**CS 5950/6030: Network Security — Fall 2005**

# MASTER

**Please note that answers are more detailed than required for full credit.**
This is in order to give you a credit for any partial answers (e.g., if you include one subcase for, say, the case of limitations of testing, you get a credit for the full case).

# Midterm Exam
# -- Version 1 --

**First Name** ________________________ **Last Name:** ________________________

**Row**___________ **Seat** ________

**>>> Please do NOT open this exam booklet until <<<**
**>>> the instructor gives a clear signal to start. <<<**

| Question Nr | Max. Score | Your Score | Comments |
|---|---|---|---|
| 1 | 5 | | |
| 2 | 10 | | |
| 3 | 15+10 | | |
| 4 | 10 | | |
| 5 | 20 | | |
| 6 | 10 | | |
| 7 | 10 | | |
| 8 | 10 | | |
| 9 | 10 | | |
| **TOTAL** | **100+10** | | |

**Notes**:

- Sign the front page now. After the signal to start, sign *all* pages.
- You can use both sides of exam booklet if needed.
- You may have 1 sheet of notes, front and back. **>>> If you plan to use an extra sheet, you must sign it now! <<<** Turn in the sheet of notes with your exam.
- If you are unsure of a question, write down your assumptions regarding the question. Remember, I cannot guess what you are thinking, let me know your thought process for at least partial credit.
- Write clearly. Illegible answers will not be graded.
- You will have 45 minutes to complete the exam.

**Good luck!**

First Name:__________________________ LAST Name:__________________________

1. [5 pts] Define in your own words Principle of Adequate Protection.

Computer items must be protected to a degree consistent with their value and only until they lose their value.

2. [10 pts] Compare stream and block ciphers using the criteria listed in the table below. Enter S (for stream cipher) or B (for block cipher) into the second column to indicate which cipher type is better under a given criterion. Enter a brief explanation into the 3rd row.

| **Criterion** | **S or B better?** | Why? |
|---|---|---|
| delay | S | **Receiver not waiting for all block characters before decoding** |
| error propagation | S | **Error in coding one char does not affect encoding of another** |
| diffusion | B | **Frequency of a *char* from P (plaintext) diffused in a *block* of C (ciphertext)** |
| **susceptibility to malicious insertion** | **B** | **Impossible to insert a char into a block without easy detection (block size would change)** |

3. [15 pts + extra]
   [15 pts] Give example of a solution for establishing a symmetric key via PKE. It can be a solution which does *not* provide sender authentication.
   [EXTRA: 10 pts] For an extra credit, show both a solution without and with sender authentication.

Note: 15 pts for giving any one of the 2 solutions below. 25 pts for giving both.

Solution without sender authentication.

- S determines secret key K
- S encrypts K with $k_{PUB-R}$: C = E( $k_{PUB-R}$ , K )
- S send C to R
- R decrypts C to get K: D( $k_{PRIV-R}$ , C ) = K
- S & R communicate using secret (symmetric) key K

Solution with sender authentication.

- S determines secret key K
- S encrypts K with both $k_{PRIV\text{-}S}$ & $k_{PUB\text{-}R}$:

  $C = E( k_{PUB\text{-}R} , E( k_{PRIV\text{-}S} , K ))$
- S sends C to R
- R decrypts C to get K: $D( k_{PUB\text{-}S} , D( k_{PRIV\text{-}R} , C ) )$
- S & R communicate using secret (symmetric) key K

First Name:_________________________ LAST Name:_________________________

4. [10 pts]
   a. [5 pts] Explain the penetrate and patch approach to judging program security (a.k.a. judging software security by fixing faults).
   b. [5 pts] What are shortcomings of the penetrate and patch approach?

<u>For a)</u>

--Red Team / Tiger Team tries to crack software.
--If it withstands the attack => security is judged as good.

<u>For b)</u>

--Too often developers try to quick-fix problems discovered by Tiger Team.
--Quick patches often introduce new faults.

5. [20 pts]
   a. [15 pts] Write a detailed scenario of a buffer overflow affecting a call stack area.
   b. [5 pts] Explain how attacker can achieve her goals.

<u>For a)</u>

--Initial stack: [data][data][...]
--Pgm executes a subroutine => return address is pushed onto stack
(so subroutine knows where to return control to when finished)
Stack: [ret_addr][data][data][...]

--Subroutine allocates dynamic buffer char sample[10]
=> buffer (10 empty spaces) pushed onto stack
Stack: [..........][ret_addr][data][data][...]

