CS 161 Midterm 2 Review

Grant Ho, Frank Li, Rishabh Poddar

RSA (Confidentiality)

Alice’s key pair

- Public key = (n, e)
- Private key = d

e, d such that

$X = X^{e.d} \bmod n$

RSA (Confidentiality)

Alice’s key pair

- Public key = (n, e)
- Private key = d

e, d such that

$$X = X^{e.d} \bmod n$$

Pick primes p, q

$$n = pq$$

$$\varphi(n) = (p-1)(q-1)$$

Pick $2 < e < \varphi(n)$

Compute d such that

$$e.d = 1 \bmod \varphi(n)$$

RSA (Confidentiality)

Alice’s key pair

- Public key = (n, e)
- Private key = d

e, d such that
$X = X^{e.d} \bmod n$

Bob encrypts M using Alice’s public key

- $C = M^e \bmod n$
- Sends C to Alice

RSA (Confidentiality)

Alice’s key pair

- Public key = (n, e)
- Private key = d

e, d such that
$X = X^{e.d} \bmod n$

Bob encrypts M using Alice’s public key

- $C = M^e \bmod n$
- Sends C to Alice

Alice decrypts C using her private key

- $M = C^d \bmod n$

RSA (Integrity + Authentication)

Alice’s key pair

- Public key = (n, e)
- Private key = d

e, d such that

$X = X^{e.d} \bmod n$

RSA (Integrity + Authentication)

Alice’s key pair

- Public key = (n, e)
- Private key = d

e, d such that
$X = X^{e.d} \bmod n$

Alice signs M using her private key

- $S = H(M)^d \bmod n$
- Publishes (M, S)

RSA (Integrity + Authentication)

Alice’s key pair

- Public key = (n, e)
- Private key = d

e, d such that
$X = X^{e.d} \bmod n$

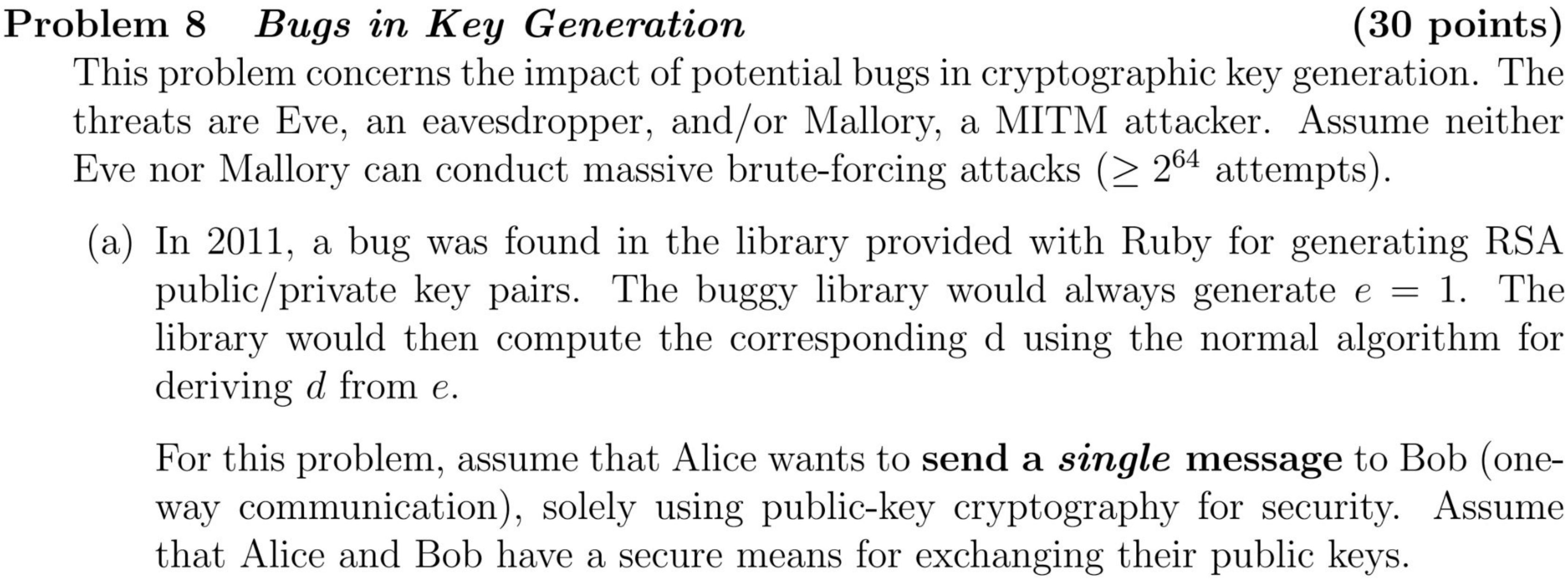
Alice signs M using her private key

- $S = H(M)^d \bmod n$
- Publishes (M, S)

Anyone can verify signature using Alice’s public key

- $H(M) = S^e \bmod n$

Crypto (Sp13 - Final)

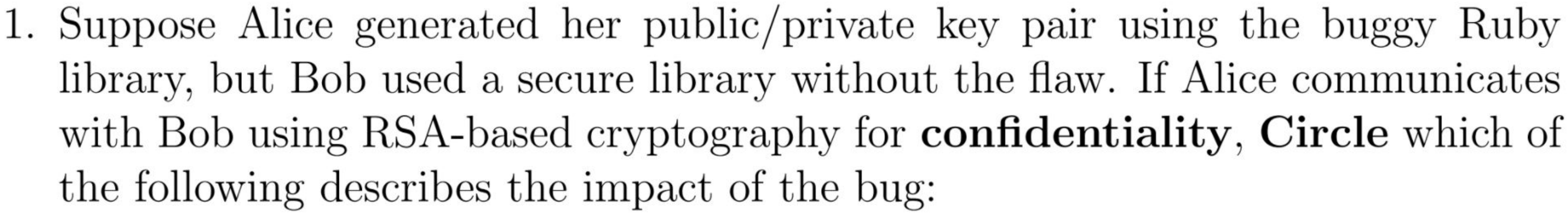


Problem 8 ***Bugs in Key Generation*** **(30 points)**

This problem concerns the impact of potential bugs in cryptographic key generation. The threats are Eve, an eavesdropper, and/or Mallory, a MITM attacker. Assume neither Eve nor Mallory can conduct massive brute-forcing attacks ($\geq 2^{64}$ attempts).

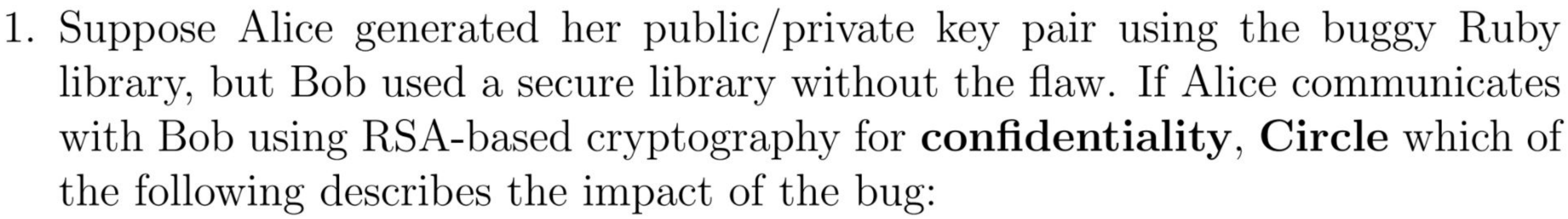
(a) In 2011, a bug was found in the library provided with Ruby for generating RSA public/private key pairs. The buggy library would always generate $e = 1$. The library would then compute the corresponding d using the normal algorithm for deriving d from e.

For this problem, assume that Alice wants to **send a *single* message** to Bob (one-way communication), solely using public-key cryptography for security. Assume that Alice and Bob have a secure means for exchanging their public keys.



1. Suppose Alice generated her public/private key pair using the buggy Ruby library, but Bob used a secure library without the flaw. If Alice communicates with Bob using RSA-based cryptography for **confidentiality**, **Circle** which of the following describes the impact of the bug:

- No effect in this scenario
- Bob can read Alice’s messages, but so can Eve
- Bob will not be able to read Alice’s messages



1. Suppose Alice generated her public/private key pair using the buggy Ruby library, but Bob used a secure library without the flaw. If Alice communicates with Bob using RSA-based cryptography for **confidentiality**, **Circle** which of the following describes the impact of the bug:

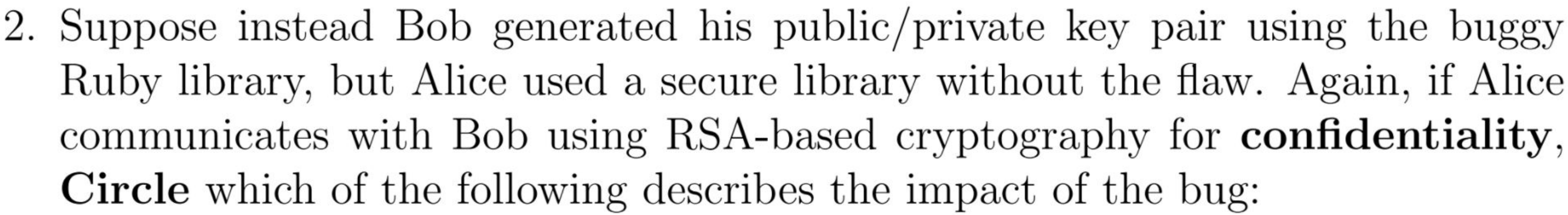
- No effect in this scenario
- Bob can read Alice's messages, b
- Bob will not be able to read Alice'

Alice's keys : n_A, e_A and d_A

If $e_A = 1$, then $d_A = 1$

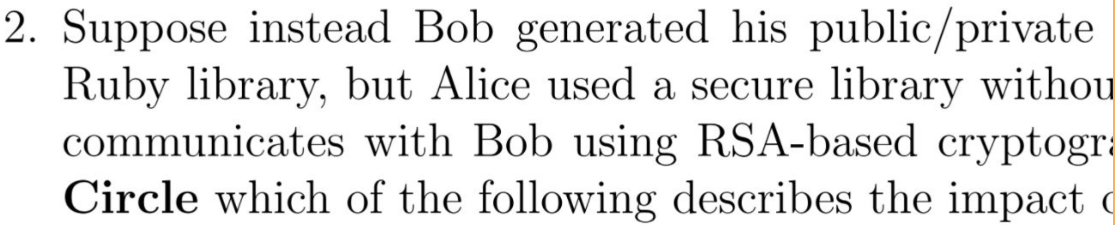
Bob's keys : n_B, e_B and d_B

$A \rightarrow B : M^{eB} \bmod n$



2. Suppose instead Bob generated his public/private key pair using the buggy Ruby library, but Alice used a secure library without the flaw. Again, if Alice communicates with Bob using RSA-based cryptography for **confidentiality**, **Circle** which of the following describes the impact of the bug:

