

TLS and HTTPS

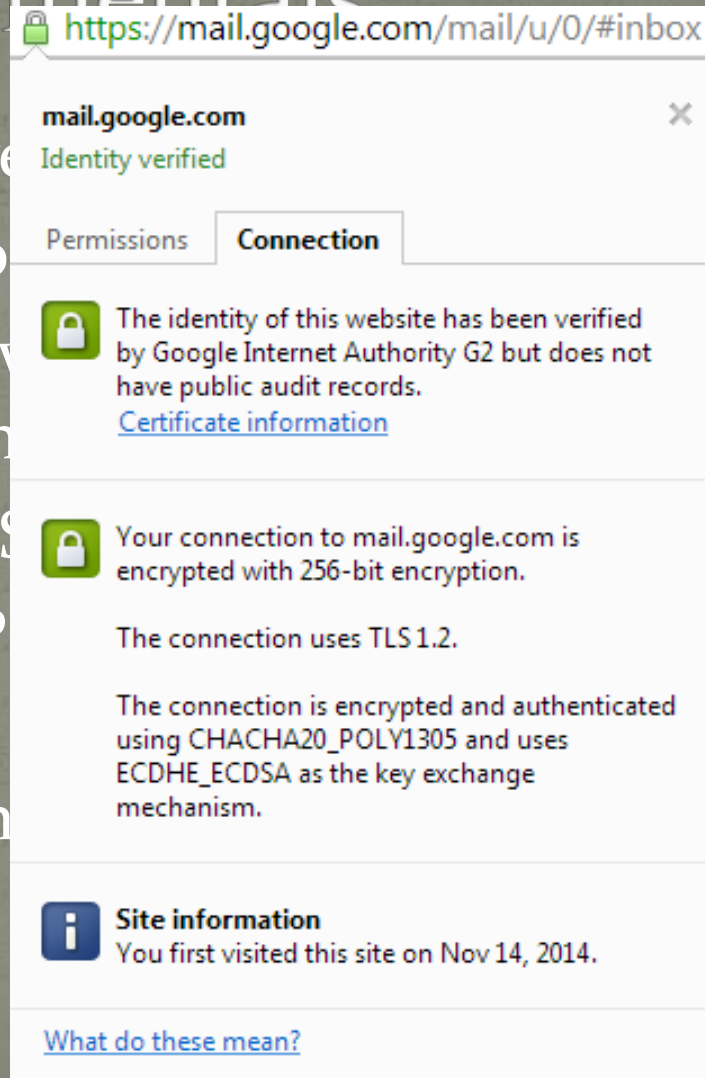
CSE 461 Section

A joke about bad weather



TLS Fundamentals

- “Transport Layer Security (TLS) is a standard protocol for securing Internet traffic (e.g. web browsing, email, and file transfers) (Application Layer),”
- Previously known as Secure Sockets Layer (SSL), which has been deprecated.
- TLS replaced SSL as the standard protocol for securing Internet traffic.
- Used for HTTP (Hypertext Transfer Protocol Secure) traffic.
- Supported by most modern web browsers.



https://mail.google.com/mail/u/0/#inbox

mail.google.com Identity verified

Permissions Connection

The identity of this website has been verified by Google Internet Authority G2 but does not have public audit records.
[Certificate information](#)

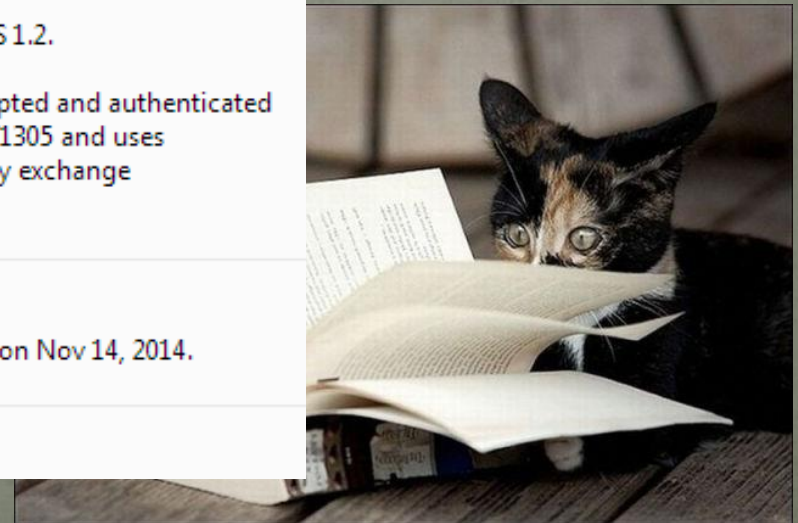
Your connection to mail.google.com is encrypted with 256-bit encryption.

