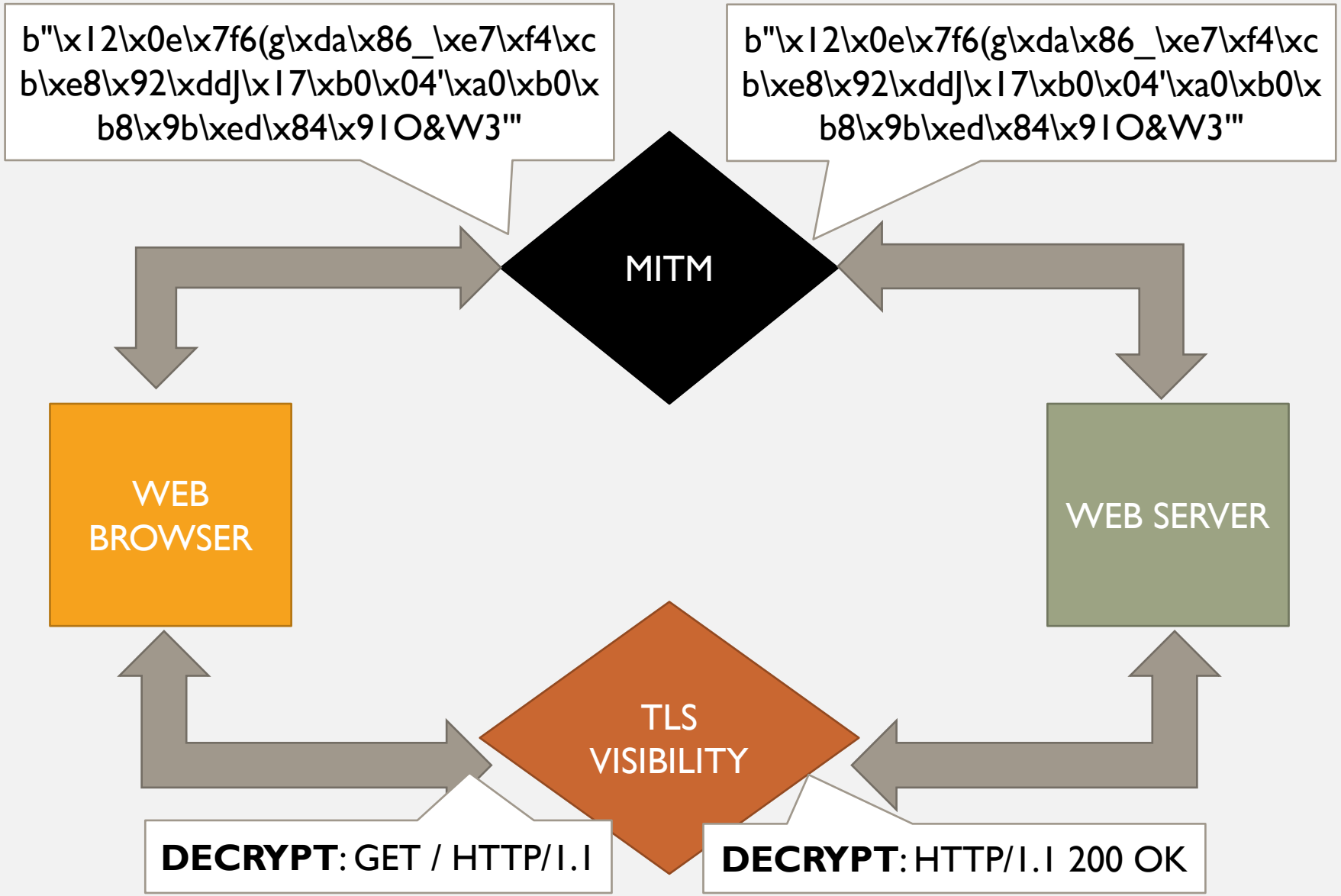
CS 361S

Spring 2020

Seth James Nielson

TLS VISIBILITY

- TLS is designed to provide END-TO-END "security"
- MITM should NOT be able to read/modify/forge data
- TLS Visibility "breaks" this for "authorized" purposes


b"\x12\x0e\x7f6(g\xda\x86_\xe7\xf4\xcb\xe8\x92\xddJ\x17\xb0\x04'\xa0\xb0\xb8\x9b\xed\x84\x91O&W3'"
b"\x12\x0e\x7f6(g\xda\x86_\xe7\xf4\xcb\xe8\x92\xddJ\x17\xb0\x04'\xa0\xb0\xb8\x9b\xed\x84\x91O&W3'"
MITM
WEB BROWSER
WEB SERVER
TLS VISIBILITY
DECRYPT: GET / HTTP/1.1
DECRYPT: HTTP/1.1 200 OK

