Lecture 8 🎋

> **Class**: TLS Optimization
>
> **Date**: 19.57 AM , 25 Nov
>
> **Author**: Lasal Hettiarachchi

Key learnings:

- DH key exchange
- ALPN
- SNI
- TSL session resumption

TSL protocol provides

- sender authentication
- non rupidiation
- Confidentiality

When it comes to security we have to be concerned about

- Security
- Cost
- Performance

TLS RSA Handshake

This uses 4 messages for the handshake

- Client Generates a symmetric key
- Encrypt with the servers public key
- Send to the server to use it as the symmetric key for the establised session
- Use servers private key to decrypt
- The client and server uses the symmetric key to encrypt and decrypt the session.

The symmetric key is used to encrypt the actual communication

Issues with this?

If the private key is compromised all the previous communication that uses the private key is revealed

In this scenario the private key is used for both authenticating the server and indirectly encrypting the messages

Foward secrecy

- If the attack is successful in the future it should not compromise todays secrets

created with craft

- RSA is not foward secret
- Compromise of keys in the future or present should not compromise the secrecy of present or past communications

The method that is used to ebsure forward secrecy is Diffie Helman and forward secrecy

Diffie Helman and forward secrecy

The problem with RSA handshake is that the session key is encrypted and sent through the unsecure channel.

In DH the concept is that they use ephemeral keys

Ephemeral keys

- When a client and a server is communicating a session key is created and discarded after the communication
- Exist only until the session exists (length of the session)
- Never stores anywhere

The current handshake has 3 optimizations

- Diffie-Hellman
- Ephemeral symmetric key
- TLS 1.3 message reduction

The above approaches provide,

- Security
- Efficiency
- Foward Security

Diffie-Hellman Key exchange

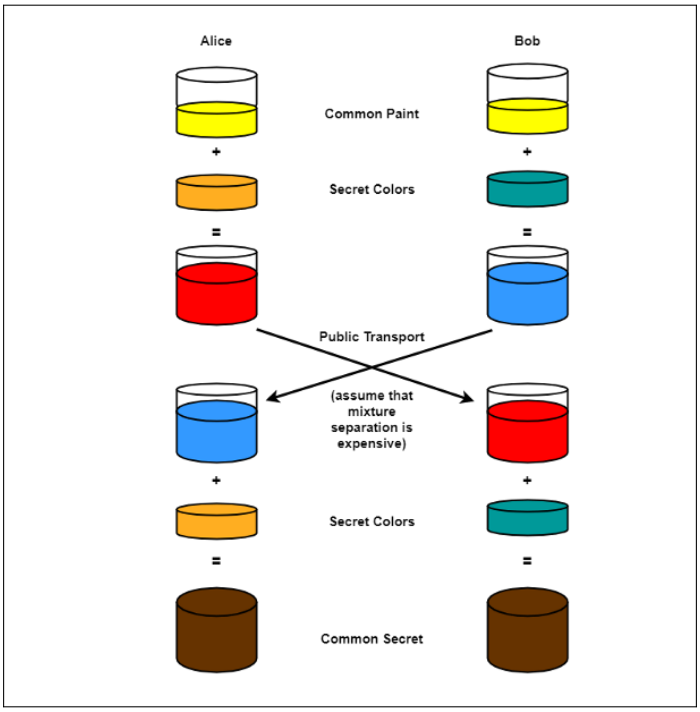
Majority of network communication uses this

Socket protocol or double wretched protocol are more advanced versions of this

- Allows two parties with no prior knowledge of each other to establish a shared secret key over an insecure channel.
- This key can then be used to encrypt subsequent communications using a symmetric-key cipher.

