CSE331: Homework 1 Solutions (Fall 2021, Radu Sion)

1.

We make use the multiplicativity in the following way:

1. Use the $\mathcal{A}$ algorithm on $E_A(M)$, if it succeeds stop else continue.
2. Repeat the following until you obtain M:
   (a) Generate a random R and compute $E_A(R)$, then compute $E_A(M) * E_A(R)$ which (by multiplicativity) gives $E_A(M * R)$.
   (b) Run algorithm $\mathcal{A}$ on $E_A(M * R)$ and, if it succeeds, obtain M from the $M * R$ it returned (by multiplying it with R^{-1}).

The probability of k repeated failures is 0.99^k which, for a large enough (but still constant) k is very close to 0.

2.

Alice prepares 20 messages one of which is the valuable one (the other 19 are garbage). She runs the oblivious transfer protocol, modified to work for 20 messages rather than just two:

1. Alice generates 20 public-key/private-key pairs and sends the 20 public ones to Bob.
2. Bob uses one of the 20 keys he received to encrypt a key he chose and sends the encrypted result to Alice.
3. Alice decrypts what Bob sent her 20 times, once with each of her private keys. One of the 20 resulting strings is Bob's chosen key but Alice does not know which one it is.
4. Alice encrypts her 20 messages, each with a different one of the bit strings she obtained in the previous step (one of which is Bob's chosen key, the 19 others are garbage). She sends the resulting 20 encrypted messages to Bob.
5. Bob decrypts each of the 20 messages with his chosen key. There is one chance out of 20 (i.e., 5%) that he will obtain Alice's valuable message.
6. Three hours later, Alice gives Bob her 20 private keys so that he can verify that she did not cheat.

3.

When Alice wants to login to Mallory's machine, the following takes place:

- M sends B as the R_M of Step 1 of LOGIN and then catches A's message to T (in Step 2 of LOGIN).
- $M : A$ sends $A, E_A(T_A, B, H(passwd), M)$ to T as Step 1 of WMF (as if $K = H(passwd), M$).
- In Step 2 of WMF, B ends up thinking that $K = H(passwd), M$. From this point on M can impersonate A with B.

Mallory tries to log on at about the same time as Carol. In what follows the notation "$M:X$" means "Mallory pretending to be X". Here is how the attack unfolds:

- $C \xrightarrow{C} B$
  and then Mallory prevents the R'_B of Step 2 of the Carol-Bob exchange from reaching Carol (Bob thinks she got it but Carol is still waiting for it).
- $M:A \xrightarrow{A} B$.
- $M:A \xleftarrow{R_B} B$.

4.

- $M{:}B \xrightarrow{R_B} C$
  so that now Carol thinks that Bob sent her R_B.

- From then on "Alice" (i.e., $M{:}A$) pretends to continue the protocol with Bob, even though it will result in Bob eventually obtaining garbage rather than R_B; this is because $M:A$ cannot do $E_{AT}(R_B)$ and instead has to send Bob garbage that looks like it could be $E_{AT}(R_B)$, e.g., $E_K(R_B)$ = the encryption of R_B by a random key K. What Bob eventually gets is $D_{AT}(E_K(R_B))$ rather than R_B, which he fails to recognize (but he does not know who caused it, and may even attribute it to Carol — more on this below).

- Carol's protocol with Bob now will ultimately result in Trent sending Bob $E_{BT}(R_B)$, which will cause Bob to allow access to $M:A$ (because Bob associates R_B with "Alice", i.e., with $M{:}A$). Bob will eventually "time out" the association between R'_B and C (because he never receives $E_{BT}(R'_B)$ from Trent) and will deny access to Carol (he may even think that Carol is the cause for the above-mentioned $D_{AT}(E_K(R_A))$ garbage that he received earlier).

6.

