TLS Performance

John Graham-Cumming

December 11, 2014

CLOUDFLARE

Two Worries

- Will using HTTPS use a great deal of CPU?
- Will using HTTPS add latency to a connection?

CLOUDFLARE

Two Worries

- Will using HTTPS use a great deal of CPU? **No**
- Will using HTTPS add latency to a connection?

CLOUDFLARE

Two Worries

- Will using HTTPS use a great deal of CPU?
- Will using HTTPS add latency to a connection?

No

CLOUDFLARE

Two Worries

- Will using HTTPS use a great deal of CPU?
- Will using HTTPS add latency to a connection?

Not much

CLOUDFLARE

Basic TLS Knowledge

- TLS has three important components
  - Key Exchange
  - Encryption
  - Integrity

CLOUDFLARE

Basic TLS Knowledge

- TLS has three important components
  - Key Exchange **At start of connection**
  - Encryption
  - Integrity

CLOUDFLARE

Basic TLS Knowledge

- TLS has three important components
  - Key Exchange
  - Encryption
  - Integrity

At start of connection

Throughout connection

CLOUDFLARE

Basic TLS Knowledge

- TLS has three important components
  - Key Exchange
  - Encryption
  - Integrity

Use public key/asymmetric schemes

CLOUDFLARE

Basic TLS Knowledge

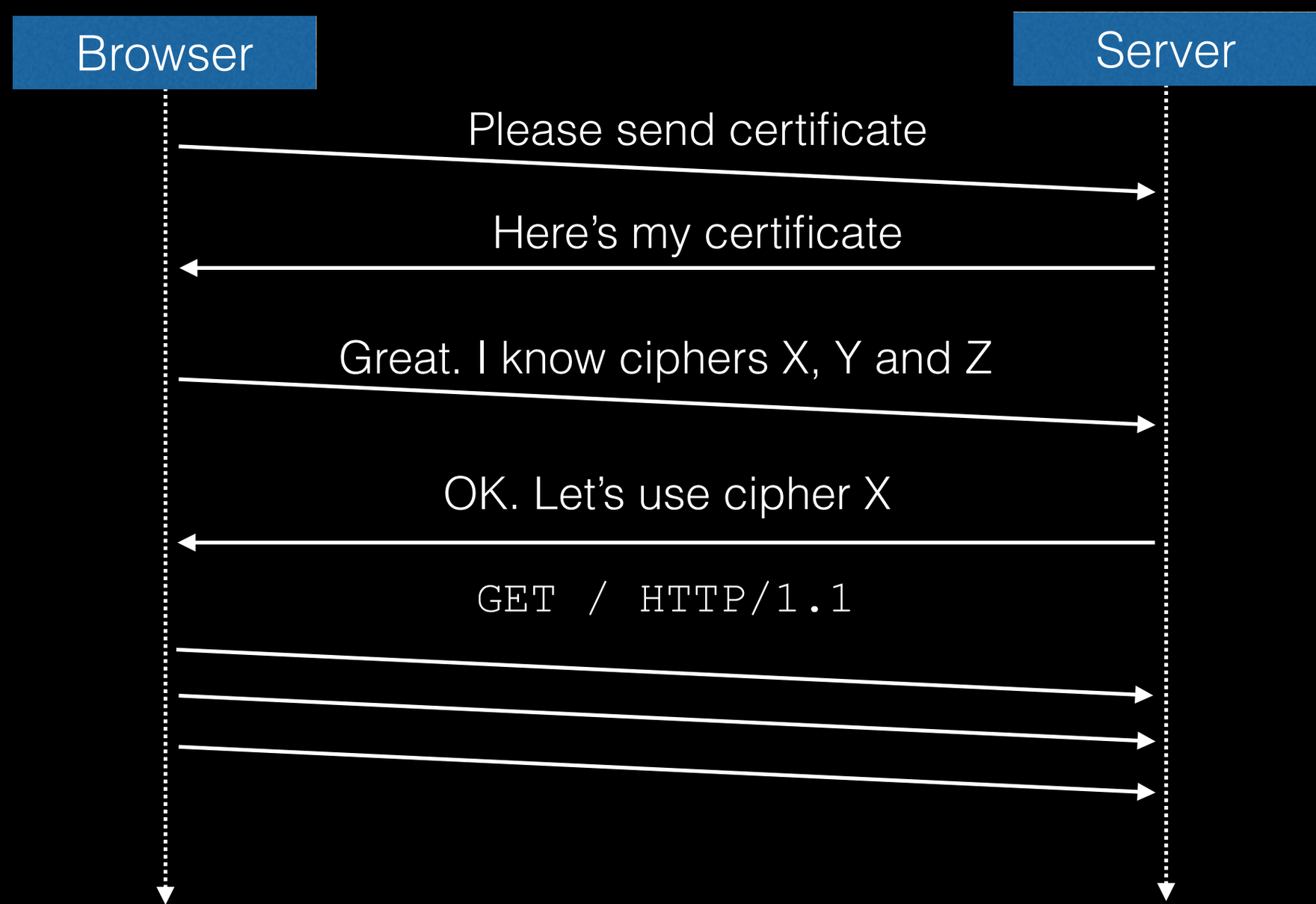
- TLS has three important components
  - Key Exchange
  - Encryption
  - Integrity

