A-3 *Give two common ways to prove/make probable that the person making a card-not-present transaction is in physical possession of the card. Compare the two alternatives in terms of security.*

Two efforts to increase security during a card-not-present transaction discussed during the course are **SET** and **3D Secure**. These two methods provide authentication that the buyer is in fact the one registered to the card used.

I find it redundant to explain the whole protocols in detail since the question in hand is just to compare the security of them which we can do without this.

3D Secure provides user authentication typically via the use of a password the cardholder must enter at the time of purchase. Other authentications can be chosen by the bank such as BankID (this author has touchID via BankID app for example). 3D Secure can also protect the cardholder against phising sites by allowing the display of a Personal Assurance Message set by the cardholder on card activation, so this person can verify that the pop-up box in which to enter the password is from the bank. I.e two factor authentication. To stop attackers from intercepting a card and choose a password or such before the user can, the enrollment into the 3D Secure system is fortified by a number of options. Either the bank can choose to let you activate the card face-to-face where they can check your ID, or via their website where its assumed only you can log into your personal pages/account management. Or a third option is available where the card is activated on the first purchase made and password is chosen at that time. This is protected by the use of personal questions assumed only the intended user knows.

SET, Secure Electronic Transaction, provides more than just two factor authentication but failed to gain traction on the market due to user friendliness issues. In this authentication method the merchant will never get a hold on the card details and the bank will never know the items ordered. The use of PKI and certificates are implemented, which is the user friendliness issues mentioned. This protocol relies on the use of the dual signature technique and encryption via the gateways public key to keep the order information from being read by the bank and the payment information from being read by the merchant. The dual signature consists a hash/digest of two already hashed parts, the payment information and the order information. with the usage of hashes and signatures this protocol also provides integrity of the data. In summary, 3D secure provides us with authentication of the cardholder, an optionally authentication of the bank. While SET provides us with two factor authentication, data integrity, and confidentiality of information. Since 3D Secure typically uses a password the strength of this protocol lies on the cardholder. If a bad/easy password is chosen this is susceptible to guessing attacks, for example if the password is the name of the pet. Or if the personal questions are easy to guess. A possible security flaw of the SET method is the breach of the PKI or loss of private key. *PS sorry for this wall of text.*

A-6 *In SET, why is the Payment Information first symmetrically encrypted and not immediately encrypted with the Gateway's public key?*

The payment information, PI, is symmetrically encrypted with DES in SET **due to the speed of this encryption algorithm**. Symmetric encryption is a lot faster than asymmetric, so encrypting everything with the gateways public key would take a lot more time. The key used by the cardholder is then encrypted with the use of the public key of the gateway and sent together with the encrypted PI. When the merchant then relays the information to the payment gateway after order verification the gateway can unwrap/decrypt the key used to encrypt PI. Then in turn decrypt the PI and verify this with OIMD in the dual signature.

A-12*How is mutual authentication between issuer and cardholder achieved in VbV (3D Secure)?*

As mentioned in question **A-3** the 3D Secure protocol offers two factor authentication from the bank side via the use of a Personal Assurance Message to be displayed with each purchase, verifying the pop-up window was issued by the bank. And from the client side via the password entered by the cardholder, which is assumed secret to only this person.

A-14*The multiplicative property of RSA provides for blind signatures. What is meant by "the multiplicative property of RSA"?*

This simply means that $E(m_1) * E(m_2) = E(m_1 * m_2)$. Where $E(x)$ is the encryption of x and $*$ is an arbitrary operation.

A-15*When requesting a blind signature, why must Alice keep r secret?*

r is the random value Alice uses to blind her message with. Consider blind RSA signing:

Alice calculates $r^e \bmod N$, where e and N are public, r is random secret value and $gcd(r, N) = 1$.

Alice calculates the blind message $m' = mr^e \bmod N$. And sends this to Bob (signer).

Bob can calculate $d = e^{-1} \bmod \phi(N)$.

Bob calculates $s' = (m')^d \bmod N$, and sends this to Alice.

Alice, who knows the inverse of r, r^{-1}, can calculate the signed message $s = m^d$ by calculating $s = s'r^{-1} \bmod N$.

From this we can see that if r is not secret an attacker can calculate $r^e \bmod N$ and r^{-1}. And thereby calculate s.

A-18 *When Alice buys something from Bob using the untraceable E-cash scheme, why is it impossible for Bob to learn the identity of Alice?*

The merchant Bob creates a binary vector of length k were the value of each indices, $1 \leq j \leq k$, is random. This vector is then given to Alice which returns for each indices, $(x_j, a_j \oplus ID, d_j)$ if the value is 0 or returns (y_j, a_j, c_j) if the value is 1.

The XOR operation protects Alice's identity from Bob. To retrieve ID one would need a_j for the same index as $a_j \oplus ID$ which never happens since Alice only sends either a_j or $a_j \oplus ID$ for each index in the binary vector.

A-28 *Compare the PayWord protocol and the Peppercoin-like protocol in the lecture notes from the point of view of the customers, both in terms of what they pay, and in terms of what they need to compute to make a purchase.*

Payword: For each merchant Alice visits see computes a hash chain at the time of the first purchase. this hash chain is sent together with merchant information and Alice's certificate signed by the bank. For each item see wishes to purchase she sends (w_i, i) to the merchant where i would be the total mount of money of all purchases made so far. When the merchant has enough micropayments from Alice he sends it to the bank who verifies the commitment and total amount of all purchases and debits Alice that amount. In summary Alice has to compute a large hash chain and is debited for the amount the items costs.
Peppercoin: In this scheme Alice only needs to send $\{T, S\}_{PRIV_A}$ to the merchant and increment S, i.e she only has to sign information about purchase and an increment a serial number S. In this scheme as with PayWord she's only debited the amount of the purchases.

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?*

Probabilistic payment uses probability to choose what amount is to be debited of the user. The probability is calculated as $\frac{\mu}{\gamma}$ where μ is a micropayment and γ is macropayment. And $1 - \frac{\mu}{\gamma}$ is the probability Alice pays nothing.

Lets say the probability is $\frac{1}{100}$. This means that if Alice makes 100 purchases of 1 SEK each she will be debited an amount of 100 SEK once and 99 times nothing averaging to 1 SEK per payment.

In the Peppercoin scheme the merchant only had to calculate a mapping function F on the information Alice sent him and relay this plus the information from Alice to the bank. whereas in the Electronic lottery tickets scheme the merchant have to calculate a hash chain for Alice the first time she buys from him. This hash, m_0, is sent to Alice which includes this in her commitment, $S = \{M, w_0, m_0, C\}_{PRIV_A}$. For every purchase made the merchant has to calculate

a new m_i. The only thing Alice has to compute is to sign the commitment. Payments made by Alice in this scheme is now calculated as following:

$$\text{If } m_i \bmod \tfrac{\gamma}{\mu} = w_i \bmod \tfrac{\mu}{\gamma} \text{ then Alice pays } \gamma \text{ SEK.}$$