CS 161 Computer Security
Spring 2010 Paxson/Wagner
MT2

PRINT your name: _____, _____
(last) (first)

SIGN your name: _____

PRINT your class account login: cs161-_____

Your TA's name: _____

Your section time: _____

Name of the person sitting to your left: _____

Name of the person sitting to your right: _____

You may consult one sheet of paper (double-sided) of notes written for this midterm, plus the sheet of paper you brought to midterm 1. You may not consult other notes, textbooks, etc. Calculators and computers are not permitted. Please write your answers in the spaces provided in the test. We will not grade anything on the back of an exam page unless we are clearly told on the front of the page to look there.

You have 80 minutes. There are 6 questions, of varying credit (100 points total). The questions are of varying difficulty, so avoid spending too long on any one question.

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

Problem 1	
Problem 2	
Problem 3	
Problem 4	
Problem 5	
Problem 6	
Total	

Problem 1. [True or false] (14 points)

Circle TRUE or FALSE. Do not justify your answers on this problem.

(a) TRUE or FALSE: If Alice has a message to send to Bob and she wants to encrypt the message using asymmetric cryptography so that no one other than Bob can read it, she does so by using Bob's public key.

(b) TRUE or FALSE: SSL and TLS provide essentially the same end-to-end security properties.

(c) TRUE or FALSE: Properly used, a MAC provides both confidentiality and integrity.

(d) TRUE or FALSE: DNSSEC uses SSL between different name servers to certify that the results of DNS queries match those that the name servers are authorized to provide.

(e) TRUE or FALSE: In the United States, if a company posts a privacy policy on their web site and fails to comply with it, they can be prosecuted for false advertising.

(f) TRUE or FALSE: An attraction of public key cryptography is that, if implemented properly, the algorithms generally run much faster than those for symmetric key cryptography.

(g) TRUE or FALSE: Memory protection, as found in a typical operating system, prevents malicious code running in kernel mode from writing to application-owned pages.

Problem 2. [Multiple choice] (18 points)

For parts (a), (b), and (c), circle **all** correct choices.

(a) TLS uses the following cryptographic techniques:

(i) Asymmetric-key cryptography.

(ii) Symmetric-key cryptography.

(iii) Cryptographic hash functions.

(iv) PKI certificates.

(v) Nonces.

(vi) None of the above.

(b) Which of the following properties must a cryptographic hash function provide?

(i) Key revocation.

(ii) Collision resistance.

(iii) A deterministic mapping from input to output.

(iv) One-to-one mapping of input to output.

(v) Difficulty of finding an input that matches a given hash.

(vi) None of the above.

(c) What risks arise when using the same key to encrypt both directions of a communication channel, that aren't present if using different keys for the different directions?

(i) Message tampering by flipping bits in the ciphertext.

(ii) Reflection attacks.

(iii) Hash collisions.

(iv) Eavesdropping attacks.

(v) Denial-of-service.

(vi) None of the above.

(d) For this part, circle only the **best** choice.

As we saw in class, WEP is vulnerable to active attacks that allow an active attacker to flip bits in the ciphertext and thereby cause unauthorized modifications to the message received by the recipient. What would be the best defense against this kind of attack?

(i) Use a different key for each direction and for each wireless device.

(ii) Protect the ciphertext using a MAC.

(iii) Encrypt using AES in Cipher Block Chaining (CBC) mode.

(iv) Encrypt using AES in Electronic Code Book (ECB) mode.

(v) Prepend a random 32-bit nonce to the packet before applying the CRC and encrypting it.

Problem 3. [Terminology] (14 points)

For each of the numbered concepts given below, list the term that best applies:

Access control list	El Gamal encryption	Private key
AES	Entropy	Public key
Asymmetry cryptography	IV	Revocation list
Certificate	Kirchoff's principle	RSA
Certificate authority	Known plaintext attack	Sandboxing
Chosen plaintext attack	MAC	Setuid
Counter mode	Message integrity	SHA256
Cryptographic hash	Nonce	SYN cookies
Dictionary attack	One-time pad	Trapdoor one-way function
Diffie-Hellman exchange	PKI	Web-of-trust

Not all terms are used.

