Advanced Computer Security, Summary for the Exam

Timo Karvi

12.2010

Modular Arithmetics

You should

- be able to prove the basic properties of modular arithmetics,
- understand and be able to construct inverse elements with respect to addition and multiplication.
- understand the concept of the primitive root and be able to find small primitive roots.

It is not necessary to remember the extended Euclidean algorithm. Pocket calculators are not needed.

Finite Fields

You should

- be able to construct finite fields of the form $GF(p^n)$,
- and to find irreducible polynomials of small degree.

Basic Problems

You should

- be able to explain the factorization problem and it relation to cryptography
- as well as the discrete logarithm problem.

AES

You should

- know the basic logic of AES and
- the functioning of SubBytes, ShiftRows and MixColumns operations.

However, it is not necessary to remember the S-box or the multiplication matrix of MixColumns.

RSA I

- You should be able to explain generally, how public and private keys are defined.
- Moreover, it is expected that you can construct public and private keys concretely when n is small.
- You should be able to explain the encryption and decryption procedures.
- It is not necessary to remember the theorems which show how RSA can be broken if some bits of the secret keys are known or if private keys are too small.
- You should be able to explain the problem of short plaintexts and how this problem is solved in practice (including OAEP).
- Remember the side channel attack against RSA!
- It is not necessary to remember the algorithm to calculate powers or requirements for the parameters.

RSA II

- Remember how digital signatures are done using RSA. Remember also the useless attack to forge signatures.

DH

- You should be able to explain the basic Diffie-Hellman method to generate keys.
- Also the man-in-the-middle attack against the basic DH.

Needham-Schroeder with a Server

- You should be able to analyze the key generating protocol of Needham and Schoeder. This means that you can find its weak points. You can explain why various parameters are needed and what extra parameters should be added so that the weakness disappears.
- You should know the concepts of forward and partial forward secrecy and resistance to key compromise impersonation. Also you can analyse some simple cases if they satisfy these concepts.

Attacks

- You should be able to explain the various attack possibilities against key agreement or security protocols.
- For reflection and typing attacks, also examples.
- You should know the possibility of certificate manipulation, but it is not necessary to remember the example.

Design Principles

- It may be possible that you have to write an essay about the design principles for cryptographic protocols.
- The same with the robust principles of public key cryptography.

Protocols

[allowframebreaks]

- Remember Bellare-Rogaway and the attack against it.
- Remember the ISO/IEC 9798 protocol 4. A simple security analysis is expected.
- It is not necessary to remember by heart Andrew's Secure RPC Protocol, but you should be able to analyse it and find its flaw, if ther protocol is shown to you.
- Similarly with Burrow's modification.
- You can explain and analyse Boyd's protocol.
- You should know the Denning-Sacco improvement of Needham-Schroeder.
- You should know the ISO/IEC 11770-3 protocols and be able to explain their differences.
- Not necessary to remember the public key version of Needham-Schroeder, but you should be able to analyse it if it is shown to you.
- The same with Station-to Station Protocol.

Group Protocols

- Practical requirements.
- GDH.2 should be known in such a way that you can even simulate it. Not necessary to remember GDH.1 or 3.
- Remember BD with broadcasts. Not necessary to remember BD without broadcasts by heart, but you should be able to simulate it, if it is shown to you.
- The concept of the key tree, calculation principles for the blinded and private keys, join and leave operations should be known.
- Authenticated GDH.2 should be known. Not necessary to know SA-GDH.2 by heart, but you should be able to simulate it, if it is shown to you.
- Some ideas about the performance of the various protocols.

HIP

- The basic idea of HIP (using identities instead of IP addresses).
- You should be able to analyse the Base Exchange Protocol, if it is shown to you.
- Trust relations in different situations (exercise 5.4).
- The use of hash chains.