

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?

$$(A \text{ says } P \rightarrow Q) \rightarrow B \text{ says } P \rightarrow A \text{ says } Q$$

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.

$$\Gamma \vdash (A \text{ says } P \rightarrow Q) \rightarrow B \text{ says } P \rightarrow A \text{ says } Q$$

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

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.

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

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

$$\text{sign}_{\text{sk}_{\text{cs}}}(\text{isbefore}(\text{exp}) \rightarrow \text{isKey}(\text{cmu}, \text{pk}_{\text{cmu}})) \quad (2)$$

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 $\text{isKey}(\text{cmu}, \text{pk}_{\text{cmu}})$. 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 \mathbf{cs} is compromised. That is, assuming an attacker \mathbf{mal} has access to \mathbf{cs} 's secret key $\mathbf{sk}_{\mathbf{cs}}$, describe what they must do to convince someone that $\mathbf{isKey}(\mathbf{cmu}, \mathbf{pk}_{\mathbf{mal}})$. Then, explain how either equation (1) or (2) (or both) should be fixed to remove this vulnerability.