1. The security goal of ensuring that a communication arrives at the recipient in a form identical to what the sender transmitted.
2. A widely used, standardized symmetric key encryption algorithm.
3. A way of checking whether the private key matching the public key in a certificate has been compromised and so the certificate should no longer be accepted.
4. A symmetric-key algorithm for ensuring that a message has not been tampered with.
5. The amount of uncertainty that an attacker faces when trying to guess an unseen value.
6. An approach by which users can build up a degree of confidence in a public key's validity without requiring a trusted root of authority.
7. An algorithm for digitally signing data with a private key such that anyone with possession of the corresponding public key can verify the signature.
8. A signed statement by a trusted authority that a given public key indeed belongs to a given party.
9. A value used in symmetric key cryptography to ensure that a new session that transmits the same text as a previous session does not result in identical ciphertext.
10. A way of constructing a stream cipher, given a block cipher.
11. The notion that the security of a well-designed cryptography algorithm should not rely upon the secrecy of the algorithm itself but only on the secret keys it uses.
12. A widely used, standardized cryptographic hash function.
13. A Unix operating system mechanism that enables a program to execute with the privileges of a different user identity rather than the identity of the user who invoked the progam.
14. A trusted third party who provides a way for one party to learn the public key of another party. Web browsers have a list of these trusted third parties, to support communication using HTTPS.

Problem 4. [Cryptography] (15 points)

SuperMail is a company designing a secure email system for protecting emails using cryptography. They have hired you to advise them on the best way to use cryptographic algorithms for this purpose. Their system generates two public/private keypairs for each user of the system (one keypair for signing, one for encryption) and provides a way for each user to securely learn the public keys of all of their contacts. In the following, you can assume that digital signatures are computed using RSA and public-key encryption is performed using El Gamal.

(a) SuperMail wants every email to be authenticated and protected from modification or tampering while it is transit from the sender to the receiver. Suppose Alice is sending an email M to Bob. Given SuperMail's design constraints, which of the following options would be a secure way to protect the authenticity and integrity of her email? Circle **all** secure choices.

(i) Alice's software should encrypt M under Bob's public key. In other words, Alice's software should send $E_{K_B}(M)$ to Bob.

(ii) Alice's software should send M along with a digital signature on M using Alice's private key. In other words, Alice should send $M, \text{Sign}_{K_A^{-1}}(M)$.

(iii) Alice's software should choose a new symmetric key k for this email, send an encryption of k under Bob's public key, and also send an encryption of M under k using a stream cipher such as RC4. In other words, Alice should send $E_{K_B}(k), M \oplus \text{RC4}(k)$.

(iv) Alice's software should choose a new symmetric key k for this email, send an encryption of k under Bob's public key, and also send an encryption of M under k using AES in CBC mode. In other words, Alice should send $E_{K_B}(k), \text{AES-CBC-Encrypt}_k(M)$.

(v) Alice's software should choose a new symmetric key k for this email. Then it should send four pieces of information: the message M, a MAC on M under the key k, an encryption of k under Bob's public key, and a digital signature on k using Alice's private key. In other words, Alice should send $M, \text{MAC}_k(M), E_{K_B}(k), \text{Sign}_{K_A^{-1}}(k)$.

(b) SuperMail wants to adds a new feature for secure transmission of confidential email. They have in mind that the sender should be able to mark an email to indicate that it is confidential. Confidential emails should be protected from eavesdropping, interception, modification, or tampering while they are in transit from the sender to the receiver. SuperMail wants to protect the authenticity, integrity, and confidentiality of confidential emails.

Let M be a confidential email that Alice wants to send to Bob, K_B be Bob's encryption public key, and K_A^{-1} be Alice's private key for signing. Which of the following options would be the best choice for protecting confidential emails? Circle only the **best** choice.