How the diffie hellman key exchange works

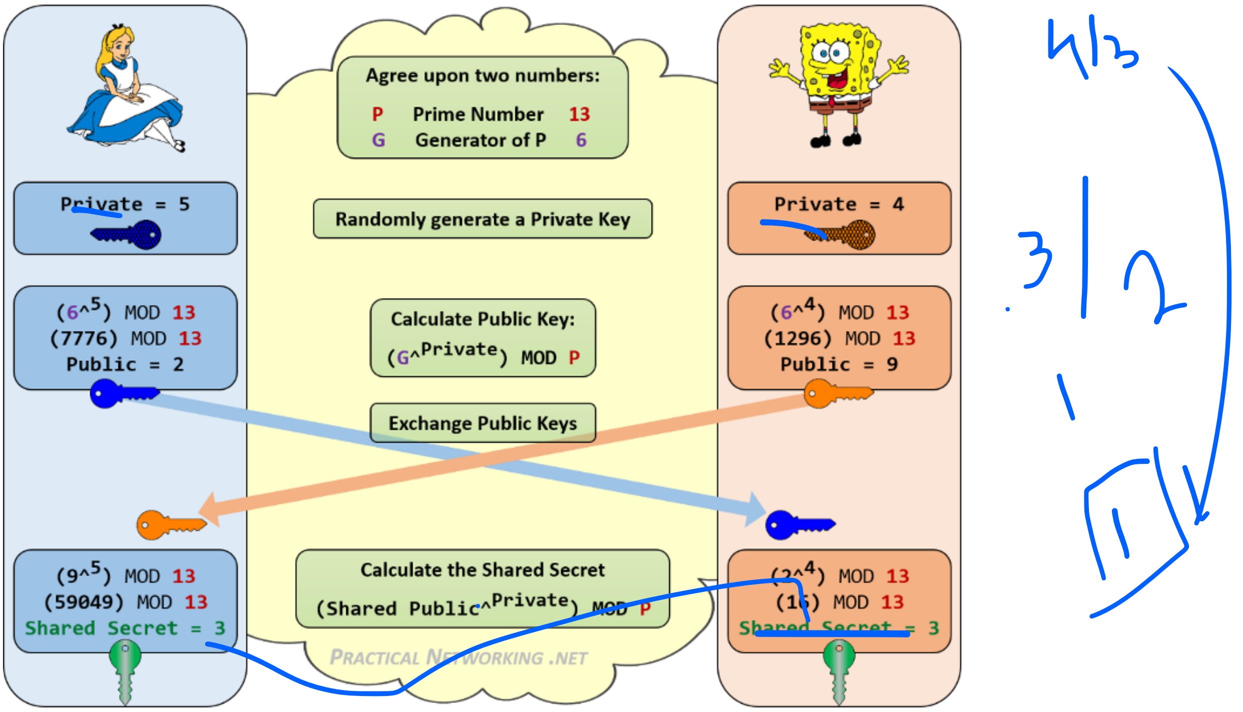
created with craft



- Alice and bob both aggrees on a paint and communicates it over the secure channel.
- That is the premaster keys
- Then Alice and Bob picks a random color. These are never sent over the communication channel.
- Alice mixes the Orange and yellow while Bob mixes the Aqua and yellow.
- Then the 2 resulting buckets are exchanged
- Theres no way to unmix the 2 colors . Meaning that u cannot duduce that Bobs random color was aqua nor Alices random color was orange. So what is done is Alice mixes her original random color with the exchanged color and the bob does the same
- The Resulting color (Brown) generated is the common secret key

This explained in math terms

created with craft



- Agreed upon 2 numbers Prime and generator of P are shared over the insecure channel
- P is a very large prime number and the security of this depends upon how large the prime number is
- Then random private keys are generated (These are never sent over the network)
- Then the generator is taken and power of it is taken to the private key on both sides and mod withthe prime number . Which generates the public key
  - Public key of alice is 2
  - Public key of Bob is 9
- These public keys are exchanged
- Then the Public key to the power private key is mod woth the prime number which generates the shared secret key

Since modulus is a one way function the method is a one way function

- This Algorithm can only be used to key agreement and not encryption and decryption
- This is one way

This can be prone to brute force with large Computational capabilities

Thus Variations of DH is created.

