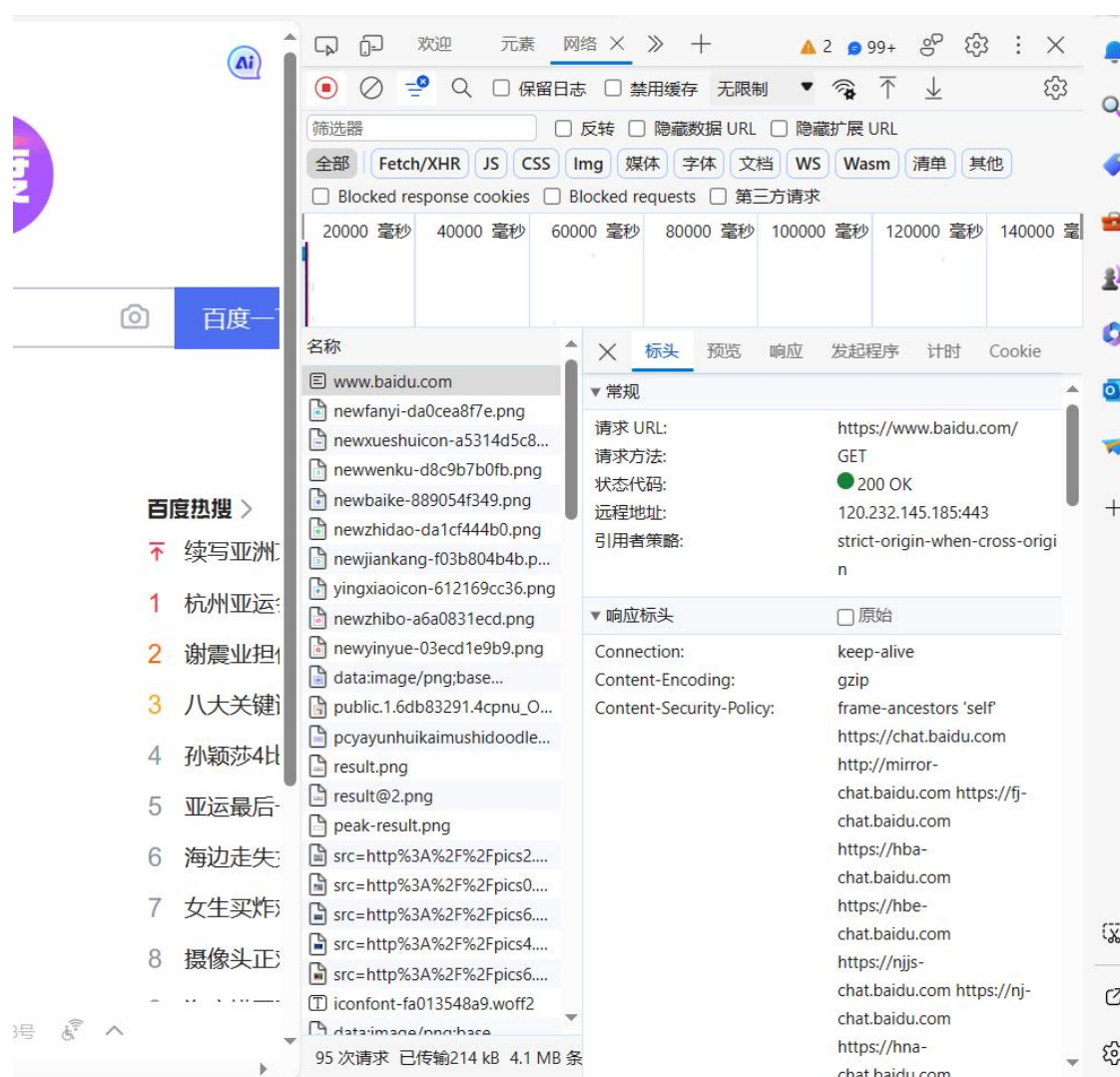


# 一、Https：通过非对称加密的方式来传递对称加密所需要的密钥

(1) 向百度一下，你就知道 (baidu.com)发送请求



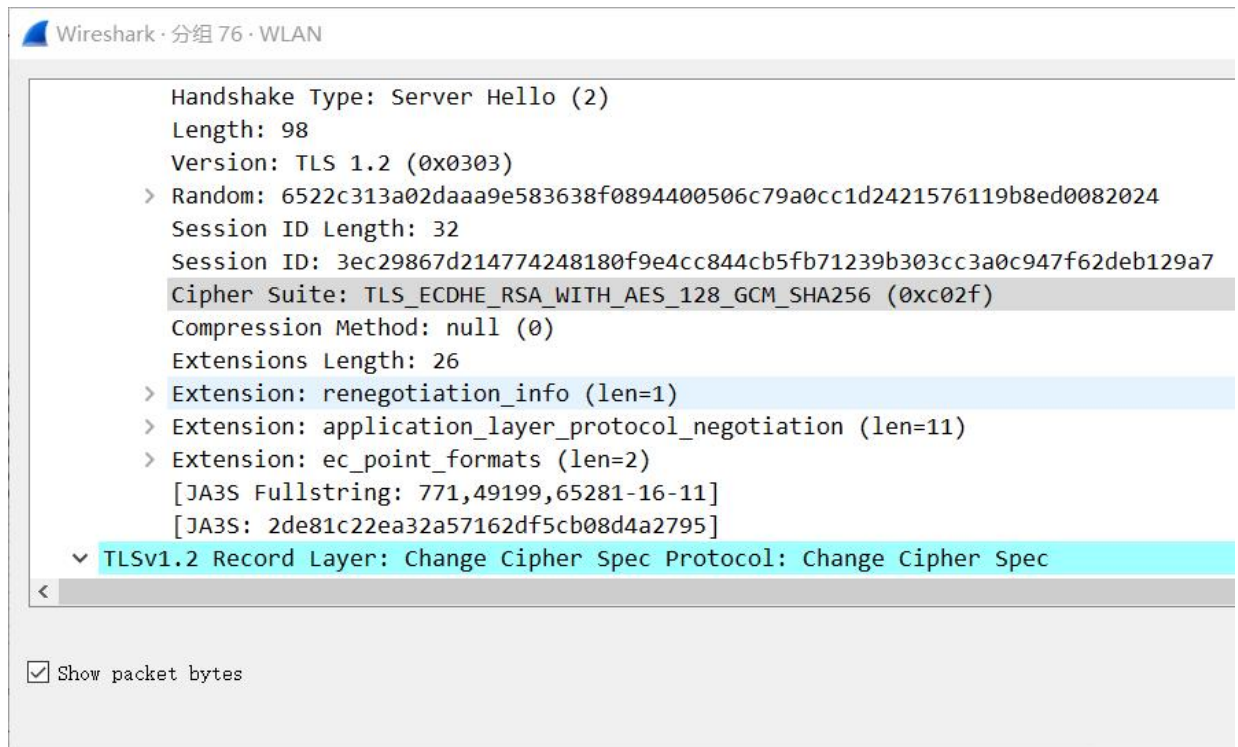
(2) 从第二条开始的三条分别带有【SYN】，【SYN,ACK】，【ACK】为三次握手。

*WLAN						
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)						
ip.addr == 120.232.145.185						
No.	Time	Source	Destination	Protocol	Length	Info
57	3.077934	100.65.178.14	120.232.145.185	TCP	66	53194 → 443 [SYN] Seq=0 Win=64240 Len=0 MS
58	3.078317	100.65.178.14	120.232.145.185	TCP	66	53195 → 443 [SYN] Seq=0 Win=64240 Len=0 MS
60	3.098834	120.232.145.185	100.65.178.14	TCP	66	443 → 53195 [SYN, ACK] Seq=0 Ack=1 Win=819
61	3.098907	100.65.178.14	120.232.145.185	TCP	54	53195 → 443 [ACK] Seq=1 Ack=1 Win=132096 L
62	3.099086	100.65.178.14	120.232.145.185	TLSv1.2	571	Client Hello
63	3.099959	120.232.145.185	100.65.178.14	TCP	66	443 → 53194 [SYN, ACK] Seq=0 Ack=1 Win=819
64	3.100013	100.65.178.14	120.232.145.185	TCP	54	53194 → 443 [ACK] Seq=1 Ack=1 Win=132096 L
65	3.100168	100.65.178.14	120.232.145.185	TLSv1.2	571	Client Hello
75	3.120266	120.232.145.185	100.65.178.14	TCP	56	443 → 53195 [ACK] Seq=1 Ack=518 Win=30208
76	3.120510	120.232.145.185	100.65.178.14	TLSv1.2	212	Server Hello, Change Cipher Spec, Encrypte
77	3.120700	100.65.178.14	120.232.145.185	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Me
79	3.122439	120.232.145.185	100.65.178.14	TCP	56	443 → 53194 [ACK] Seq=1 Ack=518 Win=30208
80	3.123184	120.232.145.185	100.65.178.14	TLSv1.2	212	Server Hello, Change Cipher Spec, Encrypte