The connection uses TLS 1.2.

The connection is encrypted and authenticated using CHACHA20_POLY1305 and uses ECDHE_ECDSA as the key exchange mechanism.

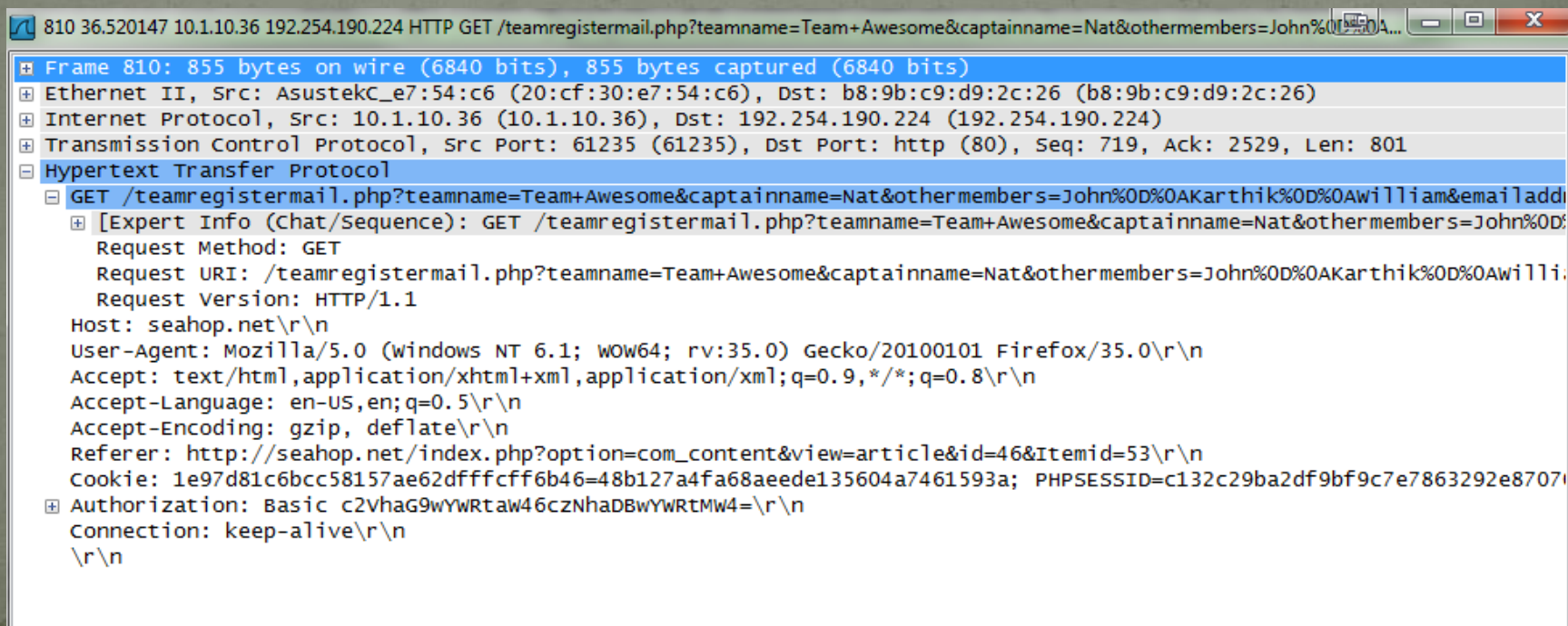
i Site information
You first visited this site on Nov 14, 2014.