- No effect in this scenario
- Bob can read Alice’s messages, but so can Eve
- Bob will not be able to read Alice’s messages



2. Suppose instead Bob generated his public/private
Ruby library, but Alice used a secure library withou
communicates with Bob using RSA-based cryptogr
Circle which of the following describes the impact c

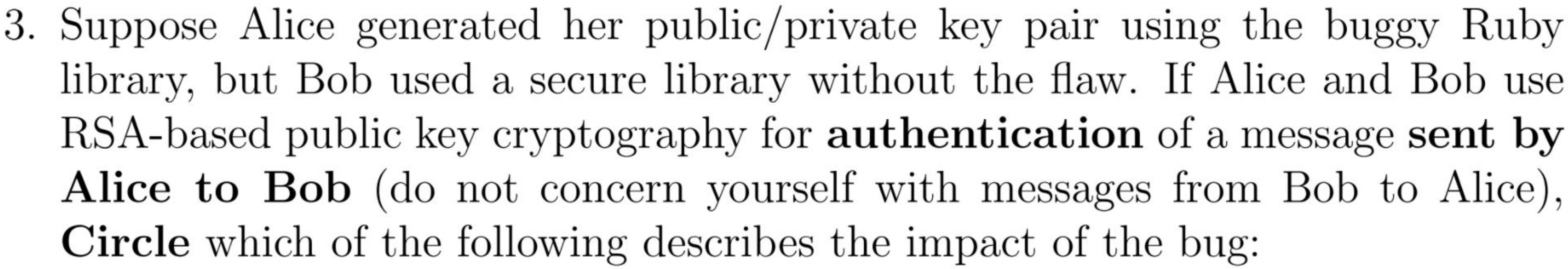
Alice’s keys : n_A, e_A and d_A

Bob’s keys : n_B, e_B and d_B

If $e_B = 1$, then $d_B = 1$

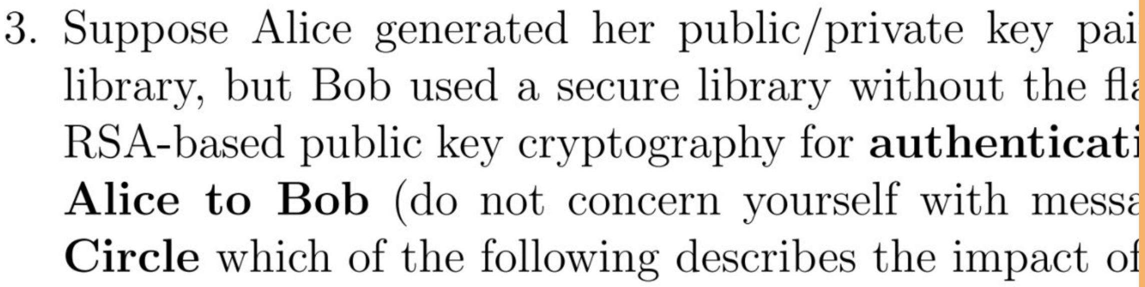
$A \rightarrow B : M^{eB} \bmod n$
$= M^1 \bmod n$

- No effect in this scenario
- Bob can read Alice’s messages, but so can Eve
- Bob will not be able to read Alice’s messages



3. Suppose Alice generated her public/private key pair using the buggy Ruby library, but Bob used a secure library without the flaw. If Alice and Bob use RSA-based public key cryptography for **authentication** of a message **sent by Alice to Bob** (do not concern yourself with messages from Bob to Alice), **Circle** which of the following describes the impact of the bug:

- No effect in this scenario
- Alice’s message will not verify to Bob as properly signed
- Mallory can construct a message that appears to have a valid signature from Alice



3. Suppose Alice generated her public/private key pai
library, but Bob used a secure library without the fla
RSA-based public key cryptography for **authenticati**
Alice to Bob (do not concern yourself with messa
Circle which of the following describes the impact of

Alice’s keys : n_A, e_A and d_A

If $e_A = 1$, then $d_A = 1$

Bob’s keys : n_B, e_B and d_B

A → B : M, S

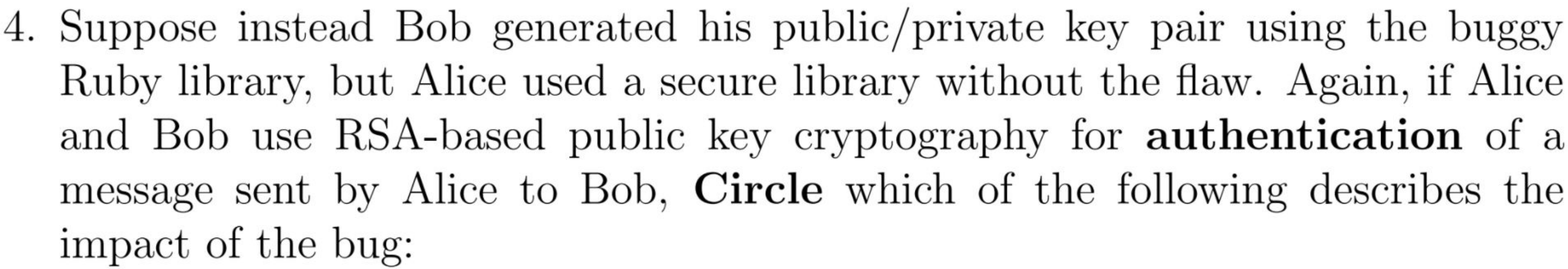
$S = H(M)^{dA} \text{ mod } n$

$= H(M)^1 \text{ mod } n$

Mallory can create (M’, S’)

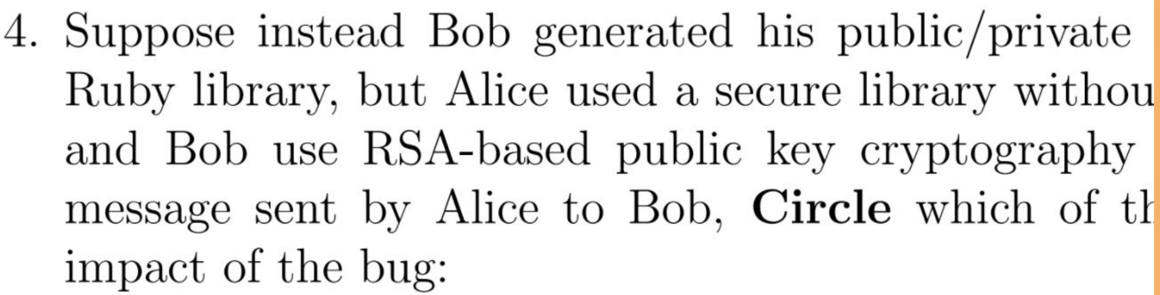
$S' = H(M') \text{ mod } n$

- No effect in this scenario
- Alice’s message will not verify to Bob as properl
- Mallory can construct a message that appears to have a valid signature from Alice



4. Suppose instead Bob generated his public/private key pair using the buggy Ruby library, but Alice used a secure library without the flaw. Again, if Alice and Bob use RSA-based public key cryptography for **authentication** of a message sent by Alice to Bob, **Circle** which of the following describes the impact of the bug:

- No effect in this scenario
- Alice’s message will not verify to Bob as properly signed
- Mallory can construct a message that appears to have a valid signature from Alice



4. Suppose instead Bob generated his public/private
Ruby library, but Alice used a secure library withou
and Bob use RSA-based public key cryptography
message sent by Alice to Bob, **Circle** which of th
impact of the bug:

Alice’s keys : n_A, e_A and d_A

Bob’s keys : n_B, e_B and d_B

If $e_B = 1$, then $d_B = 1$

A → B : M, S

$S = H(M)^{dA} \bmod n$

- No effect in this scenario
- Alice’s message will not verify to Bob as properly signed
- Mallory can construct a message that appears to have a valid signature from Alice

Crypto (Sp13 - Final)

1

Alice sends to Bob: $E_{K_A}(M \mid\mid \mathrm{Sign}_{K_A^{-1}}(\mathrm{SHA}(M)))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

1

$$\text{Alice sends to Bob: } E_{K_A}(M \,||\, \text{Sign}_{K_A^{-1}}(\text{SHA}(M)))$$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Bob needs Alice’s private key to decrypt

Crypto (Sp13 - Final)

2

Alice sends to Bob: $E_{K_B}(M \,||\, \mathrm{Sign}_{K_B^{-1}}(\mathrm{SHA}(M)))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

2

Alice sends to Bob: $E_{K_B}(M \mid\mid \text{Sign}_{K_B^{-1}}(\text{SHA}(M)))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Alice needs Bob’s private key to sign

Crypto (Sp13 - Final)

3

Alice sends to Bob: $E_{K_A}(M), \mathrm{Sign}_{K_B^{-1}}(\mathrm{SHA}(M))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

3

Alice sends to Bob: $E_{K_A}(M), \text{Sign}_{K_B^{-1}}(\text{SHA}(M))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Bob needs Alice's private key to decrypt

Alice needs Bob's private key to sign

Crypto (Sp13 - Final)

4

Alice sends to Bob: $E_{K_B}(M), \mathrm{Sign}_{K_A^{-1}}(\mathrm{SHA}(M))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

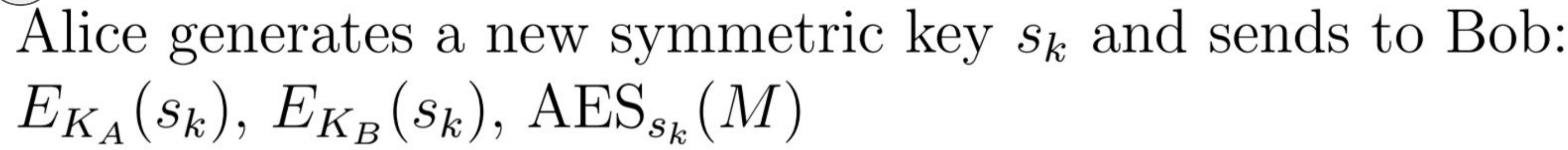
4

Alice sends to Bob: $E_{K_B}(M), \mathrm{Sign}_{K_A^{-1}}(\mathrm{SHA}(M))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

5



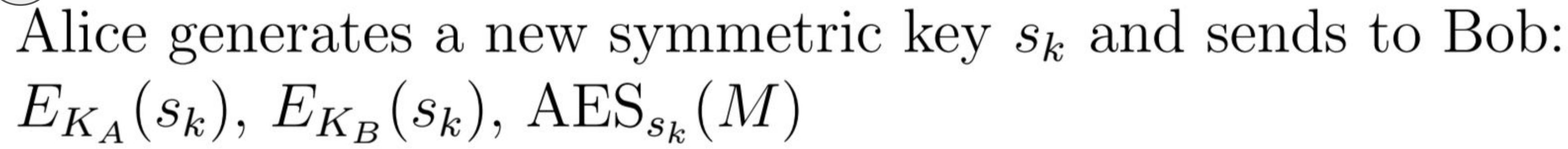
Alice generates a new symmetric key s_k and sends to Bob:
$E_{K_A}(s_k)$, $E_{K_B}(s_k)$, $\mathrm{AES}_{s_k}(M)$

