CS 161 SP19 Final Review

Day 2: Network Monitoring, Special Topics, and more...

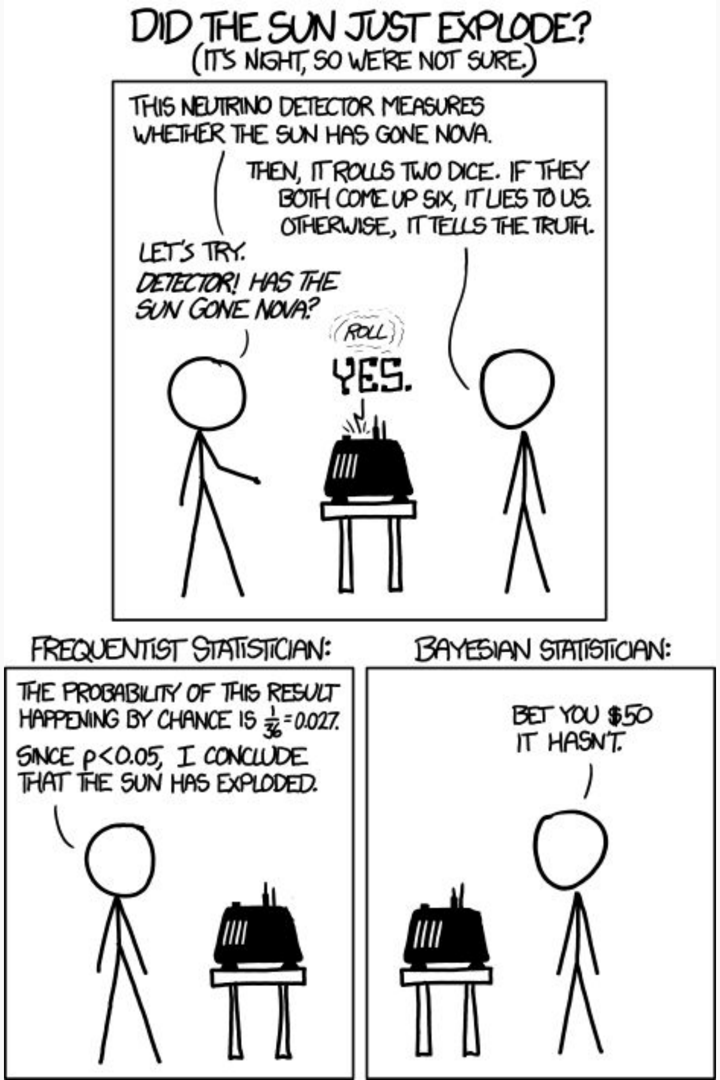
Intrusion Detection

False Positives and False Negatives

- **False positive**: detector *should not* alert, but it *does*
  - Costs $: some poor on-call sap from IT must check it's not actually an issue
- **False negative**: detector *should* alert, but it *does not*
  - Costs $$$: this means we got hacked

Is the cost of a false positive always lower than the cost of a false negative?

iOS optionally “nukes” memory if you fail to login correctly. What happens if you’re just inebriated, and can’t type properly?



Base Rate Fallacy

In general: effectiveness of a detector DEPENDS on how often attacks occur.

IDS Types

- **IDS**: intrusion *detection* system
- **Signature-Based IDS**: alert if request matches a certain pattern
- **Anomaly-Based IDS**: alert on any "weird" occurrences
- **Specification-Based IDS**: alert if request does not match a specification
- **Behavioral-Based IDS**: alert if it looks like bad things have happened
- Understand tradeoffs, pros & cons of each of these

HIDS vs. NIDS

- **HIDS** (host-based IDS): an IDS which runs on a "target" computer itself
  - Better access to *semantics*
  - Protect against local (non-network) threats
- **NIDS** (network-based IDS): an IDS which runs on different computer, monitors network for attacks
  - Cheap, easy to deploy

Questions on Intrusion Detection?

Practice Question

You are doing security things. Consider detectors A, B, and C with false positive and false negative rates shown to the left. You can also use no detector (None). A false negative costs $10,000, a false positive costs $100.

Is there enough information to determine which detector would be the best?

No! We need the base rate of attacks

Name	**FP Rate**	**FN Rate**
A	1%	10%
B	3%	5%
C	2%	6%
None	0%	100%

Practice Question

You are doing security things. Consider detectors A, B, and C with false positive and false negative rates shown to the left. You can also use no detector (None). A false negative costs $10,000, a false positive costs $100.

The base rate of attacks is 8%.

Which detector is the best and how much does it cost per request?

Name	**FP Rate**	**FN Rate**
A	1%	10%
B	3%	5%
C	2%	6%
None	0%	100%

Practice Answer

Personal strategy: put everything in terms of per 100 requests

Reminder: FP costs $100, FN costs $10,000

Name	FP Rate	FN Rate	Requests	Attacks	FPs	FNs	Cost/100
A	1%	10%	100	8	~0.9	~0.8	~$8,100
B	3%	5%	100	8	~2.8	~0.4	~$4,300
C	2%	6%	100	8	~1.8	~0.5	~$5,200
None	0%	100%	100	8	0	~8.0	~$80,000

Practice Answer

Personal strategy: put everything in terms of per 100 requests

Reminder: FP costs $100, FN costs $10,000

Name	FP Rate	FN Rate	Requests	Attacks	FPs	FNs	Cost/100
A	1%	10%	100	8	~0.9	~0.8	~$8,100
B	3%	5%	100	8	~2.8	~0.4	~$4,300
C	2%	6%	100	8	~1.8	~0.5	~$5,200
None	0%	100%	100	8	0	~8.0	~$80,000

Fall 2018 Q2

(d) An engineer worries that the robot may violate the "Three Laws of Robotics" and attacks humans. The engineer suggests that we can add a *secret* termination command to the robot: If someone speaks "cryptocurrency" *proudly and sentimentally* near a Kiwi robot, the robot will instantly turn itself off. The engineer assumes that only a few employees know this command. [2]

We now ask two questions.