- Tripple DH
-
- Extended Tripple DH
- Double Ratchet algorithm
- Signal Protocol

Security

- The computer cannot generate random numbers, instrructions have to be given. But things called pseudo random numbers can be generated which are random enough
- Chosen properly is chosing the pseudo random numbers
- Factorizing large numbers is impossible
- An efficient algorithm to solve the discrete logarithm problem would enable breaking of this

TLS Optimization

- The newest version is 1.3
- Optimizations are aditional configurations that may or may not be added
- They have advantages and trade offs
- Optimizations and features added into TLS to handle practical issues in communication
  - Must be enabled on the server sided
  - Must be supported by the client devices
- Developers must decide which ones to use
  - ALPN
  - SNI
  - Session resumption

80

20

10

ALPN (Application Layer Protocol Negotiation)

- ALPN will use dedicated ports (HTTP = 80 HTTPS = 443)
- When browsing the web the protocol used is HTTP
- When using TLS the application layer protocol messages with data is embeded into the TLS protocol
- By base standard the HTTP embeded in TLS is mapped to 443
- When you send an HTTP packet the TCP layer addresses it to 80, if you send HTTP embeded in TLS, the TCP layer addresses it to 443
- The other problem is the first 1000 ports are resolved and each port has a specific protocol
- The very moment you use TLS that doubles.
- ALPN allows us to nogotiate with the client and ask them to use 443 for TLS but we will negotiate the application layer protocol that is being embedded in.
- This allows the reuse of 443 for many application layer protocols

How does this work?

- TLS extensions with additional header feilds that are put into the TLS protocol if ALPN is enabled
- The header feild is called **protocol name list** with the list of protocols

SNI (Server Name Indication)

- Chain of Trust in **Public Key Infrastructure (PKI).** Digital certificate is issued to a website
- Imagine having one web server with multiple websites

created with craft

- What happens is the client will send a message to 443 for the ip address asking for the digital certificate
- If the server has multiple websites, there are multiple web certificates. How does he know wich certificate to send
- SNI is a method/header feild which allows the client to send the domain
- Then the web server looks at the domain names that he has and send the matching websites
- Downsides: Overhead
- Allows client to indicate which server name (web site) it wants to communicate with at the given IP address
- Server can inspect the SNI hostname sent in the ClientHello message, select the appropriate certificate

TLS Session resumption

TLS like TCP is a connection oriented protocol

- Connection oriented : before communication we do a handshake and agree on certain parameters
    - Ex: Max window size, Flow window, Session number
    - TLS has timeout: Not communicating for a certain period
- Why a timeout is required: Certain physical resources are reserved for the communication. If the connection ends without killing the connection theres no way to deallocate the resources.
    - Ex: Memory is reserved as the sliding window. size
- After a timeout the handshake has to be performed again. T he server has to spend time resuming sessions and doing the calculations to resume session
- In TLS, there is a session ID. Session resumption allows to resume the old session
- Downside: The server now has to keep track of the sessions and the associated parameters for a certain amount of time.
- This is against the ephemeral key concept of the DH protocol which times out the session keys along with the session. Also if we are storing the keys, we are violating the core concepts of forward secrecy.
- Requires the server to maintain a cache of past sessions and keys

for a small company we suggest to use

- SNI
- But ALPN doesnt have any effects
- Session resumption has to store session data and the requirements pertaining to user experience dictates this decision (Ususally session resumption is run when the server has alot of load on it)

Other than the TLS optimisations that can be enabled or disabled(ALPN,SNI..) there are things as developers that we can do.

Optimising for TLS

This means how we develop the applications and the underlying infrastructure to optimize when using TLS

If properly done it improves,

- Performance
- Security
- UX
- Operational cost

SSL/TSL offloading

- ASIC : A~~pplication specific~~ integrated circuit. (These are circuits iptimized to run certain applications contradicting to CPU which can run everythin from network processing to graphics)
- Instruction set ← FIrmware ← Drivers ← Software
- Routers are usually build using ASICS.
- In the early days asymetric key encryption was really resource intensive so normal CPUs couldnt handle
- So this was offloaded to a machine built specificallyally for TSL processing.
- Requires the public key encryption to be "offloaded" to dedicated machines
- No longer required with modern infrastructure
- Critical to enable session resumption

1-RTT TSL Handshake

- Base handshake required 2RTT(4 messages)
- TLS 1.3 is 1 RTT
- Unoptimized TSL can be worse
  - Multi RTT
  - Slow and ineffective certificate revocation checks
  - Large TLS records