[What do these mean?](#)



Purposes for TLS (1)

- When we don't use TLS, web traffic goes over unencrypted
- This includes HTTP payloads, but also HTTP headers
- Why are headers a problem too?




```
810 36.520147 10.1.10.36 192.254.190.224 HTTP GET /teamregistermail.php?teamname=Team+Awesome&captainname=Nat&othermembers=John%0D%0A...
+ Frame 810: 855 bytes on wire (6840 bits), 855 bytes captured (6840 bits)
+ Ethernet II, Src: AsustekC_e7:54:c6 (20:cf:30:e7:54:c6), Dst: b8:9b:c9:d9:2c:26 (b8:9b:c9:d9:2c:26)
+ Internet Protocol, Src: 10.1.10.36 (10.1.10.36), Dst: 192.254.190.224 (192.254.190.224)
+ Transmission Control Protocol, Src Port: 61235 (61235), Dst Port: http (80), Seq: 719, Ack: 2529, Len: 801
+ Hypertext Transfer Protocol
  GET /teamregistermail.php?teamname=Team+Awesome&captainname=Nat&othermembers=John%0D%0AKarthik%0D%0AWilliam&emailaddi
    [Expert Info (Chat/Sequence): GET /teamregistermail.php?teamname=Team+Awesome&captainname=Nat&othermembers=John%0D%
      Request Method: GET
      Request URI: /teamregistermail.php?teamname=Team+Awesome&captainname=Nat&othermembers=John%0D%0AKarthik%0D%0AWilli:
      Request Version: HTTP/1.1
      Host: seahop.net\r\n
      User-Agent: Mozilla/5.0 (windows NT 6.1; WOW64; rv:35.0) Gecko/20100101 Firefox/35.0\r\n
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
      Accept-Language: en-US,en;q=0.5\r\n
      Accept-Encoding: gzip, deflate\r\n
      Referer: http://seahop.net/index.php?option=com_content&view=article&id=46&Itemid=53\r\n
      Cookie: 1e97d81c6bcc58157ae62dfffcff6b46=48b127a4fa68aeede135604a7461593a; PHPSESSID=c132c29ba2df9bf9c7e7863292e8707
    Authorization: Basic c2VhaG9wYWRTaw46czNhaDBWYWRtMW4=\r\n
    Connection: keep-alive\r\n
    \r\n
```

Purposes for TLS (2)

- Data integrity
- Server (and client) authentication



Defcon Wall of Sheep



Wall of Sheep

login	pass	domain ip	application
h00p	tdc*****	65.154.34.164	HTTP
voltagespike@fastmail.fm	tha*****	66.111.4.52	IMAP
Jennifer.lee@post.harvard.edu	poc*****	184.73.159.65	foursquare
demblew	MIC*****	137.52.224.216	pop
wencevdn	Sla*****	128.242.245.20	Twitter (on Android)
Nokia-osso-rx-49	JOS*****	207.114.197.94	HTTP
computicu	lof*****	128.242.245.116	Twitter
reuhelix	fay*****	128.242.245.116	Twitter
vishakn@yahoo.com	hea*****	184.73.159.65	foursquare
em2827891836	622*****	207.114.197.95	HTTP
rossknapp@gmail.com	863*****	184.73.159.65	foursquare
imylongs	tes*****	128.242.245.43	TWITTER
crissti	int*****	128.242.245.148	Twitter
6062191197	pre*****	184.73.159.65	foursquare
ptkrisnan	4li*****	128.242.245.20	twitter
	fun*****	184.73.159.65	4square

TLS and CONNECT

- HTTP CONNECT is used to establish a two-way connection “tunnel” between two parties
- After this, a “triple handshake” is performed over the tunnel
- After the handshake, the two parties can communicate securely
- We’ll take a closer look at this handshake



TLS Handshake Protocol (Concept)

- What do we need to do to communicate securely?
 - Make sure we're speaking the same language
 - Prove who we are
 - Establish a secret code