Use public key/asymmetric schemes

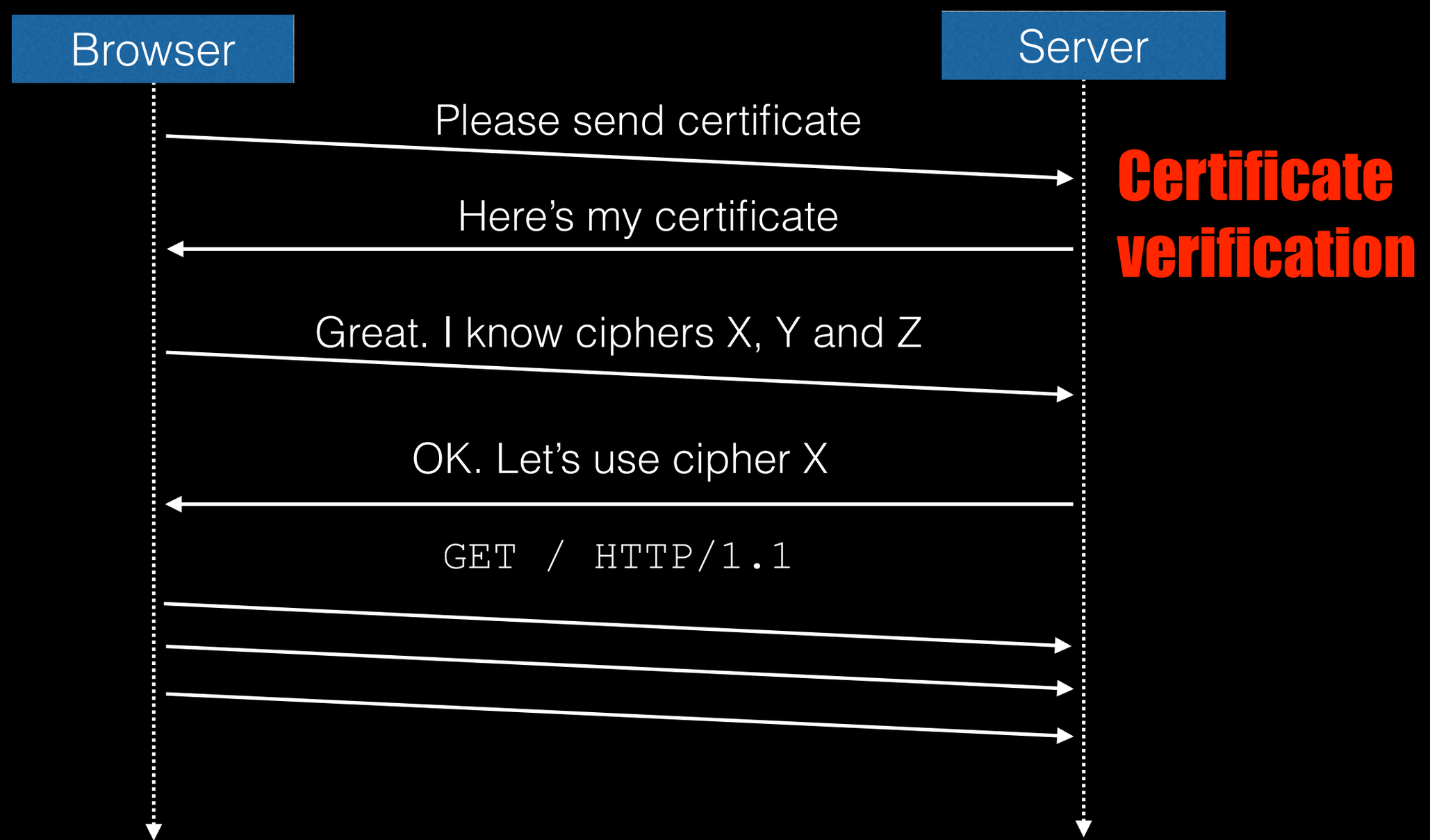
Use symmetric schemes

CLOUDFLARE

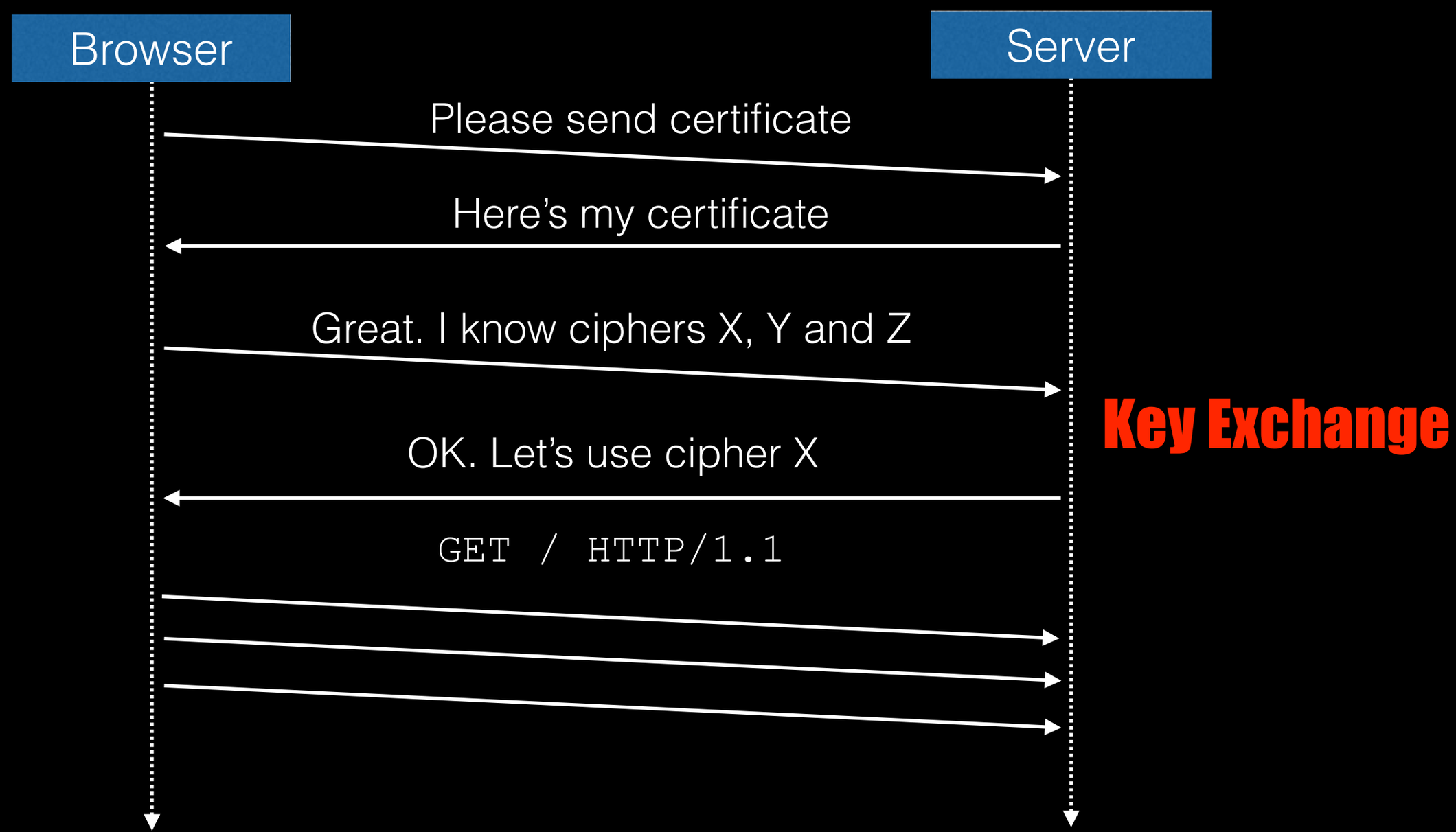
Basic TLS Handshake



Basic TLS Handshake

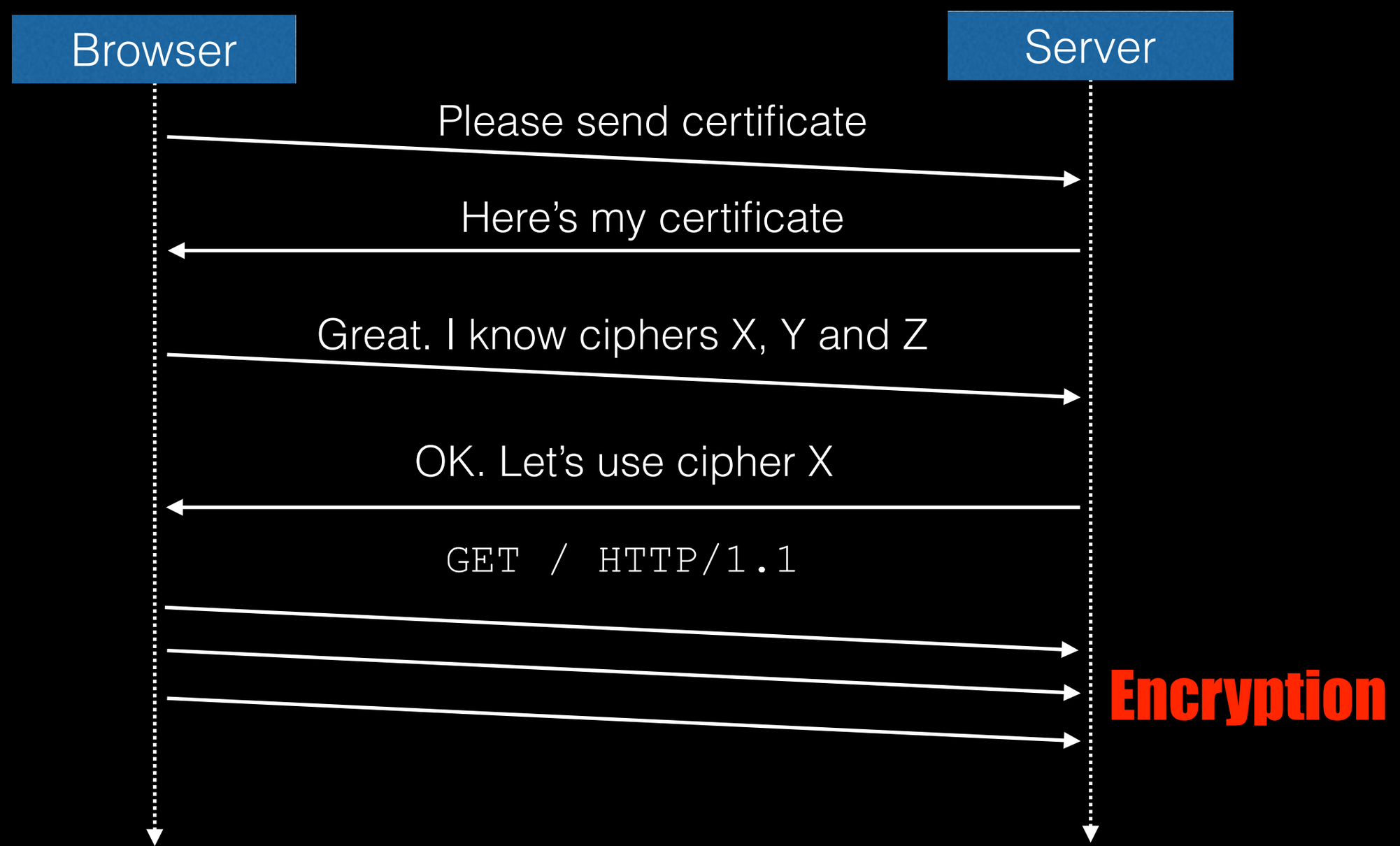


Basic TLS Handshake



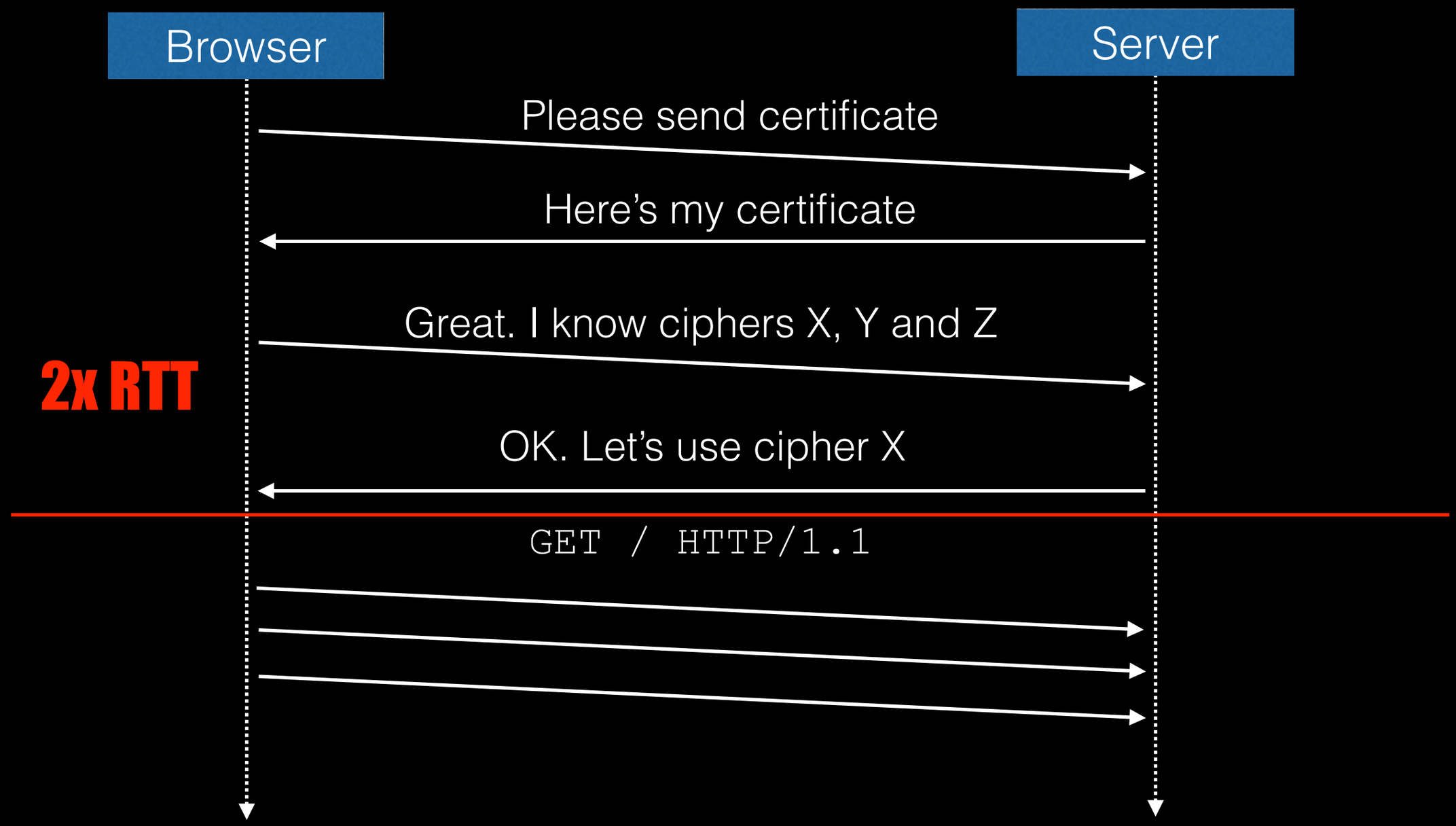
CLOUDFLARE

Basic TLS Handshake



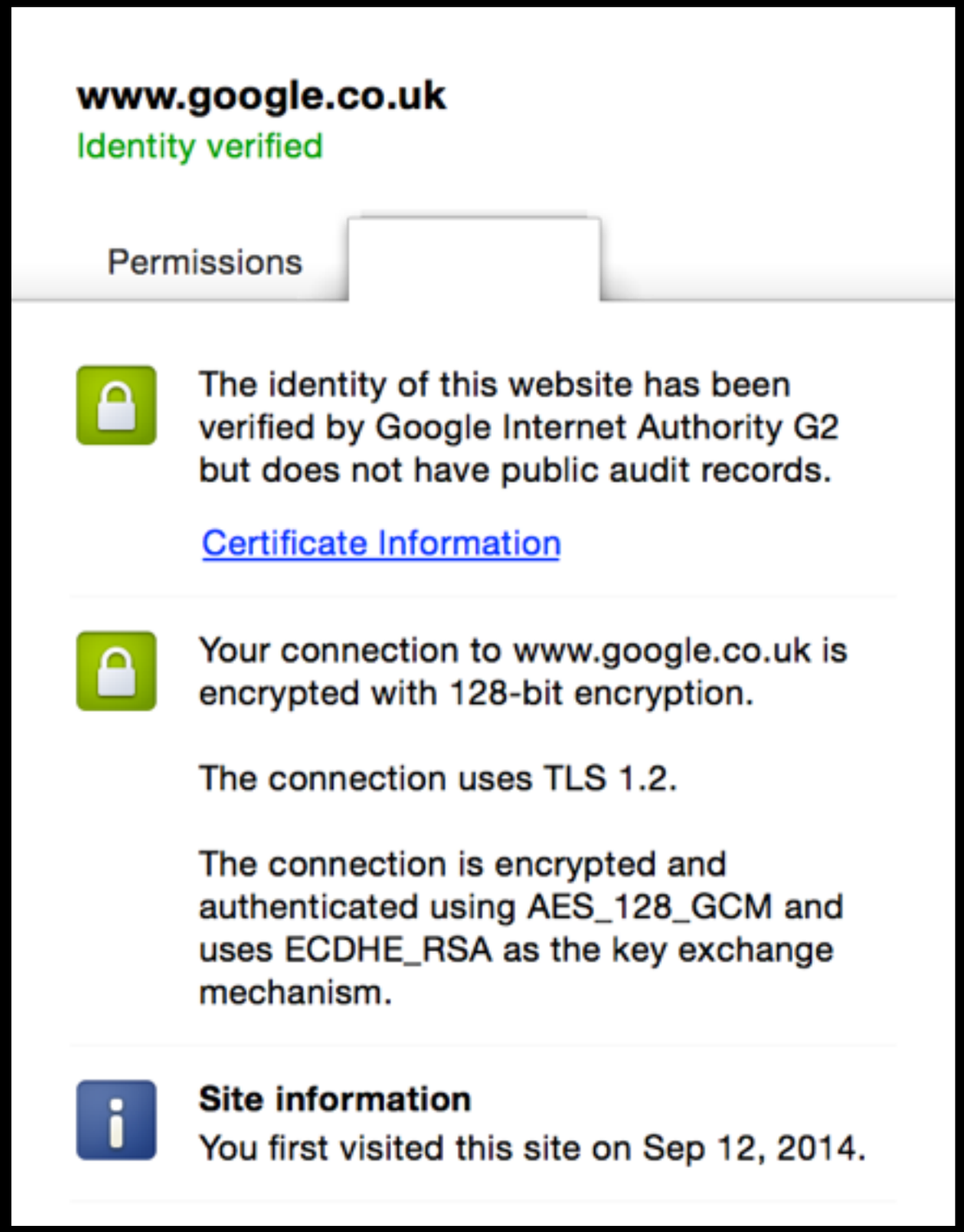
CLOUDFLARE

Basic TLS Handshake



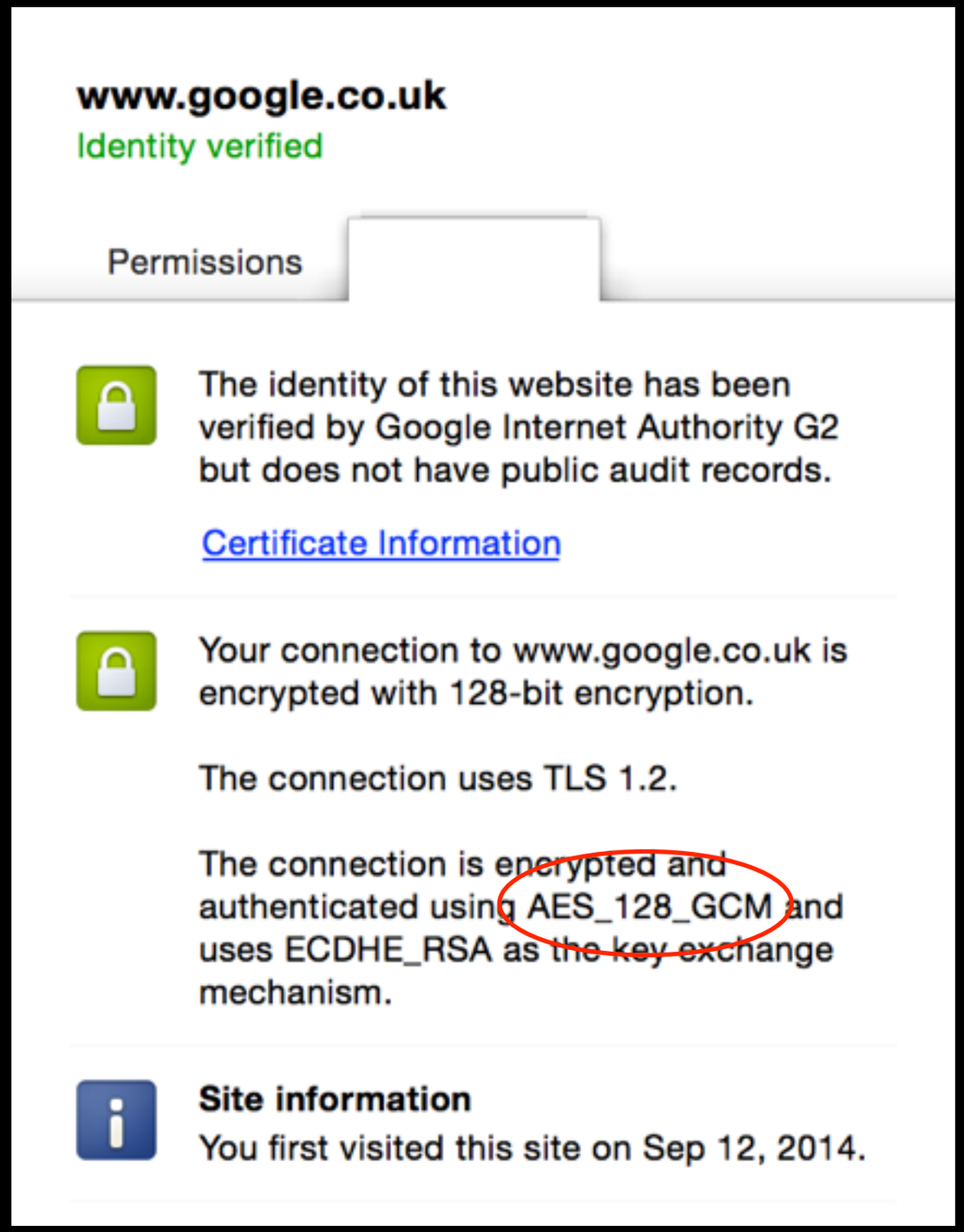
CLOUDFLARE

google.co.uk



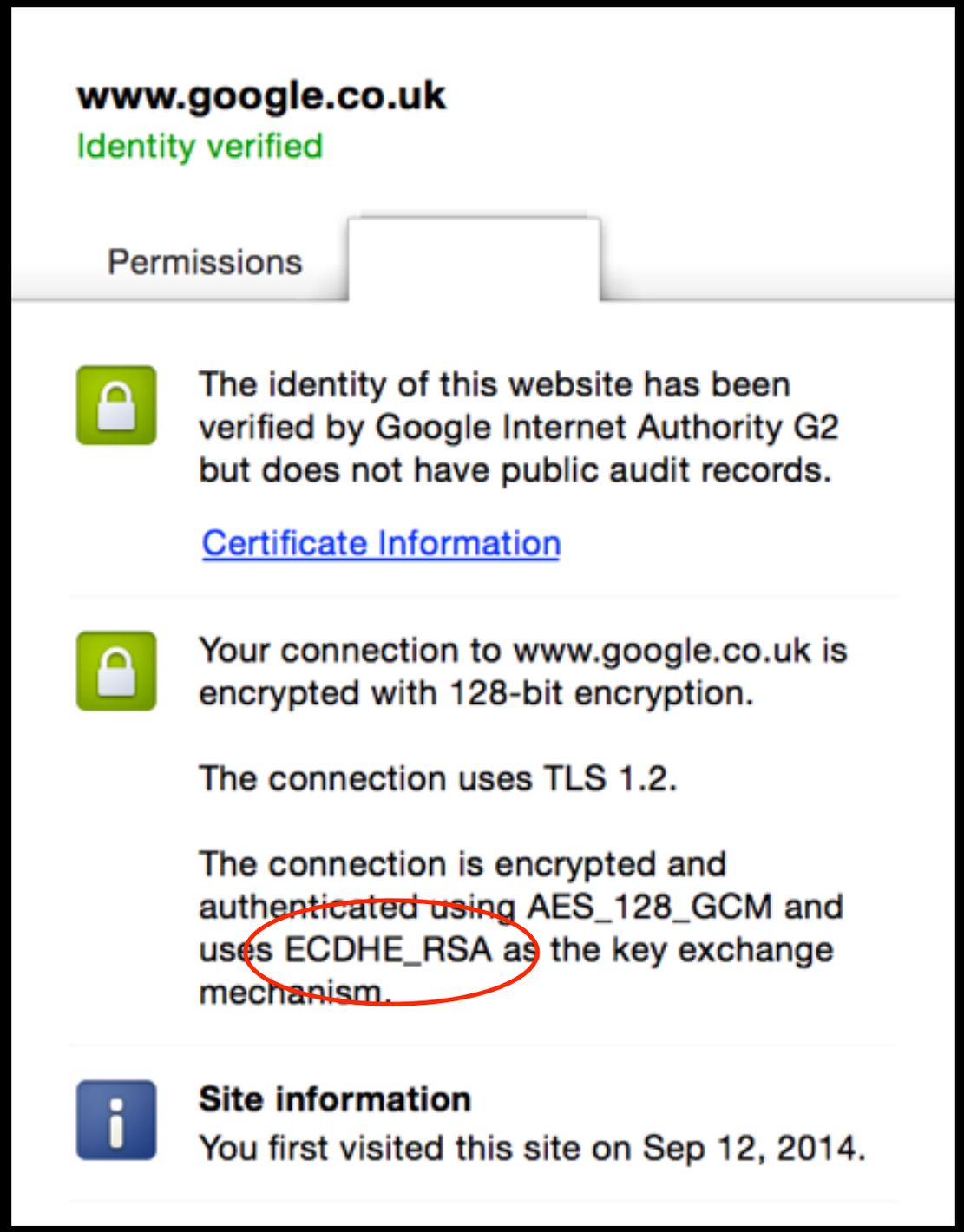
CLOUDFLARE

google.co.uk



CLOUDFLARE

google.co.uk



CLOUDFLARE

Key Exchange

```
% openssl speed rsa
                  sign    verify    sign/s verify/s
rsa  512 bits 0.000047s 0.000004s  21453.4 248742.9
rsa 1024 bits 0.000167s 0.000012s   5992.6  85880.8
rsa 2048 bits 0.001226s 0.000035s    815.3  28217.1
