Question: A-7 Is SSL required in SET? Motivate your answer.
Answer: No. SET (Secure Electronic Transaction) aimed to provide application level confidentiality and integrity protection of the transmitted information. It used PKI infrastructure, identities based on X.509 certificates, RSA signatures, symmetric encryption and special software to be installed. It is hard to believe that SSL would be required (it was also very early for SSL at the time of drafting SET) as conceptually the protocol itself is built on many of the same exact components and theory. SET was promised to deliver confidentiality, authentication, integrity and even non-repudiation by itself.

Question: A-11 The acronyms ACS and ADS are both related to VbV (3D Secure). Explain them briefly.
Answer: ACS (Access Control Server) is a authentication (often just a password validation with the issuing bank) service provided in the third domain, the so called Interoperability Domain in the 3D Secure scheme.This is often outsourced, meaning that access control is performed on a third-party web server/address. ADS (Activation During Shopping) is a scheme to offer/force non-participating users to “sign up” while in the process of using their card online. Some argue that ADS poses more risk than advantages since a) the shopping-minded card user is likely to “just get it done” and chose e.g. weak passwords (later used by ACS), or b) this just like ACS often resides at a third-party, introducing doubt or even phishing vulnerabilities into the concept.

Question: A-12 How is mutual authentication between issuer and cardholder achieved in VbV (3D Secure)?
Answer: This is in part achieved by the ACS (Access Control Server) described in question A-11 from the perspective of authentication of the cardholder by the issuer. In practice, a so called Payer Authentication Request (PAReq) is sent to the user which in essence opens an URL to an authentication website.

The other way around, the authentication of the issuer by the cardholder is more problematic as third-parties often are hosting the ACS/website on other domains than the bank’s. This is prone to phishing attacks. If the browser would visit the bank’s site the cardholder could rely on its certificate. In most practical cases, the cardholder simply has to trust a few logos and no true authentication is performed.

Question: A-14 The multiplicative property of RSA provides for blind signatures. What is meant by ”the multiplicative property of RSA”?
Answer: Mathematically speaking it means that the product of the encrypted messages is equal to the encrypted product of the plaintext messages, $E(m1) * E(m2) = E(m1 * m2)$.

Question: A-22 Briefly explain the differences between session-level aggregation, aggregation by intermediation and universal aggregation.
Answer: In the context of micropayments, these techniques are used to group together small payments into larger ones, thus reducing the fees.

- *Session-level:* All small purchases made during e.g. a user's shopping session are aggregated until the end of the session, then processed together. An example could be a user that buys/downloads many songs that only costs very little individually, but during a whole session (e.g. a day) the aggregated amount for all is significant and worth the processing fees for the merchant.
- *Universal:* Aggregation of microtransaction between multiple users and multiple merchants into macro transactions, not just one-user-to-one-merchant as above. Probabilistic payment schemes are often used (see question A-29 below).
- *Intermediation:* An intermediary (e.g. third-party payment provider) gathers all microtransactions from multiple users and/or merchants and only processes them when it has reached a certain threshold. Example could be a virtual wallet which has virtual coins purchased once, then used in small increments on multiple merchants or occasions.

Question: A-24 Explain how a hash chain, similar to the one computed in PayWord, can be used to implement a one-time-password login in e.g., Linux or Windows.
Answer: By applying a hash function many (*n*) times to a seed value, a hash chain is created. The last value (call it x) is stored in the system. Now, when the user wants to log in, he applies the hash function n-1 times to the seed and submits the value (call it p). Now the system knows it is the correct password since the hash of it (now seed hashed n times) equals the stored value x. Now x is updated to be the new value in the chain, p. Since the system expects the next value after each login, an eavesdropper has no use for the one time password. The login procedure can be repeated n-1 times, then a new seed is needed and start over.

Question: A-29 What is meant by a probabilistic payment? How does the Electronic Lottery Tickets scheme differ from Peppercoin from the user's perspective? How do they differ from the Merchant's perspective?
Answer: The concept of probabilistic payments is often used as a technique to accomplish universal aggregation (see question A-22 above) in the context of microtransactions where rare larger payments are "cheaper" to do due to the fee structure than many small ones. The idea is to instead of making a micropayment of e.g. \$x, the user makes a macro payment of \$y with the probability x/y (and pays nothing with the probability 1-x/y). Example; Let x=1, y=100. Then, user pays \$100 with probability 0.01 (1/100) and \$0 with probability of 0.99 (1-0.01). On average though, he pays exactly \$1.

The concept is denoted electronic lottery tickets and though it of course evens out in the long run, a user can be hit with (several) large payments immediately, making the system feel unfair. A user who only buys very few things, can in theory also be overcharged. An improvement, Peppercoin, aims to guarantee that a user never pays more than is "spent". The problem is moved over to the bank's side of the transaction instead. For the merchant, the two solutions are equal. He gets paid the large amount when it is time for it, probabilistically speaking.

Question: A-34 How is the difficulty in Bitcoin block hashing adapted so that it (almost) always takes about 10 minutes for the system to produce a new block, regardless of the computational power that enters the system?
Answer: Bitcoin uses a "proof of work" which regulates the computation efforts of new blocks. Specifically, a hash must be computed with a constrained result (must be lower than a threshold number). It is this number, which is re-evaluated every 14 days, that is determined to aim for a 10 minute block creation time. In other words, every two weeks, if much more computational power has entered the network, the threshold is adjusted so that the computation gets harder and the time is held at a constant 10 minutes.