Assignment 5
Proof-Carrying Authorization
Sample Solution

15-316: Software Foundations of Security & Privacy
Frank Pfenning

Due **Wednesday**, November 20, 2024
85 points

Your solution should be handed in as a file `hw5.pdf` to Gradescope. If at all possible, write your solutions in LaTeX. The handout `hw5-pca.zip` includes the LaTeX sources for some lectures and the necessary style files which provide some examples for rules, derivations, and proofs. Because we are posting it on Wednesday, it is also due on Wednesday. You may use up to two late days as usual.

1 Authorization Logic [25 points]

For each of the following, either give a derivation in the sequent calculus or give a counterexample. In the latter case, use p and q as atomic formulas and demonstrate that the counterexample is not derivable.

Task 1 (5 pts) $(A \textbf{ says } P) \wedge (A \textbf{ says } Q) \vdash A \textbf{ says } (P \wedge Q)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P, Q \vdash P}\,\mathsf{id} \quad \dfrac{}{P, Q \vdash Q}\,\mathsf{id}}{P, Q \vdash P \wedge Q}\,\wedge R}{P, Q \vdash A \textbf{ aff } (P \wedge Q)}\,\textbf{aff}}{P, A \textbf{ says } Q \vdash A \textbf{ aff } (P \wedge Q)}\,\textbf{says}L}{A \textbf{ says } P, A \textbf{ says } Q \vdash A \textbf{ aff } (P \wedge Q)}\,\textbf{says}L}{(A \textbf{ says } P) \wedge (A \textbf{ says } Q) \vdash A \textbf{ aff } (P \wedge Q)}\,\wedge L}{(A \textbf{ says } P) \wedge (A \textbf{ says } Q) \vdash A \textbf{ says } (P \wedge Q)}\,\textbf{says}R$$

Task 2 (5 pts) $A \textbf{ says } (P \wedge Q) \vdash (A \textbf{ says } P) \wedge (A \textbf{ says } Q)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P, Q \vdash P}\,\mathsf{id}}{P \wedge Q \vdash P}\,{\wedge}L}{P \wedge Q \vdash A\ \mathbf{aff}\ P}\,\mathbf{aff}}{A\ \mathbf{says}\ (P \wedge Q) \vdash A\ \mathbf{aff}\ P}\,\mathbf{says}L}{A\ \mathbf{says}\ (P \wedge Q) \vdash A\ \mathbf{says}\ P}\,\mathbf{says}R \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P, Q \vdash Q}\,\mathsf{id}}{P \wedge Q \vdash Q}\,{\wedge}L}{P \wedge Q \vdash A\ \mathbf{aff}\ Q}\,\mathbf{aff}}{A\ \mathbf{says}\ (P \wedge Q) \vdash A\ \mathbf{aff}\ Q}\,\mathbf{says}L}{A\ \mathbf{says}\ (P \wedge Q) \vdash A\ \mathbf{says}\ Q}\,\mathbf{says}R}{A\ \mathbf{says}\ (P \wedge Q) \vdash (A\ \mathbf{says}\ P) \wedge (A\ \mathbf{says}\ Q)}\,{\wedge}R$$

Task 3 (5 pts) $(A\ \mathbf{says}\ P) \vee (A\ \mathbf{says}\ Q) \vdash A\ \mathbf{says}\ (P \vee Q)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P \vdash P}\,\mathsf{id}}{P \vdash P \vee Q}\,{\vee}L_1}{P \vdash A\ \mathbf{aff}\ (P \vee Q)}\,\mathbf{aff}}{A\ \mathbf{says}\ P \vdash A\ \mathbf{aff}\ (P \vee Q)}\,\mathbf{says}L}{A\ \mathbf{says}\ P \vdash A\ \mathbf{says}\ (P \vee Q)}\,\mathbf{says}R \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Q \vdash Q}\,\mathsf{id}}{Q \vdash P \vee Q}\,{\vee}L_2}{Q \vdash A\ \mathbf{aff}\ (P \vee Q)}\,\mathbf{aff}}{A\ \mathbf{says}\ Q \vdash A\ \mathbf{aff}\ (P \vee Q)}\,\mathbf{says}L}{A\ \mathbf{says}\ Q \vdash A\ \mathbf{says}\ (P \vee Q)}\,\mathbf{says}R}{(A\ \mathbf{says}\ P) \vee (A\ \mathbf{says}\ Q) \vdash A\ \mathbf{says}\ (P \vee Q)}\,{\vee}L$$

Task 4 (5 pts) $A\ \mathbf{says}\ (P \vee Q) \vdash (A\ \mathbf{says}\ P) \vee (A\ \mathbf{says}\ Q)$

