CSE 539 Homework 2

ASU ID: 1222585234

1. (20 points) Short Answer Questions:

(a) (2.5 points) A company stores the customer's password as an MD5 hash in the database. To which of the following network password attacks through the regular login user interface is your website most vulnerable?
A. Dictionary attack
B. Rainbow table attack
C. Birthday attack
Solution. (C) Dictionary attack. Since, we only have access to the login interface and not the hashes we cannot use rainbow table attack. Also using a dictionary of possible passwords is better as compared to Birthday attack which uses random passwords to find a collision.

(b) (2.5 points) Is the following argument true or false, and why? "In Diffie-Hellman key agreement, Alice chooses a random exponent a and sends Bob A = g^a. Bob chooses a random exponent b and sends Alice B = g^b. Since Alice and Bob do not calculate the modulo before sending A and B to each other, the eavesdropper Eve can learn the shared key by computing AB".
Solution: False Eve cannot compute the shared key by multiplying A and B because:

A = g^{a}mod p.

B = g^{b}mod p.

So, AB = g^{a+b}mod p. Whereas the shared key is g^{ab}mod p.

(c) (2.5 points) Is the following argument true or false, and why? "MAC is like a tag that can be added to a piece of data, which certifies that someone who knows the secret key attests to this particular data. We can use a MAC to transform a CPA-secure encryption scheme into a CCA-secure one."
Solution: Yes, we can convert the CPA secure encryption scheme into CCA secure encryption by encrypting first then adding the MAC tag to it.

(d) (2.5 points) In the execution of Diffie-Hellman key agreement, Alice and Bob use the prime p= 2330587351 and the primitive root g= 7. Alice chooses the secret key a= 1570039966. Similarly, Bob chooses the secret key b= 505212543. What will each party send to the other? And what is the Alice & Bob's shared key?
Solution: Given the values of p, g, a, b the DHKA interactions will look like the following:

Alice chooses a = 1570039966. Calculates A = g^{a}mod p = $7^{1570039966}$mod 2330587351
A = 234802898.

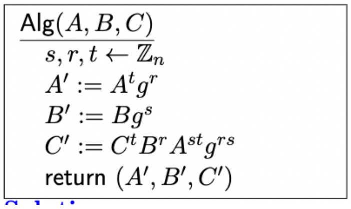
Bob chooses b = 505212543. Calculates B = g^{b}mod p = $7^{505212543}$mod 2330587351
B = 370920881.
Each of them calculates the shared key:

Alice : B^{a}mod p = g^{ab}mod p = 2330587351

Bob : A^{b}mod p = g^{ab}mod p = 2330587351

(e) (2.5 points) Suppose you are given A from a cyclic group generated by g. You are allowed to choose any A′ such that A′ ≠ A, and learn the discrete log of A′ with respect to base g. Can you use this ability to know the discrete log of A? If yes, show your algorithm. If no, explain why? Solution: So, if we are able to learn the discrete log of A’ from the cyclic group generated by g. That means we get the exponent to which g must be raised so that we get A’. Now that we have this value lets call it x. If find the value of g^{x+1}mod p then we will get another value from the cyclic group of g. Now on repeating this process we can find all the values from the cyclic group of g including the value of A.

(f) (2.5 points) Let G be a cyclic group with n elements and generator g. Let DH = {ga, gb, gab ∈ G3 | a, b ∈ Zn}, and suppose (A, B, C) ∈ DH. Show that the output distribution of the following algorithm is uniform distribution over DH.



```
Alg$(A, B, C)$
    $s, r, t \leftarrow \mathbb{Z}_n$
    $A' := A^t g^r$
    $B' := B g^s$
    $C' := C^t B^r A^{st} g^{rs}$
    return $(A', B', C')$
```

Solution: According to the the algorithm given if we substitute the values of A = g^{a} , B = g^{b} and C = g^{ab} in A’, B’ and C’ we get:

A’ = g^{at+r}

B’ = g^{b+s}

C’ = $g^{abt + br + ast + rs}$ = $g^{(at+r)(b+s)}$

That means in the original DH key agreement I can choose a = at+r and b = b+s to get the exponent of the shared key as ab = (at+r)(b+s). Since, a,b belong to integer space and so do s,r,t and g is a generator which generates G. Thus proving that the algorithm has uniform distribution over DH.

