Raluca & Nick
Fall 2016

CS 161 Computer Security

Midterm 2

PRINT your name: ____________2.2in, ____________2.2in
(last) (first)

I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that any academic misconduct will be reported to the Center for Student Conduct, and may result in partial or complete loss of credit.

SIGN your name: ____________4in

PRINT your class account login: `cs161-`____________.5in and SID: ____________2in

Your TA's name: ____________4in

Your section time: ____________4in

Exam # for person sitting to your left: ____________1.2in

Exam # for person sitting to your right: ____________1.2in

You may consult one sheet of paper (double-sided) of notes. You may not consult other notes, textbooks, etc. Calculators, computers, and other electronic devices are not permitted.

You have 80 minutes. There are 7 questions, of varying credit (110 points total). The questions are of varying difficulty, so avoid spending too long on any one question. Parts of the exam will be graded automatically by scanning the **bubbles you fill in**, so please do your best to fill them in somewhat completely. Don't worry—if something goes wrong with the scanning, you'll have a chance to correct it during the regrade period.

If you have a question, raise your hand, and when an instructor motions to you, come to them to ask the question.

Do not turn this page until your instructor tells you to do so.

Question:	True/False	Shorts	The No Such Agency's Attack Tools	Applied Cryptography	TLS
Points:	20	12	24	15	12
Score:					

Problem 1 ***True/False*** **(20 points)**

Circle True or False. Do not justify your answer.

(a) True or False: Randomizing the DNS query identifier prevents an on-path attacker (sitting between the client and the DNS server) from spoofing DNS responses.

False

(b) True or False: One defense against the Kaminsky Attack is to randomize the destination port along with randomizing the identifier field.

False

(c) True or False: An off-path attacker can intercept and modify the DNS reply sent by the DNS server to the client.

Solution: False - an off-path attacker cannot see or intercept packets.

(d) True or False: In order to determine whether a given certificate is valid, it is sufficient to only verify the signature on the given certificate.

Solution: False - you need to check that the public key you are using is correct, either by checking the signatures all the way up the certificate chain, or by checking the public key against a root public key hardcoded in the browser.

(e) True or False: If an attacker learns the internal state of an HMAC-based pRNG (HMAC-DRBG) they can reconstruct previous outputs.

Solution: False - HMAC-DRBG, even if the attacker learns the internal state they can only predict the future, not reconstruct previous outputs.

(f) True or False: If an attacker learns the internal state of an HMAC-based pRNG (HMAC-DRBG) they can predict future outputs.

Solution: True - HMAC-DRBG, if the attacker learns the internal state they can predict the future.

(g) True or False: if an attacker connected to your local wireless network using WPA2-Enterprise (like AirBears2) they can directly observe all your network traffic?

Solution: False: WPA-Enterprise does a public key operation to generate the secret between the client and the base station.

(h) True or False: You click the "Forgot password" link on the website, and the web server sends you an email with your password in it (in plaintext). Assume that this communication is over SSL/TLS so an attacker cannot eavesdrop on it. True or False: The website stores only hashed passwords.

False. This indicates the website is storing un-hashed plaintext passwords (possibly encrypted, but this is irrelevant, as a website breach would presumably leak the decrypted passwords).

(i) True or False: As long as one uses long and randomly-generated passwords, it is safe to use the same password for all your online accounts.

False. Even if the password is very secure, if one of the websites does not properly store passwords (say, it keeps them in plaintext) and is breached, an attacker will learn the password and be able to use it on all the other sites you have accounts on.

(j) True or False: Consider that a server stores hashed passwords, each salted with a large and randomly chosen salt and stored along with the salt, instead of storing only hashed passwords (with no salt). This prevents an attacker who stole the salted hashed passwords database from mounting a dictionary attack.

False, it slows it down, but does not prevent it.

