There is many possible cases. For answers not listed here will also receive either full or partial credits based on the correctness of their answers.

1 (a) [4 points] correct description of app using TS (2pt), explanation of why nonce is inappropriate (2pt)

Case 1.
The main situation where a nonce is not appropriate is when the communication is essentially one way. For example, email or other one way protocols. In such situations, there is not way for the verifier to communicate the nonce to the client.

Case 2.
Use of timestamp in authentication protocol may be appropriate in a situation where per-connection short term state information (e.g. nonce) can not be maintained. In other words, if reduction of per-connection server state is of concern timestamp should be used.

Case 3.
If round trip latency in authentication protocol is critical, then timestamp should be used instead of a nonce since timestamps in protocols offer the advantage of fewer messages (typically one). Note that a nonce could be used with performance drawbacks, but sometimes the flow of communication doesn’t allow the round trip.

1 (b) [6 points] 2pts for each situation. For each case, description of situation using nonce (1pt), explanation of why TS is inappropriate (1pt)

Situation 1.
Assume a local time clock is not available. Then reliance on timestamp may be replaced by a nonce, for instance, in an unilateral authentication protocol.

Situation 2.
The clocks at sender and receiver have to be (at least loosely) synchronized if timestamp were to be used in the protocols. In such a case, nonce would be a better option since it does not require clocks to be synchronized (e.g. geographically distributed nodes). This is an example where the clocks might not be well synchronized.

Situation 3.
An example of using nonce as a commitment in Fiat-Shamir identification protocol (zero-knowledge proof). Here, the nonce is used as challenges to provide freshness, and also as commitments to prevent cheating. Timestamp can not provide these properties.

Situation 4.
If the pattern of communication is such that the verifier already is communicating with the entity proving its identity, then there is no additional
cost to providing the nonce. For example, with mutual authentication of the servers, since the client initiates the exchange, it can provide a nonce that the server will use in proving its identity.

Situation 5.
If stronger guarantees are needed use of nonce is preferred. Timestamps are prone to post-dated clock problem. That is if an authenticator is sent before the clock is reset, that authenticator can be replayed when the time incorrectly recorded in the authenticator is reached. Also the attacker can do a suppress-replay attack if the sender’s clock is behind the receiver’s clock. That is the attacker can first intercept the message, suppress it and replay it later when the timestamp is current. If the sender’s clock is ahead of the receiver’s clock, than replay attacks are possible. Timestamp is prone to these attacks and hence provides fewer guarantees.

2.
Access control list: It corresponds to the rows of the Access control matrix. In ACL, for each object there is a list of principals and actions permitted on that object. The armed guard of the door must have a list of people who are allowed access. This is like an ACL where the object is the door and the subject are the people with the right as the permission to access. It doesn’t provide authentication. Card keys are another example. They provide authentication because they are encoded by a card identifier which is then looked by a computer in its access control list.

Advantages:
- Easy to find all the subjects who have access to a particular object.
- It can be compacted by removing null entries.

Disadvantages:
- Hard to find objects accessible to an individual subject.
- In distributed systems, the ACL must be centralised or replicated. When centralised, then it must be conatced every time to determine permissions on every access. When it is replicated, all the copies must be consistent.

Capabilities list: It corresponds to the columns of the Access Control Matrix. In Capability list, for each principle there is a list of object and actions permitted for that principal. The keys correspond to the capabilities. Keys can be simple capability Keys provide no authentication.

Advantages:
- Easy to find the rights given to a subject
- It is easy to delegate capabilities.
- They dont need to be centralised or replicated in a distributed system since a live object carries capabilities in the memory.

Disadvantages:
- Hard to find all subjects who have access the a particular object.
- It is hard to revoke capabilities.
- It must have a revokation list to be checked when a capability is presented.
Policies need to be determined dynamically depending on the time, system load, security status of the host or relationship with other host.

The subject correspond to the columns, object to the rows and the rights to the individual cell in the access control matrix.