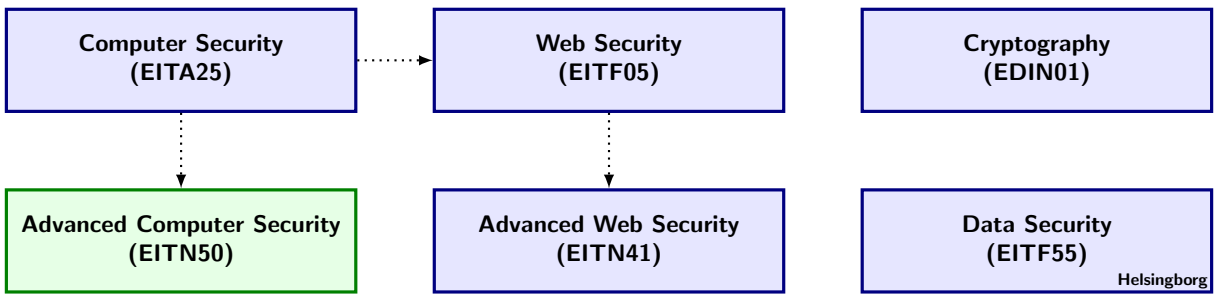
Advanced Computer Security 2019

Department of Electrical and Information Technology
Lund University

Project: Object Security

Learning goals:

- Understand how to implement the main security properties of secure channel.
- Understand the specific nature of a connection using object security.
- Get an understanding study and implement forward security.



Preparations

- Read the paper by Selander et al on object security for IoT,
- (`http://www.eit.lth.se/fileadmin/eit/courses/eitn50/Project_ObjSec/protected/Application_Layer_Security_Protocols_for_IoT.pdf`),
- Read about forward security, for example `https://scotthelme.co.uk/perfect-forward-secrecy/`.
- Read the entire project description (this document) before you begin.

REMARK: The paper of Selander et all. is a non-published paper which we may use for this course and access to the paper is protected. You should enter user: `EITN50` and as password you should use the same password that is used for the course lecture slides.

Instructions for Project Approval

The project consists of a number of assignments that guide your work and you should use the assignment numbers to structure your report. The assignments are of type A or of type B. You must do the A AND B type assignments if you aim for a grade 4 for this course. The A type assignments are mandatory.

- Indicate on the front page your group number in addition to your name(s).
- In the report give a brief architecture overview of your implementation and chosen approach.
- Document your work with logs and printouts.
- The code you deliver should include all non-standard dependencies on libraries.
- You should submit the report electronically in pdf format. Give the file the following name: "adsecxy_projectB.pdf", where xy is the number of your group and use the subject "EITN50" in the email that contains the report. Send it to `ben.smeets.lu@analys.urkund.se`.

1 Instruction

TLS and IPsec are protocols to provide a secure connection between two communicating entities. These protocols are very popular. One thing these two protocols have in common is that they are session based. For many use cases this is not a problem and even fit naturally with the nature of the application, e.g. VPN or secure connection to your bank server. However, as explained by Mattsson and Selander [1], these session based protocols are not always a good solution. Particularly for IoT devices one works now on standardizing an alternative called Object security.

The paper of Selander et all [2] gives you a technical description how things are organized in a solution for IoT devices. You should not (even try to) implement the protocols of this paper but you should use the paper as an advanced example. The solution you have to implement can be - should be - much simpler but yet capture the main ideas of an communication protocol that implements object security principles.

Assignment 1

In this project you have to implement a proof-of-concept implementation of a secure connection for two parties that should fulfill the following: your solution should

1. [A] work on the principle of object security,
2. [A] provide integrity, confidentiality, and replay protection,
3. [A] use UDP as the way to exchange data between the two parties (sending and receiving party),
4. [A] work on the principle of forward security,
5. [A] should have at least two distinct parts; handshake and (protected) data exchange,
6. [B] report should contain a sequence diagram of the protocol parts (using PLANTUML),
7. [A] actually work when we test it. The data packets should by small as one can expect for small IoT devices, say max 64 bytes,
8. [A] document and explain the design choices for your implementation,
9. [B] test with an intermediate party that acts as a cache that your receiving party can later pickup the objects from the intermediatory party.

The code should be

- documented(a listing of the code and a log print when you ran the code) in an appendix of project report
- and should be provided in source code using ordinary text files from which we can build the programs of the involved parties, each for each communicating party.

Use either Java or Python and in case you use non-default libraries/components your source code delivery should provide all code and instructions needed to build and run the programs. Avoid the use of many libraries, keep it simple. For the crypto functions that you need, you can use libraries mentioned below.

Once more, reports that do not include/satisfy the A marked items mentioned before are returned and put as 'pending' until sufficiently completed.

Examples of useful cryptography libraries are

- BouncyCastle when you program in Java, and
- PyCrypto or cryptography when you program in python.

Sequence diagrams can be very useful to explain your protocol. They can be neatly generated by using the PLANTUML program, `http://plantuml.com/sequence-diagram`. The simplest form is using this program from the command line, i.e., `java -jar plantuml.jar file1` which produces a png file with the sequence diagram. Via the aforementioned link you have examples and documentation how to use this program.

HINT: Make your solution first work without an intermediate party and then think how you modify your protocol to make it working with the caching setup.

References

[1] J Mattsson, G Selander, Object Security in Web of Things, 2014, W3 Org.

[2] G Selander, F Palombini, J Mattsson, L Seitz, Application Layer Security for the Internet of Things, unpublished.