

1	/10
2	/32
3	/36
4	/14
5	/8
Total	/100

- Do not begin the exam until you are told to do so.
- You have 50 minutes to complete the exam.
- There are 9 single-sided pages; please sign your name on *all* of them.

1. (10 points) True or False? Circle the appropriate answer.

- a. T F Block ciphers are typically less computationally expensive than public-key cryptographic schemes.
- b. T F The DES cryptosystem has been shown to be uncrackable except by brute force attacks.
- c. T F Digital signatures require the property of *nonrepudiation*, which says that a principal should not be able to spoof another principal's signature.
- d. T F The principle of complete mediation suggests that, by default, users should not be granted administrative privileges.
- e. T F Kerberos is an example of an arbitrated protocol.
- f. T F Encryption can help protect the *integrity* of data.
- g. T F It is possible to prove to someone that you know a secret without revealing *any* information to them about the secret.
- h. T F Good protocol design suggests that message formats be kept as uniform as possible to simplify the end hosts and reduce the trusted computing base.
- i. T F Cracking a message encrypted under two independently chosen keys is more difficult than cracking a message encrypted under one key.
- j. T F For shared-key protocols that employ a trusted third party (like Needham-Schroeder and Kerberos), it is necessary to distribute $O(n^2)$ keys before any session keys may be generated. (Here, n is the number of principals.)

2. Short answer

- a. (8 points) Should system administrators force users to change their passwords frequently (say, once per month)? Give one argument in favor and one argument against doing so—back up each answer by appealing to the principles of secure system design.

In favor: Security as process suggests that changing passwords proactively will reduce the window of vulnerability if passwords are changed frequently, as long as good passwords are chosen.

Against: Simplicity of design suggests that changing passwords frequently, which is more complicated and irritating to users, is undesirable—there is more work necessary to ensure that people choose good passwords. Also, people will have more trouble remembering passwords, so they will cycle through them or write them down.

- b. (8 points) Suppose that A has a very short secret s (e.g. a single bit or even a Social Security number), and she wishes to send B a message m that will not reveal s but that can be later used verify that A did know s . Explain why sending the MD5 hash ($m = MD5(s)$) or encrypting s with A 's public RSA key $m = K_A\{s\}$ would not be secure choices. Briefly, suggest a better way of creating m that doesn't involve exchanging multiple messages between A and B .

B can do a brute force search to find the value of s . A better solution is to add some salt (as with Unix passwords) to make it harder for B to do the search. It's also a good idea to add a string known to B so that A has a harder time finding a hash collision.

- c. (8 points) Recall that the five primary attacks against network protocols are: Replay, Interleaving, Reflection, Chosen Text, and Forced Delays. Pick *two* of these attacks, (briefly) describe them and give a typical countermeasure for each.
- Replay: use of challenge-response techniques embed target identity in response.
Interleaving link messages in a session with chained nonces.
Reflection: embed identifier of target party in challenge response use asymmetric message formats use asymmetric keys.
Chosen text: embed self-chosen random numbers ("confounders") in responses use zero knowledge techniques.
Forced delays: use nonces with short timeouts use timestamps in addition to other techniques.
- d. (8 points) Recall that three important qualities of a system's security are *confidentiality*, *integrity*, and *availability*. These three properties are very often interdependent.
- i. Give an example where the failure to protect confidentiality leads to a compromise of availability. Be as specific as you can.
ANS: Compromised password lets an attacker change the password, denying future access by a legit user.
 - ii. Give an example where the failure to protect integrity leads to a compromise of confidentiality. Be as specific as you can.
ANS: A buffer overflow gives an attacker complete control of a machine, possibly permitting him to view confidential files.

3. Cryptography

- a. Suppose that A and B have previously established a secret shared key. At some later point they want to establish an encrypted channel between them, but they want to verify that they are both still in possession of the same secret key. Recalling that XOR with a one-time pad of bits constitutes a perfect cipher, A proposes the following solution: A generates a string of random bits of length equal to the key's length, XORs the random bits and her copy of the key and sends the result to B . B receives the message, XORs the result with his copy of the key and sends the result back to A who can verify whether the bits B sent her match her original random string—neither A nor B transmits the key in cleartext. If the bits match, A and B must have used the same key.
- i. (4 points) Explain why this protocol is a bad idea.
An attacker can get the key by simply XORing the two messages that are exchanged.
 - ii. (6 points) Suggest a different protocol to securely accomplish the same task and briefly explain how it works.
Use a nonce challenge.

