Homework 04 and 05

ECE 443/518, Fall 2022

Boyang Wang and Jia Wang

1. *(10 points)* Let's work on an example for the garbled circuit discussed in Lecture 17 between Alice and Bob who want to compute $f = NAND(a, b)$.

   a) *(2 points)* Suppose 0 and 1 on each wire is encrypted into a 5-bit number (0 to 31). Alice chooses $A_0 = 7$, $A_1 = 17$, $B_0 = 19$, $B_1 = 3$, and $O_0 = 18$, $O_1 = 6$. What are S_A and S_B?

   b) *(2 points)* For the encryption function $e_{k_1||k_2}(x) = (k_1 + k_2 + x) \bmod 32$, show how Alice garbles the circuit. Suppose Alice chooses $a = 1$. What Alice should send to Bob as her input?

   c) *(2 points)* Suppose Bob chooses $b = 0$. Show how Bob encrypts his input with Alice's help using OT. Assume Alice's RSA public key to be $(n = 35, e = 5)$.

   d) *(2 points)* Show how Bob computes with the garbled circuit and the encrypted inputs, and then communicates with Alice to determine f.

   e) *(2 points)* Show that Bob cannot decide Alice's choice of a (assuming OT only reveals B_0 but no additional information). As a hint, is it possible for Alice to choose $A_0 = 17$, $A_1 = 7$ while sending Bob exactly the same garbled circuit and inputs?

Answer:

(a). f = NAND(a,b)

$A_0 = 7 = 00111$ $A_1 = 17 = 10001$

$B_0 = 19 = 10011$ $B_1 = 3 = 00011$

So $S_A = A_0(\text{bit } 0) = 0$

$S_B = B_0(\text{bit } 0) = 1$

(b). Encryption function is $E_{k0,k1}(x) = (k_0 + k_1 + x) \bmod 32$

In this equation combining the garbled circuit algorithm we know:

k_0 is A_a; k_1 is B_b; and X should be O_{output}, so we can rewrite it as

$E_{Aa,Bb}(O_{output}) = (A_a + B_b + O_{output}) \bmod 32$

Therefore,

$E_{A0,B0}(O_{output}) = (7+19+6) \bmod 32 = 0$

$E_{A0,B1}(O_{output}) = (7+3+6) \bmod 32 = 16$

$E_{A1,B0}(O_{output}) = (17+19+6) \bmod 32 = 10$

$E_{A1,B1}(O_{output}) = (17+3+18) \bmod 32 = 6$

So the encrypted truth table is

A	B	$E_{A,B}(O)$
7	19	0
7	3	16
17	19	10
17	3	6

Alice cannot reveal A and B directly, nor can she reveal S_A and S_B.

Hence Alice will remove all but the first bit of A and B, and rearrange rows.

A (bit 0)	B (bit 0)	$E_{A,B}(O)$
0	0	16
0	1	0
1	0	6
1	1	10

Above is the garbled tabled Alice generated and will send to Bob.

If Alice choose a = 1, then she should send $A=A_1=17$ to Bob.

(c). RSA public key = (n=35 e=5), private key = (pq=5*7, d=5)

The OT procedure is described as follow:

- Bob should receive two random number x_0 and x_1 from Alice, say $x_0=1$ and $x_1=2$.
- Bob has to choose $x_0=1$ since b=0, then he chooses randomly y=3 and computes:
  $$v = (y^5 + x_0) \bmod n = (3^5+1) \bmod 35 = 34$$
- Bob sends v = 34 to Alice, Alice must calculate the following two equations:
  - $B_0' = B_0 + ((v-x_0)^d \bmod n) = 19+(33^5 \bmod 35) = 22$
  - $B_1' = B_1 + ((v-x_1)^d \bmod n) = 3+(32^5 \bmod 35) = 5$
- And send $B_0'=22$ and $B_1'= 5$ back to Bob
- Bob will recover B_0 by $B_0 = B_0' – y = 22 - 3 = 19$

(d). Now Bob knows A=17 and $B=B_0=19$, both with a bit 0 of 1.

So he finds $E_{A,B}(O)=10$ from the last row of the garbled circuit.

Bob decrypts the output by solving $(17+19+O) \bmod 32 = 10$.

So Bob obtains O = 6. Alice will then reveal o=1.

