Problem Set 8 Solutions

Problem 8-1. Plaintext-aware encryption

(a) In iKP, what is really needed is non-malleability. Indeed assume that the encryption scheme used in the protocol is malleable. Then it could be possible given an encryption of a slip (including credit card number, order and delivery address) to compute the encryption of a related slip, for example with the same credit card number, but different order and delivery address. This could open the door to fraud. For example once I see one of your encrypted slips I can start charging stuff to your credit card number even without knowing it, by producing different slips with the same credit card number.

By requiring plaintext-awareness, the authors also enforce non-malleabilty (see part (b))

(b) Plaintext-awareness is a stronger condition than non-malleability. Indeed if a secure encryption scheme E is plaintext-aware, the sender must know what the ciphertext decrypts to. We want to prove that the scheme is also non-malleable. Given a ciphertext $c = E(m)$, suppose that I can produce a ciphertext of a related message for example $c' = E(m+1)$. Because the scheme is plaintext-aware, this means that I must know $m+1$, which means that I know m. So I computed m given only c, that is the scheme is not secure.

The argument above generalize to the concept of "related" message, if one assumes that security of the encryption scheme means that you have no informaiton about the message encrypted. In particular that you will not be able to come up with "related" messages.

Problem 8-2. Electronic Cash

Suppose tha User pays Vendor with coin $(m, S(m))$. If Vendor thinks that he may want to transfer the coin later, instead of depositing directly to the Bank , he chooses the challenge c during the payment protocol in the following way.

The Vendor prepares a pseudo coin $(m', \mathcal{S}(m'))$ and computes $c = \mathcal{H}(m', \mathcal{S}(m'))$ where $\mathcal{H}$ is a secure hash function.

During the transfer protocol OtherVendor will receive $m, S(m), c, r, m', \mathcal{S}(m')$ and will perform the additional check that $c = \mathcal{H}(m', \mathcal{S}(m'))$.

This will force Vendor to use the same pseudo-coin each time he wants to transfer the original coin, so double-transferring will be detected because Vendor with high probability will answer two challenges on the same pseud-coin.