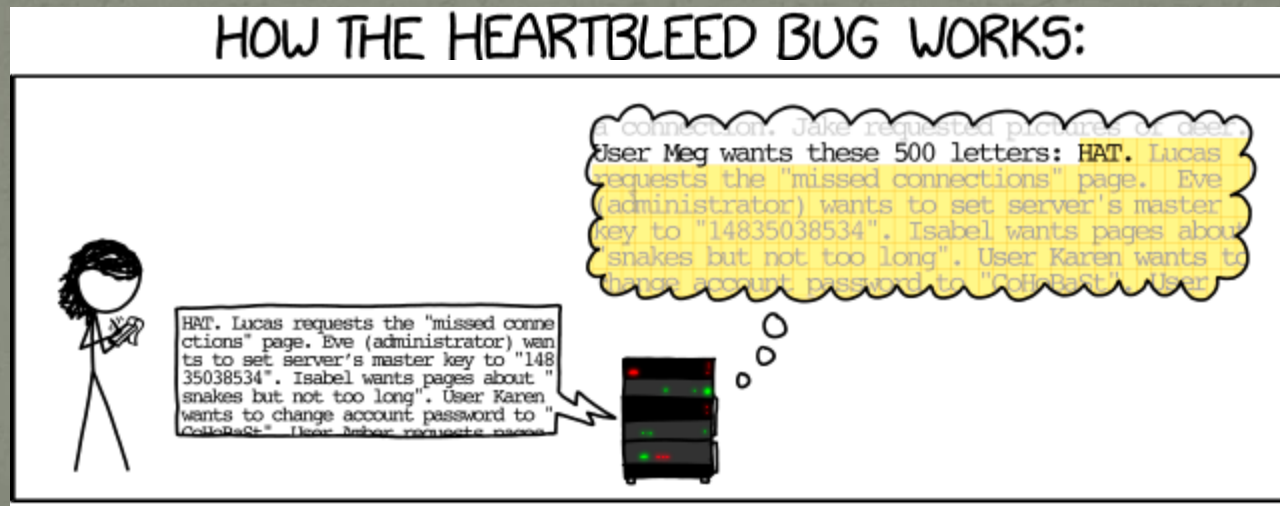


TLS Handshake Protocol (Rough Details)

- Client tells the server its protocol version and what cryptographic algorithms it can use
- Server responds with a protocol version and cryptographic algorithm to use
- Server sends its certificate to verify its identity
- Client verifies certificate and sends Pre-Master Secret, encrypted so only the server can read it
- Client and server both use that PMS to generate a Master Secret, which is used to generate encryption keys
- Communication commences

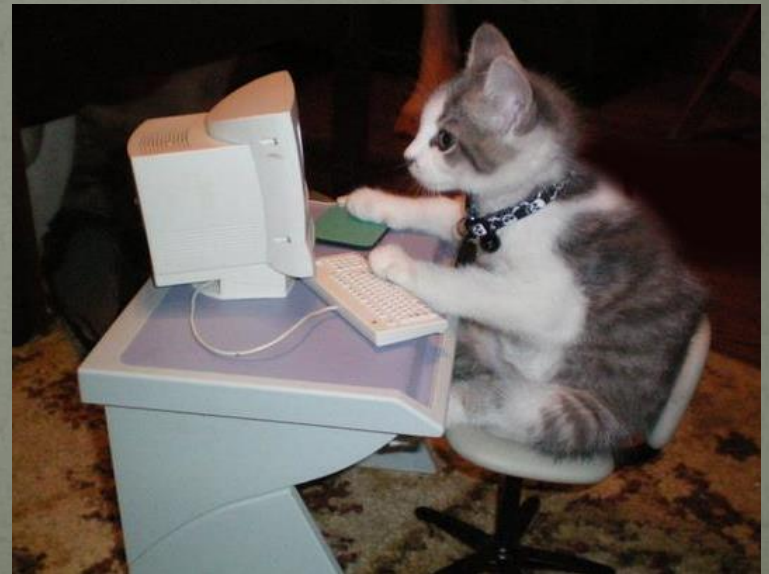
Heartbleed bug

- 2014 Bug in OpenSSL implementation of TLS
- Clients ask for a “heartbeat” message to test and keep alive communication links
- In OpenSSL, length checking wasn't properly performed on the heartbeat data



TLS Exploits

- How might data be intercepted by a MITM, even when encrypted over TLS?
 - Implementation bugs (e.g., Heartbleed, 3Shake)
 - Server/browser attacks (e.g., truncation attack)
 - “Truncate” logout packet from user
 - User’s browser tells them they’ve logged out
 - They haven’t
 - Side-channel attacks



