Part 1 a

In an event-based publish-subscribe pattern, publishes will categorise the message they publish into specific topics – subscribers will only receive messages about the topic they subscribe to. The main advantage of publish-subscribe is loose coupling – publish and subscribers are not aware of each other’s existence and can operate independently. However, this means that publishers cannot verify that all intended recipients (subscribers) have received the message. Hence, an event-based publish-subscribe pattern is not appropriate for cases where the publisher or subscriber needs to be aware of the other’s existence, for example an air traffic controller. The air traffic controller would publish a message and would assume all current aircrafts would receive the message; there is no way for the controller to confirm that all aircrafts have indeed receive the message. Vice versa, the aircraft would also have no way to know if its designated controller is currently operating – it has no visibility on to the controller’s existence.

Part 1 c)

|  |  |  |
| --- | --- | --- |
| Letter | Lamport (i) | VC (ii) |
| A | 1 | <0,0,0,1> |
| B | 1 | <1,0,0,0> |
| C | 2 | <0,0,1,1> |
| D | 3 | <0,0,3,1> |
| E | 2 | <0,0,0,2> |
| F | 4 | <0,2,3,1> |
| G | 2 | <2,0,0,0> |
| H | 3 | <0,0,0,3> |
| I | 5 | <0,3,3,1> |
| J | 6 | <0,3,4,1> |
| K | 4 | <3,0,0,3> |
| L | 7 | <0,3,5,1> |
| M | 4 | <0,0,0,4> |
| N | 8 | <4,3,5,3> |
| O | 9 | <5,3,5,3> |

Part 1 d

In PGP, trust is decentralized and derived from other users – a PGP user can either fully trust or partially trust another user. Fully trust means that the user A trusts user B’s certificate and also trust all certificates signed by user B. Partial trust results in A only trusting B’s certificate and not all certificates signed by B. In CA, trust is centralized – all users that fully trust a certain certificate authority (CA1) will completely trust all certificate authorities certified by the former (CA2, CA3 are trusted if CA1 is trusted).

For PGP to reproduce the role of CA, there will need to be a few master users. These master users would be completely trustworthy – any certificates signed by the master user are trusted. These master users can then choose which users they trust to sign off certificates on their behalf – thus forming a similar centralized hierarchy like in CA.

Part 1e

(i)

A exchanges a nonce of A (NA) with B – this nonce is encrypted in a shared key already known to both A and B. B then responds with its own nonce (NB) and a modification of A’s nonce (NA+1) – A then responds in a similar manner by modifying B’s nonce. The three messages help to set up mutual authentication – firstly, only A and B know the shared key KAB hence it is assumed that if the nonce is modified, only A or B can modify the nonces. This thus proves A and B’s identity to each other. The nonce also helps to confirm that the current communication is fresh. The last message from B to A occurs after mutual authentication has been confirmed – B then wants to provide A with a new session key to be used for their current communication, thus achieving the distribution of a session key.