(3) 点开第一个 info 为 Client Hello 的分组，可以观察到客户端支持的各种加密方式

Wireshark · 分组 62 · WLAN	
Session ID: 3ec29867d214774248180f9e4cc844cb5fb71239b303cc3a0c947f62deb129a7	
Cipher Suites Length: 32	
▼ Cipher Suites (16 suites)	
Cipher Suite: Reserved (GREASE) (0x5a5a)	
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)	
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)	
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)	
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)	
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)	
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)	
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)	
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	
<	
<input checked="" type="checkbox"/> Show packet bytes	

(4) 点开 **info** 为 **server Hello** 的分组，可以观察到服务器支持的加密方式为客户端的一个子集



(5) 分析 SSL/TLS 交换证书过程，打开带有 **server hello**, **change cipher spec** 的分组

```

▼ Handshake Protocol: Server Hello
  Handshake Type: Server Hello (2)
  Length: 98
  Version: TLS 1.2 (0x0303)
  Random: 6522d24cb172117dc39133297effc2bcfd5e0f7179ce1abc6ed3f3b6e99b9922
  Session ID Length: 32
  Session ID: 7a15b84128bc2c615487ea692fe5d4b971501f0b5ba6ff7f5eae148d7212d4df
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
  Compression Method: null (0)
  Extensions Length: 26
  Extension: renegotiation_info (len=1)
  Extension: application_layer_protocol_negotiation (len=11)
  Extension: ec_point_formats (len=2)
    [JA3S Fullstring: 771,49199,65281-16-11]
    [JA3S: 2de81c22ea32a57162df5cb08d4a2795]

```

packet bytes

## (6) 交换密钥，打开 **change cipher spec** 的分组

```

▼ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  Content Type: Change Cipher Spec (20)
  Version: TLS 1.2 (0x0303)
  Length: 1
  Change Cipher Spec Message
▼ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
  Content Type: Handshake (22)
  Version: TLS 1.2 (0x0303)
  Length: 40
  Handshake Protocol: Encrypted Handshake Message

```

## (7) 开始数据交互，即带有 **Application Data** 字样的分组。



## 二、Http

### (1) 三次握手:

访问 <http://httpbin.org/>, 得到如下图:

*WLAN						
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(V) 无线(W) 工具(T) 帮助(H)						
ip.addr == 54.172.39.167						
No.	Time	Source	Destination	Protocol	Length	Info
39	0.982450	100.67.108.232	54.172.39.167	TCP	55	50872 → 80 [ACK] Seq=1 Ack=1 Win=515 Len=1
53	1.333260	54.172.39.167	100.67.108.232	TCP	56	80 → 50872 [RST] Seq=1 Win=0 Len=0
297	6.720591	100.67.108.232	54.172.39.167	TCP	66	50929 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1
320	6.978861	54.172.39.167	100.67.108.232	TCP	66	80 → 50929 [SYN, ACK] Seq=0 Ack=1 Win=26883
321	6.979022	100.67.108.232	54.172.39.167	TCP	54	50929 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
416	8.853177	100.67.108.232	54.172.39.167	HTTP	546	GET / HTTP/1.1
434	9.125912	54.172.39.167	100.67.108.232	TCP	56	80 → 50873 [ACK] Seq=1 Ack=493 Win=143 Len=0
457	9.629734	54.172.39.167	100.67.108.232	TCP	1448	[TCP Previous segment not captured] 80 → 508
458	9.629738	54.172.39.167	100.67.108.232	TCP	1448	[TCP Out-Of-Order] 80 → 50873 [ACK] Seq=240
459	9.629739	54.172.39.167	100.67.108.232	TCP	1448	[TCP Previous segment not captured] 80 → 508
460	9.629740	54.172.39.167	100.67.108.232	TCP	1448	[TCP Out-Of-Order] 80 → 50873 [ACK] Seq=3028
461	9.629858	100.67.108.232	54.172.39.167	TCP	66	[TCP Dup ACK 416#1] 50873 → 80 [ACK] Seq=493
462	9.629961	100.67.108.232	54.172.39.167	TCP	66	[TCP Dup ACK 416#2] 50873 → 80 [ACK] Seq=493
463	9.629996	100.67.108.232	54.172.39.167	TCP	74	[TCP Dup ACK 416#3] 50873 → 80 [ACK] Seq=493
464	9.630030	100.67.108.232	54.172.39.167	TCP	66	[TCP Dup ACK 416#4] 50873 → 80 [ACK] Seq=493
465	9.631572	54.172.39.167	100.67.108.232	TCP	1448	80 → 50873 [ACK] Seq=5816 Ack=493 Win=143 Le
466	9.631577	54.172.39.167	100.67.108.232	TCP	1448	80 → 50873 [ACK] Seq=7210 Ack=493 Win=143 Le
467	9.631690	100.67.108.232	54.172.39.167	TCP	66	[TCP Dup ACK 416#5] 50873 → 80 [ACK] Seq=493
468	9.631798	100.67.108.232	54.172.39.167	TCP	66	[TCP Dup ACK 416#6] 50873 → 80 [ACK] Seq=493
469	9.639762	54.172.39.167	100.67.108.232	TCP	1283	80 → 50873 [PSH, ACK] Seq=8604 Ack=493 Win=1
470	9.639767	54.172.39.167	100.67.108.232	TCP	293	[TCP Fast Retransmission] 80 → 50873 [PSH, A
471	9.639880	100.67.108.232	54.172.39.167	TCP	66	[TCP Dup ACK 416#7] 50873 → 80 [ACK] Seq=493
				0000 00 0f e2 ff 80 ff 20 1e 88 dc :		

TCP 三次握手为带有【SYN】，【SYN, ACK】，【ACK】的三条分组

第一次握手时，客户端给服务器发送 SYN 标志位

Wireshark · 分组 297 · WLAN

Acknowledgment Number: 0  
Acknowledgment number (raw): 0  
1000 .... = Header Length: 32 bytes (8)  
> **Flags: 0x002 (SYN)**  
Window: 64240  
[Calculated window size: 64240]  
Checksum: 0x3fa5 [unverified]

<

0000	00 0f e2 ff 80 ff 20 1e 88 dc 26 ca 08 00 45
0010	00 34 40 13 40 00 80 06 00 00 64 43 6c e8 36
0020	27 a7 c6 f1 00 50 c6 db 2e 0e 00 00 00 00 80
0030	fa f0 2f a5 00 00 02 04 05 b4 01 03 03 08 01
0040	04 02