rsa 4096 bits 0.007799s 0.000123s    128.2   8119.5
```

```
% openssl speed ecdh
                                  op      op/s
  160 bit ecdh (secp160r1)   0.0002s   5706.4
  192 bit ecdh (nistp192)    0.0002s   4619.1
  224 bit ecdh (nistp224)    0.0001s   8137.1
  256 bit ecdh (nistp256)    0.0002s   4796.2
  384 bit ecdh (nistp384)    0.0006s   1539.4
  521 bit ecdh (nistp521)    0.0007s   1408.7
  163 bit ecdh (nistk163)    0.0002s   5273.2
  233 bit ecdh (nistk233)    0.0003s   3923.2
  283 bit ecdh (nistk283)    0.0005s   2070.1
  409 bit ecdh (nistk409)    0.0008s   1286.5
  571 bit ecdh (nistk571)    0.0018s    566.7
  163 bit ecdh (nistb163)    0.0002s   4572.2
  233 bit ecdh (nistb233)    0.0003s   3841.2
  283 bit ecdh (nistb283)    0.0005s   2017.7
  409 bit ecdh (nistb409)    0.0008s   1223.6
  571 bit ecdh (nistb571)    0.0021s    486.3
```

CLOUDFLARE

Encryption/Integrity

```
% openssl speed rc4
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
rc4            377761.26k   634763.28k   771228.76k   744920.27k   728186.61k
```

```
% openssl speed sha
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
sha1            74722.31k   211757.63k   447491.08k   648227.50k   778855.77k
sha256          52784.27k   117919.96k   204055.46k   233204.18k   247332.42k
sha512          42471.15k   172133.21k   268125.41k   358115.75k   355905.41k
```

CLOUDFLARE

RC4

https://www.youtube.com/watch?v=NKnZARFAhMk

CLOUDFLARE

Rough Numbers

- Key Exchange crypto takes roughly 1ms
- Symmetric crypto used for encryption is very fast
  - Easily do >100Mbps per core on commodity hardware

CLOUDFLARE

Rough Numbers