--Subroutine executes: sample[i] = 'A' for i = 10
Stack: [..........][A][data][data][...]
Note: ret_address overwritten by A!
(Assumed: size of ret_address is 1 char)
--Subroutine finishes - buffer for char sample[10] is deallocated
Stack: [A][data][data][...]
--RET operation pops A from stack (considers it return address)
Stack: [data][data][...]
--Pgm (which called the subroutine) jumps to A
=> shifts program control to where attacker wanted

<u>For b)</u>

By using the above scenario, an attacker can specify any return address A for her subroutine. This shifts program control to where attacker wanted.

(In detail:
Upon subroutine return, pgm transfers control to attacker's chosen address A—even in the OS area.
Next instruction executed is the one at address A. It could be the first instruction of a malicious program that grants the highest access privileges to its „executor.")

6. [10 pts] Describe in a sufficient detail an example of a storage covert channel

<u>This is example of one of many possible kinds of storage covert channels.</u>

Example: File lock covert channel.

- Protected variable X has n bits: X1, ..., Xn.
- Trojan within Service Program written to leak value of X.
- Trojan and Spy Program are synchronized, so can „slice" time into n intervals.
- File FX (not used by anybody else)
- To signal that Xk=1, Trojan locks file FX for interval k (1≤ k ≤ n);
  to signal that Xk=0, Trojan unlocks file FX for interval k.
- Spy Pgm tries to lock FX during each interval
- If it succeds during k-th interval, Xk = 0 (FX was unlocked);
  otherwise, Xk = 1 (FX was locked)

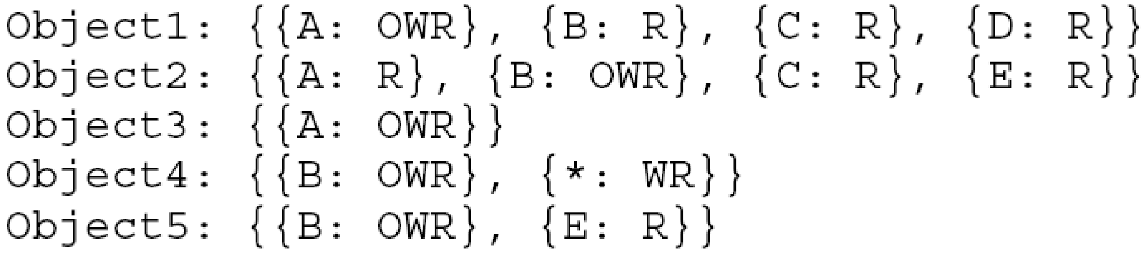
7. [10 pts]
   a. [5 pts] Explain the majority voting technique for fault tolerance.
   b. [5 pts] How can it improve security?

<u>For a)</u>
3 processors run the same software.
All outputs are compared.
Result is accepted if outputs of ≥ 2 processors agree

<u>For b)</u>
Even if an attacker penetrates one of the processors, the other two still give a good answer. Therefore, the system as a whole will give a correct result.

8. [10 pts] Explain using access control lists (ACLs) for object access control using an example ACL.

Subjects: A, B, C, D, E                Five objects—as named



```
Object1: {{A: OWR}, {B: R}, {C: R}, {D: R}}
Object2: {{A: R}, {B: OWR}, {C: R}, {E: R}}
Object3: {{A: OWR}}
Object4: {{B: OWR}, {*: WR}}
Object5: {{B: OWR}, {E: R}}
```

9. [10 pts]

a. [5 pts] Explain the group protection approach for file protection.
b. [5 pts] What are its advantages and disadvantages?

For a)

Groups are created on the basis of a need to share files.
User belongs to one group (otherwise can leak info on group's objects)
Example — In Unix: user, (trusted) group, others

For b)

Advantage:

--Ease of implementation
(OS recognizes user by user ID and group ID upon login. File directory stores for each file: File owner's *user ID* and file owner's *group ID.*)

Problems:

i) User can't belong to > 1 group
Solution: Single user gets multiple accounts
--E.g., Tom gets accounts Tom1 and Tom2
--Tom1 in Group1, Tom2 in Group2
--Problem: Files owned by Tom1 can't be accessed by Tom2 (unless they are public – available to 'others')
Problems: account proliferation, inconvenience, redundancy (e.g., if admin copies Tom1 files to Tom2 acct)

ii) A user might become responsible for file sharing if the user belongs to > 1 group.
E.g., admin makes files from all groups visible to a user (e.g., by copying them into one of user's accts and making them private user's files)
=> User becomes responsible for 'manually' preventing unauthorized sharing of his files between his different 'groups'

iii) Limited file sharing choices
Only 3 choices for any file: private, group, public