(i) Send $E_{K_B}(M), \text{Sign}_{K_A^{-1}}(K_B)$.

(ii) Send $E_{K_B}(M), \text{Sign}_{K_A^{-1}}(M)$.

(iii) Send $E_{K_B}(M), \text{Sign}_{K_A^{-1}}(E_{K_B}(M))$.

(iv) Send $E_{K_B}(k), \text{Sign}_{K_A^{-1}}(E_{K_B}(k)), E_k(M)$ where k is a new symmetric key chosen for this email and E_k represents encryption under k with a symmetric-key encryption algorithm.

(v) Send $E_{K_B}(k), \text{Sign}_{K_A^{-1}}(E_{K_B}(k)), E_k(M), \text{MAC}_k(M)$ where k is a new symmetric key chosen for this email, E_k represents encryption under k with a symmetric-key encryption algorithm, and MAC_k represents a message authentication code (MAC) using key k.

Problem 5. [Local system security and privacy] (21 points)

This problem concerns safeguarding your privacy when using your desktop computer, in the presence of the desktop system having potential security problems. For each question, provide a short answer.

(a) To maintain your privacy, before surfing to `www.twitter.com` you always instruct your browser to delete all of your cookies. The browser's documentation states that cookies are stored in the file `~/.mycookies.txt`, and you know from past experience that indeed that's where they are kept.

Suppose you are concerned that your browser has malicious code running within it, though you are confident that your operating system has not been compromised. You type `www.twitter.com` into your browser's address bar to take you to the Twitter site. Are there steps you could take (which could involve additional effort on your part) to check whether your browser sent any information to `www.twitter.com` via cookies as part of that request? Circle yes or no, then briefly explain.

(i) Yes. (ii) No.

Justification:

(b) Now consider a slightly modified situation where you are confident that your browser has not been tampered with, but you are concerned that your operating system may have been compromised.

You type `https://secrets.cs.berkeley.edu` into the browser's address bar, and your browser establishes a TLS connection to `https://secrets.cs.berkeley.edu`. That web server responds with a Web form for you to type in your username and password, and your browser sends back your answers via TLS.

Can malware running inside the operating system extract your username and password? Circle yes or no, then briefly explain.

(i) Yes. (ii) No.

Justification:

(c) Suppose now that you are confident that both your browser and your operating system have not been tampered with, but you believe a user-level process that runs as user `daemon` has been infected and is running malware. Permissions on your system allow user `daemon` to read your files (which includes the browser executable) but not to write them. You are confident that your OS correctly implements these permissions and provides process-level isolation and memory protection. However, user `daemon` is allowed by the operating system to "sniff" packets received or sent by your system's network interface card. Assume that your browser is free of bugs.

You again make a TLS connection to `https://secrets.cs.berkeley.edu`. Can the malware running as user `daemon` extract your username and password? Explain how, or why not. Circle yes or no, then briefly explain.

(i) Yes. (ii) No.

Justification:

Problem 6. [An alternate authentication scheme] (18 points)

The startup company Gingerbread, Inc. is marketing a new authentication scheme to several banks that offer online banking services. In Gingerbread's sales presentation, they suggest that banks can eliminate passwords and all existing authentication schemes and replace them with the Gingerbread web authentication scheme.

Here's the secret sauce behind their scheme. The Gingerbread employees have read the HTTP specification carefully. They discovered that when a web server sends a cookie to your web browser, there are two optional flags that the web server can set:

- If the *persistent* flag is set, the browser will store the cookie forever on persistent storage (e.g., the filesystem). Even if the user closes the browser, reboots their machine, and starts a new browser instance, the browser still remembers the cookie.
- If the *secure* flag is set, the browser will only send the cookie back over a SSL (HTTPS) connection. The browser will never send the cookie back over an unencrypted (HTTP) connection. This ensures that eavesdroppers will never be able to learn the value of a cookie set with the secure flag.

