2015

April 20, 2017

Note

Q1 a-c and Q3 don't seem to be covered in the course anymore.

Question 1

d

1. Client queries ISP for IP of `www.somesite.com`
2. ISP queries root server to find IP address of `.com` DNS server
3. ISP queries `.com` server to get IP of `somesite.com` DNS server
4. ISP queries `somesite.com` to get IP of `www.somesite.com` DNS server
5. ISP returns IP address of `www.somesite.com` to client
6. Client can now access that host

e

Yes, an organisation can have the same alias for both. An MX resource record type contains the host name of the mail server.

Answer 3

Question 2

a

Bob receives a message from Alice and wants to ensure

- Confidentiality: Someone else hasn't seen the contents of the message
- Authentication: The message originally came from Alice
- Message Integrity: Contents of the message hasn't been changed

b

- $p = 7$
- $q = 13$

i

$n = p \times q = 91$ is the modulus for the public and private keys (this is part of the public key)

$\phi(n) = \phi(91) = (13-1)(7-1) = 72$ is the number of positive integers less than and relatively prime to n

ii

e is less than n and has no common factors with $\phi(n)$, i.e. $\gcd(e, \phi(n)) = 1, e < n$

e is released as the public key

iii

$e \times d \equiv 1 \mod \phi(n)$

$d \times 5 \equiv 1 \mod 72$

$d = 29$

d is kept as the private key

iv

$9^e \mod n = 9^5 \mod 91 = 81$

Decryption (purely for demonstration): $81^d \mod n = 9^{29} \mod 91 = 9$

c

Both hash function and checksum function both return a value that is difficult to reverse. An Internet Checksum is designed to detect common errors quickly and efficiently. However, it does not attempt to prevent collisions. A hash function does attempt to minimise collicions, making it better for message integrity as the same data cannot create the same hash.

A hash function is a one way mapping of data of an arbitrary size to a fixed length string. To verify a message, you rehash the message and verify it against the hashed copy.

d

- M =message
- $H(x)$ =hash function
- K^- =private key
- K^+ =public key

1. Alice hashes the message, $H(M)$
2. Alice signs the hashed message with her private key, $S = K_A^-(H(M))$
3. Alice encrypts the message with Bob's public key, $C = K_B^+(M)$
4. Alice sends (C, S) to Bob
5. Bob decrypts the message, $\bar{M} = K_B^-(C)$
6. Bob decrypts the signed message with Alice's public key, $H(M) = K_A^+(S)$
7. Bob hashes the decrypted message, $H(\bar{M})$
8. Bob verifies his version of the hashed message matches Alice's version of the hashed message, $H(M) = H(\bar{M})$

e

Macropayments

- $1
- Strong crypto
- Credit/debit cards
- Cash
- Check

Micropayments

- < $1
- Lightweight crypto
- Information goods
    - e.g. Payments for stock quote $0.001

f

Combines two ideas

- Makes it computationally costly for network users to validate transactions
- Reward them for trying to help validate transactions

For a given transaction, after someone has validated it, they would like to broadcast this to the network. To do this, they have to find a nonce x such that when it is appended to the list of transactions and the combination is hashed, the output hash begins with a specific amount of 0s. This is made more or less difficult by varying the number of 0s.