⋄ Is accidental turning off when not supposed to *a false positive* or *a false negative*? (No explanation needed)

○ A false positive. ○ A false negative.

Fall 2018 Q2

(d) An engineer worries that the robot may violate the "Three Laws of Robotics" and attacks humans. The engineer suggests that we can add a *secret* termination command to the robot: If someone speaks "cryptocurrency" *proudly and sentimentally* near a Kiwi robot, the robot will instantly turn itself off. The engineer assumes that only a few employees know this command. [2]

We now ask two questions.

⋄ Is accidental turning off when not supposed to *a false positive* or *a false negative*? (No explanation needed)

● A false positive.

○ A false negative.

Abusing Network Monitoring

NSA Spy Tools

- Industry techniques, repurposed for mass surveillance
- **Metadata Drift Nets**: even with encryption, metadata leaks information!
  - Widescale network monitoring on metadata
- **XKEYSCORE**: NIDS plus a frontend to search for "bad" keywords
- **NSA Quantum**: inject TCP packets into HTTP
  - Usually loses its race condition since it first pings classified devices before spoofing
- As far as we are aware, none of these have ever been used on people in the US without an insane amount of paperwork
  - But if you're not in the US, then ¯_(ツ)_/¯

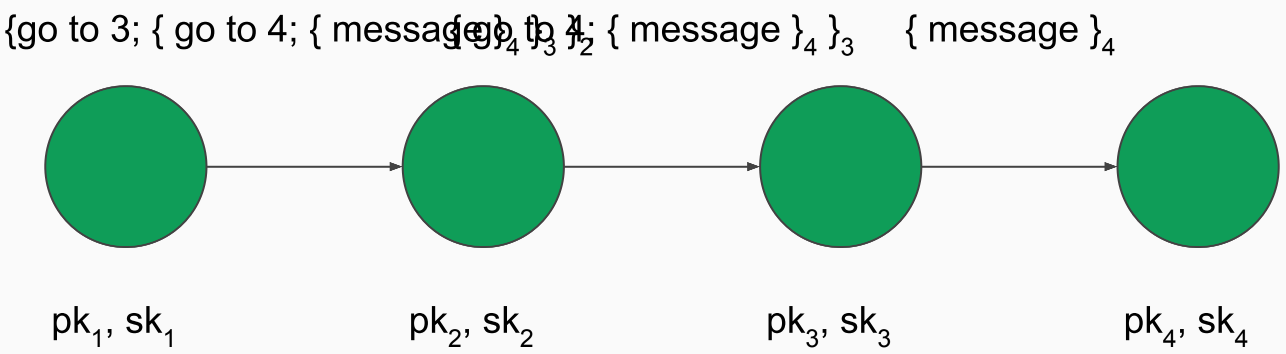
Chinese Censorship Tools

- **Great Firewall** of China: if it detects "bad" keywords, injects TCP RSTs
  - Essentially an on-path NIDS
  - Used for government censorship
- **Great Cannon** of China: injects Javascript into HTTP connections
  - Used in a DDoS attack against Github

Special Topics

Tor

- **Goal**: provide (low-latency) anonymous networking
- **Onion routing**: route message through multiple nodes with encryption
  - Each node "peels" off its layer of encryption
  - (Example below isn't quite how Tor works, diagram is focusing on just onion routing)



Tor Limitations

- Tor does not protect vs. "global passive adversary"
  - View all network traffic, do timing analysis, deanonymize nodes
- "Last-hop" node (exit node) gets unencrypted traffic
  - Effectively a man-in-the-middle
- Does not stop deanonymization in other forms
  - e.g. malware
- Does not provide availability
  - Some countries attempt to block access to Tor

Malware

- **Virus**: malicious piece of code that propagates when the user executes it
- **Worm**: standalone program that copies itself onto target system
  - Like a virus, but can travel without human interaction
- Can include a **rootkit**: kernel patch to hide its presence
- Can be used to make a **botnet**: a collection of infected machines ("bots") under the control of one entity (the "botmaster")
- **Antivirus**: basically a HIDS
  - Primarily signature- & behavior-based

Personal Defense

- All about threat modeling
- Very hard to defend "advanced persistent threats"
  - Nation-state attackers
- Focus on more likely threats:
  - "Common" criminal
  - Intimate partner threat (spousal abuse)

Hardware Attacks

- *Main Idea*: hardware protections are not 100% effective like we thought
- **Rowhammer**: repeated memory writes can cause DRAM bits to flip
  - Flipping (the right) page table bit can let you r00t the operating system
- **Meltdown** and **Spectre**: allow reading other processes' memory
  - Rely on abusing *side-channels* from "speculative" execution
  - Mainly use cache-timing side-channels

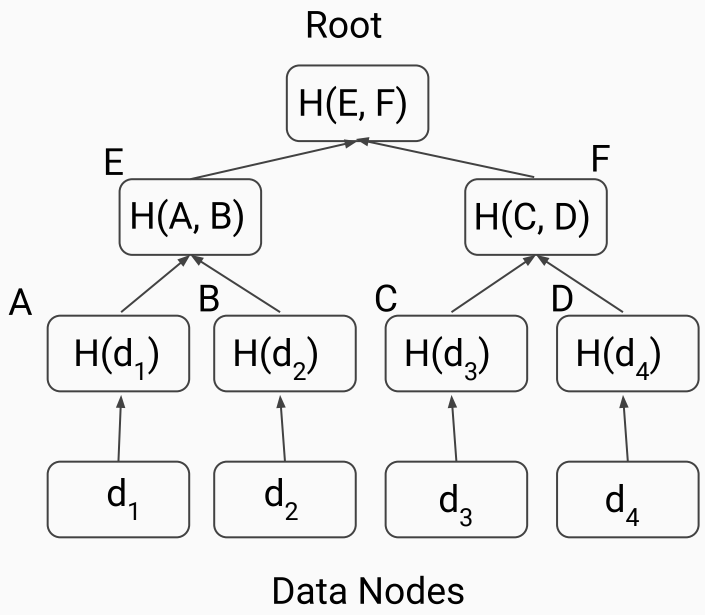
Merkle Trees

Store a file with a hash tree:

- Data is stored in the leaves
- Next level up contains the hashes of the leaves
- All higher nodes contains the hash of their children

If two files are different, then they will have different root hashes.