Assume M is single block msg

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

5



Alice generates a new symmetric key s_k and sends to Bob:
$E_{K_A}(s_k)$, $E_{K_B}(s_k)$, $\mathrm{AES}_{s_k}(M)$

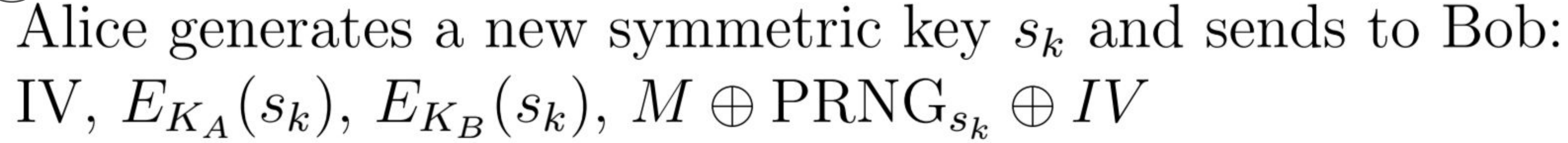
1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Only encryption; does not provide integrity or authentication

Alice doesn’t sign, hence no non-repudiation

Crypto (Sp13 - Final)

6



Alice generates a new symmetric key s_k and sends to Bob:
IV, $E_{K_A}(s_k)$, $E_{K_B}(s_k)$, $M \oplus \mathrm{PRNG}_{s_k} \oplus IV$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

6

Alice generates a new symmetric key s_k and sends to Bob:
$\mathrm{IV},\ E_{K_A}(s_k),\ E_{K_B}(s_k),\ M \oplus \mathrm{PRNG}_{s_k} \oplus IV$

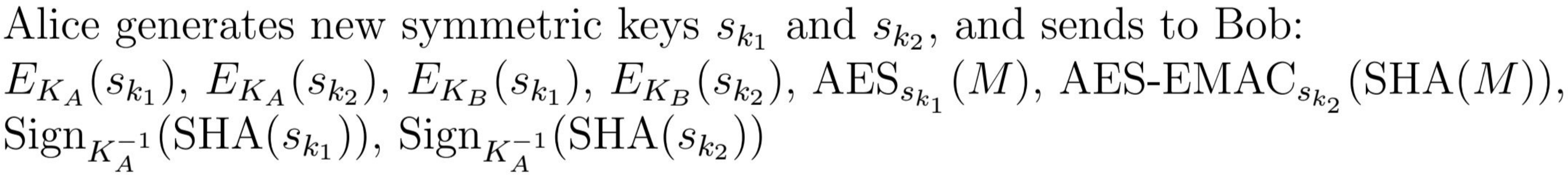
1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Only encryption (stream cipher); does not provide integrity or authentication

Alice doesn’t sign, hence no non-repudiation

Crypto (Sp13 - Final)

7

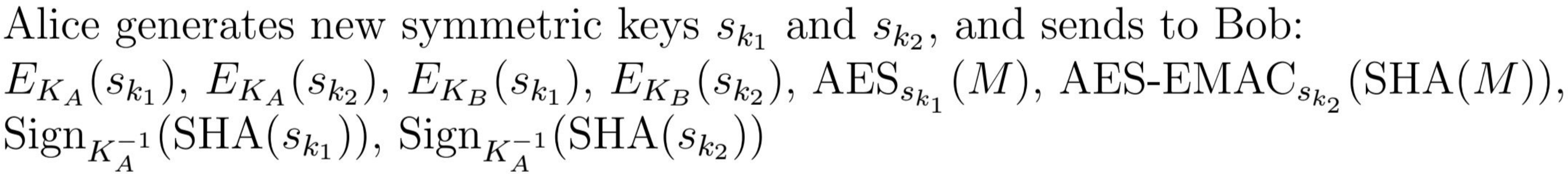


Alice generates new symmetric keys s_{k_1} and s_{k_2}, and sends to Bob:
$E_{K_A}(s_{k_1})$, $E_{K_A}(s_{k_2})$, $E_{K_B}(s_{k_1})$, $E_{K_B}(s_{k_2})$, $\text{AES}_{s_{k_1}}(M)$, $\text{AES-EMAC}_{s_{k_2}}(\text{SHA}(M))$,
$\text{Sign}_{K_A^{-1}}(\text{SHA}(s_{k_1}))$, $\text{Sign}_{K_A^{-1}}(\text{SHA}(s_{k_2}))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

7



Alice generates new symmetric keys s_{k_1} and s_{k_2}, and sends to Bob:
$E_{K_A}(s_{k_1})$, $E_{K_A}(s_{k_2})$, $E_{K_B}(s_{k_1})$, $E_{K_B}(s_{k_2})$, $\text{AES}_{s_{k_1}}(M)$, $\text{AES-EMAC}_{s_{k_2}}(\text{SHA}(M))$,
$\text{Sign}_{K_A^{-1}}(\text{SHA}(s_{k_1}))$, $\text{Sign}_{K_A^{-1}}(\text{SHA}(s_{k_2}))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

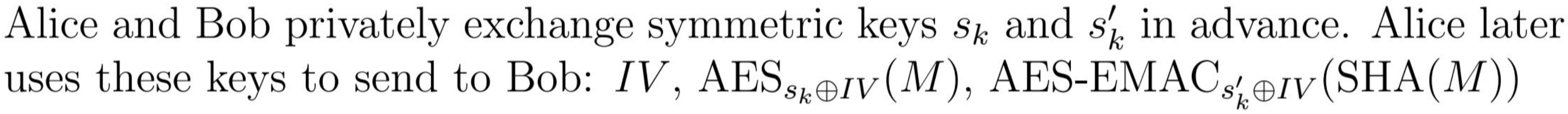
Encryption provides confidentiality

MAC provides integrity + authentication

Alice doesn’t sign ciphertext, hence no non-repudiation

Crypto (Sp13 - Final)

8



Alice and Bob privately exchange symmetric keys s_k and s'_k in advance. Alice later uses these keys to send to Bob: IV, $\text{AES}_{s_k \oplus IV}(M)$, $\text{AES-EMAC}_{s'_k \oplus IV}(\text{SHA}(M))$

1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Crypto (Sp13 - Final)

8

Alice and Bob privately exchange symmetric keys s_k and s'_k in advance. Alice later uses these keys to send to Bob: IV, $\text{AES}_{s_k \oplus IV}(M)$, $\text{AES-EMAC}_{s'_k \oplus IV}(\text{SHA}(M))$

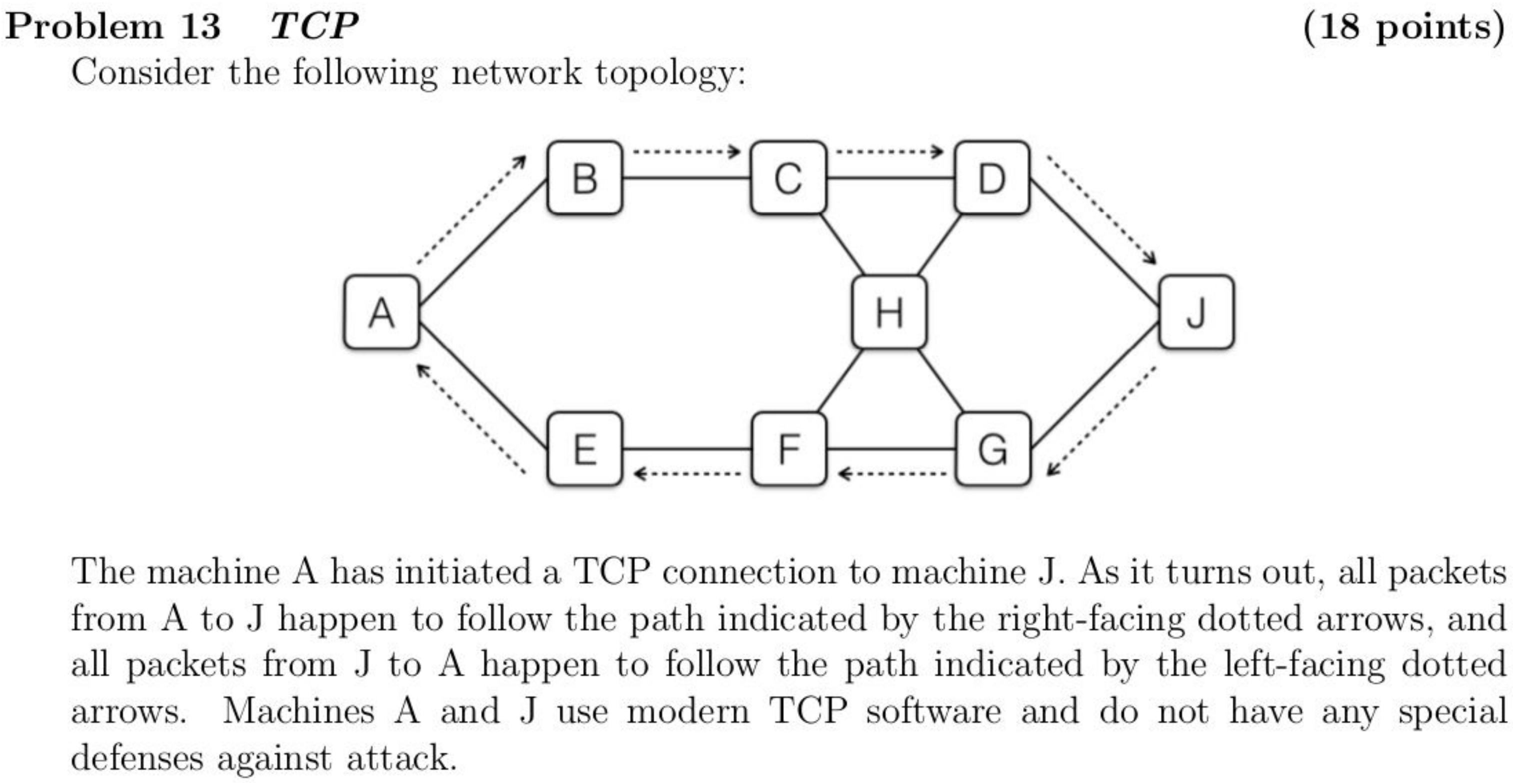
1. Broken
2. Confidentiality
3. Integrity
4. Authentication
5. Non-repudiation
6. None

Encryption provides confidentiality

MAC provides integrity + authentication

Alice doesn’t sign, hence no non-repudiation

TCP Injection Practice Question (SP 14 - Final)

