University of Applied Sciences Darmstadt
Department of Computer Science
Nicolas Buchmann and Lisa Schehlmann (Harald Baier)

Winter Term 2015/2016
2014-11-05

Security Protocols and Infrastructures, Lab 2

Part 1 (Diffie-Hellman Key Exchange)

(a) Find the smallest prime p with $p \geq 300$.

(b) Implement a function in `C, C++, C#, Java or Python` to compute the public key A of Alice: Input is (p, g, a), where a denotes Alice's private key. You may assume to work on unsigned machine data types (e.g. unsigned integers).

(c) Compute the following public keys, if the prime from part (a) and the generator $g = 5$ is used:

- Alice chooses $a = 43$ as her secret key.
- Bob chooses $b = 72$ as his secret key.

(d) Compute the common secret of Alice and Bob, where SHA-1 is used as a key derivation function (KDF) as follows: Let $K \equiv g^{ab} \bmod p$ and c be the corresponding ASCII-character of the numeric value K. Then KDF(K) is equal to SHA-1(c).

(e) Imagine you are sniffing the communication between Carl and David. You are getting the domain parameters p and $g = 5$ used by Alice and Bob. Additionally, the individual public keys of Carl and David are $C = 67$ and $D = 172$, respectively. Write a function to compute both private keys c and d (of Carl and David) and the common secret K.

Give a statement on the efficiency of your algorithm.

(f) (Optional) Make use of your implementation to solve the following DLP for Alice's private key: $p = 2148532933$, $g = 1001$, $A = 1992854757$. Take care of overflows in the data types of your programme.

Part 2 (RSA in `openssl`)

This exercise requires that you have access to `openssl`. You shall find the appropriate `openssl`-commands to solve the following tasks.

(a) Generate an RSA key pair with a 2048-bit modulus, that is the bit-length of n is 2048. Please store the key pair in a file named `keyfile.pem`. The key file must be encrypted using 256bit AES.

(b) Use the command `less keyfile.pem` to view the encoded key pair. Explain, which encoding is used.

(c) Print all components of both the public and the private key to standard output using the `openssl` command and explain their meaning.

(d) Export the public key from file `keyfile.pem` to the file `keyfile_pub.pem`. Show and explain the different components saved in `keyfile_pub.pem`.

(e) Convert the private keyfile `keyfile.pem` to a DER encoded version, which you store in `keyfile.der`. Do not use encryption for this file. Compare the length of the two files `keyfile.pem` and `keyfile.der` and explain the reason.

(f) Which of the key parts are actually encoded within the file `keyfile.der` and which are computed on the fly by the `openssl` tool?

(g) Sign the key file `keyfile.der` using SHA-512 as hash algorithm and your private RSA key. What is the length of the signature? Do you see a relation between the signature length and your RSA parameters?