- b. Consider the RSA algorithm where $p = 7$ and $q = 11$.
- i. (10 points) Find a public–private keypair. For partial credit, explain the steps you take.
 $d = (7, 77), e = (43, 77)$
 - ii. (4 points) Use the public key to encrypt the message whose numerical encoding is 2.
 $2^7 \bmod 77 = 51$
- c. (4 points) Suppose that A generates an RSA keypair with modulus $n = pq$, but her private key is compromised. Rather than generating a new modulus, A decides to create a new keypair using the same modulus. Is this safe? Explain.
- No, it is unsafe. From d and e , it is easy for an attacker to compute $(p - 1) \times (q - 1)$, since $de \equiv 1 \pmod{(p - 1) \times (q - 1)}$. Once that is known, the attacker can compute the inverse of the new public key e' easily.

- d. (8 points) Recall that the DES algorithm consists of 16 rounds of permutation and substitution sandwiched between a permutation and inverse permutation:

$$DES_K(m) = (\pi^{-1} \circ \text{round}_{16} \circ \dots \circ \text{round}_1 \circ \pi)(m)$$

Each round operates on a 64 bit input divided into two 32 bit chunks L and R , the left and right halves. Suppose that the initial permutation of m is L_0R_0 , that is $\pi(m) = L_0R_0$. Recall that the output of the i^{th} round is computed as follows :

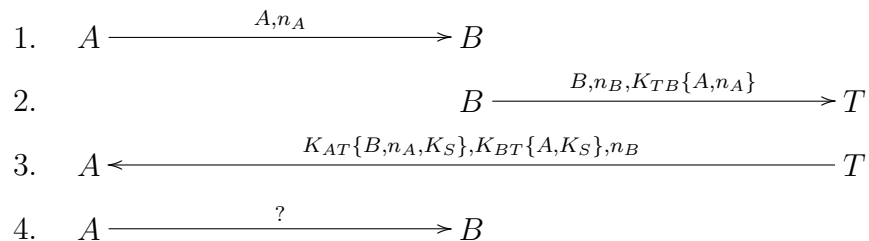
$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \end{aligned}$$

Suppose that the function $f(X, Y)$ is incorrectly implemented and instead of using S-boxes it always returns X . What output does this broken version of DES produce? (Your answer should mention the inverse permutation π^{-1} ; be as specific and concise as possible. Show your work for partial credit)

$$\pi^{-1}(R_0(L_0 \oplus R_0))$$

4. Protocols

Consider the following protocol in which A and B use a trusted third party T to perform mutual authentication and establish a session key K_S . Assume that initially A and T share the symmetric key K_{AT} and B and T share the symmetric key K_{BT} . A and B generate nonces n_A and n_B , respectively. There are four messages in the protocol, the first three of which are shown below.



- a. (6 points) What message should A send to B in step 4 to complete the protocol?
 $K_{BT}\{A, K_S\}, K_S\{n_B\}$
- b. (4 points) Is the protocol still secure if the message in step 2 is changed to have the contents: $B, n_B, n_A, K_{TB}\{A\}$? Explain why or why not (the answer to this question does not depend on your solution for part a).
 No. It is now susceptible to replay attacks—an intruder C can intercept the $K_{TB}\{A\}$ and then later use it to pretend to be B . Also, this opens the potential for a chosen plaintext attack.
- c. (4 points) Is the protocol still secure if the message in step 2 is changed to have the contents: $B, K_{TB}\{A, n_B, n_A\}$? Explain why or why not (the answer to this question does not depend on your solution for part a).
 Yes. If it was secure when the nonce n_B was public, it's certainly still secure when n_B is private.

5. (8 points) Access Control

Suppose that a certain system has n subjects (principals) and m objects (files, etc.). Assume that modifying an entry in an access control list (ACL) or a capabilities list (CAPs) takes constant time and that the lists are not sorted in any particular order. For each operation and representation of the access control matrix described below, give the *worst-case performance* of the operation in terms of n and m .

Operation	Representation	
	ACL	CAPs
Deleting all rights for a particular subject with respect to <i>one</i> given object.	$O(n)$	$O(m)$
Deleting all rights for a particular subject with respect to <i>all</i> objects.	$O(mn)$	$O(m)$