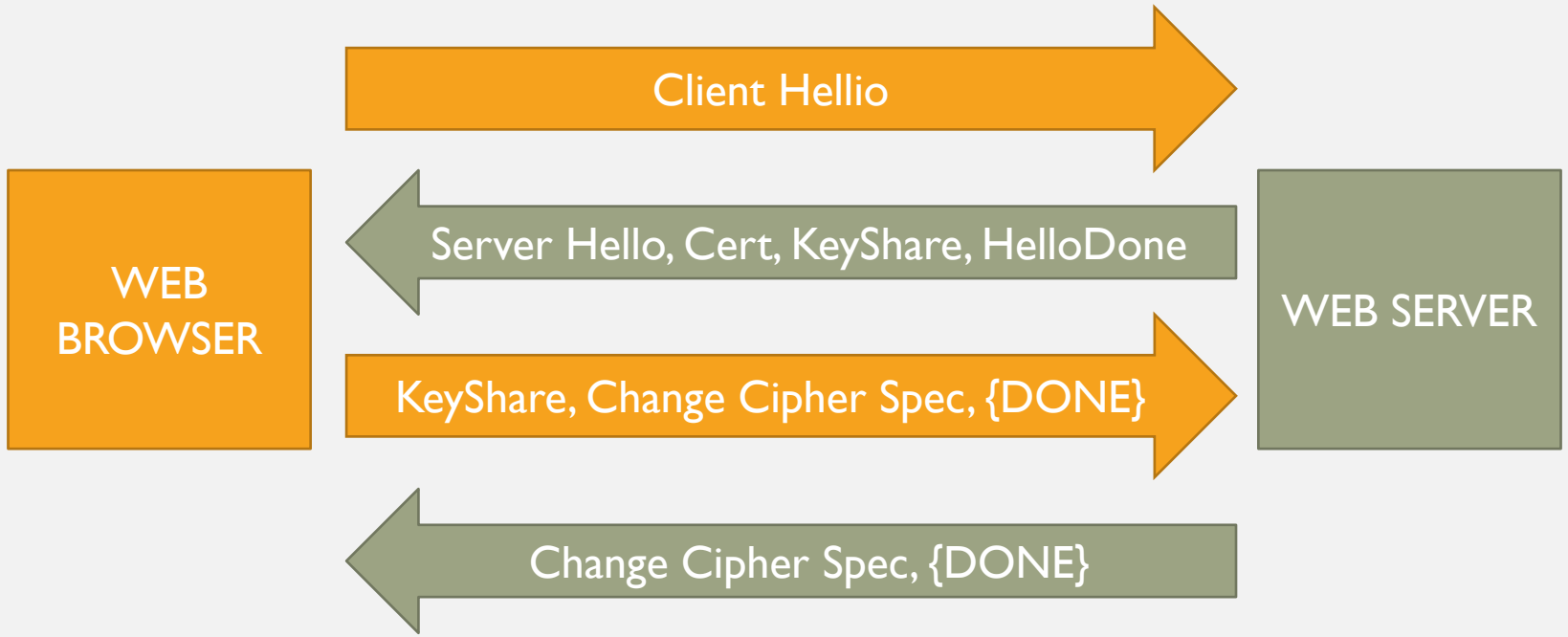
TLS 1.2 HANDSHAKE

Goal	Agree on a cipher suite for encryption, authentication, etc.
Goal	Identify the server (and optionally the client)
Goal	Create session keys for bi-directional communication