Try to mitigate setup time

- Key Exchange crypto takes < 1ms
- Symmetric crypto used for encryption is very fast
  - Easily do >100Mbps per core on commodity hardware

CLOUDFLARE

You don’t need special hardware

- Modern CPUs have crypto-specific instructions
  - Intel/AMD: AES-NI
- OpenSSL has CPU-specific optimizations for key algorithms

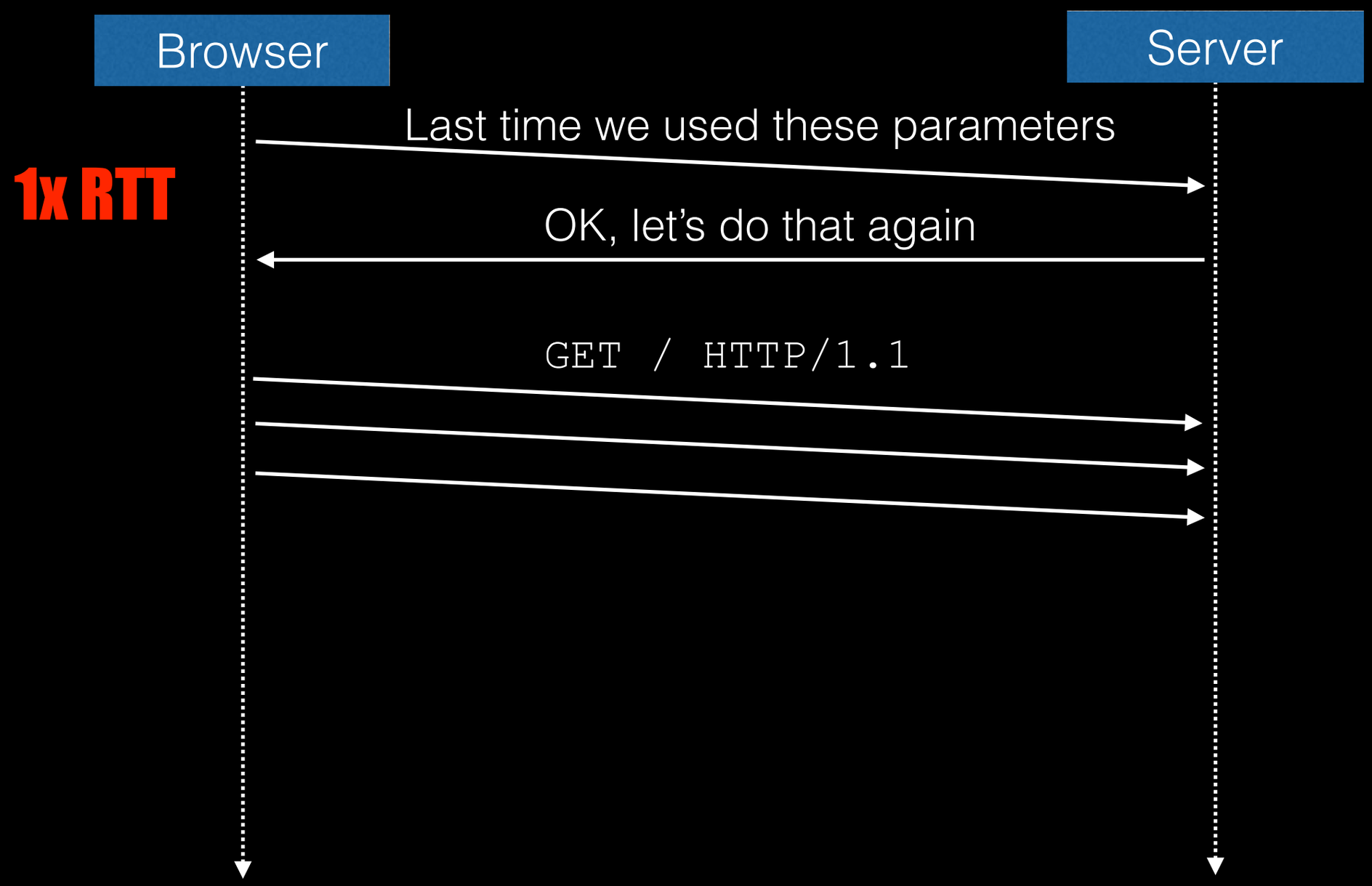
CLOUDFLARE

Speed up start up

- Session resumption
- Use of CDNs
- OCSP Stapling
- Tune web server windowing

CLOUDFLARE

Session Resumption



Session Identifiers

- Server invents a session ID and sends to client
- When client returns later, server uses session ID to find parameters
- Server must store the session ID to parameter mapping
- Session IDs/parameters need to be shared across servers
- Sessions IDs need to be expired