We can find *where* they differ in logarithmic time with respect to the file length.



Certificate Transparency

Someone (e.g. Google) centrally hosts a “log”: a record of all issued certificates in an append-only merkle tree.

Anyone can detect if host tries to edit the log by checking the root hash.

Anyone can detect if “append-only” is violated. (consistency proof)

Clients only use a certificate if it appears in the log.

Web services can monitor the log for fake versions of their own certificates. (audit proof)

Questions on Special Topics?

DoS / DDoS

What is a DoS?

"No, you don't get to have fun." -- the first DoS-er, probably

Denial-of-Service: typically a much simpler task than other exploits.

Can happen at *any* layer.

Practice:

Design a DoS attack at each layer:

- **Layer 1**:
- **Layer 2**:
- **Layer 3** (IP):
- **Layer 4** (TCP):
- **Layer 7**:

Practice:

Design a DoS attack at each layer:

- **Layer 1**: Cut the wire, jam the radio signal, etc.
- **Layer 2**: MAC Flooding
- **Layer 3** (IP): Ping (ICMP) Flooding
- **Layer 4** (TCP): Just keep initiating new handshakes (SYN Flooding)
- **Layer 7**: Login. Logout. Login. Logout. . . .

Why do these work?

What’s the common denominator?

“Flooding”

The goal is to overwhelm the victim’s resources, and prevent legitimate users from being able to perform legitimate protocols.

Abuse the latency, resource consumption, etc of these protocols ...

(c) (24 points) Now that he has tested his WiFi access, Bob then tells Alice: "I want to buy that last muffin at the counter. Let me check if I have enough money in my bank account." Eve hears this and panics! She wants the last muffin too but is waiting for her friend Mallory to bring enough cash to buy it. She is now determined to somehow stop Bob from buying that last muffin by preventing him from checking his bank account. Through the corner of her eye, Eve sees Bob start to type `https://bank.com` in his browser URL bar . . .

Describe two network attacks Eve can do to prevent Bob from checking his bank account. For each attack, describe clearly in one or two sentences how Eve performs the attack.

Attack #1:

Attack #2:

Solution:

Note that Eve cannot do an ARP or DHCP spoofing attack as Bob has already connected to the WiFi network, so already knows the IP and hardware addresses of the local network's gateway and DNS resolver. (This assumes that extraneous ARPs are not accepted by Bob's system. ARP spoofing is a viable answer for this problem if accompanied by specific mention of this consideration.)

1. TCP RST injection attack — Eve can sniff Bob's transmitted (and received) packets, so she can observe the sequence numbers of TCP packets. Thus, Eve can send a valid TCP RST packet to Bob's browser (or to the bank website), resetting the TCP connection.

2. DNS response spoofing — When Bob tries to load the bank website, his browser will generate a DNS request for the bank's domain. Eve can spoof a response with an incorrect answer, preventing Bob from loading the bank website properly.

3. DoS attack on either Bob's system or the coffee shop network. This can be done through various means, such as DNS amplification attacks directed at Bob.

TLS

SSL/TLS

- Goal is to create a **secure channel**
  - Confidentiality: nobody can see the data communicated
  - Integrity: you know the data has not been changed in-transit
  - Timeliness: you are guaranteed that old data is not being *replayed* by a MITM
- DOES NOT PROVIDE:
  - ~~Anonymity~~: easy to figure out **who** is talking with whom
  - ~~Availability~~: TLS does not prevent DoS or messages being adversarially dropped
  - ~~Padding~~*: one can tell **how much** data is being communicated
  - ~~Non-repudiability~~: not possible to **prove** that this was the message you received

TLS Hardness

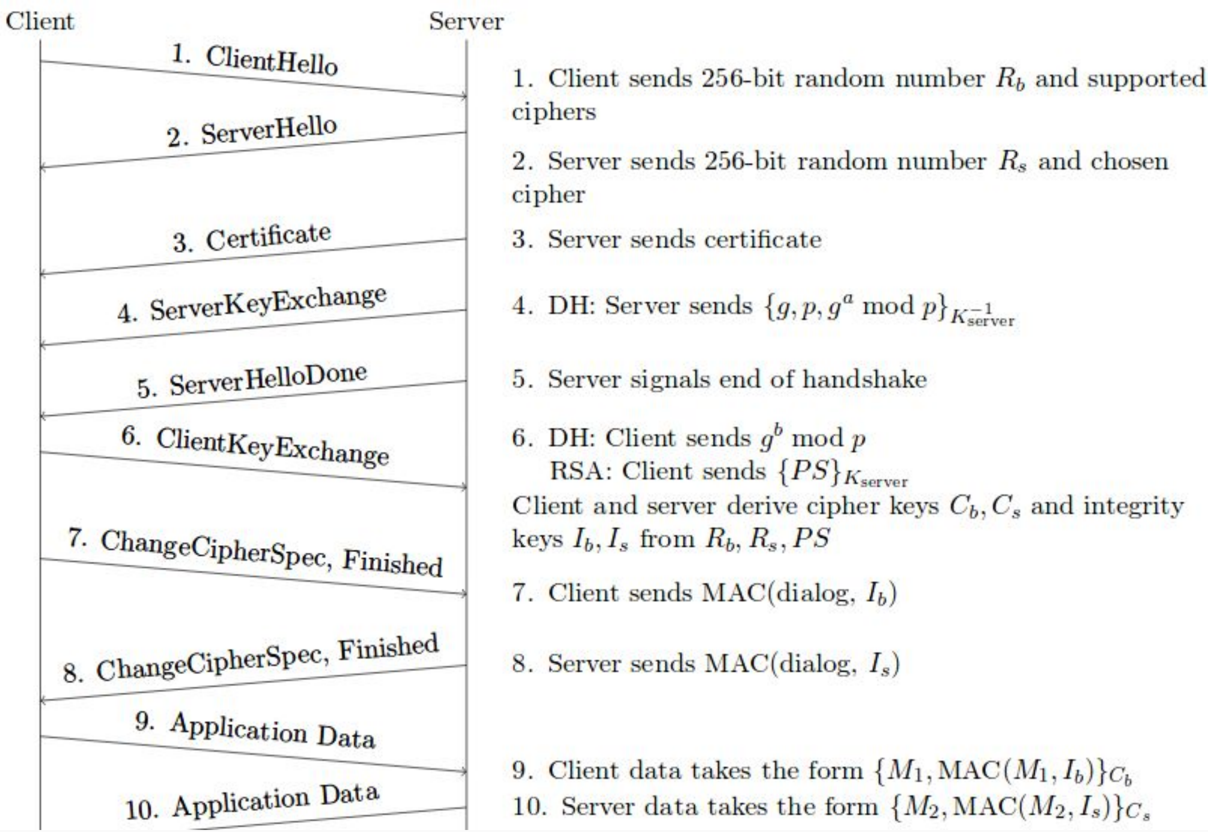
Need to agree on secret *PS*.