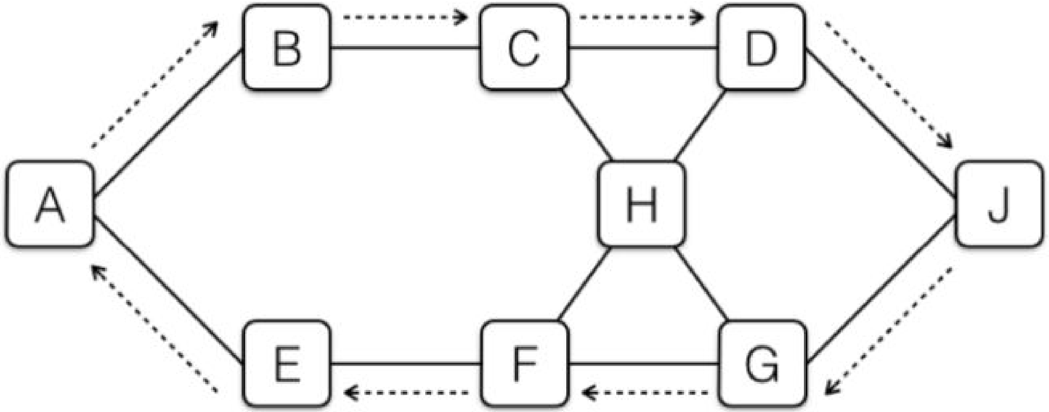


Problem 13 ***TCP*** **(18 points)**

Consider the following network topology:

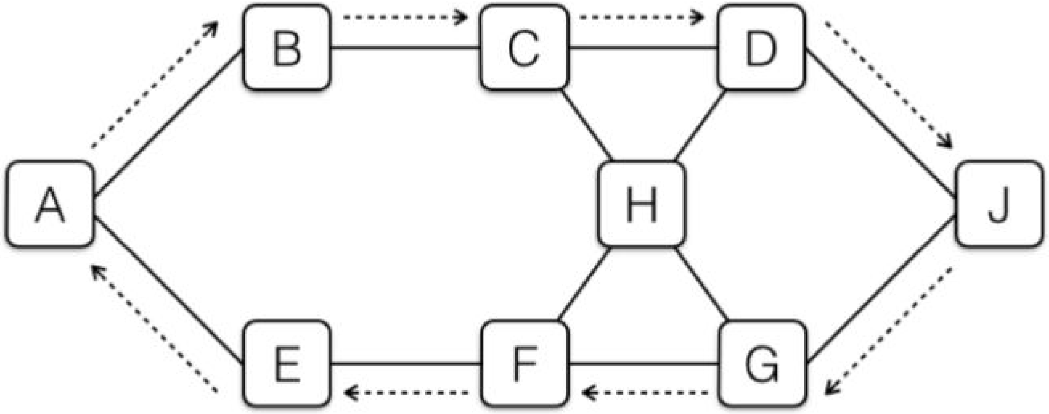
The machine A has initiated a TCP connection to machine J. As it turns out, all packets from A to J happen to follow the path indicated by the right-facing dotted arrows, and all packets from J to A happen to follow the path indicated by the left-facing dotted arrows. Machines A and J use modern TCP software and do not have any special defenses against attack.

TCP Injection Practice Question



(a) Suppose that Mallory controls (only) machine C. Can she inject RST packets destined for machine J into this TCP connection, such that they will be accepted by machine J? Why or why not?

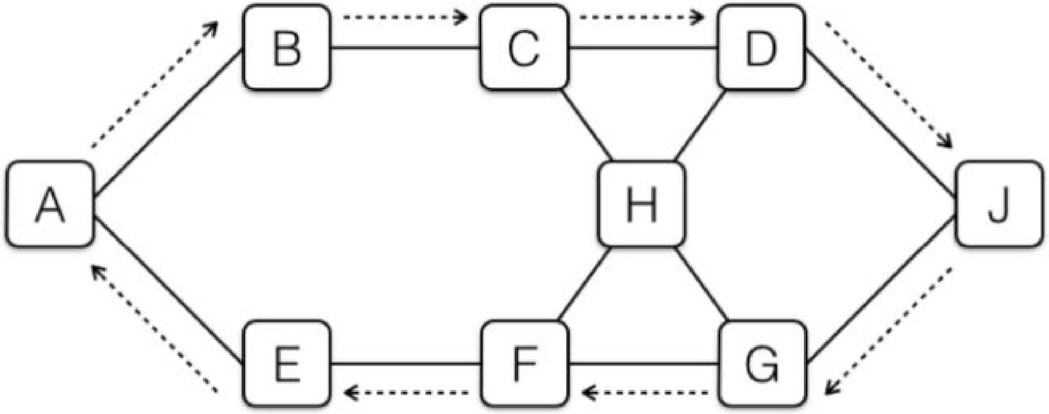
TCP Injection Practice Question



(a) Suppose that Mallory controls (only) machine C. Can she inject RST packets destined for machine J into this TCP connection, such that they will be accepted by machine J? Why or why not?

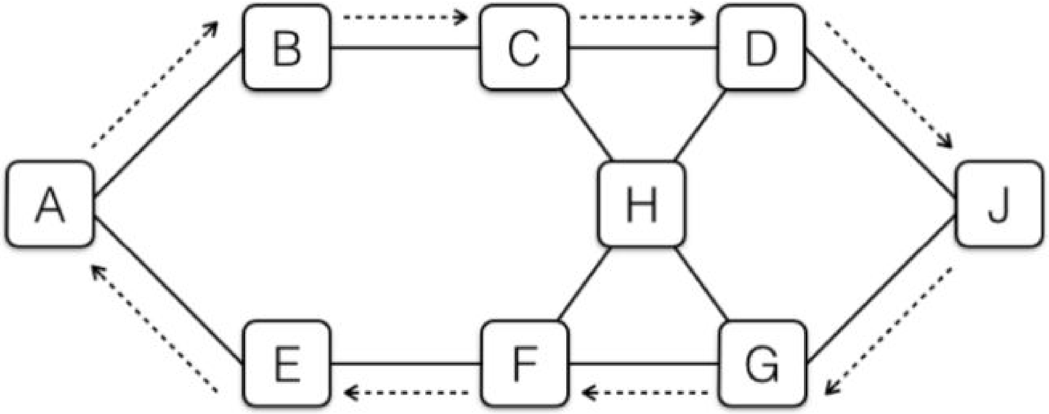
Yes, she's on-path and can see A's TCP seq #, so can inject a valid forged RST packet to J.

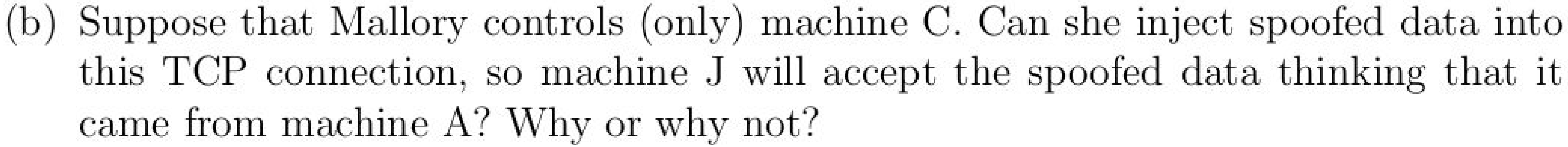
TCP Injection Practice Question



(b) Suppose that Mallory controls (only) machine C. Can she inject spoofed data into this TCP connection, so machine J will accept the spoofed data thinking that it came from machine A? Why or why not?

TCP Injection Practice Question

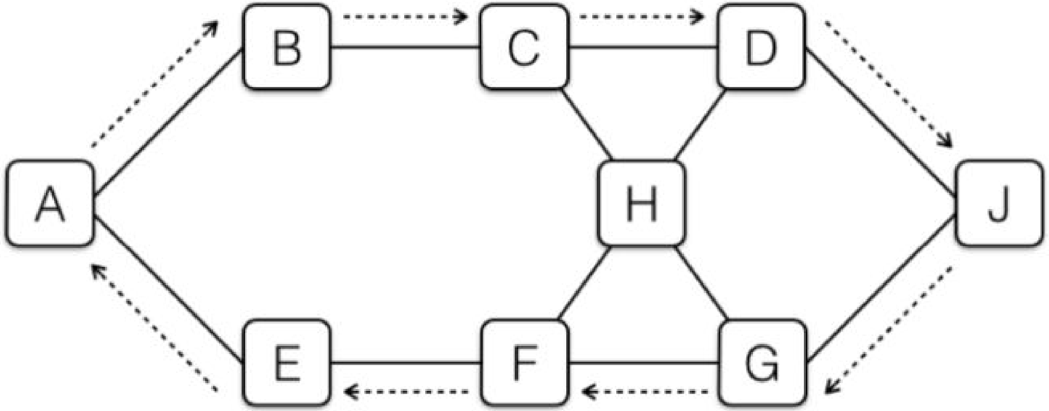


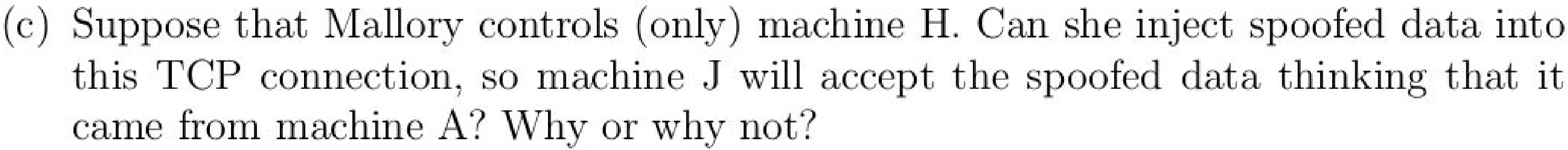


(b) Suppose that Mallory controls (only) machine C. Can she inject spoofed data into this TCP connection, so machine J will accept the spoofed data thinking that it came from machine A? Why or why not?

Yes, she’s on-path and can see A’s TCP seq #, so can inject valid forged data packet to J.

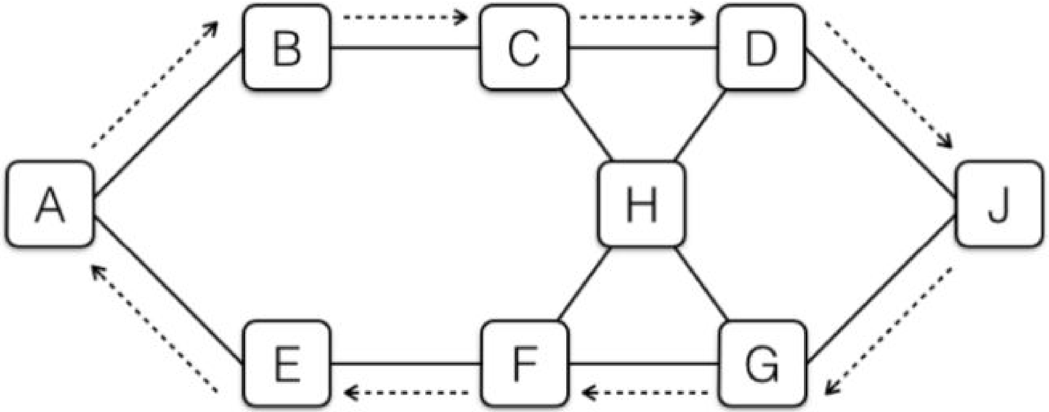
TCP Injection Practice Question





(c) Suppose that Mallory controls (only) machine H. Can she inject spoofed data into this TCP connection, so machine J will accept the spoofed data thinking that it came from machine A? Why or why not?