(e). If Alice chooses $A_0=17$, $A_1=7$, $B_0=19$, $B_1=15$, $O_0=26$, $O_1=6$,

then the encrypted truth table is

A	B	$E_{A,B}(O)$
17	19	10
17	15	6
7	19	0
7	15	16

Hence the garbled circuit is

A (bit 0)	B (bit 0)	$E_{A,B}(O)$
0	0	16
0	1	0
1	0	6
1	1	10

This is the same as the garbled circuit in the original setting.

Now Alice will choose a=0 to send Bob $A=A_0=17$.

Bob will still receive $B_0=19$ via OT.

Overall, Bob will receive the exact same data from A as the original setting.

Bob cannot decide whether Alice chooses a=0 or 1 as both cases are possible.

2. *(2 points)* (Chapter 2 Question 1, p35 in Introduction to Computer Security) Consider a computer system with three users: Alice, Bob, and Cyndy. Alice owns the file *alicerc*, and Bob and Cyndy can read it. Cyndy can read and write the file *bobrc*, which Bob owns, but Alice can only read it. Only Cyndy can read and write the file *cyndyrc*, which she owns. Assume that the owner of each of these files can execute it.

   a) Create the corresponding access control matrix.

   b) Cyndy gives Alice permission to read *cyndyrc*, and Alice removes Bob's ability to read *alicerc*. Show the new access control matrix.

(a)

	alicerc	bobrc	cyndyrc
Alice	{O,X}	{R}	---
Bob	{R}	{O, X}	---
Cyndy	{R}	{R, W}	{R, W, X, O}

(b)

	alicerc	bobrc	cyndyrc
Alice	{O,X}	{R}	{R}
Bob	---	{O, X}	---
Cyndy	{R}	{R, W}	{R, W, X, O}

3. *(3 points)* (Chapter 2 Question 2, p35 in Introduction to Computer Security) Consider the set of rights $\{read, write, execute, append, list, modify, own\}$.

   a) Using the syntax in Section 2.3, write a command $delete\_all\_rights(p, q, s)$. This command causes p to delete all rights the subject q has over an object s.

   b) Modify your command so that the deletion can occur only if p has modify rights over s.

   c) Modify your command so that the deletion can occur only if p has modify rights over s and q does not have own rights over s.

(a). Using the syntax in Section 2.3, write a command delete_all_rights (p, q, s). This command causes p to delete all rights the subject q has over an object s.

```
command delete_all_rights(p, q, s)
delete read from a[q, s];
delete write from a[q, s];
delete execute from a[q, s];
delete append from a[q, s];
delete list from a[q, s];
delete modify from a[q, s];
delete own from a[q, s];
end
```

(b). Modify your command so that the deletion can occur only if p has modify rights over s.

```
command delete_all_rights(p,q,s)
if modify in a[p,s] then
delete read in a[q,s]
delete write in a[q,s]
delete execute in a[q,s]
delete append in a[q,s]
```

```
delete list in a[q,s]
delete modify in a[q,s]
delete own in a[q,s]
end
```

(c). Modify your command so that the deletion can occur only if p has modify rights over s and q does not have own rights over s.

```
command delete_all_rights(p,q,s)
create subject tmp
enter read in a[tmp,s]
if own in a[q,s] then
delete read from a[tmp,s]
if modify in a[p,s] and read in a[tmp,s] then
delete read in a[q,s]
delete write in a[q,s]
delete execute in a[q,s]
delete append in a[q,s]
delete list in a[q,s]
delete modify in a[q,s]
delete own in a[q,s]
destroy subject tmp
end
```

4. *(5 points)* (Chapter 5 Question 2, p71 in Introduction to Computer Security) Given the security levels TOP SECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED (ordered from highest to lowest), and the categories A, B, and C, specify what type of access (read, write, both, or neither) is allowed in each of the following situations. Assume that discretionary access controls allow anyone access unless otherwise specified.

   a) Paul, cleared for (TOP SECRET, { A, C }), wants to access a document classified (SECRET, { B, C }).

   b) Anna, cleared for (CONFIDENTIAL, { C }), wants to access a document classified (CONFIDENTIAL, { B }).

   c) Jesse, cleared for (SECRET, { C }), wants to access a document classified (CONFIDENTIAL, { C }).

   d) Sammi, cleared for (TOP SECRET, { A, C }), wants to access a document classified (CONFIDENTIAL, { A }).

   e) Robin, who has no clearances (and so works at the UNCLASSIFIED level), wants to access a document classified (CONFIDENTIAL, { B }).