For RSA:

- Client sends $\{PS\}_{K_{server}}$ encrypted with RSA
- Need server's private key to decrypt

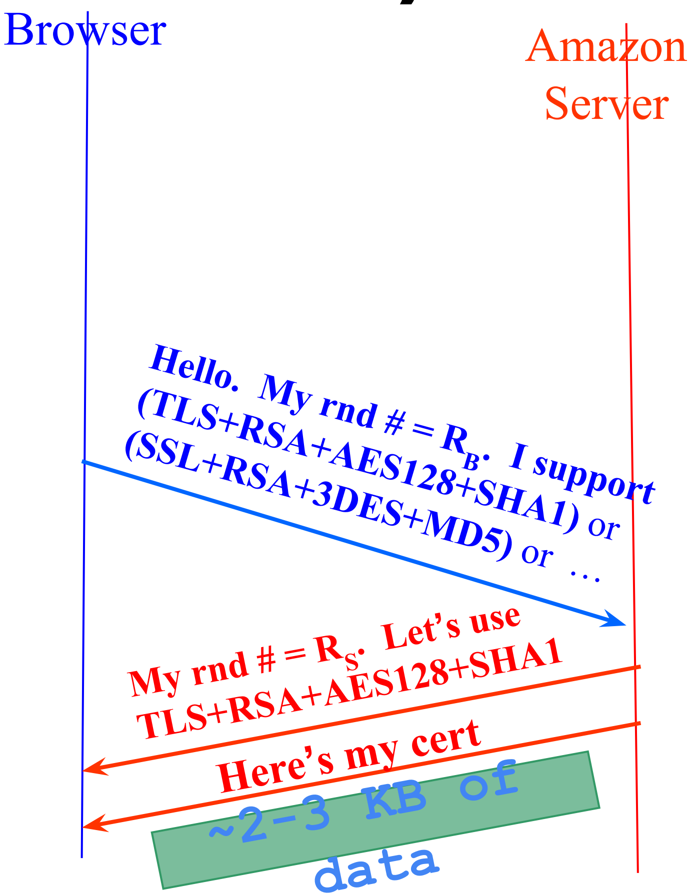
For Diffie-Hellman (all mod *p*):

- Client sends g^b. Server sends g^a. PS is g^{ab}.
- Computational Diffie-Hellman: no (polytime) attacker can get from g^a and g^b to g^{ab}.

