ECLT5820 Distributed and Mobile Systems

Assignment 1 (Topics 1-4)

Due – 11:59pm, 4th Oct. 2021 (Monday)

Please send a pdf file to eclt5820@cse.cuhk.edu.hk with Email title and file name "ECLT5820 Asg#1, Your name, Your student ID".

Q1 (**20 points**) Suppose you are building a large-scale shopping website. In order to provide good service and user experience, what characteristics will you consider in your system? For example, the website should be accessible using different devices, safe for customers to use, and robust when failures happen. Try to illustrate at least five characteristics with explanations.

Answer:

- Heterogeneity: e.g., Support PC client and mobile client.
- Scalability: The website should be scalable according to the business growth.
- Transparency: e.g., the users just use the service without knowing the address of the servers
- Openness: The website should support the popular industry standard, e.g., html.
- Security: We should consider the authentication and fraud detection issues.
- Failure Handling: We should use multiple servers to handle failures due to hardware problem and multiple telecom service to avoid the network interruption issues. We should use UPS as electricity backup.
- Concurrency: we should consider if there will be problems if a user purchase goods on multiple terminals simultaneously.

Q2 (**20 points**) Please explain your choice about whether we should use TCP protocol or UDP protocol to implement the following application-level protocols.

1) File transfer (e.g., FTP)
2) Information browsing (e.g., HTTP)
3) Domain name system (DNS)
4) Remote procedure call (RPC)
5) Streaming media (e.g., Real-time Transport Protocol)

Is UDP more reliable than TCP? Please justify your answer with details.

Answer

1) TCP or UDP. Typically, file transfer uses TCP. But there are also some UDP file transfer tools and projects, such as QUIC, Tsunami, etc.
2) TCP. Most modern websites have a state, and TCP guarantees re-transmissions aren't presented to the application.
3) UDP. Since TCP connections are very "expensive" while UDP is faster. The performance of DNS is the most important issue. Besides, a DNS response can usually fit in a single packet with lots of room to spare, thus using TCP for DNS is wasteful.
4) TCP or UDP. Generally, RPC applications will use UDP when sending data, and only fall back to TCP when the data to be transferred doesn't fit into a single UDP datagram.
5) UDP, for example, the Real-time Transport Protocol uses UDP.

No, TCP has a handshake (acknowledgement) mechanism which can detect packet loss, and it has a packet retransmission mechanism which can handle packet loss. It also provides message resequencing.

Q3 (**20 points**) Suppose you are an email user. When you are using an email system (e.g., login, sending email, or receiving email), what are some potential passive attacks and active attacks you might suffer? Try to identify at least four of them with examples. Design security proper security mechanisms to handle (either by prevention or detection) these attacks.

Answer:

1) Eavesdropping on transmissions (Passive Attack): the content of the email is leaked.
   Security mechanism: protect the email contents by encryption.
2) Denial of service (Active Attack): the email service is under DoS attacks by attackers and becomes not available to serve you.
   Security mechanism: Detect the attackers either manually or automatically and block their accesses to the mail server.
3) Masquerading (Active Attack): Someone may pretend to be my friend and send me an email.
   Security mechanism: Carefully check the mail header to make sure the email is not a masquerading from an attacker; Confirm with your friend to make sure indeed the email come from the friend.
4) Message Tempering or replaying (Active Attack): The message I received may be tempered or replayed by someone during transmission.
   Security mechanism: Message tempering can be prevented by message encryption, and message replaying can be detected by mailer for the duplication of the message, or by the recipient for legitimacy of the message from a known person (to avoid masquerading).

Q4 (**15 points**) Consider the RSA encryption algorithm. Please write down your computation details.

1) p=5, q=11, and e=7, what is the corresponding public key and a possible private key?
2) Try to encrypt '2' with the public key and decrypt '15' with the private key.
3) Assuming Alice has her key pair, denoted as PUBKEY(Alice) and PRIKEY(Alice), and Bob has his key pair, denoted as PUBKEY(Bob) and PRIKEY(Bob). Please design a security mechanism for the following application scenario: Alice sends to Bob an invitation message, and Bob replies with a confirmation. What security features can the mechanism provide? Is there any potential vulnerability of your design? Please explain.

