Assignment 7: Authorization & Trust
15-316 Software Foundations of Security and Privacy

1. **Maybe, maybe not (10 points).** Is the following formula valid in the authorization logic discussed in lecture?

   $$(A \texttt{ says } P \to Q) \to B \texttt{ says } P \to A \texttt{ says } Q$$

   If so, prove it formally. If not, explain why, and provide a set of policy assumptions Γ that would suffice to make it valid. In other words, provide Γ such that the following sequent is provable.

   $$\Gamma \vdash (A \texttt{ says } P \to Q) \to B \texttt{ says } P \to A \texttt{ says } Q$$

   Note: Your Γ *should not be* $A \texttt{ says } Q$*.*

2. **Countersignatures (15 points).** It is common practice in PKI to have the CA issue weaker certificates that rely on a *countersignature* for verification. So suppose that ca is the certificate authority and cs is the countersigner, and cmu wants a key signed. One way to accomplish this might be to have ca issue a certificate to cmu that consists of the following.

$$\mathtt{sign}_{\mathsf{sk}_{\mathsf{ca}}}(\forall x.\mathsf{cs}\ \mathtt{says}\ \mathtt{isKey}(\mathsf{cmu}, x) \rightarrow \mathtt{isKey}(\mathsf{cmu}, x)) \tag{1}$$

Then cs must issue a second certificate, which comes with an expiration date for a particular key $\mathtt{pk}_{\mathsf{cmu}}$, modeled by $\mathtt{isbefore}(exp)$, where *exp* is the expiration date of the countersignature.

$$\mathtt{sign}_{\mathsf{sk}_{\mathsf{cs}}}(\mathtt{isbefore}(exp) \rightarrow \mathtt{isKey}(\mathsf{cmu}, \mathsf{pk}_{\mathsf{cmu}})) \tag{2}$$

Note that rather than signing a public key unconditionally, the ca signs the public key conditional on a statement from the countersigner that the key is still valid. This can partially mitigate the consequences of leaked keys, because the countersignature can have a short expiration period, so after a countersigned key is leaked, the vulnerable party simply lets the countersignature expire.

(a) **(5 points).** Explain how a remote party can use *(1)* and *(2)*, along with knowledge of the ca's public key and cs's public key, to establish $\mathtt{isKey}(\mathsf{cmu}, \mathsf{pk}_{\mathsf{cmu}})$. Your answer should explain how to the *(Sign)* and *(Cert)* from lecture 18, but you do not need to write a formal proof.

(b) **(10 points).** Explain why this approach to countersigning is vulnerable in the event that cs is compromised. That is, assuming an attacker mal has access to cs's secret key $\mathtt{sk}_{\mathsf{cs}}$, describe what they must do to convince someone that $\mathtt{isKey}(\mathsf{cmu}, \mathtt{pk}_{\mathsf{mal}},)$. Then, explain how either equation *(1)* or *(2)* (or both) should be fixed to remove this vulnerability.