

CS 378 - Network Security and Privacy
Spring 2009

Homework #3

Due: 2pm CDT (in class), May 7, 2009

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/academics/conduct/>

Late submission policy

This homework is due at the **beginning of class** on **May 7**. All late submissions will be subject to the following policy.

You start the semester with a credit of 3 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 three assignments 1 day late, submit one assignment 3 days late, *etc.* After your 3 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 (5 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 (5 points)

Recall the **cloning** attack on cell phones from Homework 1. The attacker (passive or active) gathers enough information from one or two phone calls to create a clone of the caller's phone. Later on, the attacker uses the clone to impersonate the victim to the cell phone company and make untraceable calls billed to the victim's number, even when the victim's own phone is switched off.

Pear Corp. has released another version of its popular iPhone. For authentication, it uses the Digital Signature Standard (DSS). Each iPhone stores its own DSS private key x and secret random value k . Both are selected at random when the phone is manufactured and burned into tamper-proof read-only memory. The public key corresponding to x is stored in the phone company's database.

When a iPhone initiates a call, the phone company challenges it with a fresh, random challenge C . The phone executes the DSS signing algorithm and responds with $\text{sig}_x(C)$, *i.e.*, challenge C is digitally signed with the phone's private key. The phone company verifies the signature using the phone's public key, and, if verification succeeds, completes the call and bills the phone.

Is iPhone secure against cloning? Give a detailed explanation.

Problem 3

Consider a PKI (public-key infrastructure) authority that periodically issues a list of *valid* certificates. Revocation is implicit: if a certificate number is not present on the list, then it has been revoked or not issued in the first place.

Problem 3a (5 points)

Why is it important that the authority periodically re-publish the list even if no new certificates have been issued?

Problem 3b (5 points)

Is it sufficient for the list to contain only the serial numbers of valid certificates, or should it also contain something else for each certificate? Explain.

Problem 4

Consider a server-assisted mutual authentication and key establishment protocol. Assume that Alice and the Server share a pairwise symmetric key K_{AS} , while Bob and the Server share a pairwise symmetric key K_{BS} . During the protocol, the trusted Server generates a

fresh, random session key K and distributes it to both Alice and Bob as follows:

1. *Alice* \rightarrow *Bob* A, N_A N_A is fresh and random
2. *Bob* \rightarrow *Server* $B, enc_{K_{BS}}(A, N_A, N_B)$ N_B is fresh and random
3. *Server* \rightarrow *Alice* $enc_{K_{AS}}(B, N_A, K), enc_{K_{BS}}(A, K), N_B$ K is a fresh session key
4. *Alice* \rightarrow *Bob* $????$

Problem 4a (5 points)

What message does Alice send to Bob in step 4 of the protocol?

Problem 4b (5 points)

Suppose the second message of the protocol (from Bob to Server) is changed to

$$B, enc_{K_{BS}}(A, N_A), N_B$$

In other words, Bob's nonce N_B is *not* encrypted. Is the protocol still secure? Explain.

Problem 4c (5 points)

Suppose the third message of the protocol (from Server to Alice) is changed to

$$B, enc_{K_{AS}}(N_A, K), enc_{K_{BS}}(A, K), N_B$$

In other words, Bob's identity B is *not* encrypted. Is the protocol still secure? Explain.

Problem 5 (5 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 6

Problem 6a (5 points)

When SSL 2.0 was designed, U.S. export regulations prohibited export of encryption software that used keys longer than 40 bits. In the “exportable” version of SSL 2.0, the client sends to the server 40 secret bits encrypted with the server’s public key. The server then hashes these 40 secret bits with the two nonces exchanged in this SSL session to obtain a 128-bit value, which (for the purposes of this problem) is used as the key in a *deterministic* symmetric encryption scheme protecting communications between the client and server. Recall that the nonces in SSL are sent in the clear and are thus known to the attacker.

What attack is prevented (or at least made much more difficult) by using this 128-bit value derived from the 40-bit secret and public nonces instead of directly using the 40-bit secret as the key? Explain in detail.

(Hint: SSL-protected packets often contain known strings, *e.g.*, HTTP GET in Web sessions).

Problem 6b (5 points)

Does SSL/TLS defend servers against SYN flooding attacks? If yes, explain how. If no, explain why not.