02244 Logic for Security
Project on Security Protocols

- Hand-out: Feb 9, 2026
  Hand-in: via DTU Learn until Mar 16, 2026 **noon**
- We allow to work and hand-in in groups of up to 3 students.

  Each report must indicate which students are part of the group.

  The reports must be divided into sections, and each section must have one group member designated as author. This should reflect a fair distribution in report writing among the group members. Any section without such a marking of one single author will count as not submitted.
- The report must indicate which resources have been used to perform the work. This includes text books, research papers, information found on the web, the use of any AI tools, detailed suggestions from teachers, and results of discussions or cooperation with other students.
- Page Limit: 15 pages
- Along with your report, you have to submit the AnB files and the your lab logbook (see below). These are separate files and do not count into the page limit of the report.
- Presentation: In the meeting on Mar 16, we ask groups to present the protocols they have designed, so we can discuss the different solutions, common mistakes, and insights gained.

Open Authorization

Scenario: A has some online resources, including an online storage of photos taken on holidays at a server B. A now wants to make a photo book for friends and use a photo book service P to print such book from a selection of photos. Instead of downloading the photos from B and uploading them to P, A wants to give P *authorization* that P can access A's photos at B.

Of course, A does not want to give P access to other resources like A's email box, and only read-access to the photos (maybe even just a particular folder at server B).

A does not have a public/private key pair, but uses an identity provider (similar for instance to MitID) to authenticate herself and authorize a party like P to access a resource like B on her behalf. For this reason, A shall have a password that is a shared secret with the identity provider, but this password is not a good cryptographic secret, so protocol must not use it as an encryption key. Everybody **except** A has a public/private key pair, and everybody **including** A knows the public key of the identity provider. The identity provider knows all public keys.

We assume that the identity provider is an honest agent, and all parties trust the identity provider, i.e., any statements signed by the identity provider (e.g., what the public key of some agent is) is accepted by other parties as the truth.

Development & Lab Logbook The protocol should be gradually developed, where you may start with a very simplified "bare bones" version and refine and improve it during the course, using the content from each week. There is a list of suggestions below for each week.

It is required to make a lab logbook with the documentation of this development, namely a "snapshot" of the protocol in each development step, including versions where an attack is discovered against the protocol or something is not yet working. Together with it, have a short description that notes the most important points about this version:

- What has changed to previous version? What's the idea?
- What are modeling considerations? (E.g., assumptions and goals)
- What are problems (E.g., attacks found, things that could not be modeled, excessive runtime of OFMC...)
- Discussions/thoughts of group members, or input from the teachers.

The logbook has to be delivered as an attachment of your report, so it is mandatory to do; besides that, if you do this properly it can make writing the report so much easier, because you can directly write it as a story that tells why exactly you designed it in this a particular way. The report should outline this development (rather than just showing the final protocol).

Week 2: Dolev and Yao Make an **informal** outline, i.e., try to describe how you want to design the protocol in natural language or with diagrams without considering OFMC/AnB. The reason is that in a first step you should not be limited by what OFMC can do and what can be expressed in AnB.

Try a first version in AnB. You may make some simplifications, e.g., also the public keys (so you do not need an identity provider). Think how the data on B that P can access can be modeled in an easy way.

Special task of the week: make a static analysis: given that the protocol run between honest agents and the intruder only observes the network traffic but does not send messages, what can the intruder find out according to the Dolev-Yao model?

Week 3: Lazy Intruder Make the AnB protocol more comprehensive: A must not have initially a secure key but rely on the identity provider to authenticate her or her requests; note that A has a shared secret password with the identity provider and knows the public key of the identity provider. (One should not use the password for encryption.) For simplicity you may assume that A initially knows the public keys of everybody.

Special task of the week: Use the lazy intruder technique from the lecture to show that the role of P is executable, i.e., instantiating P with the intruder and all other roles with honest agents, show that the intruder can play role P (i.e., send every outgoing message, when knowing all previous incoming messages). It is required that you follow exactly the lazy intruder method from the lecture and denote the sequence of steps.

Week 4: Typing and Implementation Make use of formats as shown in the lecture to replace all concatenations and make make sure that the protocol is type-flaw resistant as defined in the lecture.

Moreover, this week we should get rid of the requirement that A initially knows the public key of everybody. But, to make the protocol not more difficult, there should be a *separate* protocol where A asks the identity provider about the public key of another party. (A of course initially knows the public key of the identity provider.)

Special task of the week: Show that the protocols are type-flaw resistant, i.e., that for any two messages in SMP of the protocols that are not themselves variables, if they have a unifier, they have the same type.

Week 5: Channels

Special task of the week: In this week we want to use TLS channels where only the server side is authenticated. As indicated in the lecture, we can use the pseudonymous channel notation of OFMC to model transmissions over a TLS channel (although pseudonymous channels do not ensure freshness of the transmitted message). Make a variant of your protocol that replaces as much cryptography as possible by the pseudonymous channel.

Week 6: Privacy This week is reserved for catching up and writing on the report.

Special task of the week: verify that your protocol is secure even if the password between A and identity provider is a guessable secret.

Week 7: Abstraction Please present your solution in class this week!

Hand-in:

- Report with individualized sections as explained above it. It should include
  - Description the development of your protocol, including any interesting attacks that you have run into using OFMC.
  - Description of the final protocol in AnB, including explanation of initial knowledge, goals.
  - Description of any assumptions/simplifications you have made, or remaining problems.
  - The answer to the special task for each week.
- The Lab Logbook
- The AnB file(s)