Mallory tries to log on as "Alice" at the same time as Carol is trying to log on to Mallory's machine. Here is how the attack unfolds:

- $A \xrightarrow{A} M$
- $M:A \xrightarrow{A} B$
- $M:A \xleftarrow{R_B} B$
- $A \xleftarrow{R_B} M$
- $A \xrightarrow{E_{AT}(R_B)} M$
- $M:A \xrightarrow{E_{AT}(R_B)} B$
  which eventually leads to $M:A$ gaining access to Bob's machine (and to Alice gaining access to Mallory's machine).

7.

Mallory is himself with Alice but pretends to be Alice with Bob:

- $A \xrightarrow{E_M(A,R_A)} M.$
- $M:A \xrightarrow{E_B(A,R_A)} B.$
- $M:A \xleftarrow{E_A(R_A,R_B)} B.$
- $A \xleftarrow{E_A(R_A,R_B)} M.$
- $A \xrightarrow{E_M(R_B)} M.$
- $M:A \xrightarrow{E_B(R_B)} B.$

8.

Alice and Bob each generates a random key in a commutative cryptosystem; if it is a 2-key system like RSA then each party keeps secret both of their keys (so Alice knows neither one of Bob's two keys, Bob knows neither one of Alice's two keys).

1. Alice sends $E_A(x)$ to Bob.
2. Bob sends to Alice $E_B(E_A(x))$ and a randomly permuted version of the five $E_B(y_i)$'s $(i = 1, \ldots, 5)$.
3. Alice computes $E_A(E_B(y_i))$, for $i = 1, \ldots, 5$, and checks whether any of them equals the $E_B(E_A(x))$ that she received in Step (2).

9.

Let $r_1, r_2, \ldots$ be the output of the biased one. Obtain the unbiased one as follows: Pair off adjacent bits $r_i r_{i+1}$ (for $i = 1, 3, 5, \ldots$) and output a 1 if $r_i = 1$ and $r_{i+1} = 0$, output a 0 if $r_i = 0$ and $r_{i+1} = 1$ (no output is generated if $r_i = r_{i+1}$). The probability of a 1 is the same as the probability of a 0 $(= p_0 p_1)$.

10.

(1) C' modifies her own card so it contains $AcNr(C)$ rather than $AcNr(C')$. Then C' uses the card and, when challenged by the ATM, C' simply enters $PIN(C')$ and is allowed by the ATM to withdrawn the cash from $AcNr(C)$.

(2) There is no practical way for the bank to distinguish between theft by a C' and the case when C is lying (i.e., when there is no thief C').

(3) One possible solution: Each record of a transaction would also contain $PIN(C)$ as entered by C; this way when C' enters $PIN(C')$ when stealing from C she also leaves a record that contains her $PIN(C')$ in the ATM, allowing the bank to identify C' after the complaint of C. This solution has the disadvantage that it requires that no two customers have the same PIN number.
Another possible solution would be for the ATM to store the PINs together with the list of valid account numbers, so that C' cannot even withdraw the money because the ATM would realize that the pair $AcNr(C), PIN(C')$ does not match the pair $AcNr(C), PIN(C)$ that it stores (in that case the ATM would also automatically take a photographic picture of C' to establish her identity, so that this scheme would work even if other customers have the same PIN number as C').

(4) Instead of $E_B(PIN(C))$ the card would store $E_B(f(AcNr(C), PIN(C)))$ where $f(x, y)$ could be the concatenation of x and y, or their bitwise XOR, or .... The ATM protocol would now first apply f to the account number (read from the card) and the PIN (keyed in by the customer) and verify that encrypting the result by E_B results in the same thing as the second item on the card's magnetic strip (if they don't match then the ATM could take a picture of the customer ... etc).

(5) By going through the hundreds of ATMs in the city in a single day, withdrawing $200 from each one, and leaving the country before the bank discovers (at midnight) the mis-deed. The way to prevent this is for the bank to abandon the off-line operation and switch to on-line operation (so that an amount withdrawn is immediately subtracted from the customer's account balance).