This is not derivable. We demonstrate this using two concrete atomic formulas p and q. Because $\mathbf{says}L$ is not applicable, the only possibilities are ${\vee}R_1$ and ${\vee}R_2$. We show one; the other is symmetric. Each step below is either the only possible one, or uses an invertible rule. Since $q \vdash p$ is not derivable, the original sequent at the bottom is also not derivable.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\,\mathsf{id}}{p \vdash A\ \mathbf{aff}\ p}\,\mathbf{aff} \qquad \dfrac{\overset{\mathsf{XXX}}{q \vdash p}}{q \vdash A\ \mathbf{aff}\ p}\,\mathbf{aff}}{p \vee q \vdash A\ \mathbf{aff}\ p}\,{\vee}L}{A\ \mathbf{says}\ (p \vee q) \vdash A\ \mathbf{aff}\ p}\,\mathbf{says}L}{A\ \mathbf{says}\ (p \vee q) \vdash A\ \mathbf{says}\ p}\,\mathbf{says}R}{A\ \mathbf{says}\ (p \vee q) \vdash (A\ \mathbf{says}\ p) \vee (A\ \mathbf{says}\ q)}\,{\vee}R_1$$

Task 5 (5 pts) $A\ \mathbf{says}\ \forall x.\, P(x) \vdash \forall x.\, A\ \mathbf{says}\ P(x)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P(y) \vdash P(y)}\ \mathsf{id}}{\forall x.\, P(x) \vdash P(y)}\ \forall L}{\forall x.\, P(x) \vdash A\ \mathbf{aff}\ P(y)}\ \mathbf{aff}}{A\ \mathbf{says}\ \forall x.\, P(x) \vdash A\ \mathbf{aff}\ P(y)}\ \mathbf{says}L}{A\ \mathbf{says}\ \forall x.\, P(x) \vdash A\ \mathbf{says}\ P(y)}\ \mathbf{says}R}{A\ \mathbf{says}\ \forall x.\, P(x) \vdash \forall x.\, A\ \mathbf{says}\ P(x)}\ \forall R^y$$

2 Trust [30 points]

We informally define that A **trusts** B if whenever B affirms P then A also affirms P. Unfortunately, we cannot express this directly in our authorization logic because the right-hand side of the definition

$$A\ \mathbf{trusts}\ B \triangleq \forall P.\, B\ \mathbf{says}\ P \to A\ \mathbf{says}\ P$$

quantifies over all possible propositions. In authorization logic, quantifiers only range over principals and constants from some finite domain (like rooms or files or homework assignments, etc.).

Instead we assume that there is a *fixed preorder of principals* where $A \leq B$ means that A trusts B. We assume that this relation is reflexive and transitive.

Task 6 (15 pts) Define rules in the sequent calculus for authorization logic that incorporate the trust relationship. Your premises may directly refer to $A \leq B$ for principals A and B. You may modify existing rules $\mathbf{says}R$, $\mathbf{says}L$, and $\mathbf{aff}$, and you may add new rules. Other inference rules should remain unchanged.

We choose to modify the existing rules, appealing to $A \leq B$. Since the right rule is invertible, it should not involve any choice so we stay within A's perspective. The left rule cannot always be applied, and so involves a choice. If A trusts B, then A accepts everything that B says as true.

$$\dfrac{\Gamma \vdash A\ \mathbf{aff}\ P}{\Gamma \vdash A\ \mathbf{says}\ P}\ \mathbf{says}R \qquad \dfrac{A \leq B \quad \Gamma, P \vdash A\ \mathbf{aff}\ Q}{\Gamma, B\ \mathbf{says}\ P \vdash A\ \mathbf{aff}\ Q}\ \mathbf{says}L$$

$$\dfrac{\Gamma \vdash P}{\Gamma \vdash A\ \mathbf{aff}\ P}\ \mathbf{aff}$$

We could also (slightly redundantly, I believe but have not proved) change the right rule, which is also a perfectly good solution.

$$\dfrac{A \leq B \quad \Gamma \vdash B\ \mathbf{aff}\ P}{\Gamma \vdash A\ \mathbf{says}\ P}\ \mathbf{says}R$$

Task 7 (5 pts) Prove $A\ \mathbf{says}\ P \wedge B\ \mathbf{says}\ Q \vdash A\ \mathbf{says}\ (P \wedge Q)$ in your system, provided $A \leq B$.