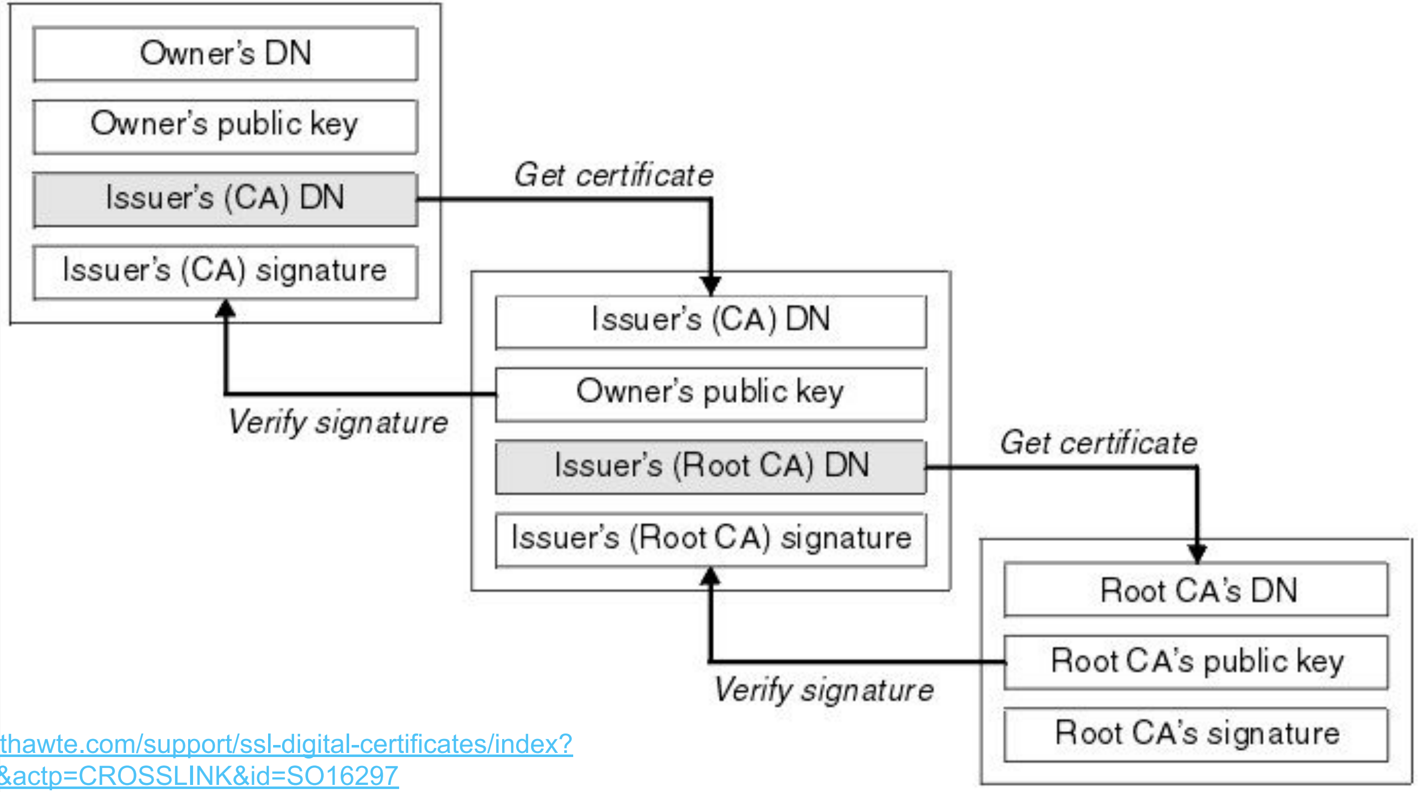


HTTPS Connection (SSL / TLS)

- Browser (client) connects via TCP to Amazon's **HTTPS** server
- Client picks 256-bit random number R_B, sends over list of crypto protocols it supports
- Server picks 256-bit random number R_S, selects protocols to use for this session
- Server sends over its certificate
- ***Client now validates cert***



Certification / Certificate Chain

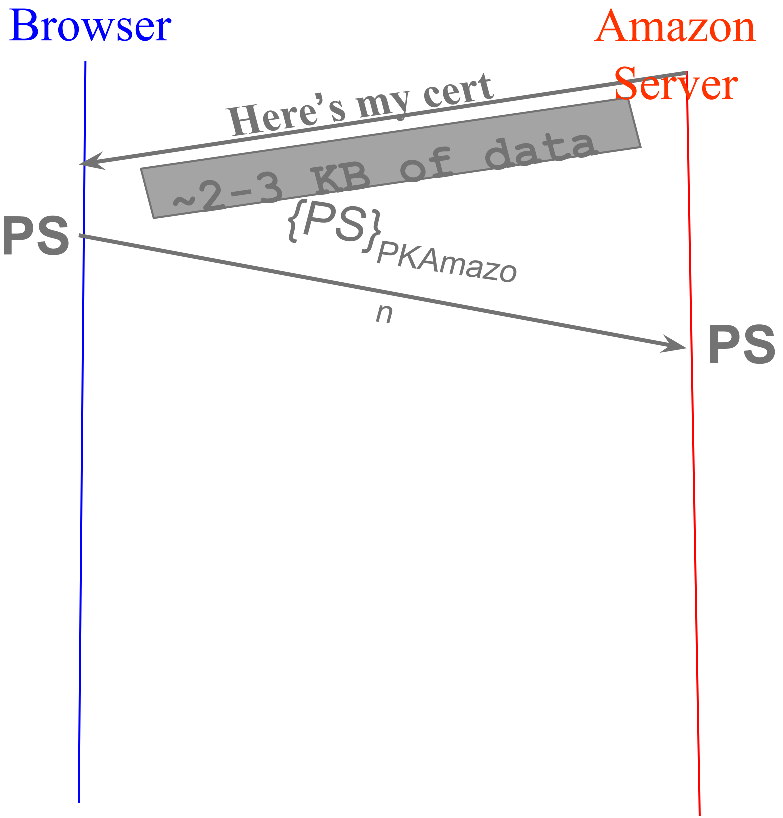


Source:
https://search.thawte.com/support/ssl-digital-certificates/index?page=content&actp=CROSSLINK&id=SO16297

PS via RSA

Q: Forward secrecy?
A: No forward secrecy because attacker can decrypt PS and knows R_B, and R_S and computes secrets

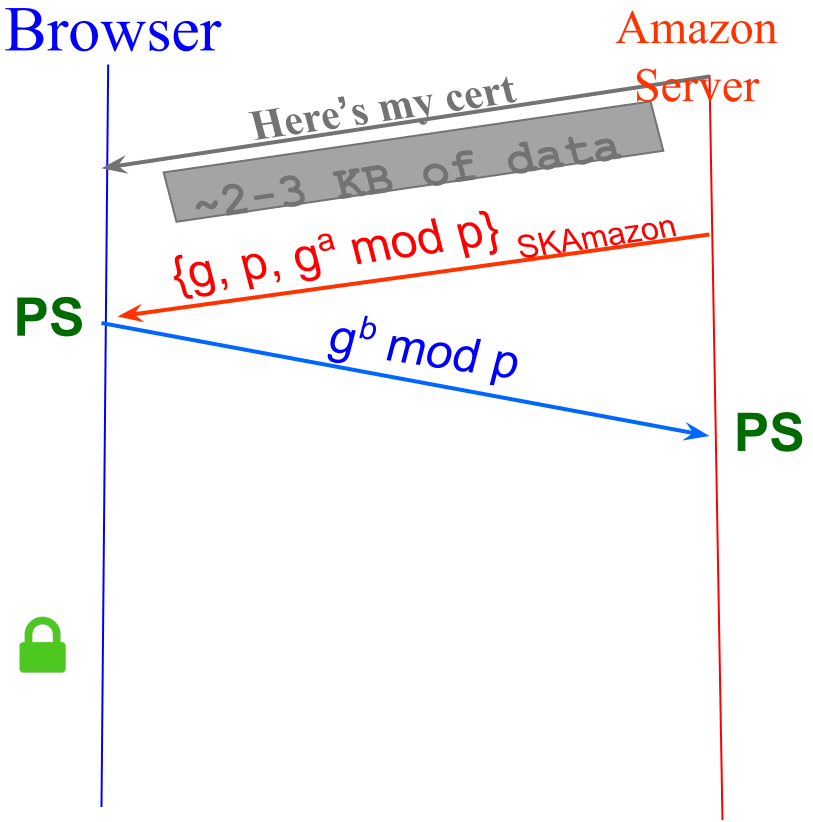
- Client generates premaster secret (PS), encrypts it with server's public key, and sends it to server
  – "If you really have the RSA private key, prove it to me by decrypting this PS."
- Both sides use PS (along with R_b, R_s) to derive **symmetric** keys.