(g) (2.5 points) In a RSA cryptosystem, you intercept the ciphertext M = 10 sent to a user, and you know that public key is (e, n) = (17, 55). What is the decoded message (or plaintext) m? Solution: For RSA we need,

p and q are the prime factors of n = 55 which are 5 and 11.

d is the euler totient which computed from p,q and e as d = e mod^{-1}(p-1*q-1).

d = 33

m =M^{d} mod n = 10^{33} mod 55

m = 10

(h) (2.5 points) Alice receives a message which is encrypted with her public key. Can Alice be sure that this message is from Bob? and why?

Solution: No, Alice cannot be sure that the message is from Bob. In public key infrastructure Bob will encrypt the message with Alice's public key so that only Alice can decrypt it. But there is no way for Alice to know that the message was from Bob.

2. (30 points) Passwords: Storage and Security

Consider the following protocol, where the client begins holding a password w of 32-bit length. Given a cryptographic hash function H : {0, 1}⋆ → {0, 1}32, a large prime number p, and primitive root g:

(i) The client chooses a random exponent a and computes A = g^a mod p. The client also computes h = H(w)
(ii) The server chooses a random exponent b, and sends B = g^b mod p to the client along with a random challenge r.
(iii) The client computes K = H(B^a||h||r) and sends it to the server along with A, where || is the concatenation operator.
(iv) The server stores K in its database.

(a) (15 points) Show how the server can perform a dictionary attack on the password.

From the above key exchange protocol we can say that the server has the values of A, b, r, K. To perform a dictionary attack the server will:

1. Have a dictionary of possible passwords and their hashes.
2. Find K' = H(A^b || h' || r) where h' is a hash from the dictionary. A^b is the same as B^a.
3. Now if there is a match between K and K' then we know the corresponding password for that value.

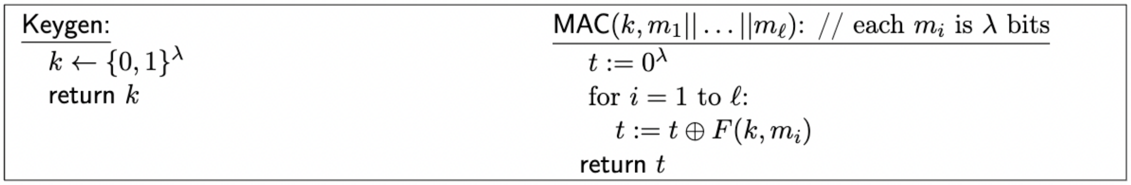
(b) (15 points) If the client also sends h to the server in Step (i), show how the server accepts K from the client before storing it?

If the server sends h in step (i) then by the end of the exchange the server will have A, b, r, K, h. Now the server can calculate the value of K on its own by K = H(A^b || h || r).
Further the server will compare this K and the received K to check if they are the same. If the server gets a match then it accepts the value of K and stores it.

3. (20 points) Authentication

Let Σ be a MAC scheme, we say that Σ is a secure MAC if the adversary knows valid MACs corresponding to various messages, she cannot produce a valid MAC for a different message. Show that the following scheme is not a secure MAC (i.e., show that the adversary can produce a valid MAC for a different message) where $F() : \{0, 1\}^{\lambda} \times \{0, 1\}^{\lambda} \rightarrow \{0, 1\}^{\lambda}$ is a secure block cipher (PRF).

(a) (10 points)



```
Keygen:
  $k \leftarrow \{0,1\}^\lambda$
  return $k$
```

```
MAC($k, m_1||\ldots||m_\ell$): // each $m_i$ is $\lambda$ bits
  $t := 0^\lambda$
  for $i = 1$ to $\ell$:
    $t := t \oplus F(k, m_i)$
  return $t$
```

Solution: According to the above function if we expand the MAC equation for a message m = m1||m2||m3

Then we get t = $0^\lambda \oplus F(k,\ m1) \oplus F(k, m2) \oplus F(k, m3)$

Since we know that if we XOR elements in an equation in different order the resulting value remains the same.

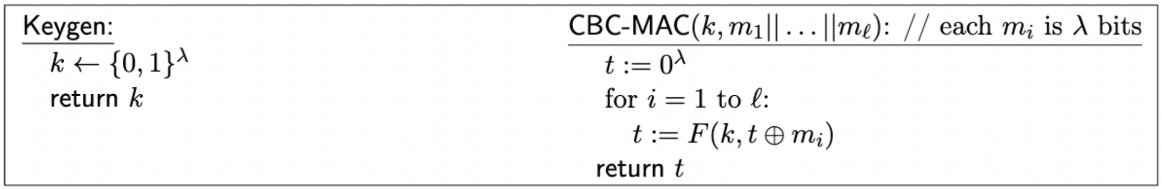
So for a message m' which is a permutation of the blocks of the actual message m.

m' = m2||m3||m1 we will get the new MAC as:

t' = $0^\lambda \oplus F(k,\ m2) \oplus F(k, m3) \oplus F(k, m1)$

So, t = t' this shows that the given MAC scheme is not secure.