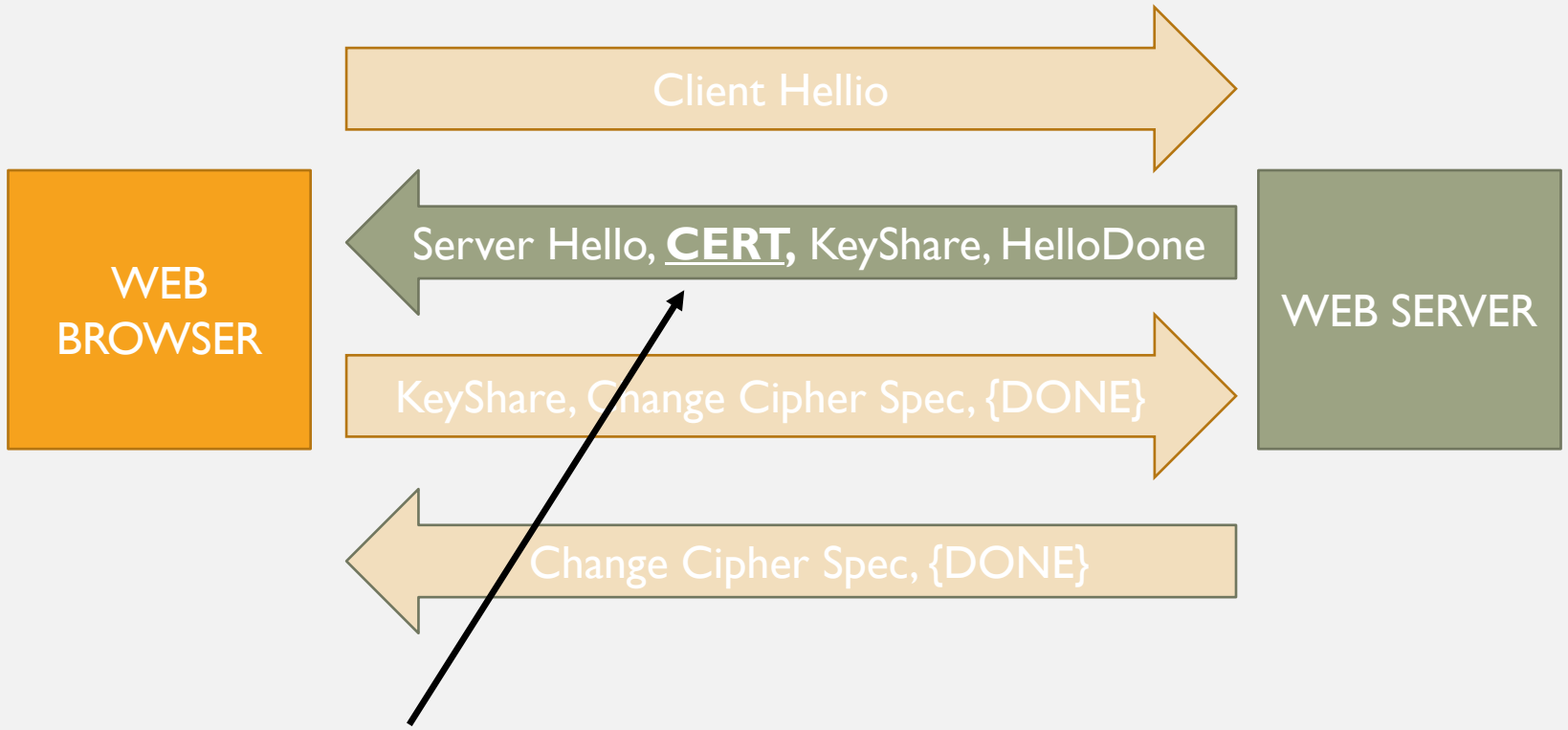
TLS 1.2 HANDSHAKE REVIEW

Step	Client	Direction	Message	Direction	Server
1			Client Hello	>	
2		<	Server Hello		
3		<	Certificate		
4		<	Server Key Exchange		
5		<	Server Hello Done		
6			Client Key Exchange	>	
7			Change Cipher Spec	>	
8			Finished	>	
9		<	Change Cipher Spec		
10		<	Finished		

END-TO-END HANDSHAKE VISUALIZATION #2



AUTHENTICATION



The "Certificate" message includes ONE OR MORE certificates.

CERTIFICATE VERIFICATION

WEB BROWSER

Verify "amazon.com" is the URL
Verify the validity period
(Other Verification)
Who issued the cert?

CERTIFICATE

Subject CN:	**amazon.com**
Not Valid Before:	**2001**
Not Valid After:	**2030**
Issued By:	**amazon CA**
Signature Blob:	**<sig>**

CERTIFICATE CHAINS

The certificate for the Host may be signed by an INTERMEDIATE Certificate Authority

Because the web browser probably doesn't have this intermediate cert, the TLS handshake includes both certificates.

Subject CN: amazon CA
…
Issued By: GlobalSign
Signature Blob: <sig>

Subject CN: amazon.com
…
Issued By: amazon CA
Signature Blob: <sig>

ROOT CA CERTIFICATES

Certificate chains MUST have a ROOT

A Root Certificate is SELF SIGNED

Browsers trust a set of root certificates AXIOMATICALLY

Certificate chains must have a trust chain to one of these roots.

TRUSTING DIFFIE HELLMAN

Recall that DH keys are EPHEMERAL

The Server's cert includes a long-term public key

The Server's DH key is signed by this key pair

IF the client trusts the cert, THEN it can validate the DH key

TLS BULK TRANSPORT

Both Client and Server derive keys

Encryption keys AND MAC keys

MAC's ensure continuous authentication

WHEN A TLS MESSAGE IS RECEIVED:

The sender is "proved" by the MAC

The MAC is "proved" via MAC key derived from DH

Server's DH key "proved" authentic by cert signature

Certificate "proved" authentic by chain to trusted root

IT ALL DEPENDS ON THE CERT

IF a browser trusts MY certificate to be Amazon's certificate

- THEN the browser will trust my DH public key

IF the browser trusts my DH public key

- THEN the browser will derive the same MAC key I do

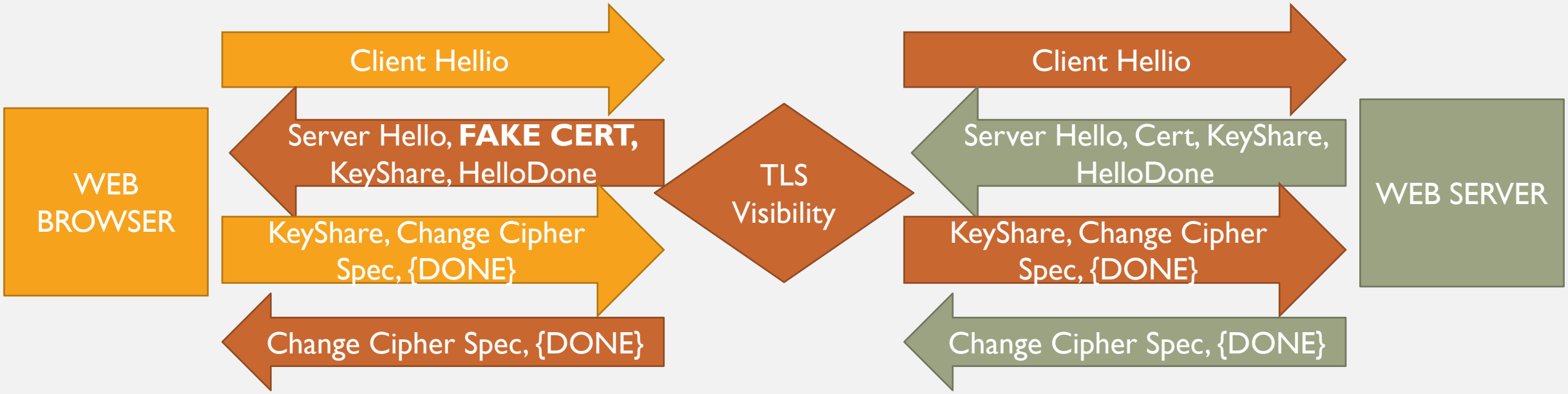
IF the browser derives the same MAC key I do

- THEN the browser will believe my messages are from Amazon

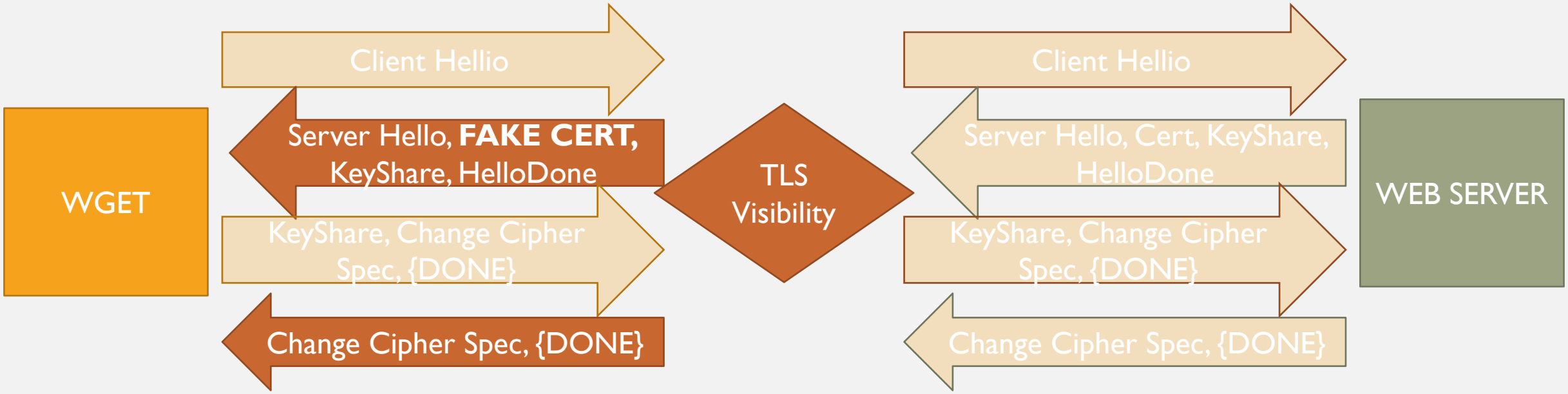
TLS VISIBILITY

- Typically, a browser/client MUST have a new root CA installed
- This root CA is a self-signed certificate from the Visibility appliance
- The appliance can now generate ANY cert and the browser believes it!
- We will discuss the huge security concerns in a later lecture

TLS VISIBILITY HANDSHAKE VISUALIZATION



YOUR LAB:



You will have to read the client messages, but the only TLS packets you have to create are for the MITM

You do NOT write the TLS connection to the web server.