

CS 378 - Network Security and Privacy
Fall 2007

Homework #3

Due: 3:30pm CST (in class), December 6, 2007

YOUR NAME: _____

Collaboration policy

No collaboration is permitted on this assignment. Any cheating (*e.g.*, submitting another person's work as your own, or permitting your work to be copied) will automatically result in a failing grade. The Computer Sciences department code of conduct can be found at <http://www.cs.utexas.edu/users/ear/CodeOfConduct.html>

Late submission policy

This homework is due at the **beginning of class** on **December 6**. All late submissions will be subject to the following policy.

You start the semester with a credit of 4 late days. For the purpose of counting late days, a "day" is 24 hours starting at 2pm on the assignment's due date. Partial days are rounded up to the next full day. You are free to divide your late days among the take-home assignments (3 homeworks and 2 projects) any way you want: submit four assignments 1 day late, submit one assignment 4 days late, *etc.* After your 4 days are used up, no late submissions will be accepted and you will automatically receive 0 points for each late assignment.

You may submit late assignments to Vitaly Shmatikov (TAY 4.115C—slide under the door if the office is locked). **If you are submitting late, please indicate how many late days you are using.**

Write the number of late days you are using: _____

Homework #3 (50 points)

Problem 1 (8 points)

You generate an RSA modulus $n = pq$, RSA public key e and the corresponding private key d . Later, you discover that your private key d has been compromised. Instead of generating a new modulus, you decide to re-use the same modulus n and simply generate a new public/private key pair e' and d' . Is this safe? Explain.

Problem 2

Because of the known risks of the UNIX password system, the SunOS 4.0 documentation recommends that the password file be removed and replaced with a publicly readable file `/etc/publickey`. An entry in this file for user U consists of the user's identifier ID_u , the user's public key K_u and the ciphertext C_u . The ciphertext C_u contains the private key S_u corresponding to K_u , encrypted using DES with a key derived from the user's login password P_u , *i.e.*, $C_u = \text{enc}_{P_u}(S_u)$. When U logs in and types in ID_u and some password P' , the system decrypts C_u to obtain S_u .

Problem 2a (5 points)

How does the system verify that the user supplied correct password, *i.e.*, that $P' = P_u$?

Problem 2b (5 points)

How can the attacker stage a brute-force guessing attack on this system? Assume that the attacker has access to the `/etc/publickey` file.

Problem 3 (8 points)

The IPsec architecture document states that when two transport mode SAs are bundled to allow both AH and ESP protocols on the same end-to-end flow, only one ordering of security protocols seems appropriate.

Should AH (authentication) be applied before ESP (encryption), or ESP before AH? Explain your reasoning.

Problem 4 (8 points)

When sending encrypted IPsec traffic through a firewall, why does there need to be an extra IP header? Why can't the firewall simply encrypt the packet, leaving the original source and destination unchanged.

(Hint: consider a destination network which has multiple gateways to the Internet, and the fact that Internet routing can route packets through any of the gateways.)

Problem 5 (8 points)

A certificate contains an identity, a public key, and signatures attesting that the public key belongs to the identity. Other fields that may be present include the organization (*e.g.*, university, company, or government) to which the identity belongs and perhaps sub-organizations (*e.g.*, college, department, branch, office).

What security purpose do these fields serve, if any?

Problem 6 (8 points)

Consider the following protocol which enables Alice and Bob to establish a shared symmetric key K with the help of a trusted server S .

Both Alice and Bob know the server's public key K_S . Alice randomly generates a temporary secret K_A , while Bob randomly generates the new key K to be shared with Alice. The protocol then proceeds as follows:

$$\begin{aligned} Alice &\rightarrow Server && enc_{K_S}(K_A) \\ Bob &\rightarrow Server && enc_{K_S}(K) \\ Server &\rightarrow Alice && K \oplus K_A \\ Alice &&& \text{computes key } K \text{ as } K_A \oplus (K \oplus K_A) \end{aligned}$$

To summarize, Alice sends her secret to the Server encrypted with the Server's public key, while Bob sends the newly generated key, also under encryption. The Server XORs the two values together and sends the result to Alice. As a result, both Alice and Bob know K .

Suppose that evil Charlie eavesdropped on Bob's message to the Server. How can he, with the help of his equally evil buddy Don, extract the key K that Alice and Bob are using to protect their communications?