(b) (10 points)



```
Keygen:
  $k \leftarrow \{0,1\}^\lambda$
  return $k$
```

```
CBC-MAC($k, m_1||\ldots||m_\ell$): // each $m_i$ is $\lambda$ bits
  $t := 0^\lambda$
  for $i = 1$ to $\ell$:
    $t := F(k, t \oplus m_i)$
  return $t$
```

Solution: The given block cipher is an example of CBC-MAC.

So if we consider a message string m = m1 || m2 || m3 and encrypt this message.

We get C = c1 || c2 || c3. And t = F(k, m3 ⊕ F(k,m2 ⊕ F(k, 0 ⊕ m1))) = c3

Now during decryption if we replace a part of the cipher text and just keep the last block the same.

I.e. C' = c1' || c2' || c3.

Now after decryption we will get a altered message P = p1 || p2 || p3.

Where, p3 = c2' ⊕ Finv(k, c3) ——- Finv() is the decryption function.

If we pass this P through our CBC-MAC function:

t' = F(p3 ⊕ F(k, p2 ⊕ F(k, p1)))

Now, c1' = F(k, p1), c2' = F(k, p2 ⊕ c1')

t' = F(k, p3 ⊕ c2') = F(k, c2' ⊕ Finv(k, c3) ⊕ c2') = F (Finv (k, c3))

t' = c3 = t

Thus showing that the given MAC scheme is not secure.

4. (30 points) Key Exchange

(a) (15 points) Alice and Bob want to perform five instances of Deffi-Helman key agreement (DHKA). Based on the DHKA construction, they should choose a and b exponents randomly each time. However, Alice and Bob use random exponents a and b in the first DHKA instance, then a + i − 1 and b + i − 1 in the i-th instance, where i ∈ {2, 3, 4, 5}. An eavesdropper Eve observes all of these DHKA interactions (e.g., messages sent/received to/from Alice and Bob in each i-th instance). Eve also knows that Alice and Bob generated the keys in the above manner. She later knows the 3-rd DKHA key. Show how she can compute the other four DHKA keys?

Solution: For i = {2,3,4,5} the 4 keys that eve has to find are:

1. K1 =$g^{(a+1)(b+1)}$ mod p
2. K2 = $g^{(a+2)(b+2)}$ mod p
3. K3 = $g^{(a+3)(b+3)}$ mod p
4. K4 = $g^{(a+4)(b+4)}$ mod p

and the first key is $g^{(a)(b)}$.

Since, Eve has observed the interactions she has $g^{a}$and g^{b} and has found the 3rd key.

K2 = $g^{(a+2)(b+2)}$ mod p

Now if we expand K2 we get $g^{ab + 2a + 2b + 4}$ mod p.

On dividing K2 by $g^{(2a)+(2b)+4}$ which can be easily calculated from $g^{a}$and g^{b} we get,

K = $g^{ab + 2a + 2b + 4 - 2a - 2b - 4}$ mod p

K = g^{ab} mod p which is the first key in DHKA.

Now using K, $g^{a}$and g^{b} Eve can get other keys as follows:

1. K1 = K x g^{a} x g^{b} = $g^{(a+1)(b+1)}$ mod p
2. K2 = K1 x g^{a} x g^{b} = $g^{(a+2)(b+2)}$ mod p
3. K3 = K2 x g^{a} x g^{b} = $g^{(a+3)(b+3)}$ mod p
4. K4 = K3 x g^{a} x g^{b} = $g^{(a+4)(b+4)}$ mod p

So, Eve has all the 5 DHKA keys.

(b) (15 points) Another variant of Diffie-Hellman key exchange schemes is to allow one party to determine the shared key. The first few steps are presented as follows. What should Alice do in Step (iii) in order to compute the same key chosen by Bob?

(i) Alice chooses a random exponent a and computes A = g^a mod p. Alice sends A to Bob

(ii) Bob chooses a random exponent b, and computes B = A^b mod p. Bob sends B to Alice.

(iii) Alice ?

Solution: From the above variant of the DHKE,

Alice has : A^{b}mod p = g^{ab}mod p and a

Bob has : g^{a}mod p and b

Considering that Bob has decided to keep the secret key as g^{b}mod p.

Alice has to take the a^{th} root of A^{b}mod p to compute the secret key.

So, $A^{b} \times g^{1/a}$ mod p = $g^{ab/a}$mod p = g^{b}mod p.