Side-channel TLS Attacks (1)

- Some data is leaked even with encryption
 - Packet send timing
 - Payload size
 - AJAX interfaces that load content dynamically provide insight into what the user is typing



Side-channel TLS Attacks (2)



Go to classic Google.

am|

amazon

aol

Advanced search
Language tools



"am"
(SSL)



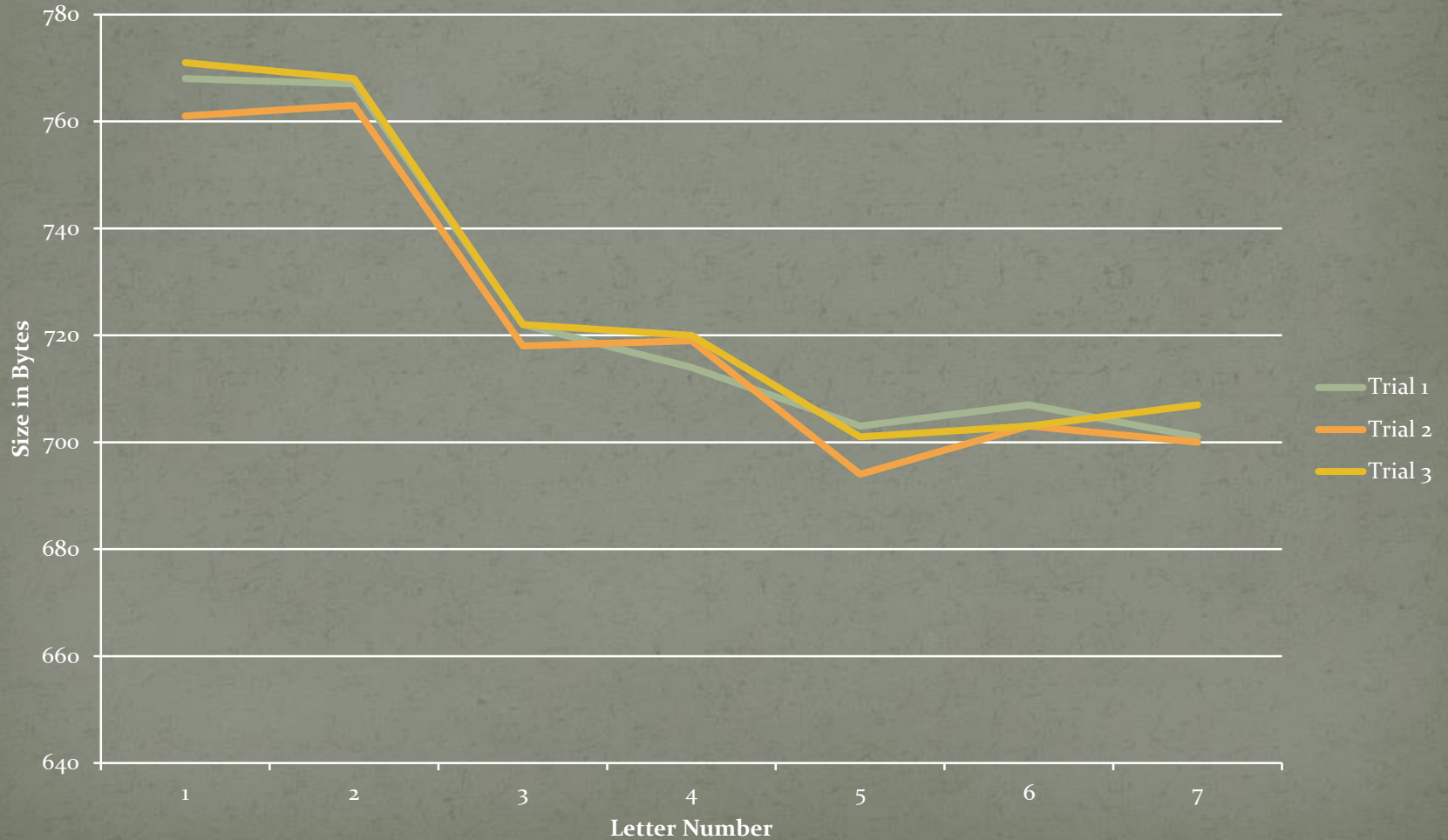
es

Google Search

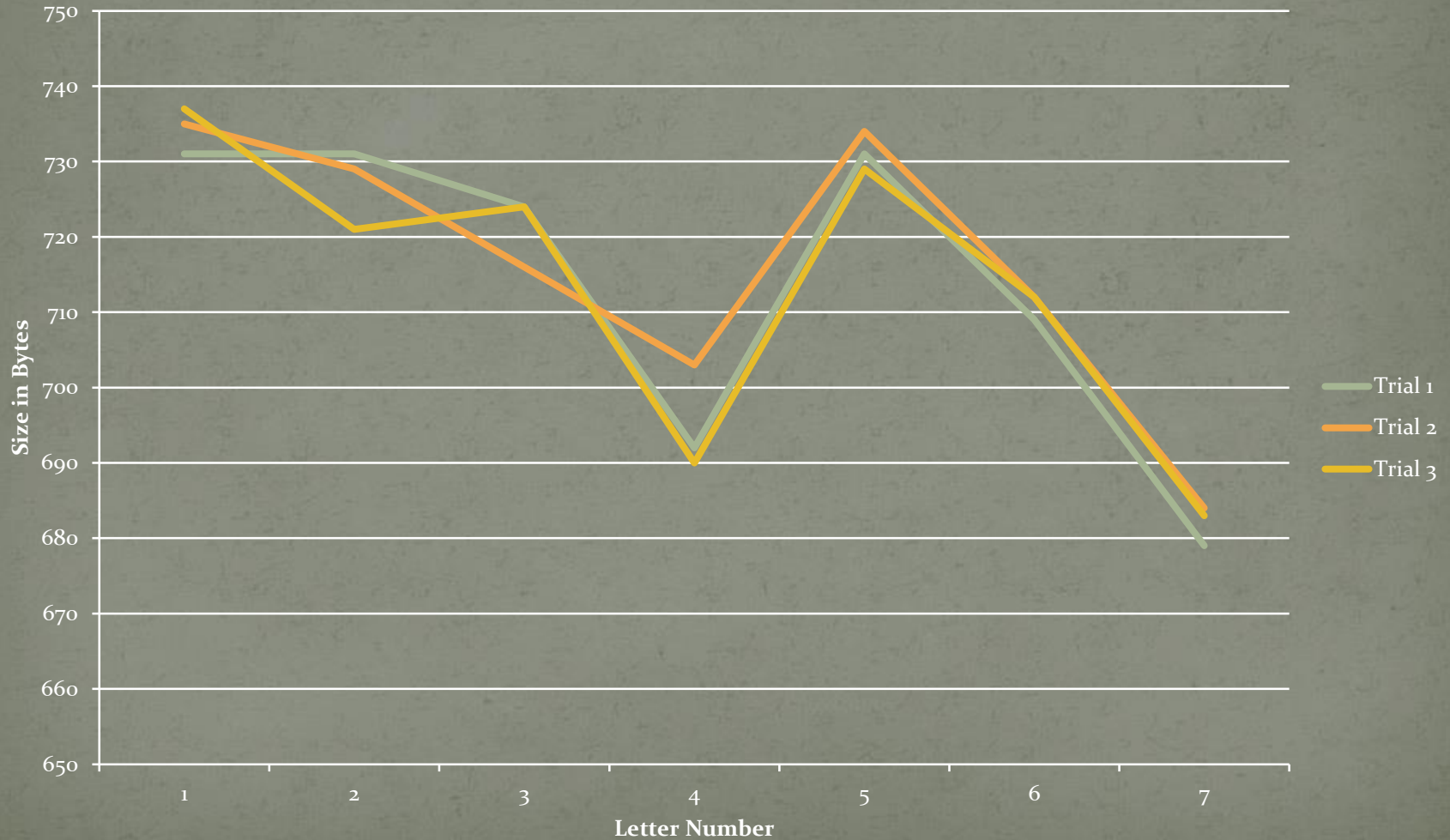