$$
\dfrac{
\dfrac{}{A \leq A}
\qquad
\dfrac{
A \leq B
\qquad
\dfrac{
\dfrac{
\dfrac{}{P, Q \vdash P}\,\textsf{id}
\quad
\dfrac{}{P, Q \vdash Q}\,\textsf{id}
}{P, Q \vdash P \wedge Q}\,{\wedge}R
}{P, Q \vdash A\ \mathbf{aff}\ (P \wedge Q)}\,\mathbf{aff}
}{P, B\ \mathbf{says}\ Q \vdash A\ \mathbf{aff}\ (P \wedge Q)}\,\mathbf{says}L
}{
\dfrac{
\dfrac{A\ \mathbf{says}\ P, B\ \mathbf{says}\ Q \vdash A\ \mathbf{aff}\ (P \wedge Q)}{A\ \mathbf{says}\ P \wedge B\ \mathbf{says}\ Q \vdash A\ \mathbf{aff}\ (P \wedge Q)}\,{\wedge}L
}{A\ \mathbf{says}\ P \wedge B\ \mathbf{says}\ Q \vdash A\ \mathbf{says}\ (P \wedge Q)}\,\mathbf{says}R
}\,\mathbf{says}L
$$

Task 8 (10 pts) Give a counterexample to A **says** $P \wedge B$ **says** $Q \vdash B$ **says** $(P \wedge Q)$ provided $B \not\leq A$. You should be able to use your rules to prove that the counterexample cannot be derived. If not, explain for partial credit why it is difficult or impossible.

The counterexample just uses atomic formulas p and q and shows that the proposed formula is not derivable. First, we apply rules that are invertible, or the only application ones.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{B \leq B}
\qquad
\begin{array}{c}\vdots \\ A\ \mathbf{says}\ p, q \vdash B\ \mathbf{aff}\ (p \wedge q)\end{array}
}{A\ \mathbf{says}\ p, B\ \mathbf{says}\ q \vdash B\ \mathbf{aff}\ (p \wedge q)}\,\mathbf{says}L
}{A\ \mathbf{says}\ p \wedge B\ \mathbf{says}\ q \vdash B\ \mathbf{aff}\ (p \wedge q)}\,{\wedge}L
}{A\ \mathbf{says}\ p \wedge B\ \mathbf{says}\ q \vdash B\ \mathbf{says}\ (p \wedge q)}\,\mathbf{says}R
$$

At this point, **says**L is not applicable since B does not trust A (by assumption $B \not\leq A$). So only the **aff** rule applies, and we fail after a few more steps.

$$
\dfrac{
\dfrac{
\begin{array}{c}\textsf{XXX} \\ A\ \mathbf{says}\ p, q \vdash p\end{array}
\qquad
\dfrac{}{A\ \mathbf{says}\ p, q \vdash q}\,\textsf{id}
}{A\ \mathbf{says}\ p, q \vdash p \wedge q}\,{\wedge}R
}{A\ \mathbf{says}\ p, q \vdash B\ \mathbf{aff}\ (p \wedge q)}\,\mathbf{aff}
$$

At the underivable sequent, so rule applies.

3 Groups and Access Control [30 points]

In the AFS file system used on the `linux.andrew` machines there are two kinds of principals: individuals (identified by their Andrew id) and groups (identified by a name distinct from an individual id). We formalize a variant of the Andrew file system access control with the following policies:

(i) Any principal may create a new group they own simply by saying so. Assume that adding such an affirmation to the policy would fail if the group already exists.

(ii) The owner of a group is always one of its members.

(iii) Only the owner of a group may add members to it. We are not concerned with revoking group membership.

(iv) The owner of a file may grant a principal read or write access simply by saying so.

(v) If a group has read or write access to a file, each member of the group has read or write access, respectively, to the file.

In order to formalize this policy in authorization logic, we use the following vocabulary:

- Principals $A, B, C, \ldots$ could denote an individual or a group.
- Groups $G, \ldots$ if we know a principal must be a group.
- $admin$. The principal who administers policy, affirming rules for groups and access control. Depending on the configuration, $admin$ could be a group or an individual.
- Mode M, which is either $read$ or $write$.
- File names F.
- $\mathsf{ownsGroup}(A, G)$. Principal A owns group G.
- $\mathsf{ownsFile}(A, F)$. Principal A owns file F.
- $\mathsf{member}(A, G)$. Principal A is a member of group G.
- $\mathsf{mayOpen}(A, F, M)$. Principal A may open file F in mode M.

