Opgaveregning 11 (21-04-2020)

## Chapter 8

### Review question 9

**In what way does a hash provide a better message integrity check than a checksum (such as the Internet checksum)?**  
Hvis der bare er ændret en enkelt lille ting i det der bliver hashet, så får man en vidt forskellig hash, hvilket gør det nemt at se noget er galt. Udover det så kan forskellige stykker data have den samme checksum, hvilket kan skabe problemer.

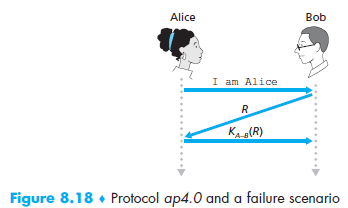
### Review question 10

**Can you “decrypt” a hash of a message to get the original message? Explain your answer.**Nej, det er muligt men det ville tage utroligt lang tid, så det er slet ikke værd at prøve. Dette er også grunden til at hash funktioner er så brugte.

### Review question 16

**What is the purpose of a nonce in an end-point authentication protocol?**Det er at undgå “reply” attacks, hvor man bare gensender beskeder, dette er ikke muligt med nonce, da dette vil sørge for at beskeder er forskellige hver gang.

### Problem 15

**Consider our authentication protocol in Figure 8.18 in which Alice authenticates herself to Bob, which we saw works well (i.e., we found no flaws in it). Now suppose that while Alice is authenticating herself to Bob, Bob must authenticate himself to Alice. Give a scenario by which Trudy, pretending to be Alice, can now authenticate herself to Bob as Alice. (*Hint*: Consider that the sequence of operations of the protocol, one with Trudy initiating and one with Bob initiating, can be arbitrarily interleaved. Pay particular attention to the fact that both Bob and Alice will use a nonce, and that if care is not taken, the same nonce can be used maliciously.)**  
Det er muligt hvis Bob tager kontakt først og ”authenticate” ham selv først, derefter kan Trudy ”extract” noncen og derefter lade som om hun er Alice.

### Problem 18

**Suppose Alice wants to send an e-mail to Bob. Bob has a public-private key pair (*K* B +, *K* B -), and Alice has Bob’s certificate. But Alice does not have a public, private key pair. Alice and Bob (and the entire world) share the same hash function H(#).**

1. **In this situation, is it possible to design a scheme so that Bob can verify that Alice created the message? If so, show how with a block diagram for Alice and Bob.**

Nej det er ikke muligt fordi at alle kan se Bob’s public key og derfor kan alle sender beskeder til ham. Derfor er det ikke muligt at vide om det er Alice der sender beskederne.

1. **Is it possible to design a scheme that provides confidentiality for sending the message from Alice to Bob? If so, show how with a block diagram for Alice and Bob.**

Ja dette er muligt da Alice kan sende beskeder til man ved at ”encrypte” beskeder med hans public key, hvorefter det kun er ham der kan ”decrypt” dem med hans private key.

### Problem 23

**When Bob signs a message, Bob must put something on the message that is unique to him. Bob could consider attaching a MAC for the signature, where the MAC is created by appending his key (unique to him) to the message, and then taking the hash. Will it cause any problem when Alice would try verification?**Ja det ville det, da Alice nu ville kende hans private key, hvilket gør at den ikke længere er private og derfor vil systemet bryde ned.