PS via Diffie-Hellman

Q: Forward secrecy?
A: Has forward secrecy because shared secret never sent over the network! If attacker has SK_{Amazon}, cannot find a.

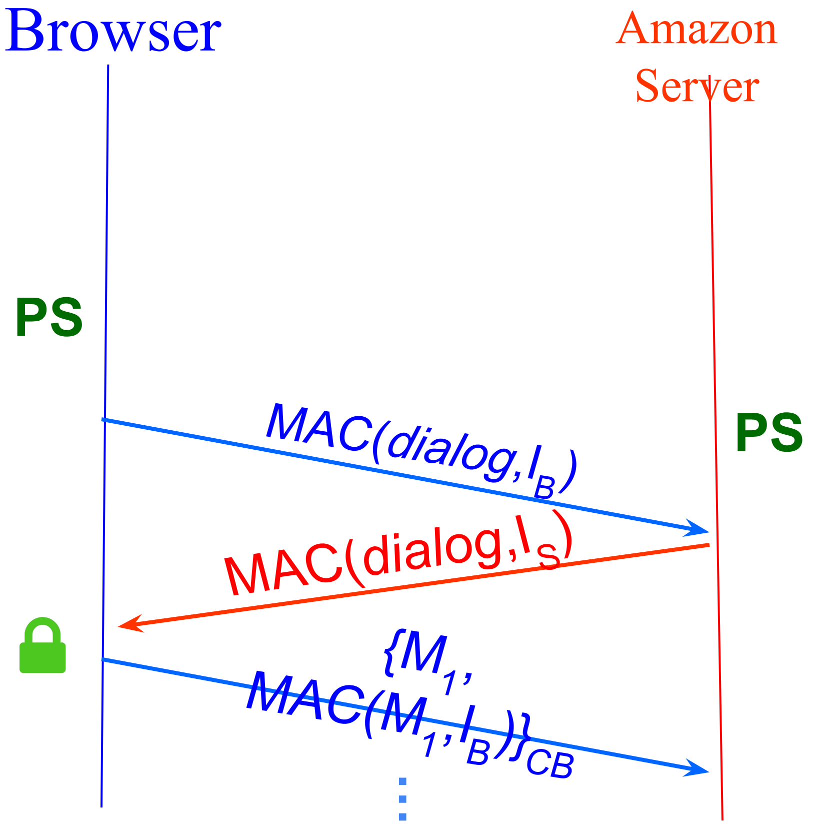
- For Diffie-Hellman, server generates random a, sends public params and g^a mod p
  - **Signed** with server's private key
- Browser verifies signature using PK from certificate
  - (provides security against classic Diffie-Hellman MITM)
- Browser generates random b, computes **PS** = g^{ab} mod p, sends g^b mod p to server
- Server also computes **PS** = g^{ab} mod p
- Both sides use PS to derive symmetric keys.



Symmetric Key Generation

- Both sides now have PS.
- Using PS, both sides derive four keys:
  - C_b - encryption key for messages from **b**rowser to the server
  - C_s - encryption key for messages from **s**erver to the browser
  - I_b, I_s - integrity keys (directional as above)

- Both sides now compute a MAC on the *dialog* (all messages from both sides).
  - If a MITM tampered with the earlier steps (say, R_b, R_s, cipher suite agreement, &c), then they will be detected now.



Questions on TLS?

Spring 2017 - Final

(b) Thanks to strong cryptography, a TLS connection to your bank is secure even if your home router's TCP/IP implementation has a buffer overflow vulnerability.

○ **True** ○ **False**

Spring 2017 - Final

(b) Thanks to strong cryptography, a TLS connection to your bank is secure even if your home router's TCP/IP implementation has a buffer overflow vulnerability.

● True ○ False

- A key property of TLS is how it provides end-to-end security: two systems can communicate using TLS without having to trust any of the intermediaries that forward their traffic. Thus, even if an attacker completely pwns your home router, the worst they can do to you is deny you service to your bank.

Spring 2017 - Final

(a) (6 points) Suppose an attacker steals the private key of a website that uses TLS, and remains undetected. What can the attacker do using the private key? **Mark ALL that apply.**

- ○ Decrypt recorded past TLS sessions that used RSA key exchange.
- ○ Successfully perform a MITM attack on future TLS sessions.
- ○ Decrypt recorded past TLS sessions that used Diffie–Hellman key exchange.
- ○ None of these.

Spring 2017 - Final

(a) (6 points) Suppose an attacker steals the private key of a website that uses TLS, and remains undetected. What can the attacker do using the private key? **Mark ALL that apply.**

● Decrypt recorded past TLS sessions that used RSA key exchange.

● Successfully perform a MITM attack on future TLS sessions.

○ Decrypt recorded past TLS sessions that used Diffie–Hellman key exchange.

○ None of these.

- RSA key exchange offers no forward secrecy, so all past sessions can be decrypted
- With the private key, a MITM can forge the server's signature. The MITM can negotiate a separate TLS connection to client and server, masquerading as the server to the client and vice versa

Fall 2018 Q2

Assumptions: A Kiwi robot connects to `api.kiwicampus.com` to receive commands via TLS, and validates certificates, just as a web browser does. The robots have correctly functioning clocks.

