Understanding Cryptography

Homework No.6 **Due Date: 1400.03.31**

Chapter 11

1. Birth-date and the collision challenge!

a) What is the minimum number of students in a class needed to have at least two students with the same birth-date with probability more than $1/2$?

b) If a year has N days and the number of students is K, find the probability of having at least two students with the same birth-date as a function of K and N.

c) If we want to observe collision in a hash function with outputs of size n bits wit probability more than $1/2$, how many random messages do we need?

(Hint: You can use the inequality $1 - x \leq e^{-x}, \quad x > 0$)

2.

We consider three different hash functions which produce outputs of lengths 64, 128 and 160 bit. After how many random inputs do we have a probability of $\varepsilon = 0.5$ for a collision? After how many random inputs do we have a probability of $\varepsilon = 0.1$ for a collision?

Chapter12

3.

We study two methods for integrity protection with encryption.

3.1. Assume we apply a technique for combined encryption and integrity protection in which a cipher text c is computed as

$$c = ek(x||h(x))$$

Where $h()$ is a hash function. This technique is not suited for encryption with stream ciphers if the attacker knows the whole plaintext x. Explain *exactly* how an active attacker can now replace x by an arbitrary x' of his/her choosing and compute c' such that the receiver will verify the message correctly. Assume that x and x' are of equal length. Will this attack work too if the encryption is done with a one-time pad?

3.2. Is the attack still applicable if the checksum is computed using a keyed hash function such as aMAC:

$$c = e_{k_1}(x||MAC_{k_2}(x))$$

Assume that $e()$ is a stream cipher as above.

Chapter 13

4.

People at your new job are deeply impressed that you worked through this book. As the first job assignment you are asked to design a digital $pay - TV$ system which uses encryption to prevent service theft through wiretapping. As key exchange protocol, a strong $Diffie–Hellman$ with, e.g., 2048 −bit modulus is being used.
However, since your company wants to use cheap legacy hardware, only DES is available for data encryption algorithm. You decide to use the following key derivation approach:

$$K^i = f(K_{AB} \parallel i)$$

Where f is an irreversible function.

4.1. First we have to determine whether the attacker can store an entire movie with reasonable effort (in particular, cost). Assume the data rate for the TV link is$1\ Mbit/s$, and that the longest movies we want to protect are 2 hours long. How many Gbytes (where $1M = 106$ and$1G = 109$) of data must be stored for a 2 hour film (don't mix up bit and byte here)? Is this realistic?

4.2. We assume that an attacker will be able to find a DES key in 10 minutes using a $brute - force$ attack. Note that this is a somewhat optimistic assumption from an attacker's point of view, but we want to provide some medium-term security by assuming increasingly faster key searches in the future.

How frequently must a key be derived if the goal is to prevent an offline decryption of a $2 - hour$ movie in less than 30 days?

5.

CrypTool

1. Use hash tools within CrypTool to do the following exercises;
   1. Use CrypTool's built-in HMAC algorithm to generate a MAC for your text of choice. The parameters the algorithm should receive include: an arbitrary hash function, an HMAC variant of different keys, your first name as the first key and your last name as the second key. The HMAC should be generated accordingly.
   2. Now compute the HMAC of the same text but this time, use CrypTool's hash functions directly instead of using the built-in MAC. Obviously, you shall use the same hash function, you chose in the previous part, and the same key pair.

Programming

Do the following exercise by writing codes in your favorite programming language. Please be noted that you may use available codes on the internet only to draw inspiration but not to copy. Also, please provide brief reports on your codes in which you include your sample inputs and your program's output to them.

1. Using SHA256, implement the HMAC algorithm in your favorite programming language. You should implement all parts except the hash algorithm on your own. Compare your code's final results with those of built-in HMAC implementations.

2. In the following exercises, write a client server program for Key Establishment in your favorite programming language, where the server serves as a Certificate Authority. Take the following points, derived from chapter **13.3.2** of your textbook, into account:

- Clients ask for and receive certificates from the server over an authenticated channel after their identity is verified by the server; However, the channel they communicate over among themselves is not authenticated.
- After receiving their certificates, the clients use asymmetric cryptography to establish a shared key between themselves.
- The certificates' format doesn't need to be as complicated as X.509 Certificates. The basic structure in page 345 of your text book is acceptable.

i. Write a client server program in which you run the server program in tandem with, at least, two replicas of the client program; Clients should establish a shared key after receiving their certificates from the server, exchanging them with one another, and verifying them.

ii. Change the program in the previous part in a way that enables CA chaining; You should simultaneously run, at least, two replicas of the serve program along with, at least, two replicas of the client program, where each client receives its certificate from a different server. You should manage the link among the servers according to pages 349 and 350 of your textbook, so that the clients can still verify each other's certificates.

Please be noted that you may not use any built-in libraries to do this exercise except for the key generation algorithms.