Esecurity: secure internet & e-passports, summer 2011

Michael Nüsken, Raoul Blankertz

1. Exercise sheet
Hand in solutions until Sunday, 10 April 2011, 23:59

For future exercises it might be important to use b-it computers. So please register an account for the b-it. (Ask at the infodesk for the procedure.)

A word on the exercises. They are important. Of course, you know that. You need 20% of the credits to be admited to the final exam. As an additional motivation, you will get a bonus for the final exam if you earn more than 60% or even more than 80% of the credits. The bonus does not help passing the exam, but if you pass the bonus will increase your mark by up to two thirds.

Exercise 1.1 (Secure email). (6 points)

(i) Send a digitally signed email with the subject $\boxed{4}$

```
[11ss-esec-handin] hello
```

to us at

```
11ss-esec-handin@lists.bit.uni-bonn.de
```

from your personal account. The body of your email must be nonempty and the signature must be verifiable and correct. [It is a good idea to verify this by sending a blind carbon copy (Bcc) to oneself.]

With Thunderbird I recommend using `enigmail` and `gpg`. In any case make sure to register your key eg. at `http://gpg-keyserver.de/`.

Choose yourself among this and possible other solutions. In any case use a pgp key pair.

(ii) Find the fingerprint of your own PGP key. Bring two printouts of it and an identification document to the next tutorial. (Do not send us an email with it. Guess, why!) $\boxed{2}$

Note: Future exercise hand-ins will only be accepted via signed email. Then a bonus point will be awarded for a correct signature and a malus for a missing or invalid signature.

The following cryptographic protocols are already implemented in many programming languages. Choose an environment of your liking. Do not reinvent the wheel. For example, you can download the program `Cryptool` from `http://www.cryptool.de/` for the Windows operating system (a license for educational purposes has been granted).

Note: Those parts of the protocols, that are not fully specified in the instructions of this exercise, are to be chosen by you. Comment your code properly and assign meaningful names to the variables.

Exercise 1.2 (Repetition public key crypto: RSA and RSA-signature). (10 points)

In this exercise we shall simulate the start of an electronic conversation of Alice and Bob, using RSA signatures and subsequent key exchange.

$\boxed{4}$ (i) Create 1024-bit RSA keys for both Alice and Bob.

$\boxed{3}$ (ii) Let Alice compose a short text, hash it, and compute an RSA-signature for it. Let Bob verify the signature.

$\boxed{3}$ (iii) Let Alice generate a random-128 bit string k, which she wants to use as a common key with Bob. Take the necessary steps to make this key known to Bob.

Exercise 1.3 (Repetition symmetric crypto: AES and CBC). (6 points)

Use the 128-bit key k generated in the previous exercise for AES.

$\boxed{3}$ (i) Let Alice send an encrypted meaningful 128-bit message to Bob. Let Bob decrypt the message.

$\boxed{3}$ (ii) Let Bob encrypt a meaningful 512-bit message using Cipher-Block-Chaining Mode (CBC) and your student ID as initialization vector (IV). Let Alice decrypt the message.

Exercise 1.4 (Repetition: hybrid crypto). (6 points)

$\boxed{3}$ (i) How is the idea of a hybrid crypto system implemented in GnuPG? What are the reasons?

$\boxed{3}$ (ii) Read PHONG Q. NGUYEN, *Can We Trust Cryptographic Software? Cryptographic Flaws in GNU Privacy Guard v1.2.3*. How does the used implementation for RSA differ from the textbook version? What are the consequences?