

Homework #3

Due date & time: 1:30pm on November 6, 2007. Hand in at the beginning of class (preferred), or email to the TA (chen131@purdue.edu) by the due time.

Late Policy: Late homeworks will not be accepted.

Additional Instructions: The submitted homework must be typed. Using Latex is recommended, but not required.

Problem 1 (25 pts) Cryptanalysis.

- (3 pts) Explain what do ciphertext-only attacks, known-plaintext attack, and chosen plaintext attack mean?
- (4 pts) Explain how to break the substitution cipher under a ciphertext-only attack? Explain the simplest way to break it under a known-plaintext attack?
- (10pts) Suppose that everyone in the world is using the DES algorithm, show how to use a *chosen plaintext attack* such that after an expensive but doable initialization step, everyone's key can be recovered in very little time. Write pseudo code for the initialization step and the function to recover everyone's key.

Hint: Ideas for the answer (but not the pseudo-code) are in slides for lecture 18.

- (8pts) Answer each of the four questions. How effective would the above attack be under *known-plaintext* (instead of chosen plaintext) attacks? What if everyone is likely to encrypt some common strings in their messages? What if everyone uses the ECB encryption mode? What if everyone uses the CBC encryption mode with the IV randomly chosen?

Problem 2 (5 pts) If the useful life of DES was about 20 years (1977-1999), how long do you project the useful life of AES to be? Justify your answer.

Problem 3 (15 pts) There are three desirable properties for cryptographic hash functions: Preimage resistant, Second preimage resistant, and Collision-resistant. For each of the following applications of hash functions, explain which of these three properties are needed and which are not.

- Alice poses to Bob a tough math problem and claims she has solved it. Bob would like to try it himself, but would yet like to be sure that Alice is not bluffing. Therefore, Alice writes down her solution, appends some random bits, computes its hash and tells Bob the hash value (keeping the solution and the random bits secret). This way, when Bob comes up with the solution himself a few days later, Alice can verify Bob's solution is correct before revealing her solution and the random bits to prove that she had a solution earlier.

- Passwords are stored in a password file, in hashed form. To authenticate a user, the password presented by the user is hashed and compared with the stored hash. A user who gains read access to the password file should not be able to log in by this method. (Assume that the mischievous user does not modify the system in any way before trying to log in.)
- A system administrator, concerned about possible breakins, computes the hash of important system binaries and stores the hash values in a read-only file. A program periodically recomputes the hash values of the files containing the system binaries, and compares them to the stored values. A malicious user who is able to overwrite one of the “protected” files should not be able to change the file without detection.
- Cryptographic signatures are produced by computing a hash of a message, then applying a signature function to the hash of the message. Suppose Eve has a list of messages m_1, \dots, m_n , and their signatures computed using Bob’s signing key, but does not have Bob’s signing key. Assuming that the signature function is not susceptible to attack, it should not be possible for Eve to present Bob’s signature on any message other than m_1, \dots, m_n .
- Suppose that Eve works for a Certificate Authority. She does not have access to the special hardware that computes digital signatures, but she knows the hash function. In addition, Eve can get messages signed, but every message that is signed automatically goes into a log file that Eve cannot change. Eve should not be able to produce a certificate signed by the Certificate Authority that does not appear in the log file.

Problem 4 (10 pts) Suppose Alice and Bob are users of one organization, which maintains a directory of RSA public keys for each user. Alice and Bob communicate regularly using authenticated, encrypted emails. That is, when Alice sends an email to Bob, it will be encrypted using Bob’s public key and digitally signed using Alice’s private key, which can be verified using Alice’s public key. The email software will look up the public keys from the directory. Eve is able to break into the server that stores the directory, and alter the directory file. How should Eve alter that file so that she can read encrypted emails sent between Alice and Bob, and forge messages from either?

Problem 5 (10 pts) A web server requires each user to log in. However, the implementers of the web site are worried about storing passwords on the server, since they are afraid someone might break in and steal them. Therefore, they decide to use a clever idea. When a user creates an account, the account number is stored on the server and the user’s password is stored in a cookie on the user’s machine. Then, when the user tries to log in later, the server compares the password typed in by the user with the password stored in the user’s cookie.

- (a) Assuming that the implementers have not thought of any other clever ideas, how would you log into another users account without knowing their password?
- (b) What methods could you use to keep passwords in cookies, but prevent the attack you devised in part (a)?

Problem 6 (15 pts) The UNIX `crypt` function is a hash function that only looks at the first eight bytes of the input message. For example, `crypt(helloworld)` returns the same value as `crypt(hellowor)`.

Some web sites use the following authentication method to authenticate users: (1) the user types in a user-id and a password P into his web browser, (2) the site, upon verification of the password P , computes $T = \text{crypt}(\text{user-id}||K)$, where $||$ denotes string concatenation, and K is a ℓ -byte site secret key $\ell \leq 8$, (3) the site sends a cookie back to the user containing T , (4) the user can use T to authenticate himself to the site in future connections.

Show that by choosing clever user-id's (of varying length) an attacker can expose the site's secret key K in time approximately 128ℓ . More concretely, the user creates an account, logs in and receives the corresponding T ; he then creates another account (with a different user-id, logs in and receives another T . By repeating this sufficient times, the user recovers K completely. We are assuming there are 128 possible values for each character in a string.

Hint: Try to recover one character of K for each account created. The attack is described in the paper "Dos and Don'ts of Client Authentication on the Web" in USENIX Security Symposium, 2001. Reading the paper is allowed.

Problem 7 (20 pts) Read the following paper (available from the lectures & handouts page), and answer the questions below.

- R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-Based Access Control Models. IEEE Computer, 29(2):38–47, February 1996.

People often claim that RBAC is natural to support policies such as separation of duty and least privilege. Give your thoughts on such claims. In particular, how can one justify such claims? How can one criticize such claims?

"Separation of privilege" and "least privilege" are two of the eight principles identified by Salzer and Schroeder. How does RBAC relate to the other six principles?