CLOUDFLARE

Session Tickets

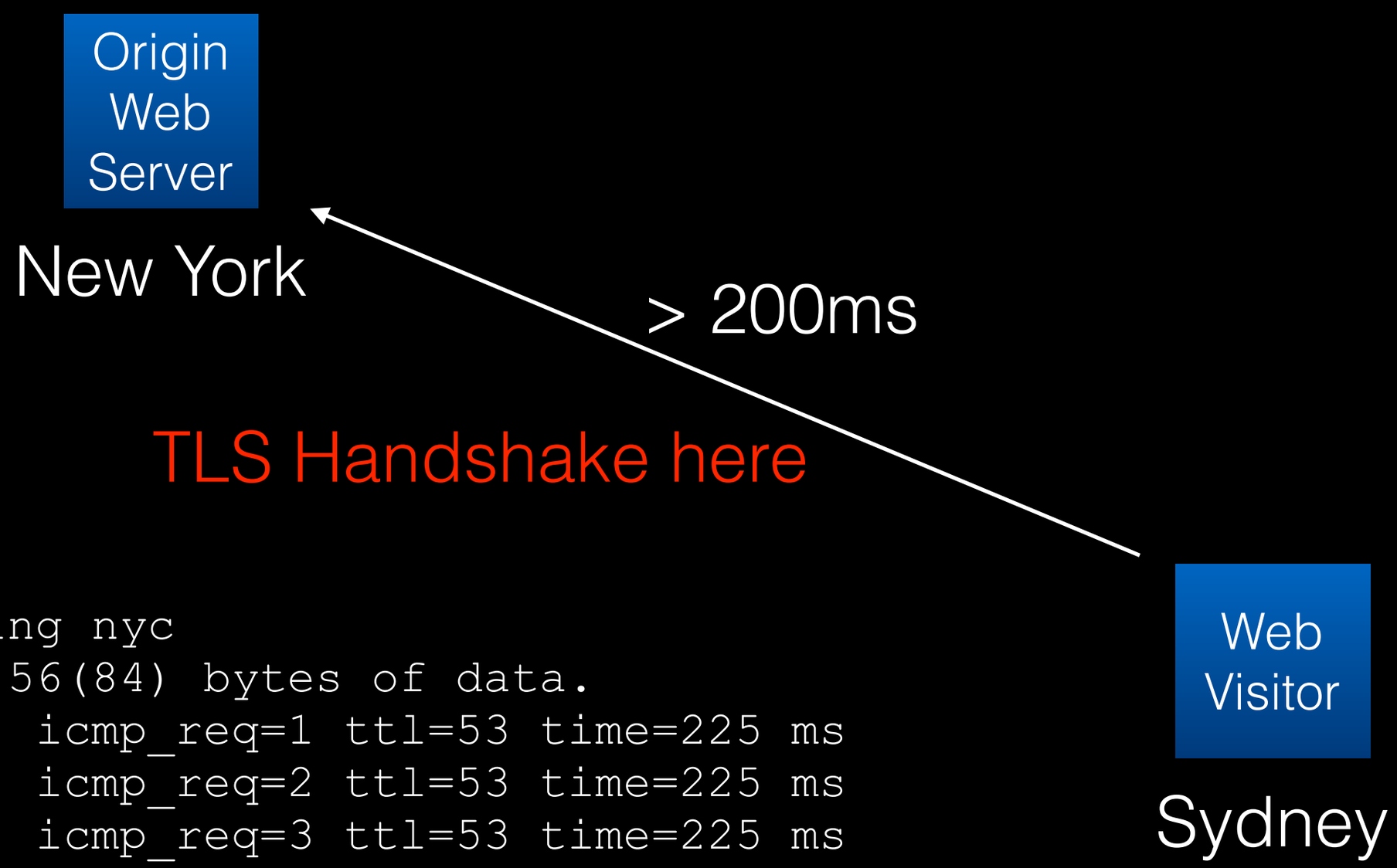
- Server encrypts session parameters and sends to client
- Encrypted parameters are opaque to client (binary blob)
- On restart client sends blob to server; server decrypts and session resumes
- Servers need to have a shared encryption key for tickets

CLOUDFLARE

```
% openssl s_client -connect google.co.uk:443 -tls1 -tlsextdebug -status
[…]
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-RC4-SHA
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1
    Cipher    : ECDHE-RSA-RC4-SHA
    Session-ID: 5F3287224163C89146A376C0B520E0AE5DD872A1FDCB3196BF01EE9B647AC61C
    Session-ID-ctx:
    Master-Key:
90A74DCF42A197050337A5969705632F33073050B7980E92085FE71125DFA3BB291AF080E109975B
564929D30A30C765
    Key-Arg   : None
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 100800 (seconds)
    TLS session ticket:
    0000 - 01 50 71 68 74 8d c1 bf-37 0f 3a 17 b8 5c f8 ed    .Pqht...7.:..\..
    0010 - bb 29 7b 99 9b 3d d3 72-f4 e4 cf 24 65 81 ad d0    .){..=.r...$e...
    0020 - 08 db fe dd 96 5c db c9-4c ba f0 d2 c8 c1 53 28    .....\..L.....S(
    0030 - b0 c6 3d 0b be c0 f9 3b-96 5a 26 28 21 92 21 c2    ..=....;.Z&(!.!.
    0040 - bb d8 99 96 57 3e 40 f8-b3 58 75 10 40 e5 2c 58    ....W>@..Xu.@.,X
    0050 - 5c 4e 7d 39 b3 53 ee 6e-e3 0f 33 83 ad 62 d1 34    \N}9.S.n..3..b.4
    0060 - 36 c3 14 97 be eb aa 03-f3 cd 6c f1 22 e0 c2 0b    6.........l."...
    0070 - ca 62 24 ef 95 ad 6d 3d-42 5b bb 0b b8 ac c4 b2    .b$...m=B[......
    0080 - e4 ee 6d dc 85 29 ed fc-ef 5c 33 20 ab c6 7c 87    ..m..)...\3 ..|.
    0090 - 24 73 a8 a5 c3 79 1d 19-ff f3 c4 19 c3 42 e8 f2    $s...y.......B..
    00a0 - 12 18 7a b5                                        ..z.
```