These two flags do not change any other aspect of how browsers handle cookies. In the Gingerbread scheme, the bank's web site will use SSL (HTTPS) for all operations; the bank's web server does not respond to unencrypted (HTTP) connections. When the user U signs up for online banking, the web server picks a new random 128-bit value A_U (called the *authenticator*), sets a secure persistent cookie on the user's browser containing the 128-bit authenticator A_U, and remembers the association (N_U, A_U) between the user's account number N_U and the 128-bit authenticator A_U for that user. The bank also asks the user to bookmark the bank's secure web site (at a HTTPS link), and instructs the user to use that bookmark when they want to access their account in the future. When the user visits the bank's website again in the future, the user's browser will send the secure persistent cookie containing the user's authenticator A_U to the bank's web server. The bank's web server can use the authenticator A_U to identify and authenticate the user U, and provide the user access to their bank account.

You are employed at a local bank, BankOBytes. Your bank manager is trying to decide whether to adopt Gingerbread's scheme on the bank web site, `www.bankobytes.com`. Your manager shares with you some of the things he's heard about Gingerbread through the rumor mill and would like to know which of these statements are accurate. For each statement listed below, circle either "ACCURATE" or "INACCURATE" according to whether the statement is an accurate characterization of Gingerbread's scheme.

(a) ACCURATE or INACCURATE: The Gingerbread scheme provides better security against phishing than password-based authentication: in the Gingerbread scheme, since users don't know their own secret authenticator A_U, users can't be easily fooled into typing their authenticator into a phishing web site.

(b) ACCURATE or INACCURATE: The Gingerbread scheme has a serious security flaw: since every web site can read all cookies stored in the browser, if the user visits a malicious third-party web site, the malicious web site could learn the user's authenticator A_U and then impersonate the user and access the user's bank account.

(c) ACCURATE or INACCURATE: The Gingerbread scheme has a serious security flaw: it can be easily broken by an attacker who simply tries to connect to the bank's web server many times, trying a different guess at A_U each time.

(continued on next page)

(d) ACCURATE or INACCURATE: One shortcoming of the Gingerbread scheme is that if the user leaves their computer logged in, then others who have physical access to the user's computer (e.g., the user's family members or the user's roommates) can impersonate the user and obtain access to the user's bank account.

(e) ACCURATE or INACCURATE: One feature of the Gingerbread scheme is that if the user wants to visit their bank website, but forgets to use the bookmark and types in `http://www.bankobytes.com` into the browser address bar, then this mistake will not compromise the user's security. From the standpoint of security, this is a good human factors engineering, because it means that this kind of user error does not cause a serious security breach.

(f) ACCURATE or INACCURATE: A potential security risk in the Gingerbread scheme is that, if Gingerbread's scheme is widely adopted by many banks, attackers might start trying to attack the process by which users initially sign up for online banking, instead of trying to steal the user's password.

(g) ACCURATE or INACCURATE: A potential privacy risk in the Gingerbread scheme is that, if Gingerbread's scheme is widely adopted by many banks, advertising networks could use the cookie set by the bank to track users as they browse third-party web sites, without the consent of the user or the user's bank.

(h) ACCURATE or INACCURATE: The only benefit of using HTTPS connections (instead of HTTP) in the Gingerbread scheme is ensuring that the user's browser is talking to the correct bank and not an imposter.

(i) ACCURATE or INACCURATE: If the bank preferred to minimize the amount of data that needs to be stored on the bank web server, there is an alternative that is also secure but needs less state on the bank web server: instead of making A_U a random 128-bit value, the bank could form A_U as the concatenation of the user's account number N_U and a MAC on the account number under some key k stored on the bank web server. In other words, in the alternate scheme, the bank web server stores a single key k (never revealed to anyone and the same for all users), and when a user signs up for online banking, the bank sets a cookie containing $A_U = (N_U, \text{MAC}_k(N_U))$ on the user's browser. This alternative is also secure and avoids the need to store the association (N_U, A_U) on the bank server.