Huisen Zhan
hszhan
20457524

1.(a) $n = pq$
$p = 2ga+1$
$q = 2gb + 1$
$n = (2ga+1)(2gb+1)$
$= 4g^2ab + 2ga + 2gb + 1$

(b) $n = 4g^2ab + 2ga + 2gb + 1$
$4abg^2 + (2a + 2b)g + 1 - n = 0$
$g = (-(2a + 2b) + \sqrt{(2a+2b)^2 - 16ab(1-n)})/8ab$
or
$g = (-(2a + 2b) - \sqrt{(2a+2b)^2 - 16ab(1-n)})/8ab$

(c) we know a, b, $n = 4g^2ab + 2ga + 2gb + 1$, and we get g in (b), put them toghter, we can get the factors of N.

2.(a) Yes, the browser will notice the Phishing attack. Because during TLS connection setup, the client side will check the host name in in the certificate match the host name
of the web site that client wants to access. And Mallory just sends the original CIBC certificate, doesn't modfiy it, the two host name will not be matched.

(b) No, the victim's browser won't notice. As above the client only check whether the host names match or not, in this situtation, they are matched.

(c) No, the victim's browser won't notice. Mallory can create the signing key that CIBC generates, and send it to CA. CA won't find this is fake, and use it to create a CIBC certificate useing information that Mallory provides. And victim's DNS cache is poisoned. So the client checking methods of TLS won't sound alerm.

(d) This is because these browsers do not properly support the secure HTTPS communication protocols TLS 1.1 or 1.2 by default. In old version IE and FireFox, even if the browser will warn user, but user still can choose contine to go to that website.

3.(d) Because the key that GunPG generates keys are big and unwieldy. The fingerprints can help people verify that they get the right key. And a key has several properties like the name and email of the key owner, key type, key expiration date and so on. When you are searching for a public key and you find multiple keys with the same properties, the only way to identify the key you are looking for is to compare the fingerprints of the two keys with the fingerprint of the key you require. Attacker can create a fake key and launch an imperson attack, if user don't check fingerprints.

4.(a) Tracker is the query of staff, then q(C) is Lucille, the rest to queries are satff and not satff, then we have:
q(C) = q(satff or Lucille) + q(not satff or Lucille) - q(staff)- q(not staff)

(b) we can use SELECT COUNT(*) to find out Rachel's occupation in company by
SELECT COUNT(*) FROM Occupation = staff or name = Rachel,
SELECT COUNT(*) FROM Occupation = not staff or name = Rachel,
then we start to give the range of salary to find out Rachel. for example, if Rachel is in staff, we can use
SELECT COUNT(*) FROM Employee WHRER(Occuption = staff or name = Rachel) and salary > n.
SELECT COUNT(*) FROM Employee WHERE Occuption = staff and and salary > n to find what range Rachel is. And we continue to decrease the range. When the output is less than k, we can add some output in the query to make sure it more than k, and delete it after. For example, we know Rachel is staff and her salary is between 5000 to 6000, and less than k, we can use
SELECT COUNT(*) FROM Employee WHRER name = not Rachel and 5500<salary<5700
SELECT COUNT(*) FROM Employee WHERE Occupation = not staff and 5500<salary<5700 to find out Rachel's salary.

5.(a) The average case is O(NbW/2), the worst case is O(NbW).

(b) This server could use encrypt-then-MAC. This would provide an increased level of integrity by preventing the leakage of the information required to do an oracle attack.

(c) Since the padding scheme of our server is : for a padding of length n, the padding starts with a byte that has value n (the length of the padding), followed by n − 1 bytes with value 0.
For 3.1
Change line 5
5. for n = b down to 2 do
(a) take r = r1 ... rb−n(rb−n+1 ⊕ n)rb−n+2 .... rb
(b) if O(r|y) = 0 then stop and output (rb−n+1 ⊕ 0)...(rb ⊕ 0)

For 3.2
change line 1
take rk = ak ⊕ 0 for k = j, ... , b

add line between 4 and 5
for n = b - j down to 1 do
(a) take r = r1 ... (rn ⊕ 1)rn+1...(rj-1 ⊕ i)rj...rb
(b) if O(r|y) = 0 then stop and output rj-1 ⊕ i