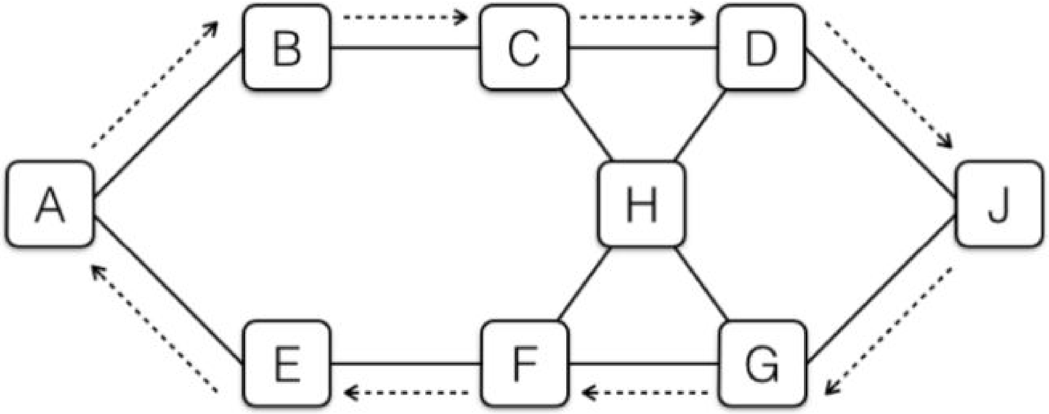
TCP Injection Practice Question



(c) Suppose that Mallory controls (only) machine H. Can she inject spoofed data into this TCP connection, so machine J will accept the spoofed data thinking that it came from machine A? Why or why not?

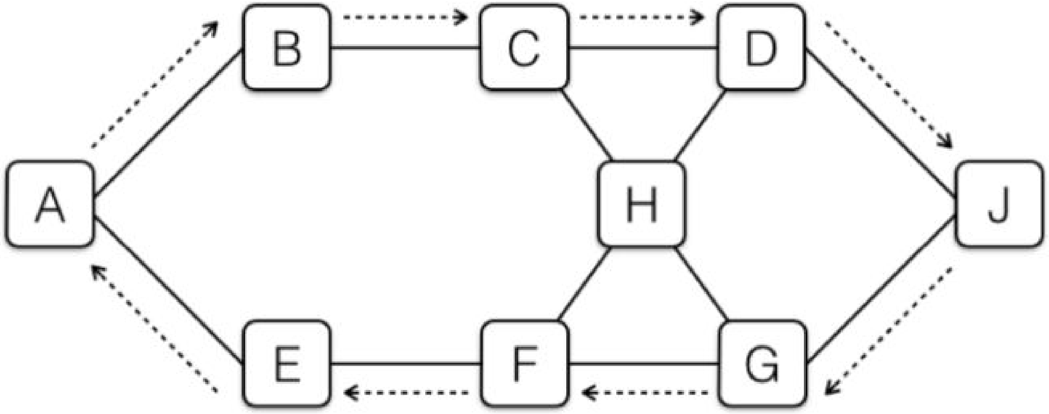
No, she's off-path and must blindly guess A's TCP seq #.

TCP Injection Practice Question



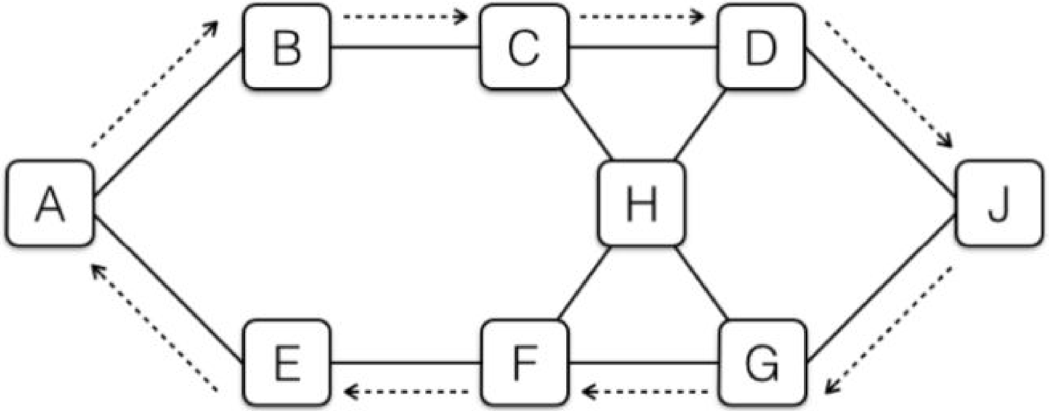
(d) Suppose that Mallory can eavesdrop on all packets that go through machine C (but cannot inject forged packets from C). Also Mallory can run software on machine F that lets her inject forged packets from F (but cannot eavesdrop on packets going through F). Can Mallory injected spoofed data into the TCP connection, so that it will be accepted by machine J as though it came from A? Why or why not?

TCP Injection Practice Question



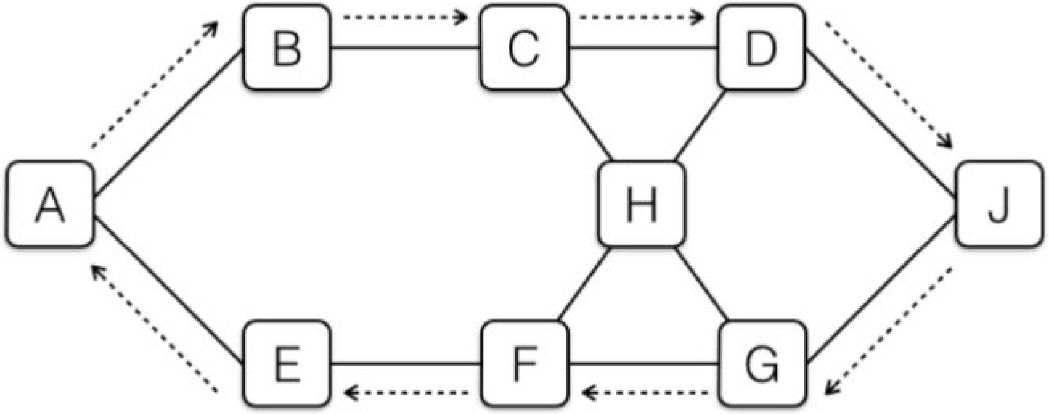
Yes, she's on-path and can see A's TCP seq # at C, and can inject spoofed data packets with correct seq #s from F to J.

TCP Injection Practice Question



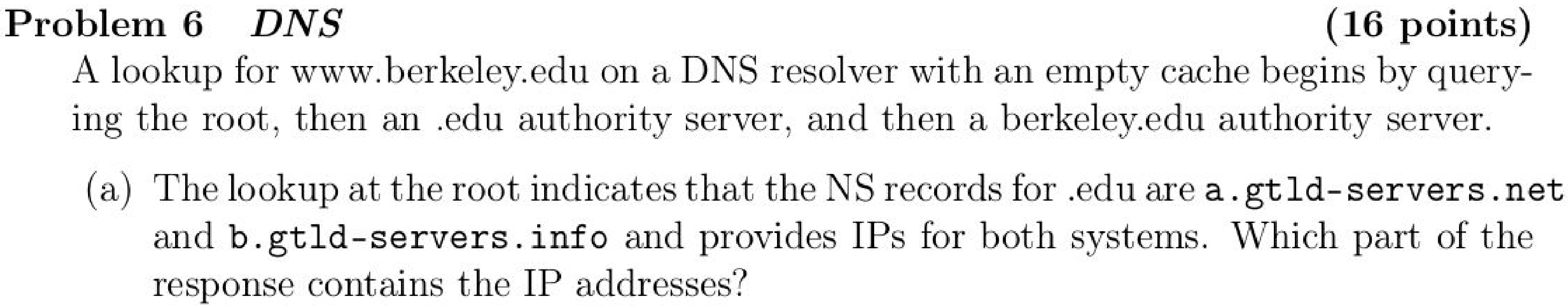
(e) Suppose that Mallory can eavesdrop on all packets that go through machine F (but cannot inject forged packets from F). Also Mallory can run software on machine C that lets her inject forged packets from C (but cannot eavesdrop on packets going through C). Can Mallory injected spoofed data into the TCP connection, so that it will be accepted by machine J as though it came from A? Why or why not?

TCP Injection Practice Question



Yes, b/c she’s on-path, and can still see seq #s in both directions at F (remember that the TCP acknowledgement field has the seq # of the receiver). So valid spoofed data packets can be injected at C sent to J.

DNS Practice Question (FA16 - Midterm 2)

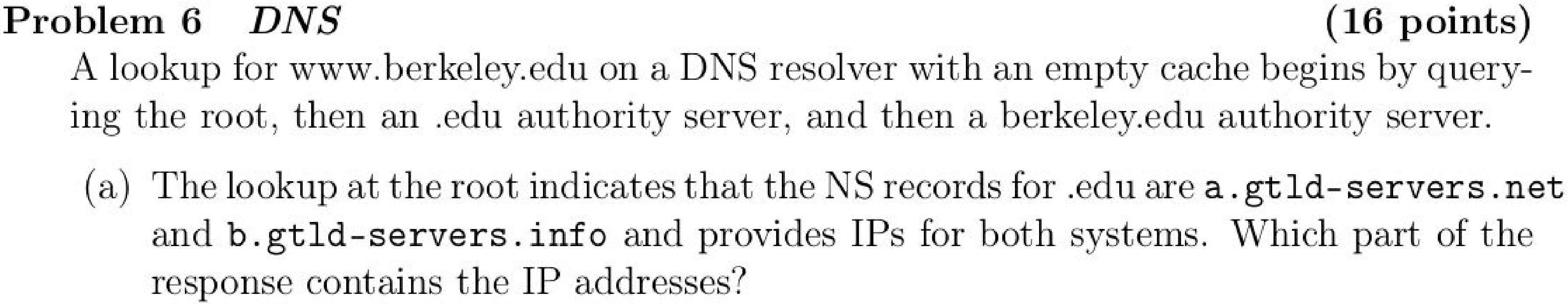


Problem 6 ***DNS*** **(16 points)**

A lookup for www.berkeley.edu on a DNS resolver with an empty cache begins by querying the root, then an .edu authority server, and then a berkeley.edu authority server.