Problem 2 *Shorts* (12 points)

(a) An eavesdropper sees a target log into a WPA2-PSK secured network and sees the 4-way handshake. The eavesdropper sees

- The network's passphrase
- The access point's ANonce
- The client's SNonce

What else does she need to compute the pairwise transport key? (Write only the item(s) in your best answer; do not write a laundry list of tries as we will penalize that).

Solution: The client and AP's MAC addresses and the network's SSID

(b) The Free Birdseed #1: Wile-E-Coyote of ACME Inc is seeding a cryptographic pRNG. He doesn't know much about security, so he's not sure what would be a good way to come up with a seed. Which of these, used alone, would be sufficiently secure? Circle zero or more options, and briefly explain why unselected options are unsuitable.

1. The process ID of Mr Coyote's application
2. A cryptographic hash of the application binary
3. Wind speeds across the Bay Bridge as reported by ACME weather app (It's number four on the App Store, "but we're really hoping to bump that up with our upcoming redesign," according to the developer.)
4. Precise timing (including microseconds) and coordinates taken from a minute of hectic flailing of the user's mouse
5. The sum of two numbers modulo a large fixed prime p, each generated from a source of randomness, knowing that one of these sources is actually broken, but the other works as expected

Solution: 4 and 5 are suitable.

1 does not have enough entropy. It is also predictable, but not constant or entirely deterministic. It is not public, so an attacker cannot just *find out* the process ID, but they can easily brute force it by guessing.

2 is constant.

3 is public. As for predicting it, weather models may model large-scale patterns, but aren't precise enough. There is still a lot of entropy in this source, but again, it is publiclly posted, so an attacker can just look up the exact values.

We also accept that 4 is not suitable with the explanation that it is too painful for users.

(c) The Free Birdseed #2: Mr Coyote comes up with a more convoluted process instead. When his application starts up, it will:

1. Instantiate a PRNG with the user ID as the seed
2. Use the PRNG to establish an encrypted connection to the BIG-SEED server which returns a large random string S
3. Reseed the PRNG with S

Suppose an attacker can eavesdrop on the application's communications over the network, but cannot tamper with them. Describe how the attacker can predict future random numbers generated by Mr Coyote's application.

Solution: The attacker can decrypt and obtain S by guessing process ID.

Problem 3 ***The No Such Agency's Attack Tools*** **(24 points)**

The No Such Agency maintains numerous network presences, including off-path, on-path, and full man-in-the-middle positions, and all these positions have the capability of creating arbitrary packets and sending them into the network. But overall the NSA prefers using the least powerful attack for the job at hand, so they will prefer an off-path attack over an on-path attack, and an on-path attack over a man-in-the-middle attack, as it is easier to be an on-path attacker and easier still as an off-path attacker.

For each attack scenario, which type of attack does the NSA select and why can't the NSA use a less powerful attack? None could also be an option.

(a) The NSA can generate only a single query against a target DNS server they seek to cache poison, and they want to reliably cache poison the target.

Solution: The NSA has to use an on-path attack, because they need to see the DNS request to get the port/transaction info to generate a malicious reply.

(b) The NSA seeks to create a UDP request which appears to come from an arbitrary IP to the remote server. This request will compromise the remote server, and the NSA doesn't need to see the reply.

Solution: An off-path attack, since they don't need to see anything about the server.

(c) The NSA seeks to create a TCP connection which appears to be from an arbitrary IP to a remote server. This request will compromise the remote server, and the remote server uses the current time to generate the initial sequence number.

Solution: An off-path attack, since they can predict the initial sequence number.

(d) The NSA seeks to create a TCP connection which appears to be from an arbitrary IP to a remote server. This request will compromise the remote server, and the remote server uses a secure RNG to generate the initial sequence number.

Solution: An on-path attack, since the attacker can't predict the initial sequence number.

(e) The NSA seeks to inject content into an existing active TCP connection between the victim and a web server. The NSA knows this victim is very paranoid and records raw traffic and requires that the victim be unable to determine that the NSA modified this traffic.