Simple security property says that a subject can write to object if subject compartment dominates object compartment. *-property says that subject can write to object if object compartment dominates subject compartment. Let (L,C) and (L', C') be compartments for different entities. ((*L,C*) *dominates* (*L',C'*) $\Leftrightarrow$ $L' \leq L$ and $C' \subseteq C$) is the principle we are going to apply to specify what type of access that the following sentences have.

a. Paul, cleared for (TOP SECRET, { A, C }), wants to access a document classified (SECRET, { B, C }).
Paul **cannot read** and **cannot write** to the document because Paul's clearance level does not dominate document's classification level and vice versa.

b. Anna, cleared for (CONFIDENTIAL, { C }), wants to access a document classified (CONFIDENTIAL, {B }).
Anna **cannot read** and **cannot write** to the document because Anna Paul's clearance level does not dominate document's classification level and vice versa.

c. Jesse, cleared for (SECRET, { C }), wants to access a document classified (CONFIDENTIAL, { C }).
Jesse **can read** document because Jesse Paul's clearance level dominates document 's classification level, but Jesse **cannot write** to the document because document's classification level does not dominate Jesse's clearance level.

d. Sammi, cleared for (TOP SECRET, { A, C }), wants to access a document classified (CONFIDENTIAL, {A }).
Sammi **can read** document because Sammi Paul's clearance level dominates document's classification level, but Sammi **cannot write** to the document because document's classification level does not dominate Jesse's clearance level.

e. Robin, who has no clearances (and so works at the UNCLASSIFIED level), wants to access a document classified (CONFIDENTIAL, { B }).
Robin **cannot read** document because Jesse Paul's clearance level does not dominate document's classification level, but Robin **can write** to the document because document's classification level dominates Jesse's clearance level.

5. *(4 points)* Suppose there are two COI classes {Bank of America, Citibank, Bank of the West} and {Shell Oil, Union'76, Standard Oil, ARCO} in an investment house using Chinese Wall Model. Alice would like to read 4 objects following the sequence a, b, c, d. Assume CD(a)=Citibank, CD(b)=ARCO, CD(c)=Citibank, CD(d)=Standard Oil. Show whether each read is granted and why.

The Chinese Wall model consists of four components:

Objects (O):	Data belonging to a company
Company Dataset (CD):	Consists of individual companies
Conflict of Interest (COI) class:	Contains companies' datasets of companies in competition
Subjects (S):	People who access objects

In principle, the model implements dynamically changing access rights.

COI class 1	COI class 2
Bank of America	Shell Oil
Citibank	Union' 76
Bank of the West	Standard Oil
	ARCO

Alice is Subject S in this case and wants to read Citibank, ARCO, Citibank, Standard Oil in sequence
1, When Alice read Citibank, she should be granted access since this is her first access.
2, When she tries to access to ARCO, she should be granted access since this is her first access
3, When Alice read from Citibank again, she should be granted access
4, When Alice read from Standard Oil, the access is denied, since that she has already read something from the same COI class.

6. *(6 points)* *alice*, *bob* and *cyndy* are three users on a Linux system. There are three groups *alice*, *bob* and *cyndy* as well and each group has the user with the same name as the member. In addition, assume the user *bob* is also the group member of the group *alice*. Explain the following permissions by showing what rights each user have.

   a) *(2 points)* A file F with *alice* as both owner and group owner, and permission 'rw-rw-r--'.

   b) *(2 points)* A file E with *alice* as owner and *cyndy* as group owner, and permission 'rwxr-x---'.

   c) *(2 points)* A directory D with *alice* as owner and *bob* as group owner, and permission 'rwxr-x--x'.

	user (owner)			group member			others		
	read	write	exec	read	write	exec	Read	write	Exec
(a)	■	■		■	■		■		
(b)	■	■	■	■		■			
(c)	■	■	■	■		■			■

For (a), no one can execute it, alice can read and write it as owner; bob can read and write it as group member; cyndy can only read it as others.
For (b), alice can read, write, and execute as owner; bob cannot do anything as others (although bob belongs to the group alice, he cannot use group member permission since the group owner of the file is cyndy); cyndy can read and execute as group member.

For (c), alice can list(r), modify (w), and enter (x, access files and subdirectories with known name) D as owner; bob can list (r) and enter (x) D as group member; cyndy can only enter (x) D as others.