(a) The lookup at the root indicates that the NS records for .edu are `a.gtld-servers.net` and `b.gtld-servers.info` and provides IPs for both systems. Which part of the response contains the IP addresses?

DNS Practice Question



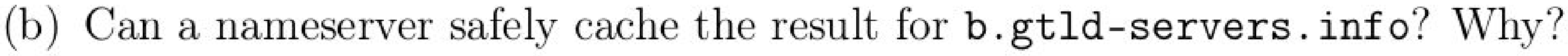
Problem 6 ***DNS*** **(16 points)**

A lookup for www.berkeley.edu on a DNS resolver with an empty cache begins by querying the root, then an .edu authority server, and then a berkeley.edu authority server.

(a) The lookup at the root indicates that the NS records for .edu are `a.gtld-servers.net` and `b.gtld-servers.info` and provides IPs for both systems. Which part of the response contains the IP addresses?

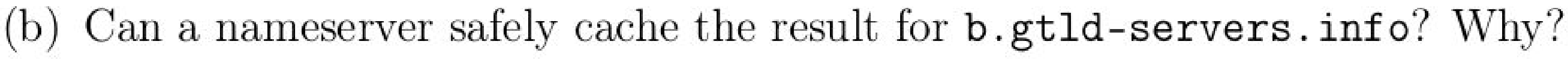
The additional records

DNS Practice Question



(b) Can a nameserver safely cache the result for `b.gtld-servers.info`? Why?

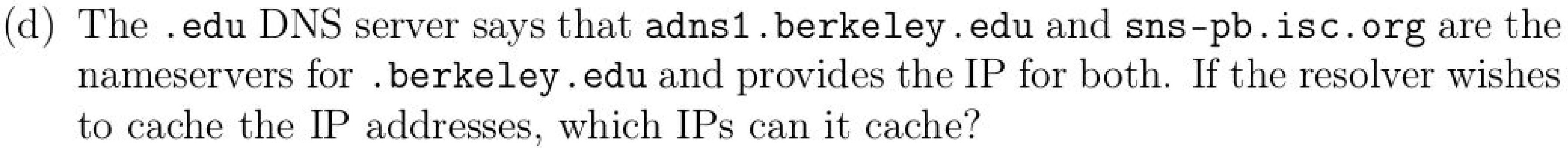
DNS Practice Question



(b) Can a nameserver safely cache the result for `b.gtld-servers.info`? Why?

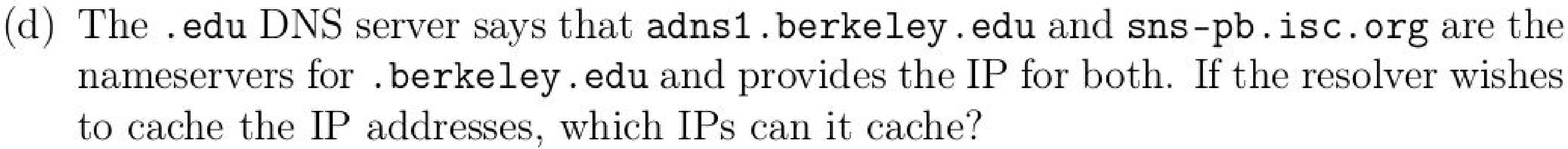
Yes, it’s in the root’s bailiwick.

DNS Practice Question



(d) The `.edu` DNS server says that `adns1.berkeley.edu` and `sns-pb.isc.org` are the nameservers for `.berkeley.edu` and provides the IP for both. If the resolver wishes to cache the IP addresses, which IPs can it cache?

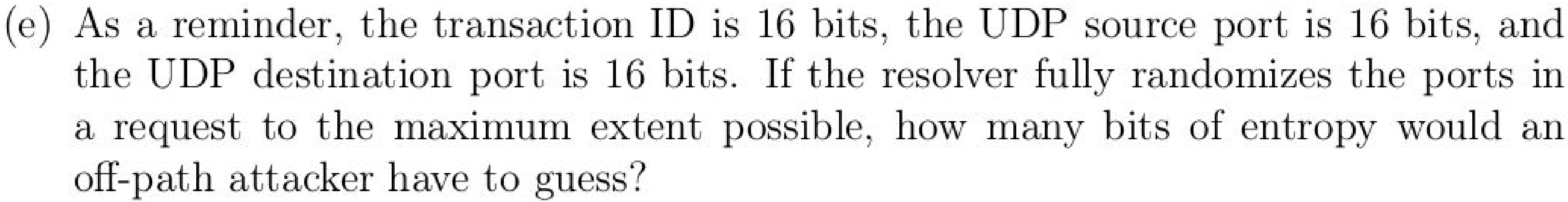
DNS Practice Question



(d) The `.edu` DNS server says that `adns1.berkeley.edu` and `sns-pb.isc.org` are the nameservers for `.berkeley.edu` and provides the IP for both. If the resolver wishes to cache the IP addresses, which IPs can it cache?

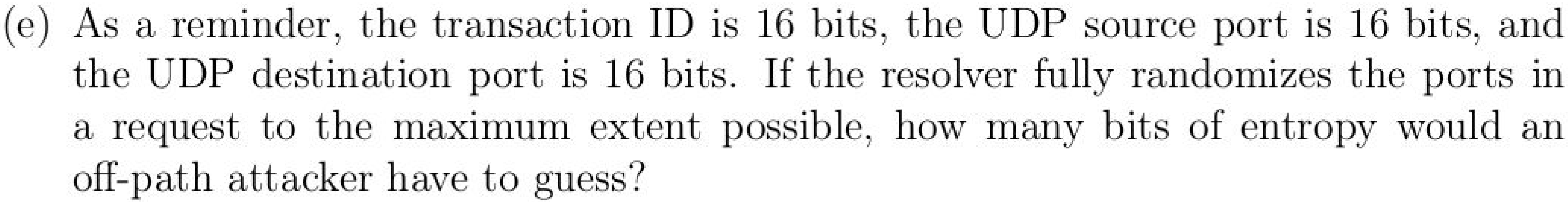
Only the IP of adns1.berkeley.edu is saved, since the other is not in the .edu nameserver’s bailiwick.

DNS Practice Question



(e) As a reminder, the transaction ID is 16 bits, the UDP source port is 16 bits, and the UDP destination port is 16 bits. If the resolver fully randomizes the ports in a request to the maximum extent possible, how many bits of entropy would an off-path attacker have to guess?

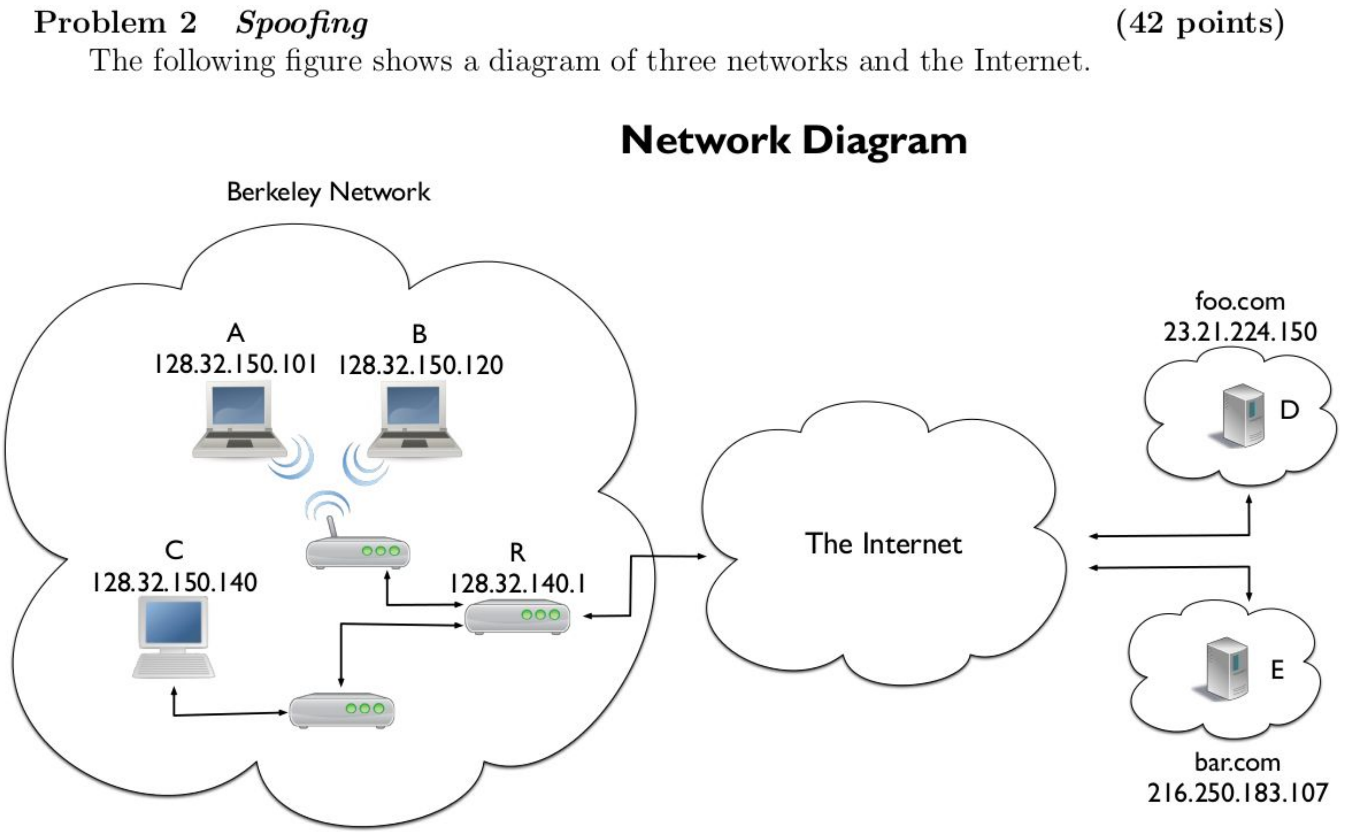
DNS Practice Question



(e) As a reminder, the transaction ID is 16 bits, the UDP source port is 16 bits, and the UDP destination port is 16 bits. If the resolver fully randomizes the ports in a request to the maximum extent possible, how many bits of entropy would an off-path attacker have to guess?

32 bits, since you don’t randomize the destination port.

