Technische Universität München
Informatik VIII
Netzarchitekturen und Netzdienste
Prof. Dr.-Ing. Georg Carle
Dr. Heiko Niedermayer

Dipl.-Inform. Ralph Holz

# Exercises for lecture „Netzsicherheit“
# Assignment 3, WS10/11

| | |
|---|---|
| Hand-out: | Thursday November 25th 2010 |
| Deadline: | Thursday December 2nd 2010 |
| Exercise course: | Wednesday December 8th 2010 |

## Task 1: Weaknesses of SHA-1 and New Cryptographic Hash Functions

In the years 2005-2009, SHA-1 came under increasing pressure as methods for collision attacks were published. As a response, the National Institute of Standards and Technology in the USA started a public competition to devise a successor, SHA-3.
In this task, we will investigate the weaknesses of SHA-1 and see how practical they really are. We will also have a brief look at the competition for SHA-3.

a) Brush up your knowledge about pre-image resistance properties of cryptographic hash functions. Then do some research on the Web. What is the current state-of-the-art result

a. for finding collisions?
b. for breaking 1st pre-image resistance of SHA-1?
c. for breaking 2nd pre-image resistance of SHA-1?

b) Assume you are an attacker. You have intercepted a message with a digital signature: *(m, sig)* with $sig = K_{priv}(SHA1(m))$. You would like to alter the message without the receiver noticing, but you cannot break the signature algorithm itself.

a. If you can break 1st pre-image resistance of SHA1, what can you do?
b. If you can break 2nd pre-image resistance of SHA1, what can you do?

c) Now assume you are a rich and evil genius and you have the power of a large computer cluster at your disposal. You can use the methods you have found in a). Assume $10^{10}$ computational steps of a method can be done in one second. How long does it take to

a. find an input value that allows you to flip one or more bits in the message? You do not care which bits.
b. find an input value that allows you to change the message to something you want?

d) You may or may not be convinced that a successor for the SHA-1/SHA-2 families is needed. Have a look at the competition now (it is on NIST’s Web site).

a. How is it organized, i. e. how are entrant candidates evaluated?
b. SKEIN is an entrant to the competition. Download v1.3 of the paper at `http://www.schneier.com/skein.pdf`.
Read the abstract and Section 2.1. What are the building blocks (components) of SKEIN?
c. Attacks on practically all algorithms, including SKEIN, have been published in the current round of the competition. What is the currently best attack on SKEIN, and which properties does it attack?

## Task 2: Analysing Protocols

The goal of this task is to let you analyse a cryptographic protocol.
Imagine Alice, Bob and Mallory are students of computer science. At the beginning of the semester, they have all pair-wise exchanged their public keys (securely!). However, while Alice and Bob have become good friends, Mallory is secretly very jealous of Alice and only pretends to be friends with the two. Neither Alice nor Bob realise this and will normally happily communicate with her.
Imagine Alice and Bob meet for a coffee at the end of class. Bob is really happy and tells Alice:
*"I have designed a new and simple cryptographic protocol for all students in the semester. It will allow us to authenticate and then establish a common session key. It's really good, look!"*

(1) B chooses a nonce $N_B$ and sends it to A, plus an encrypted symmetric key $K_B$:
$B \rightarrow A$: $N_B$, $K_{pub,A}(K_B)$

(2) A responds with a nonce $N_A$, a signature and an encrypted symmetric key $K_A$:
$A \rightarrow B$: $N_A, N_B, Sig_A(N_A,N_B,B), K_{pub,B}(K_A)$

(3) B accepts and replies with a new nonce $N_B$':
$B \rightarrow A$: $N_B', N_A, A, Sig_B(N_B', N_A, A)$

Bob continues: *"This ensures the following. When the protocol is complete*

1. *B can be sure that A created message 2 specifically as a response to B's first message. Thus, it must be A with whom B has executed the protocol!*
2. *The other way around, A can be sure that she is communicating with B because only B can create the signature in the third message!*
3. *Furthermore, we have a session key ($K_A||K_B$) that only B and A know and that they can use for the communication in the following session.*"