CLOUDFLARE

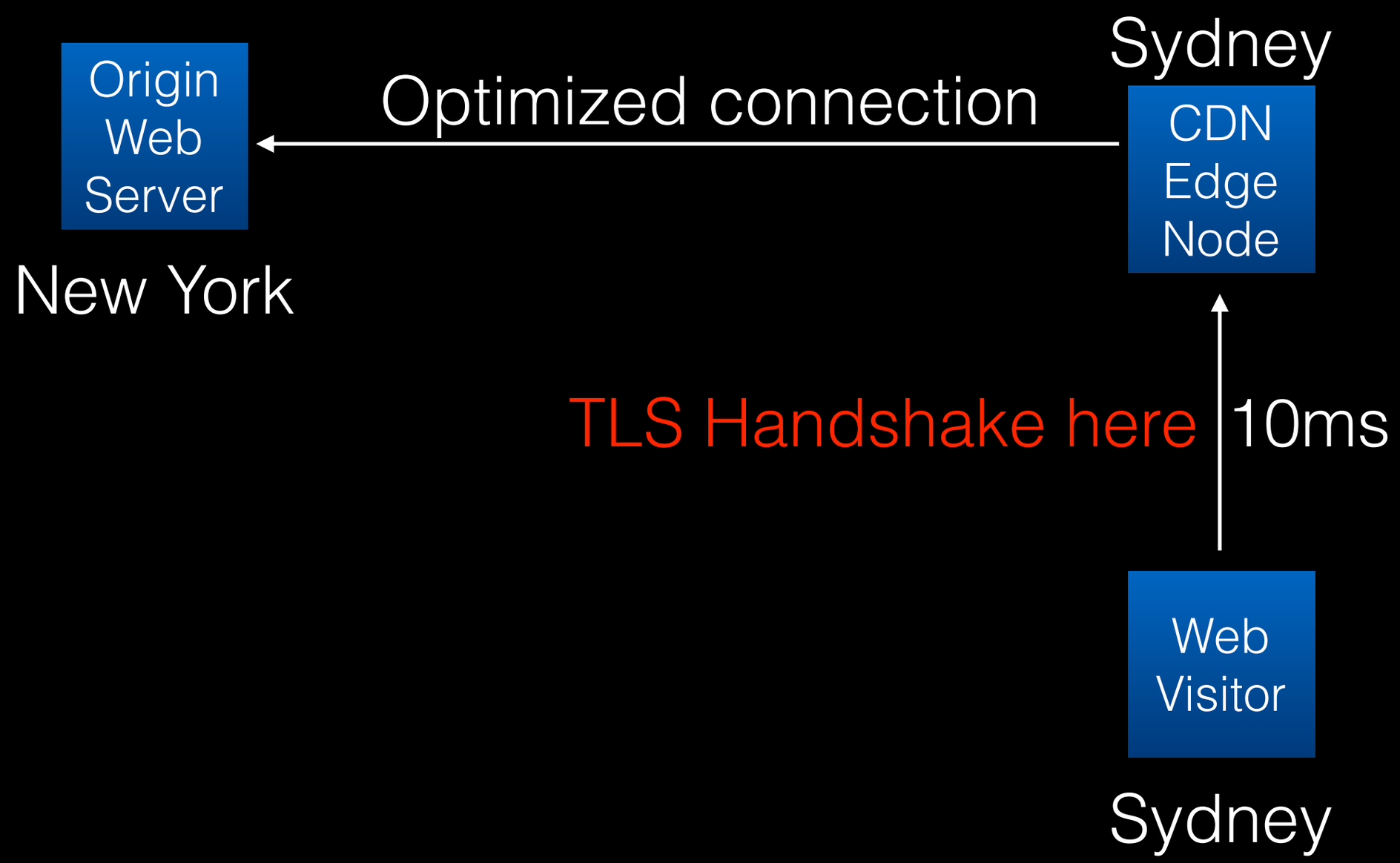
Use a CDN that supports TLS




```
syd:~$ ping nyc
PING nyc 56(84) bytes of data.
64 bytes: icmp_req=1 ttl=53 time=225 ms
64 bytes: icmp_req=2 ttl=53 time=225 ms
64 bytes: icmp_req=3 ttl=53 time=225 ms
64 bytes: icmp_req=4 ttl=53 time=225 ms
```

CLOUDFLARE

Use a CDN that supports TLS



OCSP

11	*REF*	172.16.90.156	63.245.216.134	TCP	76	41683 → https [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=28896863 TSecr=0 WS
12	0.150458000	63.245.216.134	172.16.90.156	TCP	62	https → 41683 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
13	0.150530000	172.16.90.156	63.245.216.134	TCP	56	41683 → https [ACK] Seq=1 Ack=1 Win=29200 Len=0
14	0.150869000	172.16.90.156	63.245.216.134	SSL	236	Client Hello
15	0.151051000	63.245.216.134	172.16.90.156	TCP	62	https → 41683 [ACK] Seq=1 Ack=181 Win=64240 Len=0
16	0.306432000	63.245.216.134	172.16.90.156	TCP	1516	[TCP segment of a reassembled PDU]
17	0.306461000	172.16.90.156	63.245.216.134	TCP	56	41683 → https [ACK] Seq=181 Ack=1461 Win=32120 Len=0
18	0.306516000	63.245.216.134	172.16.90.156	TCP	1516	[TCP segment of a reassembled PDU]
19	0.306524000	172.16.90.156	63.245.216.134	TCP	56	41683 → https [ACK] Seq=181 Ack=2921 Win=35040 Len=0
20	0.306588000	63.245.216.134	172.16.90.156	TLSv1.2	1512	Server Hello, Certificate, Server Hello Done
21	0.306594000	172.16.90.156	63.245.216.134	TCP	56	41683 → https [ACK] Seq=181 Ack=4377 Win=37960 Len=0
22	0.333772000	172.16.90.156	63.245.216.134	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
23	0.334232000	63.245.216.134	172.16.90.156	TCP	62	https → 41683 [ACK] Seq=4377 Ack=523 Win=64240 Len=0
24	0.338777000	127.0.0.1	127.0.1.1	DNS	79	Standard query 0x5f62 A ocsp.verisign.com
25	0.338833000	172.16.90.156	172.16.90.2	DNS	79	Standard query 0x58e9 A ocsp.verisign.com
26	0.338927000	127.0.0.1	127.0.1.1	DNS	79	Standard query 0x58c9 AAAA ocsp.verisign.com
27	0.338954000	172.16.90.156	172.16.90.2	DNS	79	Standard query 0x688e AAAA ocsp.verisign.com
28	0.347170000	172.16.90.2	172.16.90.156	DNS	175	Standard query response 0x58e9 CNAME ocsp.ws.symantec.com.edgekey.net CNAME e8218.ce.ak
29	0.347202000	172.16.90.2	172.16.90.156	DNS	218	Standard query response 0x688e CNAME ocsp.ws.symantec.com.edgekey.net CNAME e8218.ce.ak
30	0.347290000	127.0.1.1	127.0.0.1	DNS	175	Standard query response 0x5f62 CNAME ocsp.ws.symantec.com.edgekey.net CNAME e8218.ce.ak
31	0.347325000	127.0.1.1	127.0.0.1	DNS	218	Standard query response 0x58c9 CNAME ocsp.ws.symantec.com.edgekey.net CNAME e8218.ce.ak
32	0.353628000	172.16.90.156	23.43.75.27	TCP	76	51280 → http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=28896951 TSecr=0 WS=
33	0.358124000	23.43.75.27	172.16.90.156	TCP	62	http → 51280 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
34	0.363392000	172.16.90.156	23.43.75.27	TCP	56	51280 → http [ACK] Seq=1 Ack=1 Win=29200 Len=0
35	0.363818000	172.16.90.156	23.43.75.27	OCSP	495	Request
36	0.364075000	23.43.75.27	172.16.90.156	TCP	62	http → 51280 [ACK] Seq=1 Ack=440 Win=64240 Len=0
37	0.369827000	23.43.75.27	172.16.90.156	TCP	1504	[TCP segment of a reassembled PDU]