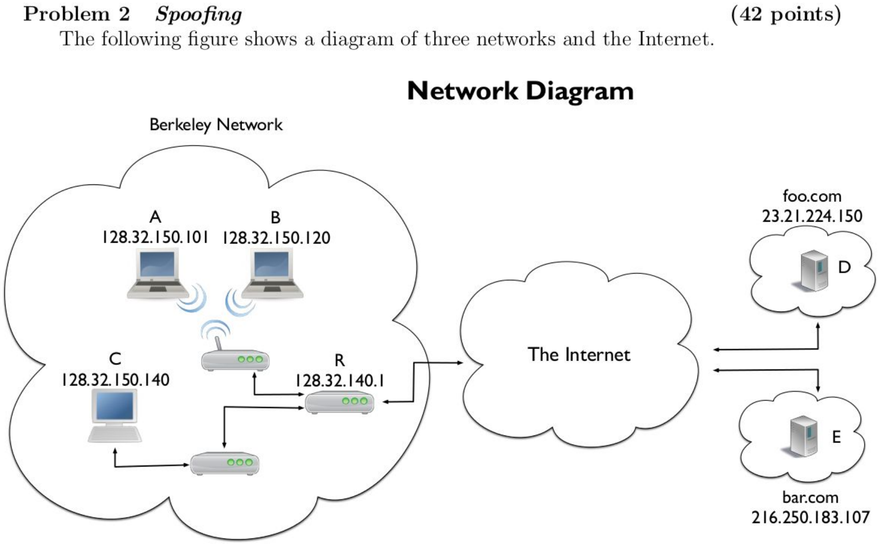
Spoofing (SP13 - Midterm 1)



Problem 2 ***Spoofing*** **(42 points)**

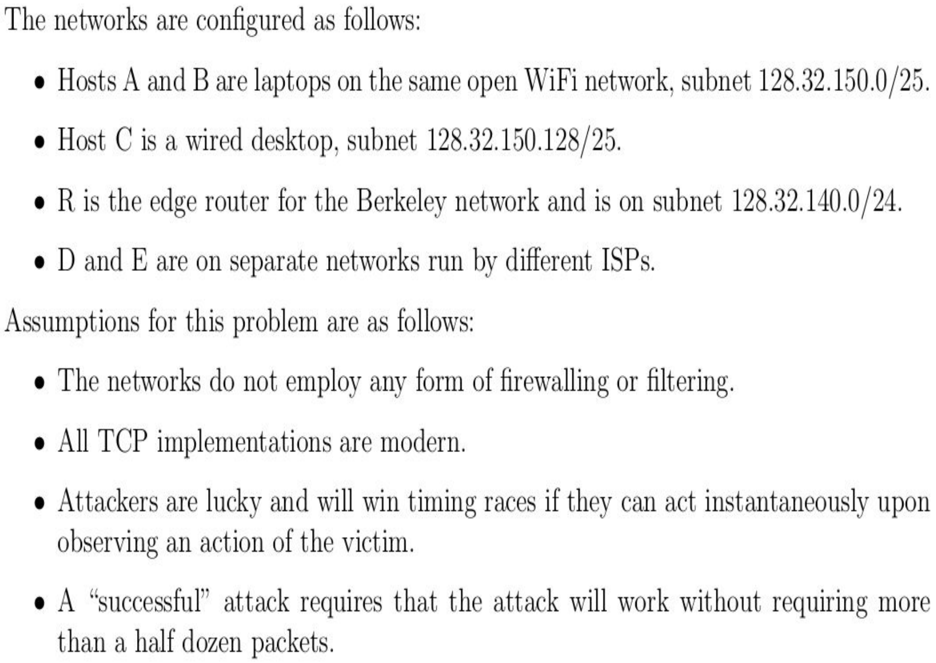
The following figure shows a diagram of three networks and the Internet.

Spoofing



Problem 2 ***Spoofing*** **(42 points)**

The following figure shows a diagram of three networks and the Internet.



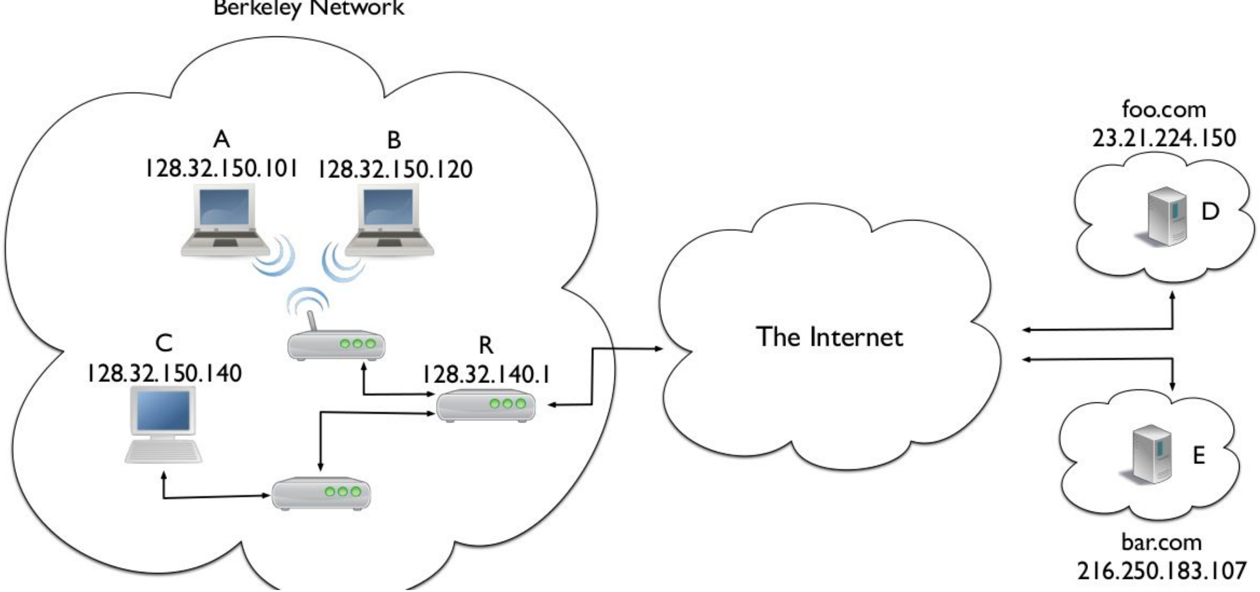
The networks are configured as follows:

- Hosts A and B are laptops on the same open WiFi network, subnet 128.32.150.0/25.
- Host C is a wired desktop, subnet 128.32.150.128/25.
- R is the edge router for the Berkeley network and is on subnet 128.32.140.0/24.
- D and E are on separate networks run by different ISPs.

Assumptions for this problem are as follows:

- The networks do not employ any form of firewalling or filtering.
- All TCP implementations are modern.
- Attackers are lucky and will win timing races if they can act instantaneously upon observing an action of the victim.
- A "successful" attack requires that the attack will work without requiring more than a half dozen packets.

Spoofing



(a) (12 points) Attacker has the ability to successfully spoof DHCP offers directed at the victim, such that the victim will accept the offer as genuine:

		Attackers					
		A	B	C	D	R	*None*
Victims	A	–					
	C			–			
	D				–		

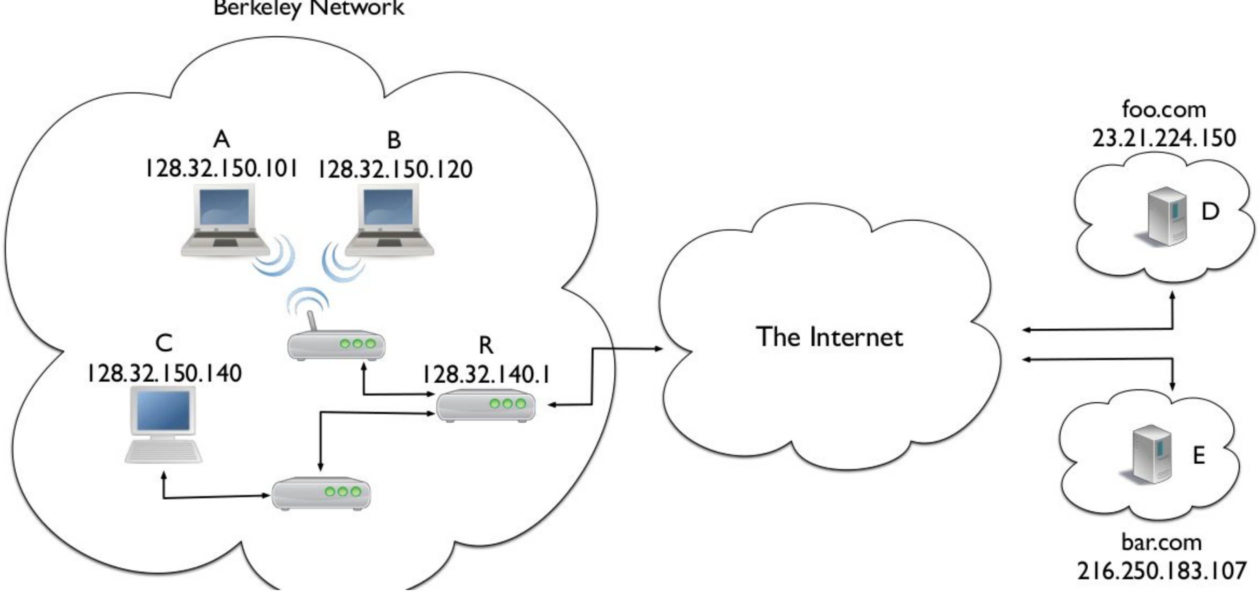
Spoofing

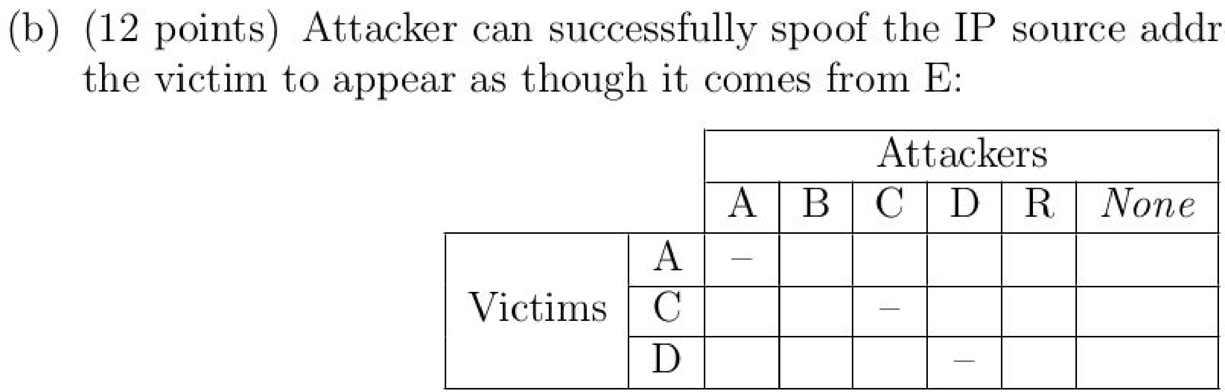
(a) (12 points) Attacker has the ability to successfully spoof DHCP offers directed at the victim, such that the victim will accept the offer as genuine:

		Attackers					
		A	B	C	D	R	*None*
Victims	A	–	X				
	C			–			X
	D				–		X

Solution: To successfully spoof a DHCP offer requires the ability to see the original DHCP request. This in turn requires an attacker connected to the same subnet as the victim. In the network considered in this problem, only A and B are connected to the same subnet.

