02244 Logic for Security
Project on Information Flow

- Hand-out: March 18, 2024
- Hand-in: via DTU Learn until May 6, 2024 **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
- You may submit source codes as additional files, but the report itself should covey all important points about your solution without having to read your source codes.
- Presentation: In the meeting on May 6th, we ask groups to present the protocols they have designed, so we can discuss the different solutions, common mistakes, and insights gained.

The Setting

Pretend that you are working for a fictitious startup IT company called *Online Auctions Solutions A/S*, or *OAS* for short. They plan to provide a software for small auction houses to augment their traditional auctions with online bidding.

Since some objects trade at a substantial price, it is important that the system is secure and that OAS can convince auction houses that it is secure. For this reason, the system shall be delivered to auction houses with source code that includes an information flow policy, i.e., a lattice of security labels and annotations of all program variables with such security labels, and a proof that the source code never violates the information flow policy.

Commission Bids

A particular use case are commission bids: this is when a customer does not want to participate live during the auction, but set a maximum beforehand, and the auction

house bids “in commission” up to this maximum on behalf of the customer during the auction.

For instance, given there is an item with a starting price of 500 Kr, and there is a commission bid from bidder A for 500 Kr, a commission from bidder B for 700 Kr, and we have a bidder C who is live at the auction, than the bidding might look like this, if raises are in steps of 50 Kr:

```
start: auction house bids for B: 550 Kr
C bids 600 Kr
auction house bids for B: 650 Kr
C bids 700 Kr
auction house bids for B: 700 Kr
C bids 750 Kr
going once ... going twice ... sold
```

Notice that the starting price is 550 Kr, because of the bids from both A and B before the auction started: B must bid at least 550 Kr to outbid A. Now C bids 600 Kr, and so on behalf of B, the auction house bids 650 Kr. C bids 700 Kr and so it gets now a bit tricky, because B has also bid 700 Kr–in commission. Since B’s bid was given before C’s bid, the auction house now bids 700 Kr on behalf of B, in a sense “overriding” C’s bid. Even though this may appear a bit strange, this is the common way auction houses usually handle this situation. Note that it in fact tells to C that the maximal bid of B is indeed 700 Kr (because, if B’s bid were higher, then the auction house would have bid 750 Kr for B). Finally, C bids 750 Kr, outbidding B, and since nobody else bids, C is the winner.

It is crucial that, before the auction, no customer should learn about the existence of any bids other than their own, and even during the auction one does not learn more then the bids that have been placed by others, in particular not what the maximum bid of any customer is (except in the special case described above, where a live party during bidding hits exactly the amount of a commission bid). Moreover, the bidders should all be pseudonymous to each other: each bidder has a unique bidder number and one can observe only the bidder number of each bid. This allows of course to see which bids come from the same bidder, but nobody except the auction house learns more about the identity of a bidder.

Reputation System

Another use case is reputation systems. Generally, auction houses may have a policy that a first-time customer has a limit on the value they can bid, because one would like to avoid the legal hassle if a bidder actually cannot pay the items they have won (or if they have even given a false name upon registration). Of course, returning customers may have accordingly higher limits. Besides that, auction houses may accept references from other auction houses: if A is considered a “good customer” at auction house B, but is new to auction house C, then A may use auction house B as a reference when registering with C to also get “good customer” status at C immediately. This requires of course that C trusts B in this matter, and that both A and B are actually willing to share this information with C.

Your Task

The management of the OAS company is convinced that the method to make the program information flow secure is the approach of Myers and Liskov:

- A. C. Myers, B. Liskov: *A Decentralized Model for Information Flow Control*, ACM Symposium on Operating Systems Principles (pp 129-142), ACM Press, 1997.

It should be emphasized that the management is really convinced of this and will not accept any alternative models that you might come up with. So please make sure that you follow Myers-Liskov!

Your task is to develop a *mockup* for this new system, where mockup means: we are interested in a concept how the software in principle works, we do *not* care about any user interfaces/GUI and security protocols between entities. Thus, it is a single (not-distributed) program where all inputs are given as

- program variables that are initially fixed with example values
- function parameters (and the function called with example values)
- or as command line parameters

and similarly, all outputs of the program are

- written into program variables
- are return arguments of functions
- or written on the screen

Also combinations of these options are allowed.

What is crucial are the following items:

- A definition of a lattice of security labels in the style of Myers and Liskov. This in turn requires a definition of the actors/roles in the program.
- The labeling of all inputs, outputs, and variables of the program with such labels.
- Any declassifications needed, in the style of Myers and Liskov.
- An information flow analysis of the program, i.e., that there are no information flows that the security policy forbids.

Management expects you to start with a rather simple design and enlarge it as you go. Quality is much more important than quantity: it is crucial that whatever the mockup supports is done well and formally verified, it is much less important that it covers a lot of scenarios and features. For the programming language, the management of OAS allows Java or Python.

The Report

The work must be documented in a report, again maximal 15 page, that documents the following:

- A description of the participants/roles in the system.
- A description of the security goals that you want to ensure, both confidentiality and integrity.
- Your design choices and reasoning, in particular the design of the security lattice and how this actually ensures your security goals.
- The justification for all declassifications.
- An excerpt of the information flow analysis for your code, maybe demonstrating it for the most interesting (for information flow) piece of code.

This is an individualized group report, i.e., the report has to be partitioned into sections/subsections such that **each part has a unique author** and the report must that is clearly denote the authorship.

Each report must indicate which resources have been used to perform the work. This includes text books, research papers, information found on the web, detailed suggestions from teachers, and results of discussions or discussions with other students not part of the group.