Solution: A full man-in-the-middle, as an on-path attacker can't stop the legitimate reply from the server.

(f) The NSA seeks to inject content into a TLS web connection that uses RSA key exchange. The NSA has a copy of the server's key. The NSA knows the victim is not recording raw traffic and so can't detect additional replies from the server. The cryptography is otherwise secure.

Solution: An on-path attack, since that is sufficient to decrypt the TLS connection.

(g) The NSA seeks to inject content into a TLS web connection that uses DHE key exchange. The NSA has a copy of the server's key. The NSA knows the victim is not recording raw traffic and so can't detect additional replies from the server. The cryptography is otherwise secure.

Solution: A full man-in-the-middle, since an observer can't deduce the key from DHE.

(h) The NSA seeks to inject content into a TLS web connection to Google, where the client is using Chrome. The NSA has a forged certificate for the server signed by a different Certificate Authority than the one Google uses.

Solution: None. Chrome doesn't accept any CA, but only those specified as belonging to Google.

Problem 4 ***Applied Cryptography*** **(15 points)**

Mr Wile-E-Coyote needs to deploy a system to authenticate user passwords on a web server that does *not* use TLS to transmit the password from the user using an encrypted channel. He needs to develop alternate approaches where the web browser downloads JavaScript from the server to perform password calculations. In what follows, a passive attacker is an attacker that does not modify any of the traffic it sees.

(a) Mr Coyote's first implementation has the javascript which the web browser downloads from the server compute H(passwd) and send it to the server. Does this protocol prevent a passive observer from directly seeing the user's password? Why?

Yes, its hashed

(b) Does this protocol prevent a passive observer from logging in as the user? Why?

No. It effectively makes H(passwd) the actual password used to communicate with the server. A dictionary attack on the hash is entirely unnecessary.

(c) Mr Coyote's second implementation has the Javascript compute H(passwd) which is used as a seed for a pRNG. This pRNG then generates a private key that is used to encrypt a channel to the server. Does this protocol prevent a passive observer from logging in as the user? Why?

Yes. This creates a shared secret that requires knowing either passwd or H(passwd) which a passive observer can't monitor.

(d) Is this implementation secure against a man-in-the-middle? Why or why not?

No. The adversary could replace the JavaScript with a version that steals the user's password.

Note that there is no key exchange here that an attacker could pose as the server in, and that guessing the password is not a MITM attack (anybody can do that).

Note further that changing the JavaScript to a version which uses an insecure hash function will not work (though it was given full points anyway)—as the server will still use the original hash function, so a MITM would be able to see client->server communications but not server->client, and further, the server would fail to decrypt the client's messages.

(e) Mr Coyote, tired of dealing with poor solutions, decides to switch his web server to TLS with a certificate purchased from SureSign. He is worried about other certificate authorities, notably Verislime. Does certificate pinning give Mr Coyote protection against an attacker who can compromise a different certificate authority?

Yes. Certificate pinning allows specifing a specific CA as responsible for a domain

Problem 5 ***TLS*** **(12 points)**

An attacker is trying to attack the company WoSlime and its users. Assume that users always visit WoSlime's website with an HTTPS connection, using Diffie-Hellman and AES encryption. (You may assume that WoSlime does not use certificate pinning) For each of the following attack scenarios, circle all of the options that an attacker could achieve in that attack scenario, and give no further explanation.

