Understanding Cryptography

Homework No.6 **Due Date: 01.03.18**

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 $1 - x \le e^{-x}, \quad x > 0$)

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?

3.

Let p be a prime number and g a generator of z_p^* , consider the function $h: z \to z_p^*$ where $h(m) = g^m \ mod\ p$.

3.1. is h collision resistant? Explain your answer?

3.2. If we assume the difficulty of the discrete logarithm problem in z_p^*, can you explain why this function is on way?

4.

We study two methods for integrity protection with encryption.

4.1. 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?

4.2. 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.

5.

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.

6.

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.

7.1. 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 manyGbytes(where $1M = 106$and$1G = 109$) of data must be stored for a2hour film (don't mix up bit and byte here)? Is this realistic?

7.2. 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 30days?

7. Write a client server program in your favorite programming language, where the client and the server can send arbitrary messages with at most 4096bit length. Use cryptographic mechanisms to guarantee the integrity and confidentiality of exchanged messages over the insecure communication channel. Take following points into account:

a) The adversary can read the source codes of client and server. Additionally, he can access and read files on the system. So, you shall not hardcode or save any of your secrets within your codes or in any of the files on your system.

b) The attacker can eavesdrop the channel. Therefore, to ensure the confidentiality of messages, you might want to use a mechanism with which the sender converts a message into an alternative form unreadable by the attacker using a shared secret, which is only known to authorized parties, and the receiver reverts the message to its original form using the same secret. Take care about the strength of the mechanism you choose and pay attention to weaknesses such as determinism and malleability.

c) As the channel is not initially secure and neither can you hardcode the shared secret, you shall use a mechanism through which one of the parties chooses the secret and sends it to the other party in a secure manner. Here, to guarantee the confidentiality of the exchanged secret, you should opt a cryptographic mechanism with which the sending party converts the secret, itself, to a form, again, obscure to the adversary. But, this time, the only person who can convert the secret to its original form is the receiving party who has a private secret, which helps it about returning the shared secret to its initial form.

d) Finally, besides eavesdropping, the attacker can actively modify the packets exchanged between parties, even if he doesn’t have the slightest idea about their contents. To prevent this from happening and guarantee the integrity of the messages, you should attach to the message block, a unique value derived from the message, itself. This value should be the output of a cryptographic function that receives the message as input. It should, further, have this property that if the adversary changes any of the original message’s bits the value associated with the new message should completely differ from the attached value. So that, this way, the other side can recognize the message has been modified on the way. Moreover, it should be computationally infeasible for the attacker to guess the content of the message from this attached value, or find a message different from the original message that, if given to the same cryptographic function as input, results in the same value.

For simplicity, assume that capabilities of the attacker are limited to those mentioned above, and he cannot do anything further, such as impersonating communicating parties, and so on.