According to some statistics, there were 127 issued certificates for `*.kiwicampus.com`, 56 of them were still valid at the time we made the exam.

(a) TRUE or FALSE: If an attacker obtains the private key for ***any*** one of the 127 certificates, an ***in-path*** attacker can send forged signals to a Kiwi robot.

○ TRUE ○ FALSE

◇ Why? (10 words max)

(b) TRUE or FALSE: If an attacker obtains the private key for ***any*** one of the certificate authorities trusted by the Kiwi robot, an ***in-path*** attacker can send forged signals to a Kiwi robot who will accept.

○ TRUE ○ FALSE

◇ Why? (10 words max)

Fall 2018 Q2

Assumptions: A Kiwi robot connects to `api.kiwicampus.com` to receive commands via TLS, and validates certificates, just as a web browser does. The robots have correctly functioning clocks.

According to some statistics, there were 127 issued certificates for `*.kiwicampus.com`, 56 of them were still valid at the time we made the exam.

(a) TRUE or FALSE: If an attacker obtains the private key for ***any*** one of the 127 certificates, an ***in-path*** attacker can send forged signals to a Kiwi robot.

○ TRUE ● FALSE

◇ Why? (10 words max)

Solution: (2 points) Expired certificates will be rejected

(b) TRUE or FALSE: If an attacker obtains the private key for ***any*** one of the certificate authorities trusted by the Kiwi robot, an ***in-path*** attacker can send forged signals to a Kiwi robot who will accept.

● TRUE ○ FALSE

◇ Why? (10 words max)

Solution: (2 points) Can create bogus certificate

SP 17 Final

(a) (24 points) Suppose the client and server use RSA to exchange the premaster secret. Mallory intercepts the ClientKeyExchange message and replaces PS with a fake value PS'. Assume that Mallory can modify the messages *after* ClientKeyExchange as well, if required. Which of the following are true? **Mark ALL that apply.**

- ○ Mallory will be able to decrypt the application data sent by the client to the server.
- ○ Mallory will be able to decrypt the application data sent by the server to the client.
- ○ The server will detect the tampering when it receives ClientKeyExchange.
- ○ Mallory can avoid detection until the server receives Finished from the client, at which point she'll be detected.
- ○ Mallory can avoid detection until the client receives Finished from the server, at which point she'll be detected.
- ○ None of these

SP 17 Final

(a) (24 points) Suppose the client and server use RSA to exchange the premaster secret. Mallory intercepts the ClientKeyExchange message and replaces PS with a fake value PS'. Assume that Mallory can modify the messages *after* ClientKeyExchange as well, if required. Which of the following are true? **Mark ALL that apply.**

- ○ Mallory will be able to decrypt the application data sent by the client to the server.
- ○ Mallory will be able to decrypt the application data sent by the server to the client.
- ○ The server will detect the tampering when it receives ClientKeyExchange.
- ○ Mallory can avoid detection until the server receives Finished from the client, at which point she'll be detected.
- ● Mallory can avoid detection until the client receives Finished from the server, at which point she'll be detected.
- ○ None of these

(b) (24 points) Turns out that Brewed Awakening's network has no encryption. Alice warns Bob that its not safe to use this connection, but Bob disagrees. Bob connects to the WiFi, and tests that he has Internet connectivity by going to `https://kewlsocialnet.com`. It loads without issues. Bob says the Alice: "See, no problem! That access was totally safe!"

If Bob is correct and the access to `kewlsocialnet.com` was safe, explain why he is correct. If he is not correct, provide a network attack against Bob.

(b) (24 points) Turns out that Brewed Awakening's network has no encryption. Alice warns Bob that its not safe to use this connection, but Bob disagrees. Bob connects to the WiFi, and tests that he has Internet connectivity by going to `https://kewlsocialnet.com`. It loads without issues. Bob says the Alice: "See, no problem! That access was totally safe!"

If Bob is correct and the access to `kewlsocialnet.com` was safe, explain why he is correct. If he is not correct, provide a network attack against Bob.

Solution: Bob is correct.

Bob is visiting an HTTPS website, which uses TLS to provide an end-to-end secure channel. As Bob's browser did not encounter any certificate warnings, then unless there's been a CA breach or some other CA issue, the network connection has confidentiality, authentication, and integrity.

We allowed full credit for solutions that specified that Bob was incorrect and provided a valid approach for undermining his HTTPS connection to the site, including the threat of obtaining fraudulent certs from misbehaving CAs.

We allowed only partial credit for solutions that framed attacks that would work in the situation if TLS did not provide all of the strong security properties that it does. These solutions received more credit if they clearly stated that the attack is relevant for Bob's *subsequent* connections, rather than his test connection. These solutions received less credit if they were simply stating that because the WiFi network is unencrypted, an attacker could read Bob's private information, since use of HTTPS prevents that.

SP 17 (Final)

(c) (24 points) Recall that ClientHello contains a nonce R_b, along with C, the cipher suites supported by the client. ServerHello contains a nonce R_s along with C_{ser}, the cipher suite chosen by the server. Which of the following modifications to the TLS protocol would prevent Mallory from conducting *any* downgrade attacks on the cipher suites? **Mark ALL that apply.**

- ○ ServerKeyExchange includes $[R_b]_{K_{\text{server}}^{-1}}$, $[C]_{K_{\text{server}}^{-1}}$
- ○ ServerKeyExchange includes $[R_b, C]_{K_{\text{server}}^{-1}}$
- ○ ServerKeyExchange includes $[C]_{K_{\text{server}}^{-1}}$
- ○ ServerKeyExchange includes $[C_{\text{ser}}]_{K_{\text{server}}^{-1}}$
- ○ ServerKeyExchange includes $[C]_{K_{\text{server}}^{-1}}$, $[C_{\text{ser}}]_{K_{\text{server}}^{-1}}$
- ○ ServerKeyExchange includes $[C \,||\, C_{\text{ser}}]_{K_{\text{server}}^{-1}}$
- ○ ServerKeyExchange includes $[R_b \,||\, C \,||\, C_{\text{ser}}]_{K_{\text{server}}^{-1}}$
- ○ None of these