(a) If the attacker obtains the private key of a certificate authority trusted by users of WoSlime, the attacker could:

1. [✓ **impersonate the WoSlime web server to a user**]
2. discover some of the plaintext of data sent during a past connection between a user and WoSlime's website
3. discover all of the plaintext of data sent during a past connection between a user and WoSlime's website
4. replay data that a user previously sent to the WoSlime server over a prior HTTPS connection
5. none of the above

(b) If the attacker obtains a copy of WoSlime's certificate, the attacker could:

1. impersonate the WoSlime web server to a user
2. discover some of the plaintext of data sent during a past connection between a user and WoSlime's website
3. discover all of the plaintext of data sent during a past connection between a user and WoSlime's website
4. replay data that a user previously sent to the WoSlime server over a prior HTTPS connection
5. [✓ **none of the above**]

(c) If the attacker is a man in the middle on a HTTPS connection between a user and WoSlime's website, the attacker could:

1. impersonate the WoSlime web server to this user
2. discover some of the plaintext of data sent during *this* connection
3. discover all of the plaintext of data sent during *this* connection
4. [✓ **discover the amount of plaintext data sent during *this* connection**]
5. discover all of the plaintext of data sent during a *past* connection between a user and WoSlime's website
6. replay data that a user previously sent to the WoSlime server over a prior HTTPS connection

7. none of the above

(d) Suppose the attacker obtains the private key that was used by WoSlime's server during a past connection between a victim and WoSlime's server, but not the current private key. Also, assume that the certificate corresponding to the old private key has been revoked and is no longer valid. This attacker could:

1. impersonate the WoSlime web server to this user
2. discover all of the plaintext of data sent during a *current* connection (one where the current private key is used) between a user and WoSlime's website
3. **(due to an ambiguity this was also accepted**

   [✓ discover all of the plaintext of data sent during a *past* connection (one where the old private key was used) between a user and WoSlime's website]
4. [✓ **none of the above**]

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

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

Yes. It is in-bailywick (aka root is authoritative for) .info

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

Surprisingly, the additional records!

(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 for `adns1.berkeley.edu`, since `sns-pb.isc.org` is not in bailywick

(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?

32b, since you can't randomize the destination port. (Slightly less also acceptable

(f) If the resolver randomly capitalizes the request (e.g. `wwW.BerKeLEy.eDU`), what is the minimum additional entropy added to requests for berkeley domains?

11 bits, since you can't randomly capitalize a name like 1.berkeley.edu etc...

Problem 7 ***A simpler "TLS" for secure Messaging*** **(11 points)**

Consider that Alice and Bob want to communicate securely, but there is an *on-path* attacker who aims to modify or read their conversations. They each have a public-key pair (PK_A, SK_A) for Alice and (PK_B, SK_B) for Bob. Unfortunately, Alice does not know Bob's public key, Bob does not know Alice's public key, there is no certificate authority here and none of them have any certificate on their keys.

Fortunately, when Alice and Bob setup the channel for communication, the attacker was not ready to mount an active attack and he/she can only observe the packets between the two. Namely, the attacker is an eavesdropper, but not a man-in-the-middle attacker. After the setup phase, the attacker starts figuring out how to modify the traffic sent between the two of them, and acts as a full man-in-the-middle attacker.

How can Alice and Bob communicate securely? Answer the questions below by simply presenting each message sent between Alice and Bob with no explanation. Use the notation Enc() for encryption (either symmetric or asymmetric), Sign() for signing, Verify() for verifying, and MAC() for mac-ing (but you don't have to use all these algorithms). For each of these algorithms specify concretely the key to be used and the plaintext on which to use it.

(a) What messages do they send each other in the setup phase?

Alice sends a message to Bob saying "please give me your PK". She receives PK_B untampered by the attacker who is passive during the setup phase. Alice then chooses symmetric keys k_1 and k_2 and sends $Enc(PK_B, k_1)$ and $Enc(PK_B, k_2)$. Bob obtains k_1 and k_2.

(b) After the setup phase, Alice wants to send messages $m_1, \ldots, m_n$ to Bob. Write down what she sends. Make sure that your solution prevents against replay attacks, namely, the attacker cannot send an old message to Bob instead of a new message and convince him it is a new message. You only need to worry about the messages sent from Alice to Bob here.

Set counter = 0. For every message send, $C = Enc(k_1, m||counter++)$, $T = MAC(k_2, C)$, where || is concatenation. Send (C,T).