Lab: Tools of the Trade

A lab on the practicalities of security

Daniel Bosk

February 28, 2020

1 Introduction

The main purpose of security is that it should be used in applications and systems produced. This laboratory assignment focuses on the practical parts of security. You have probably read a lot about different cryptographic mechanisms, e.g. AES [1] and CBC [2]. You know that you should use huge prime numbers in RSA [5], but how do you actually choose these in practice? In this lab you are supposed to answer questions such as this and get to know some implementations of what you previously only knew in theory.

1.1 Scope and Aim

The main idea of the assignment is that the participants choose different practical issues, find solutions to them, and finally present them for the class. This way everyone will get knowledge of a variety of practical problems facing developers.

The intended learning outcomes are as follows, after completion of this assignment you will be able to:

- *find* and *apply* available implementations of security mechanisms, e.g. from RFCs or third-party libraries.

The next section covers what you must read before you understand this assignment and how to do the work. Section 3 covers the work to be done, i.e. how you should learn this. Section 4 covers how it will be examined, i.e. how you show that you have fulfilled the intended learning outcomes given above.

2 Theory

Before starting this assignment you must have a wide grasp of the theory of security. If you do not, then you will not know of all available mechanisms. Hence you will neither know of all practicalities you will have to solve to use these as a developer.

3 Assignment

Now, you should come up with a question related to practice which you would like to have an answer for. There are a lot of questions of this type, for example:

- In a Diffie-Hellman key exchange you need a generator for a group, how do you find this one?
- How do you choose the RSA prime factors when generating a key?
- There is no randomness in a computer program — since these are fully deterministic — but how do we then get randomness to do cryptography using a computer?
- How do you actually use SHA-256 [3] or bcrypt [4] to protect a password: how should you use them, what values should you use?
- How do you use anonymous credentials (Identity Mixer) on HyperLedger Fabric?
- How do you use a Trusted Execution Environment, *e.g.*, Intel SGX enclaves? What can you do with it?

These are just examples, feel free to pick other questions. Please discuss possible questions with the course tutor. Remember: the questions must be oriented towards solving problems in practice. Once you have settled for a question, post it in the course forum. This way no one else will try to find the answer to the same question.

Now you will go find the answer of the question. Since this is a problem about practice, this means someone has already solved the problem. Thus, a good place to look is probably in a related implementation, especially in its documentation. Here you can see how someone solved this problem, you can see references to standards documenting how to do it — which is a preferable source for information.

When you have solved the problem, you should prepare a presentation for the class. This presentation should contain at least the following:

- What the question is.
- Why this is an interesting question.
- What the answer to the question is.
- What you have to support your claims (preferably references to standards, RFCs and software library documentation).
- A short usage example (i.e. a demo program).

4 Examination

As you will prepare a presentation, this will be presented for the class (check the course schedule for the date of this presentation). You are required to have some slides to make your presentation more comprehensible. Your presentation should be at most 15 minutes long, and it must have some technical depth — in particular, you must motivate your findings. After the presentation, you must make your presentation available for others for future reference.

Acknowledgements

References

[1] Joan Daemen and Vincent Rijmen. "The block cipher Rijndael". In: *Smart Card Research and Applications*. Springer. 2000, pp. 277–284.

[2] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*. Special Publication 800-38A. National Institute of Standards and Technology, 2001. URL: `http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf`.

[3] Information Technology Laboratory. *Secure Hash Standard (SHS)*. Federal Information Processing Standards Publication 180-4. National Institute of Standards and Technology, Mar. 2012. URL: `http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf`.

[4] Niels Provos and David Mazières. "A Future-Adaptable Password Scheme". In: *Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference*. 1999. URL: `https://www.usenix.org/conference/1999-usenix-annual-technical-conference/future-adaptable-password-scheme`.

[5] Ronald L Rivest, Adi Shamir, and Len Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126. URL: `https://dl.acm.org/citation.cfm?id=359342`.