SP 17 (Final)

c) (24 points) Recall that ClientHello contains a nonce R_b, along with C, the cipher suites supported by the client. ServerHello contains a nonce R_s along with C_{ser}, the cipher suite chosen by the server. Which of the following modifications to the TLS protocol would prevent Mallory from conducting *any* downgrade attacks on the cipher suites? **Mark ALL that apply.**

- ○ ServerKeyExchange includes $[R_b]_{K^{-1}_{\text{server}}}$, $[C]_{K^{-1}_{\text{server}}}$
- ○ ServerKeyExchange includes $[R_b, C]_{K^{-1}_{\text{server}}}$
- ○ ServerKeyExchange includes $[C]_{K^{-1}_{\text{server}}}$
- ○ ServerKeyExchange includes $[C_{\text{ser}}]_{K^{-1}_{\text{server}}}$
- ○ ServerKeyExchange includes $[C]_{K^{-1}_{\text{server}}}$, $[C_{\text{ser}}]_{K^{-1}_{\text{server}}}$
- ● ServerKeyExchange includes $[C \mid\mid C_{\text{ser}}]_{K^{-1}_{\text{server}}}$
- ● ServerKeyExchange includes $[R_b \mid\mid C \mid\mid C_{\text{ser}}]_{K^{-1}_{\text{server}}}$
- ○ None of these

TLS Limitations/Issues

- The system requires us to trust ALL Certificate Authorities
- Certificate management is complicated
- TLS can’t protect against logical errors on hosts.

WPA2

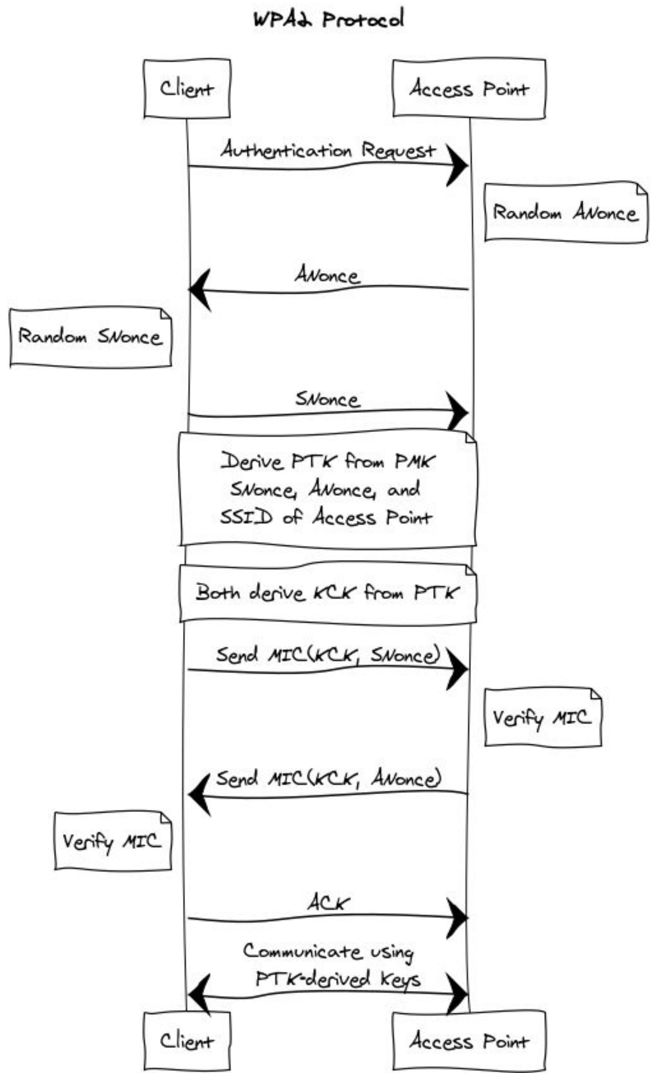
Local security

- Goal is to secure Layer 2 (between local hosts) communication
- Two major types:
    - WPA2-PSK: everyone has a shared "WiFi password", used to derive keys to communicate
        - Anyone who knows the WiFi password can see your traffic.
    - WPA2-Enterprise: everyone has a different username and WiFi password
        - Like AirBears2
        - **Main advantage:** other people cannot derive your *keys*, because they do not have your password!

WPA2-PSK is like TLS

They're basically the same:

- Use of nonces to prevent replay attacks
- No need for certificates or to agree on a PS here -- the WiFi password / PMK is sufficient for shared knowledge
- MIC(KCK, nonces) is like the MAC on the dialogues for TLS



Questions on WPA2?

Problem 6 ***Coffee Shop Worries*** **(54 points)**

Alice and Bob just arrived at Brewed Awakening, the local coffee shop. Eve is already there, enjoying a cup of tea.

(a) (6 points) Alice wants to connect to Brewed Awakening's WiFi network. Under which protocols would her connections be safe from sniffing attacks by other coffee shop visitors, such as Eve? **Mark all that apply.**

not in scope, but if you care: WEP is like WPA but it sucks

- ○ WEP
- ○ WPA2 - Enterprise mode
- ○ WPA2 - Personal mode
- ○ None of these

Problem 6 ***Coffee Shop Worries*** **(54 points)**

Alice and Bob just arrived at Brewed Awakening, the local coffee shop. Eve is already there, enjoying a cup of tea.

(a) (6 points) Alice wants to connect to Brewed Awakening's WiFi network. Under which protocols would her connections be safe from sniffing attacks by other coffee shop visitors, such as Eve? **Mark all that apply.**

○ WEP

● WPA2 - Enterprise mode

○ WPA2 - Personal mode

○ None of these

Solution: Only "WPA2 - Enterprise mode" provides per-connection secret keys with the WiFi access point to secure each connection separately.

Good luck! <3