ing Lucky



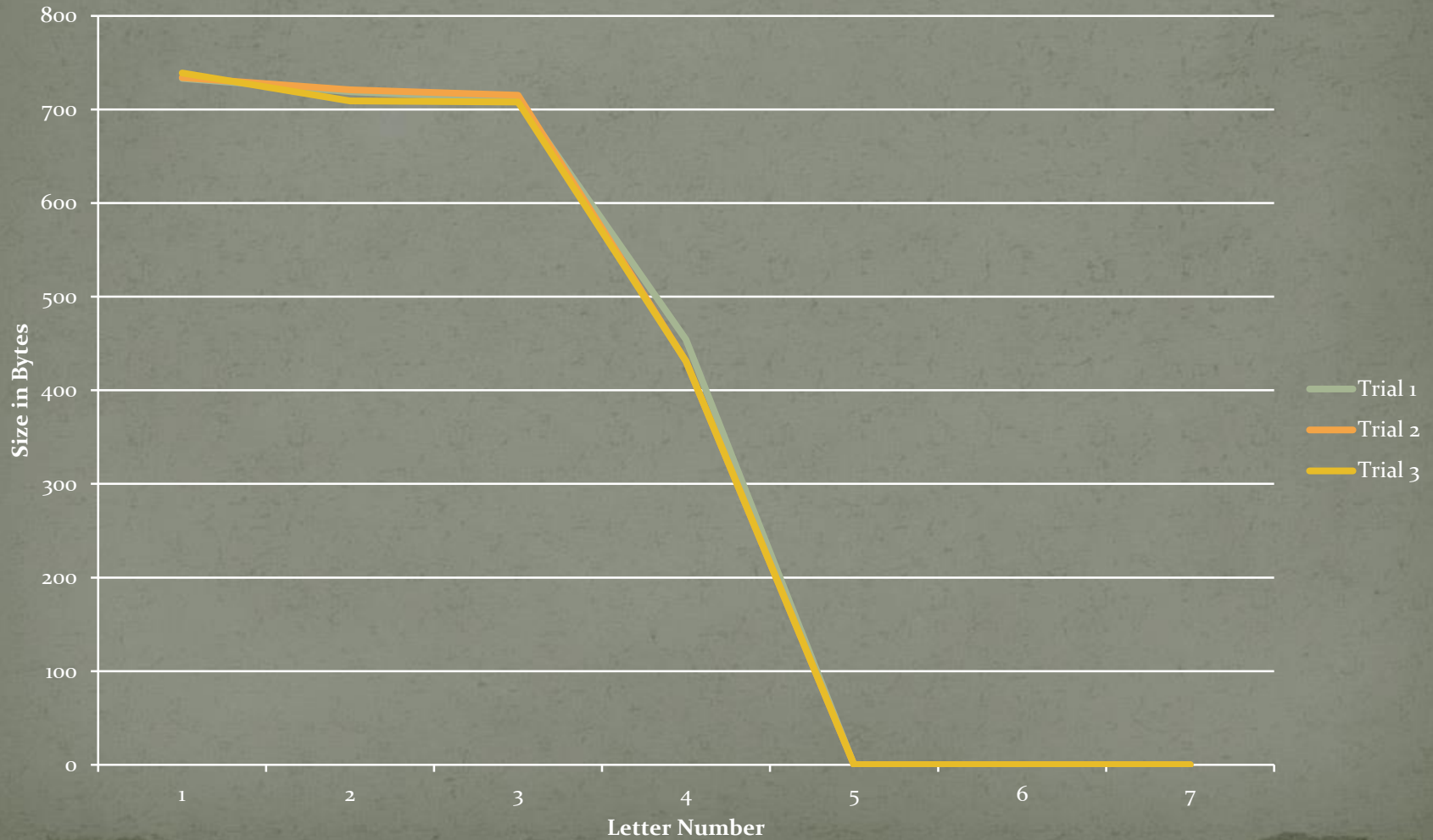
Autocomplete Packet Sizes for "hackers"



Autocomplete Packet Sizes for "benaloh"



Autocomplete Packet Sizes for "xvwqxzx"



Questions?