CLOUDFLARE

OCSP Stapling

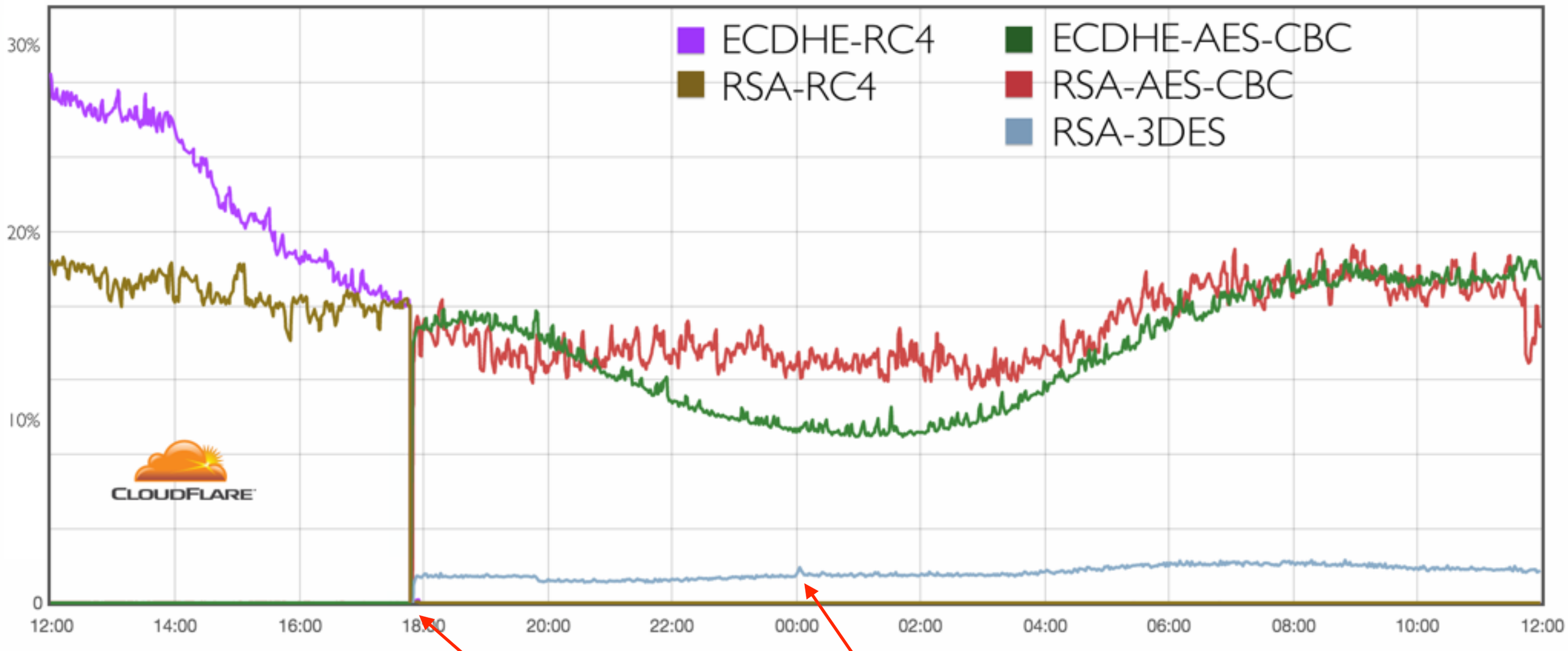
```
% openssl s_client -connect reddit.com:443 -tls1 -tlsextdebug -status
[…]
OCSP response:
======================================
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: B6A8FFA2A82FD0A6CD4BB168F3E7501031A77921
    Produced At: Dec 11 07:25:46 2014 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 3C482CAA7D028BACB016CF642BB22B236A62C380
      Issuer Key Hash: B6A8FFA2A82FD0A6CD4BB168F3E7501031A77921
      Serial Number: D643E3AAA0416C90D4FE41FFEE11FD87
    Cert Status: good
    This Update: Dec 11 07:25:46 2014 GMT
    Next Update: Dec 15 07:25:46 2014 GMT
```

CLOUDFLARE

Taming the BEAST POODLE

- BEAST attack targets CBC-based ciphers
  - Solution: switch away and use RC4 instead
  - Oops: RC4 now considered broken
  - Real solution: use TLS 1.2 with AES-GCM/AES-CBC

CLOUDFLARE



Windows XP users

0.0009% visitors using RC4

CLOUDFLARE

Taming the BEAST POODLE

- POODLE attack targets SSLv3 (not TLS)
  - Solution: disable SSLv3
  - Affected 1.12% of Windows XP users
- POODLE-bis can affect TLS
  - Poor implementation in some F5/A10 load balancers
  - Upgrade F5/A10 (https://www.imperialviolet.org/2014/12/08/poodleagain.html)

CLOUDFLARE

Recommended Configuration

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

ssl_ciphers EECDH+AES128:RSA+AES128:EECDH+AES256:RSA
+AES256:EECDH+3DES:RSA+3DES:EECDH+RC4:RSA+RC4:!MD5;

ssl_prefer_server_ciphers on;
```

CLOUDFLARE

Recommended Configuration

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

ssl_ciphers EECDH+AES128:RSA+AES128:EECDH+AES256:RSA
+AES256:EECDH+3DES:RSA+3DES:EECDH+RC4:RSA+RC4:!MD5;

ssl_prefer_server_ciphers on;
```

CLOUDFLARE

TLS False Start

- Start sending application data (`GET / HTTP/1.1`) immediately after sending *Finished* TLS frame
- Saves waiting for server to send *Finished* back and saves 1x RTT
- Some servers break on this, so browsers have different heuristics for handling
- Solution: deploy NPN/ALPN and Perfect Forward Secrecy

CLOUDFLARE

Goal 1x RTT TLS

- New connections: need TLS False Starts
- Repeat visitors: deploy Session Tickets/IDs
- Deploy OCSP stapling

CLOUDFLARE

Or use a CDN that does all that

8	0.100661000	127.0.1.1	127.0.0.1	DNS	138	Standard query response 0x22d5
9	*REF*	172.16.90.156	162.159.243.33	TCP	76	58044 → https [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=29059274 TSecr=0 WS=128
10	0.003649000	162.159.243.33	172.16.90.156	TCP	62	https → 58044 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
11	0.003745000	172.16.90.156	162.159.243.33	TCP	56	58044 → https [ACK] Seq=1 Ack=1 Win=29200 Len=0
12	0.004005000	172.16.90.156	162.159.243.33	TLSv1.2	226	Client Hello
13	0.004175000	162.159.243.33	172.16.90.156	TCP	62	https → 58044 [ACK] Seq=1 Ack=171 Win=64240 Len=0
14	0.012109000	162.159.243.33	172.16.90.156	TLSv1.2	5060	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
15	0.012150000	172.16.90.156	162.159.243.33	TCP	56	58044 → https [ACK] Seq=171 Ack=5005 Win=37960 Len=0
16	0.014592000	172.16.90.156	162.159.243.33	TLSv1.2	218	Client Key Exchange, Change Cipher Spec, Hello Request, Hello Request
17	0.014850000	162.159.243.33	172.16.90.156	TCP	62	https → 58044 [ACK] Seq=5005 Ack=333 Win=64240 Len=0
18	0.015023000	172.16.90.156	162.159.243.33	TLSv1.2	121	Application Data
19	0.015075000	172.16.90.156	162.159.243.33	TLSv1.2	453	Application Data
20	0.015227000	162.159.243.33	172.16.90.156	TCP	62	https → 58044 [ACK] Seq=5005 Ack=398 Win=64240 Len=0
21	0.015906000	162.159.243.33	172.16.90.156	TCP	62	https → 58044 [ACK] Seq=5005 Ack=795 Win=64240 Len=0
22	0.019566000	162.159.243.33	172.16.90.156	TLSv1.2	298	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
23	0.020511000	162.159.243.33	172.16.90.156	TLSv1.2	129	Application Data
24	0.020571000	172.16.90.156	162.159.243.33	TCP	56	58044 → https [ACK] Seq=795 Ack=5320 Win=40880 Len=0

```
▶ Frame 14: 5060 bytes on wire (40480 bits), 5060 bytes captured (40480 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 162.159.243.33 (162.159.243.33), Dst: 172.16.90.156 (172.16.90.156)
▶ Transmission Control Protocol, Src Port: https (443), Dst Port: 58044 (58044), Seq: 1, Ack: 171, Len: 5004
▼ Secure Sockets Layer
  ▶ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ▶ TLSv1.2 Record Layer: Handshake Protocol: Certificate
  ▶ TLSv1.2 Record Layer: Handshake Protocol: Certificate Status
  ▶ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
  ▶ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

CLOUDFLARE