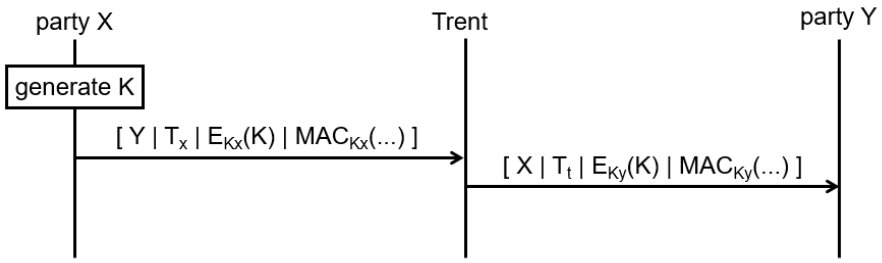
Key Exchange Protocols

Challenge-1: The wide-mouth-frog protocol

Consider the following key transport protocol:

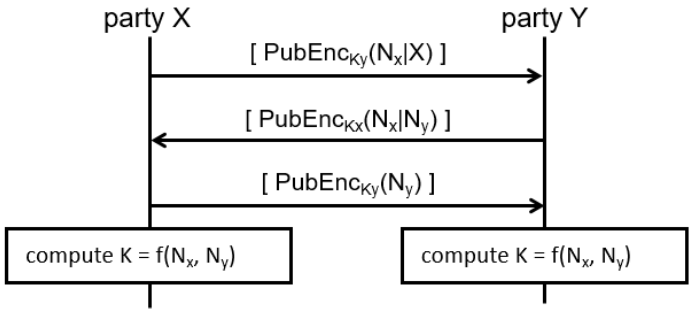


This protocol is known in the literature as the wide-mouth-frog protocol. It is similar to one of the protocols that we designed in the class. The difference is that the first message here does not include the ID of party X, and Trent replaces the timestamp of X with his own timestamp T_t (i.e., the current value of his own clock when the second message is sent to party Y).

Is this protocol secure? Assume the protocol is run by Alice and Bob, as usual. Try to construct an attack, where Mallory tricks Bob to accept a key as fresh, while the key might actually be old.

Challenge-2: The public-key Needham-Schröder protocol

Consider the following key agreement protocol:



This protocol is known in the literature as the public-key Needham-Schröder protocol. It was originally designed for partner authentication only, and it solves that task well. However, later it was noticed that the nonces N_x and N_y never appear in clear on the channel, so their values seem to be known only by parties X and Y. This led to the idea of deriving a key from them by using some one-way function *f*.

Prove that this was a bad idea by constructing an attack. Assume that the protocol is started by Alice with Mallory, and show that Mallory can take advantage of this to impersonate Alice towards Bob. At the end, Bob should believe that he established a key with Alice, while in fact, Mallory will know that key.