a3 documentation
Boris Kravchenko
bkravche
20332359

Written Questions

1.
    a. The user may commit an action on the website that gives away their identity, such as logging into an account or clicking a facebook “like” button, which links the identity of the user to the website visit. If the user has to login every time they visit this site using Tor, their visits can be very easily linked.

    b. An adversary can set up a Tor exit node that reports a very high bandwidth, thus attracting more users to use it in their route, therefore increasing the probability of encountering a connection that doesn’t use end-to-end encryption. The adversary can then listen in to the exit node-to-website traffic for those unencrypted connections and this will compromise the ignorant user’s privacy.

    c. The university can measure the average number of packets sent and received between the a controlled computer on the university network and Alice’s chosen entry node into the Tor network when fetching the home page for each of the three sites. Using this information, the university can watch the connection between Alice’s laptop and the Tor entry node and infer each of Alice’s visits to Facebook by the bandwidth being used.

    d. The University can correlate volume and timing information for the laptop-to-firewall and Tor-to-university connections. That is, if Bob’s browser downloads the advertisement, the university can infer with very good probability that it was in fact Bob by analyzing the volume and timing of traffic on those two connections as the ad is being downloaded.

2.
    a. Change the values of the first 16 digits of the IV. For example, I used the IV of 11111111111111110000000000000000 and it got me the user pppppppp. This attack works because upon decryption of the first block, the IV is XORed with the ciphertext of the first block, and since the message fits entirely into the first block, playing around with the IV causes the plaintext of the message to change. Thus, in this example, the attacker has complete control over what plaintext the database receives.

    b. Confidentiality and integrity. I would pad the message with some random data. This random data would end up in the first block, and the actual message would

end up in the second block. Since the value of the IV has no effect on the plaintext of all blocks after the first, this simple fix would prevent the attacker from changing the important bits of the message.

3.

a. -- Alice ---

```
CREATE VIEW Alice_info AS
SELECT *
FROM Accounts
WHERE CustomerName='Alice'

GRANT SELECT
ON VIEW Alice_info
To Alice

-- Clerks --

CREATE ROLE clerk

GRANT SELECT (CustomerName, AccountNumber, Balance),
        UPDATE(Balance)
ON TABLE Accounts
TO clerk

GRANT clerk TO clerk1, … clerkN          --list of users who are clerks

-- Managers--

CREATE ROLE manager

GRANT INSERT, SELECT, UPDATE(CreditRating)
ON TABLE Accounts
TO manager

GRANT manager TO man1...manN          --list of users who are managers
```

b. Consider the following example:

```
CREATE VIEW testView AS
SELECT *
FROM Accounts
WHERE Balance < 100000

GRANT SELECT, UPDATE
ON VIEW testView
To Alice
```

Alice then updates a the balance of a record with a balance is less than 100000 to a balance greater than 100000. This record is no longer part of her view, thus it no longer shows up upon a SELECT.

4. Users should check fingerprints of keys by talking to the person in real life, for example over the phone. This allows the user to reliably authenticate that the key in fact was generated by the person that claims to have generated it. If a user neglects to do this, the user could be affected by man in the middle or impersonation attacks are possible, because anyone can generate a key with a certain email address.