Advanced cryptography: Symmetric primitives, winter 2014/15

MICHAEL NÜSKEN

1. Exercise sheet
Hand in solutions until Monday, 27 October 2014, 13:59

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 50% of the credits to be admitted to the final exam. As an additional motivation, you will get a bonus for the final exam if you earn more than 70% or even more than 90% 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). (4 points)

(i) Send a digitally signed email with the subject [2]

`[13ws-ac] hello`

to me at

`nuesken@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 at `http://pgp.mit.edu/`.

Choose yourself among this solution and possible others. 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 an email with it. Guess, why!) [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, `openssl`, a UNIX command line tool, or `Cryptool`, which you can download from `http://www.cryptool.de/` (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). (6 points)

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

2 (i) Create a 2048-bit RSA keys for both ALICE and BOB.

2 (ii) Let ALICE compose a short text, hash it, and compute an RSA-signature for it. Let BOB verify the signature.

2 (iii) Let ALICE generate a random-128 bit string k, which she wants to use as a common key with BOB. Let ALICE encrypt k and send it to BOB.

Exercise 1.3 (Repetition symmetric crypto: AES and CBC). (4 points)

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

2 (i) Let ALICE send an encrypted meaningful 128-bit message to BOB. Let BOB decrypt the message.

2 (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.