Alice is not as easily convinced. She knows that authentication & key establishment protocols can be vulnerable in very subtle ways. She takes a good long look at the protocol and then declares: *"I am afraid it's broken. An attacker can inject messages such that A would falsely assume she has run the protocol with B, while in fact she was talking to the attacker. This violates your second claim. The key establishment is also broken; the attacker can learn the session key."*
Bob is down-hearted, so Alice takes pity and explains to him why the protocol is vulnerable. Can you do the same?

a) We begin by attacking the "authentication property", i.e. the second claim.
   Assume that Mallory (M) can control all messages in the network, i. e. read, delete, modify etc. She is only limited by the cryptographic functions, which we assume to be perfect. Mallory will act as a kind of woman-in-the-middle, but will be more active than usual and initiate protocol runs herself.
   Here are hints to guide you through the task:
   a. Recall that Alice and Bob view Mallory as a normal participant, i.e. when contacted by her, they will normally execute the protocol with her.
   b. M will have to execute two concurrent protocol runs: first one with A, and then one with B.
   c. Let M start the protocol runs. Her first message is: M → A: $N_M$
   d. A will answer to this message. Reuse a field from this answer to let M initiate another protocol run, this time with B.
   e. Show how the protocol completes then with the "authentication property" violated: A will believe that she has completed the protocol with B, although it really was M.

b) Which protocol field in which protocol message causes the vulnerability, and why? Change it so the "authentication property" is not violated anymore.

c) Finally, we are going to look at the key establishment. Why is the method that Bob proposes (sending encrypted symmetric keys) also broken? What can be done to fix it? Write down the new secure protocol.

## Task 3: Modifying Kerberos

We have reviewed the Kerberos protocol in depth in the lecture. The goal of this task is to discuss some extensions.

a) Change the Kerberos Protocol so that it does not use timestamps and therefore needs no synchronised clocks.
b) Argue why the Kerberos Protocol does not achieve the property of „Forward Secrecy“.
c) Extend the Kerberos Protocol so that the communication between Alice and the service S1 becomes „forward secure“.

## Task 4: Looking at Some Source Code for RSA

Cryptographic libraries are available for practically every important programming language. Interestingly, the general way you use them is often similar. In this task, we will have a brief look at cryptopp (`http://www.cryptopp.com/`) , a library for C++.

Consider the code snippet for RSA encryption below and answer the following questions.
Hint: the Wiki on the cryptopp homepage will help you with the library-specific questions.

a) Why must the string **`encryptThis`** not exceed a certain length when used with RSA?
b) What is OAEP? What attacks is it meant to prevent?
c) What purpose does the object **`sbbCipherText`** have?
d) Why is the encryption operation initialised with a random number generator?
e) Using cryptopp, give an example how to calculate a SHA-2 hash value of a string “hello”, using the 256 bit version of SHA-2.
Giving the relevant statements is sufficient. We do not test your C++ skills here: your code does not have to compile, nor do you need to correctly include the headers etc. Of course, if you want, you can also send `holz@net.in.tum.de` a working source (also attach a compiled binary then, please).
f) RSA is computationally quite expensive. How is it thus used in practice – e.g. by PGP, but also other tools using RSA?

```
// Code snippet

// Explain this
string encryptThis =
     "Must be shorter than the size of the RSA key minus OAEP bits.";

// Explain this
 RSAES_OAEP_SHA_Encryptor pubkey(
     FileSource("pubkey", true, new Base64Decoder)) );

// Explain this
SecByteBlock sbbCipherText(pubkey.CipherTextLength(encryptThis.size()));

// Explain this
AutoSeededRandomPool rng;
pubkey.Encrypt(
    rng,
    (byte const*) encryptThis.data(),
    encryptThis.size(),
    sbbCipherText.Begin());
```