Week 2 / Exercise 1

P: DLP is hard in G
Q: Forging schnorr: given X and R it is difficult to compute m_1, m and s_1, s_2 such that (s_1, R) and (s_2, R) are valid signatures.

We'll try to prove the contrapositive:

$\neg Q \rightarrow \neg P$

ie, if we can forge these signatures, then it means DLP is not hard.

Assume we have a schnorr forger:

$f(X, R) \rightarrow (s, m)$ where s is a valid signature for message m.

This means that we have a function that outputs the following:

$f(X, R) \rightarrow (k + H(R, m) \cdot x, m)$

We do not know k or x.

We will do this twice, to generate two signature message pairs:

$f(X, R) \rightarrow \begin{matrix} s_1 \\ m_1 \end{matrix} = k + H(R, m_1) \cdot x$

$\begin{matrix} s_2 \\ m_1 \end{matrix} = k + H(R, m_1) \cdot x$

We compute $s_1 - s_2$:

$= (k + H(R, m_1) \cdot x) - (k + H(R, m_2) \cdot x)$

$= x [H(R, m_1) - H(R, m_2)]$

We know the values of m_1, m_2 and R (it's public) and we can hash and subtract them to get:

$a = H(R, m_1) - H(R, m_2)$

We now divide by a to solve for x (assuming $a \neq 0$ because $m_1 \neq m_2$):

$= x$

Given our schnorr forger and the pair X, R, we have been able to "easily" (2x hashes, 1x division and 2x invocations of forger f) solve for privkey x from X, thus breaking DLP (x from $g^x = X$).

We have proven $\neg Q \rightarrow \neg P \iff P \rightarrow Q$!