☒ Show packet bytes

第二次握手时，服务器给客户端发送 SYN，ACK 标志位

Wireshark · 分组 320 · WLAN

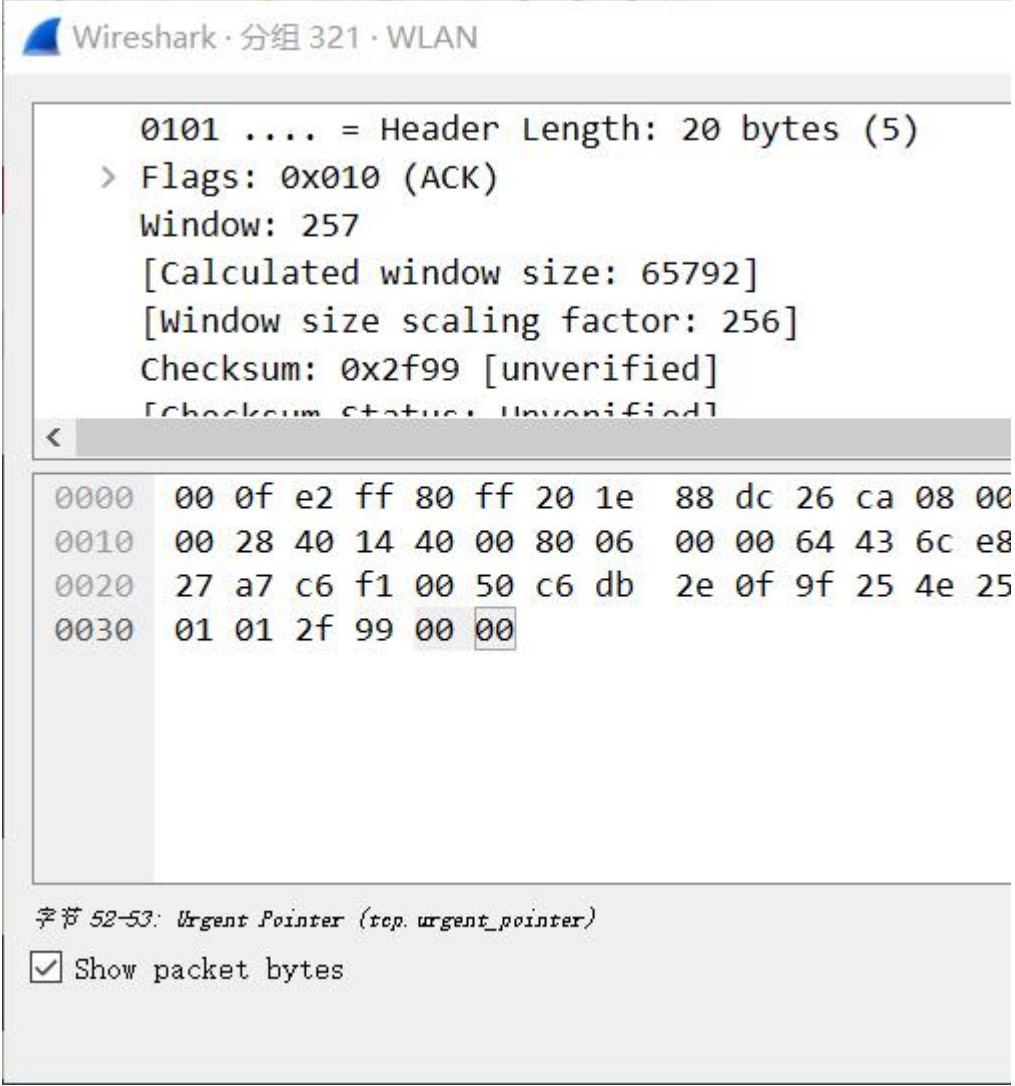
Acknowledgment Number: 1 (relative ack number)  
Acknowledgment number (raw): 3336252943  
1000 .... = Header Length: 32 bytes (8)  
> **Flags: 0x012 (SYN, ACK)**  
Window: 26883  
[Calculated window size: 26883]  
Checksum: 0x2d5e [unverified]

<

0000	20 1e 88 dc 26 ca 00 0f e2 ff 80 ff 08 00 45 04
0010	00 34 00 00 40 00 e8 06 63 41 36 ac 27 a7 64 43
0020	6c e8 00 50 c6 f1 9f 25 4e 24 c6 db 2e 0f 80 12
0030	69 03 2d 5e 00 00 02 04 05 5e 01 01 04 02 01 03
0040	03 08

☒ Show packet bytes

第三次握手时，客户端给服务器发送 ACK 标志位



## (2) 四次挥手：实为三次挥手，服务器的确认和挥手合为一个分组

第一次挥手：客户端向服务器挥手

100.67.108.232 54.172.39.167 TCP 54 51963 → 80 [FIN, ACK] Seq=2513 Ack=153250 Win=131840 Len=0

第二次挥手：服务器对客户端的挥手确认以及对客户端挥手

54.172.39.167 100.67.108.232 TCP 56 80 → 51963 [FIN, ACK] Seq=153250 Ack=2514 Win=33536 Len=0

第三次挥手：客户端确认挥手

100.67.108.232 54.172.39.167 TCP 54 51963 → 80 [ACK] Seq=2514 Ack=153251 Win=131840 Len=0