Spoofing

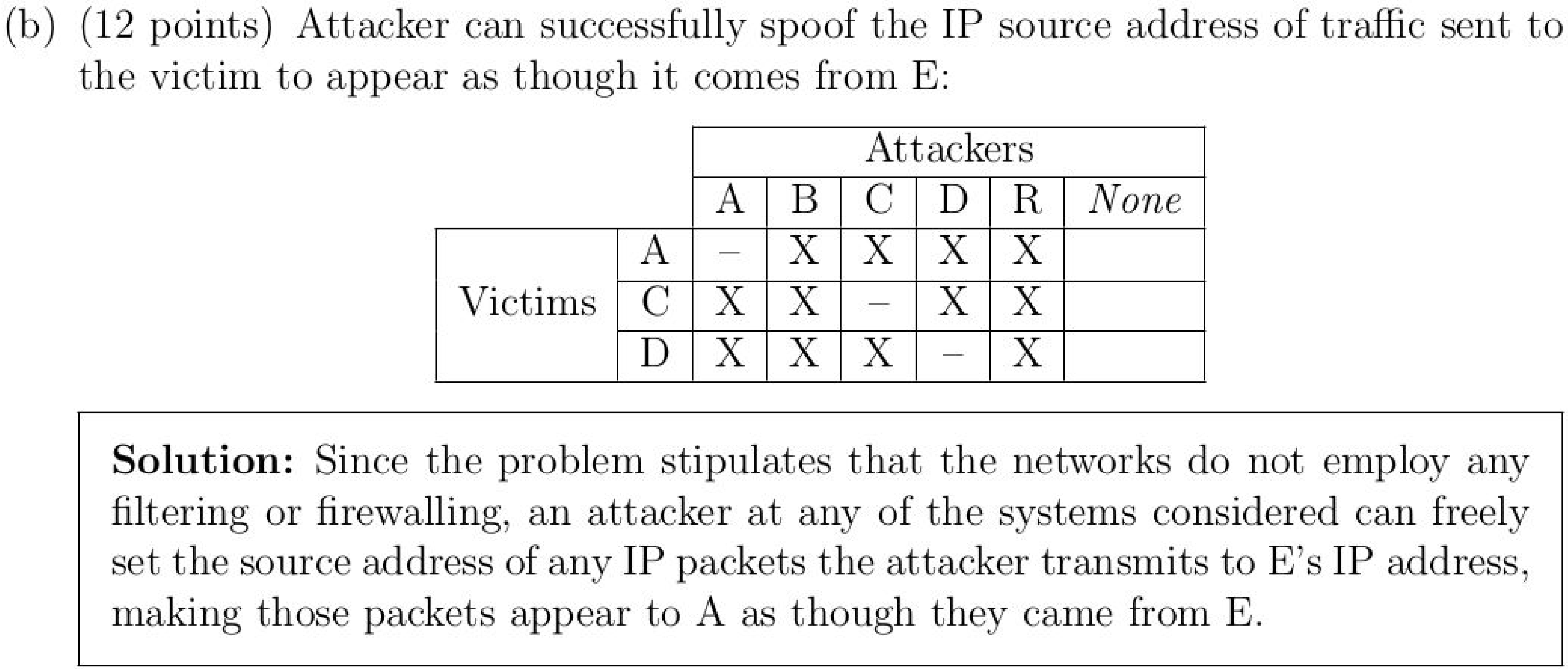




(b) (12 points) Attacker can successfully spoof the IP source address of traffic sent to the victim to appear as though it comes from E:

		Attackers					
		A	B	C	D	R	*None*
Victims	A	–					
	C			–			
	D				–		

Spoofing

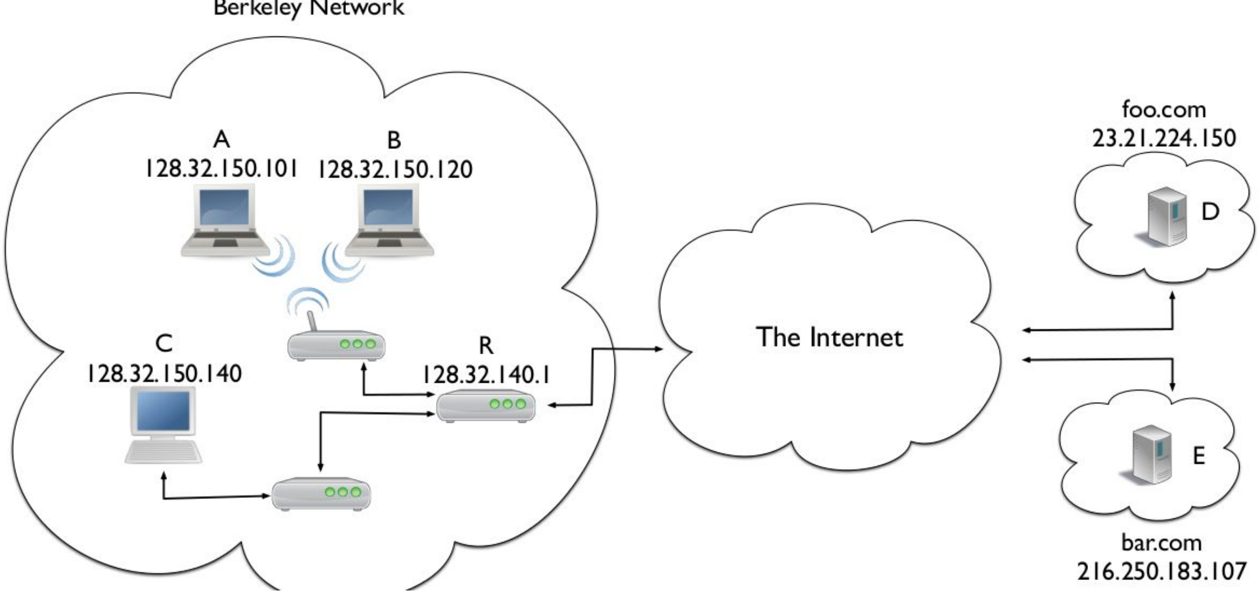


(b) (12 points) Attacker can successfully spoof the IP source address of traffic sent to the victim to appear as though it comes from E:

		Attackers					
		A	B	C	D	R	*None*
Victims	A	–	X	X	X	X	
	C	X	X	–	X	X	
	D	X	X	X	–	X	

Solution: Since the problem stipulates that the networks do not employ any filtering or firewalling, an attacker at any of the systems considered can freely set the source address of any IP packets the attacker transmits to E's IP address, making those packets appear to A as though they came from E.

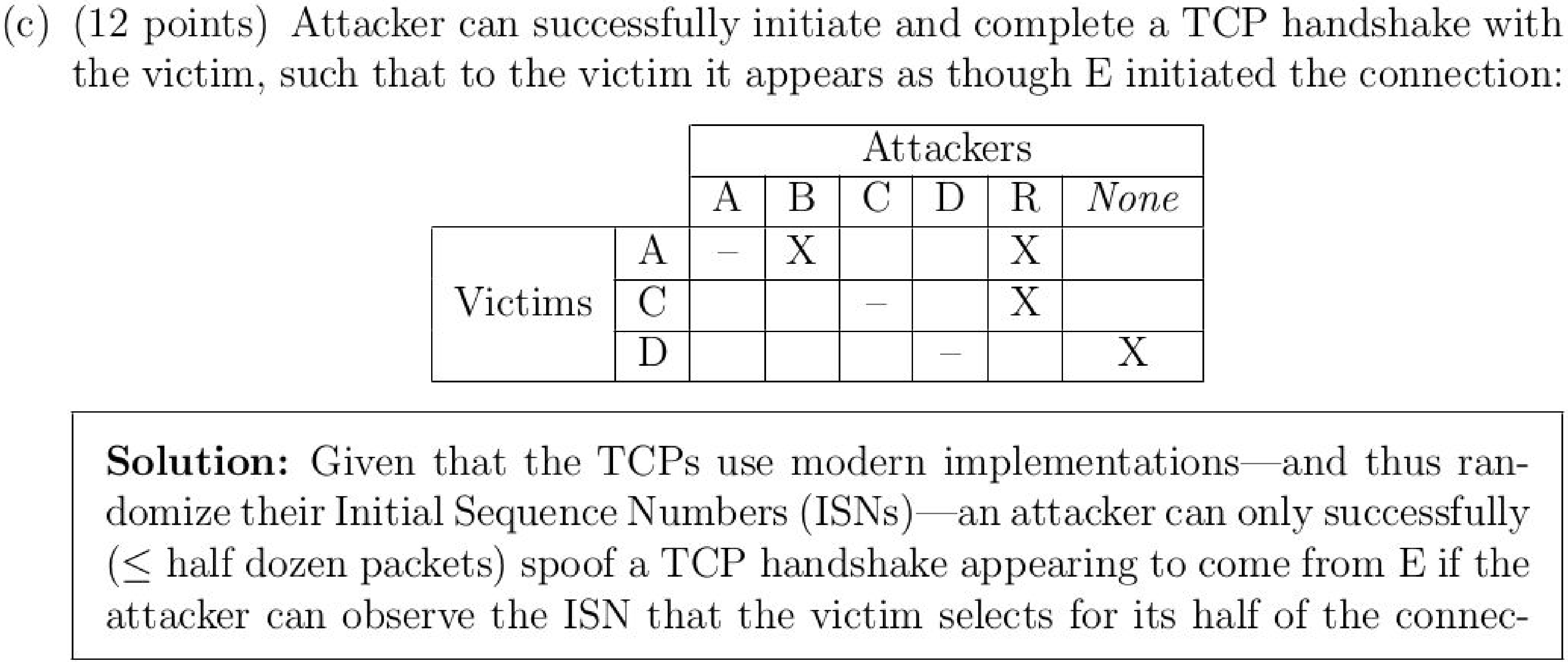
Spoofing



(c) (12 points) Attacker can successfully initiate and complete a TCP handshake with the victim, such that to the victim it appears as though E initiated the connection:

		Attackers					
		A	B	C	D	R	*None*
Victims	A	–					
	C			–			
	D				–		

Spoofing



(c) (12 points) Attacker can successfully initiate and complete a TCP handshake with the victim, such that to the victim it appears as though E initiated the connection:

		Attackers					
		A	B	C	D	R	*None*
Victims	A	–	X			X	
	C			–		X	
	D				–		X

Solution: Given that the TCPs use modern implementations—and thus randomize their Initial Sequence Numbers (ISNs)—an attacker can only successfully ($\leq$ half dozen packets) spoof a TCP handshake appearing to come from E if the attacker can observe the ISN that the victim selects for its half of the connec-