Protocol description

- Client is Prover
- Server is Verifier

We have a group ‘$\langle g \rangle$‘ whose order is $\mathtt{n}$. Secrets are ‘x_1‘ and ‘x_2‘, and public keys are ‘$h_1 = g^{x_1}$‘ and ‘$h_2 = g^{x_2}$‘.

Prover knowing ‘x_1‘	Prover knowing ‘x_2‘	Message	Verifier
‘$c_2, r_2, u_1 \in_R \mathbb{Z}_n$‘	‘$c_1, r_1, u_2 \in_R \mathbb{Z}_n$‘		
‘$a_1 = g^{u_1}$‘	‘$a_1 = g^{r_1} h^{-c_1}$‘		
‘$a_2 = g^{r_2} h_2^{-c_2}$‘	‘$a_2 = g^{u_2}$‘		
		‘ $\xrightarrow{a_1, a_2}$ ‘	
			‘$c \in_R \mathbb{Z}_n$‘
		‘ $\xleftarrow{c}$ ‘	
‘$c_1 = c - c_2 \mod n$‘	‘$c_2 = c - c_1 \mod n$‘		
‘$r_1 = u_1 + c_1 x_1 \mod n$‘	‘$r_2 = u_2 + c_2 x_2 \mod n$‘		
		‘ $\xrightarrow{c_1, c_2, r_1, r_2}$ ‘	
			‘$g^{r_1} \stackrel{?}{=} a_1 h_1^{c_1}$‘
			‘$g^{r_2} \stackrel{?}{=} a_2 h_2^{c_2}$‘

When implemented properly, this results in the client proving to the server that it knows either ‘x_1‘ or ‘x_2‘. Specifically, the client only needs to know a single one, and the server doesn't know which one.

It this passes, the server will then send the flag to the client.