Answer:

(1)
n = p * q = 5 * 11 = 55
z = (p-1) * (q-1) = 4*10 = 40
Compute d with Extended Euclidean Algorithm (not required):
$d \equiv e{-}1 \pmod{z}$
40 = 7 * 5 + 5
7 = 5 * 1 + 2
5 = 2 * 2 + 1
1 = 5 – 2 * 2 = 5 – 2 * (7 - 5) = 5 – 2 * 7 + 2 * 5 = 3 * 5 – 7 * 2 = 3 * (40 – 5 * 7) – 7 * 2 = 40 * 3 -17 * 7
d = -17 mod 40 = 23

Public Key: (7, 55)
Private Key: (23, 55)

(2)
CipherText = 27 mod 55 = 128 mod 55 = 18

PlainText = 15^{23} mod 55
15^2 mod 55 = 225 mod 55 = 5
15^4 mod 55 = 5 * 5 mod 55 = 25
15^8 mod 55 = 25 * 25 mod 55 = 625 mod 55 = 20
15^{16} mod 55 = 20 * 20 mod 55 = 400 mod 55 = 15
((15^{16} mod 55) * (15^4 mod 55) * (15^2 mod 55) * (15 mod 55) = (15 * 25 * 5 * 15) mod 55 = 20

(3)
An example mechanism:
1: Alice encrypts the invitation message with PUBKEY(Bob) and sends the cipher text to Bob.
2: Bob decrypts the cipher text with PRIKEY(Bob) and gets the plain text of invitation message.
3: Bob then encrypts the confirmation with PUBKEY(Alice) and sends the cipher text to Alice.

4: Alice decrypts the cipher text with PRIKEY(Alice) and gets the plain text of confirmation.

Feature:
Confidentiality.

Vulnerability:
Masquerades: Bob cannot verify whether the invitation message is from Alice or not. Because PUBKEY(Bob) is publicly available, some malicious guys can also create the similar message, and pretend to be Alice.

Q5 (**25 points**) Please answer the following two questions.

1) Name three advantages and three disadvantages of distributed systems over centralized ones.
2) What are Naming service and Trading service? What are the differences between them? What are the pros and cons of them when comparing with each other?

Answer:

(1)
Advantages:

- Economics – microprocessors offer a better price/performance than mainframes
- Reliability – if one machine crashes, the system as whole can still survive
- Incremental growth – computing power can be added in small increments

Disadvantages:

- Software – complex software
- Networking – the network can saturate or cause other problems
- Security – easy access also applies to secret data

(2)
Naming is concerned with defining external names for components, so that the components can be identified by Name Server by means of the external name. This approach is very much the same as the identification of participants of the telephone network by means of the white pages.

Trading is concerned with locating components based on the services the component has to offer. This approach is very much the same as the identification of company phone numbers based on the yellow pages.

Difference:

Naming service	Trading Service
Object ("services") are registered by name only	Services are registered by service type and property values
Clients look up by specifying the name only	Clients look up by specifying the service type and desired property values

A reference to one matching service is returned if one is found	A list of reference to all matching services in return
The matching must be exact	Matching is configurable - exact match - partial match
Comparable White Page with telephone book	Comparable Yellow Page with of telephone book

Naming

Pos	Cons
Simple to implement	Need to specify the external names, if the name is not clear. It will be difficult to access the service.
No ambiguity about the result	Cannot know other alternatives which provide similar services
Faster in searching as only one object should match the name	Client always has to identify the server by name
Relatively shorter response time to the client	Inappropriate if client just wants to use a service at a certain quality but does not know from whom

Trading

Pos	Cons
Need not to remember the exact name, just give attributes can access the component	Complex implementation
Can have wider range of choice	The results may not be useful
More suitable of the provider of service is not important	Slower in searching as need to search all matched objects