Optimizing the connection reuse

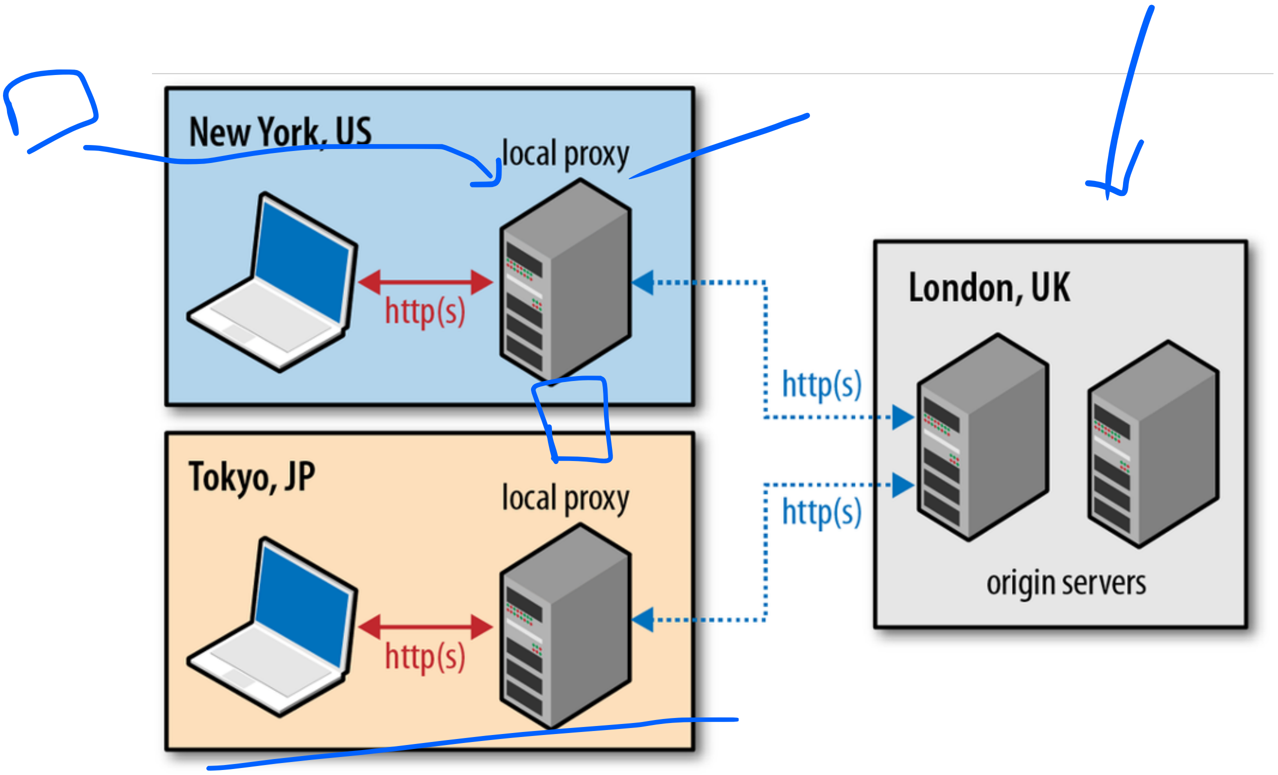
- Similar to session resumption in some ways
- Must be done for TCP and. TLS
  - Verify server and proxy configurations allow keepalive connections
  - Audit your connection timeout settings and change from default
  - Use your logs and analytics to determine the optimal timeout values.

The time out time depends on the service. For something like snapchat the time out time should be small. For something like wikipedia the time out time must be large

CDN usage

Content delivery network

created with craft



CDN takes data from the main server and stores in a local proxy server. For subsequent requests for the similar content the client will not go to the origin server but can access through the local proxy.

- Communication done by the proxy server to users: Lot of intermittent connections to clients
- Communication done by the proxy server to origin server : Stable connection. Should have a very long timeout connection from proxy to the origin
- CDNs are extremely good for static data like movies and songs. But for data that are changing such as websites, something called uncached origin fetch is used.