Task 9 (10 points) Write out the access control policy explained above in authorization logic. Label your affirmations so they can be used in proof terms.

$$
\begin{array}{lll}
v_1 & : & admin \ \textbf{says} \ \forall A.\, A \ \textbf{says} \ \mathsf{ownsGroup}(A, G) \to \mathsf{ownsGroup}(A, G) \\
v_2 & : & admin \ \textbf{says} \ \forall A.\, \forall G.\, \mathsf{ownsGroup}(A, G) \to \mathsf{member}(A, G) \\
v_3 & : & admin \ \textbf{says} \ \forall A.\, \forall B.\, \forall G.\, \mathsf{ownsGroup}(A, G) \to A \ \textbf{says} \ \mathsf{member}(B, G) \to \mathsf{member}(B, G) \\
v_4 & : & \mathsf{admin} \ \textbf{says} \ \forall A.\, \forall B.\, \forall F.\, \forall M. \\
 & & \quad \mathsf{ownsFile}(A, F) \to A \ \textbf{says} \ \mathsf{mayOpen}(B, F, M) \to \mathsf{mayOpen}(B, F, M) \\
v_5 & : & \mathsf{admin} \ \textbf{says} \ \forall G.\, \forall F.\, \forall M.\, \forall A.\, \mathsf{mayOpen}(G, F, M) \to \mathsf{member}(A, G) \to \mathsf{mayOpen}(A, F, M)
\end{array}
$$

Task 10 (10 points) Write out the necessary affirmations by *fp* or $admin$ to express each of the following. Label your affirmations so they can be used in proof terms.

1. Individual *fp* owns group $\mathit{316\_ta}$ and files $\mathit{316\_roster}$ and $\mathit{316\_grades}$.
2. Individuals $myra$ and $hemant$ are members of $\mathit{316\_ta}$.

3. Group *316_ta* may read file *316_roster* and read and write file *316_grades*.

$$
\begin{array}{lll}
w_1 & : & \mathit{fp}\ \textbf{says}\ \mathsf{ownsGroup}(\mathit{fp}, \mathit{316\_ta}) \\
 & & \wedge\ \mathit{admin}\ \textbf{says}\ (\mathsf{ownsFile}(\mathit{fp}, \mathit{316\_roster}) \wedge \mathsf{ownsFile}(\mathit{fp}, \mathit{316\_grades})) \\
w_2 & : & \mathit{fp}\ \textbf{says}\ (\mathsf{member}(\mathit{myra}, \mathit{316\_ta}) \wedge \mathsf{member}(\mathit{hemant}, \mathit{316\_ta})) \\
w_3 & : & \mathit{admin}\ \textbf{says}\ (\mathsf{mayOpen}(\mathit{316\_ta}, \mathit{316\_roster}, \mathit{read}) \\
 & & \wedge\ \mathsf{mayOpen}(\mathit{316\_ta}, \mathit{316\_grades}, \mathit{read}) \\
 & & \wedge\ \mathsf{mayOpen}(\mathit{316\_ta}, \mathit{316\_grades}, \mathit{write}))
\end{array}
$$

Task 11 (10 points) Write out a proof term (as defined in Lecture 17) showing *myra* may write file *316_grades*. You should use labels from the affirmations in the two previous tasks.

We put term arguments to proofs in [brackets], as in the concrete syntax for PCA in Lab 3. We also record a couple of formulas in comments, for readability.

$$
\begin{array}{ll}
\{ & \\
\quad \textbf{let}\ \{x_3\}_{\mathit{admin}} = v_3\ \textbf{in} & \\
\quad \textbf{let}\ y_1 = x_3[\mathit{fp}][\mathit{myra}][\mathit{316\_ta}](w_1.\pi_1)\{\textbf{let}\ \{z\}_{\mathit{fp}} = w_2\ \textbf{in}\ z.\pi_1\}_{\mathit{fp}}\ \textbf{in} & \%: \mathsf{member}(\mathit{myra}, \mathit{316\_ta}) \\
\quad \textbf{let}\ \{y_3\}_{\mathit{admin}} = w_3\ \mathit{in} & \\
\quad \textbf{let}\ y_4 = y_3.\pi_2 & \%: \mathsf{mayOpen}(\mathit{316\_ta}, \mathit{316\_grades}, \mathit{write}) \\
\quad \textbf{let}\ \{x5\}_{\mathit{admin}} = v_5\ \textbf{in} & \\
\quad x_5[\mathit{316\_ta}][\mathit{316\_grades}][\mathit{write}][\mathit{myra}]\ y_4\ y_1 & \\
\}_{\mathit{admin}} & \\
: & \\
\mathit{admin}\ \textbf{says}\ \mathsf{mayOpen}(\mathit{myra}, \mathit{316\_grades}, \mathit{write})) &
\end{array}
$$