Intro to Computer Security

Assignment 4 - Part 1

Page No.
Date

Name : Rahul Verma
email : rverma4@binghamton.edu
B-Number : B00892091

Ans1) Given : Security Clearance : (U1, S)
Security Level : (O1, TS), (O2, S), (O3, C), (O4, U)

In Mandatory Access Control : No Read Up - A user A can read only those objects whose security level <= Security Level of A & No write-down - A user A can create only objects whose security level >= the security level of A.

(1) U1 can read : O2, O3, O4.

(2) U1 can write : O1, O2.

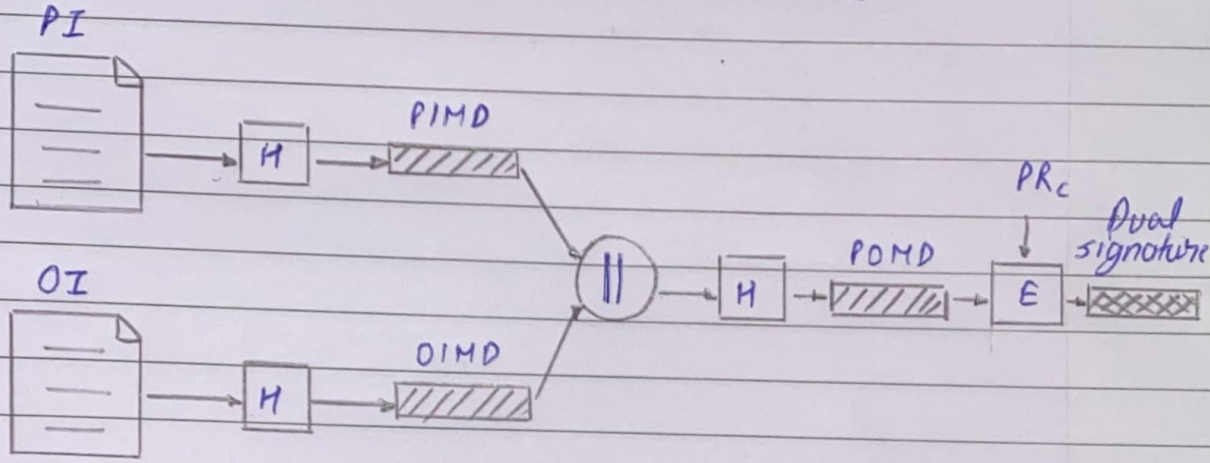
Page No.
Date

Ans 2)

(1) We use a dual signature for this,

- Signed concatenated hashes of OI & PI
- Encrypt the final hash with the customer's private key.

$$DS = E(PR_c, [H(H(PI) \| H(OI))])$$



Dual signature : $DS = E(PR_c, [H(H(PI) \| H(OI))])$

(2) Merchant: the dual signature DS, OI, the message digest for PI (PIMD), and the pub-key of the customer

- The merchant computes $D(PU_c, DS)$, if the same as $H(PIMD \| H(OI))])$, then the signature is verified.

(3) Bank: the dual signature DS, PI, the message digest for OI (OIMD), and the public-key of the customer

- The bank can compute $D(PU_c, DS)$, if the result is the same as $H(H(PI) \| OIMD))])$, then the bank has verified the signature.

Page No.
Date

Ans3) Given SQL Query:

```
SELECT * FROM u WHERE login = '" + userName + "'
and password = '" + password + "';
```

→ If we give the input as: a' or 't' = 't'; --
where, (') : Close the user input field and
(--): Comments out the rest of the line)

The query will become:

```
SELECT * FROM u WHERE login = 'a' or 't' = 't'; --
and password = '';
```

Here, the password part is commented so the query becomes:

```
SELECT * FROM u WHERE login = 'a' or 't' = 't';
```

And as t = t, the WHERE clause is true, so the query becomes:

```
SELECT * FROM u;
```

Therefore, the attackers will be able to access the information of ~~att~~ table u, without knowing their password.

Page No.
Date

Ans4) Given: The output of "f abcde" is 169348921.

→ This happens because the input (abcde) is longer than the size of buffer, so the input (abcde) overwrites the memory location that stores the value of x (123).

Therefore, the output of "f abcde" is 169348921.