Figure 1 ***(Problem 1)***

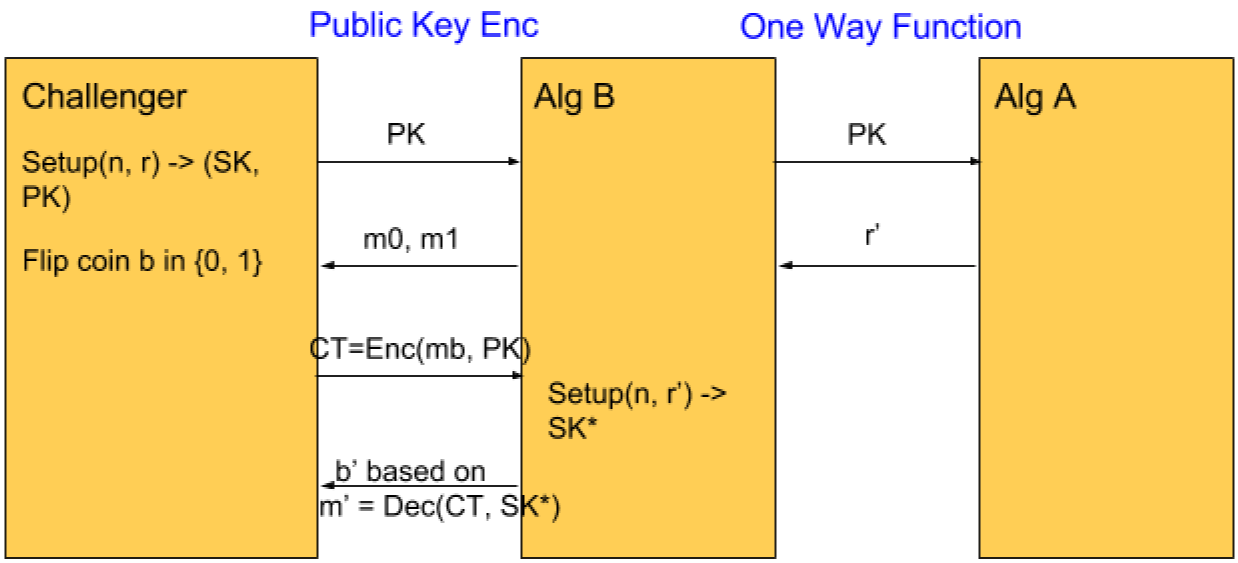
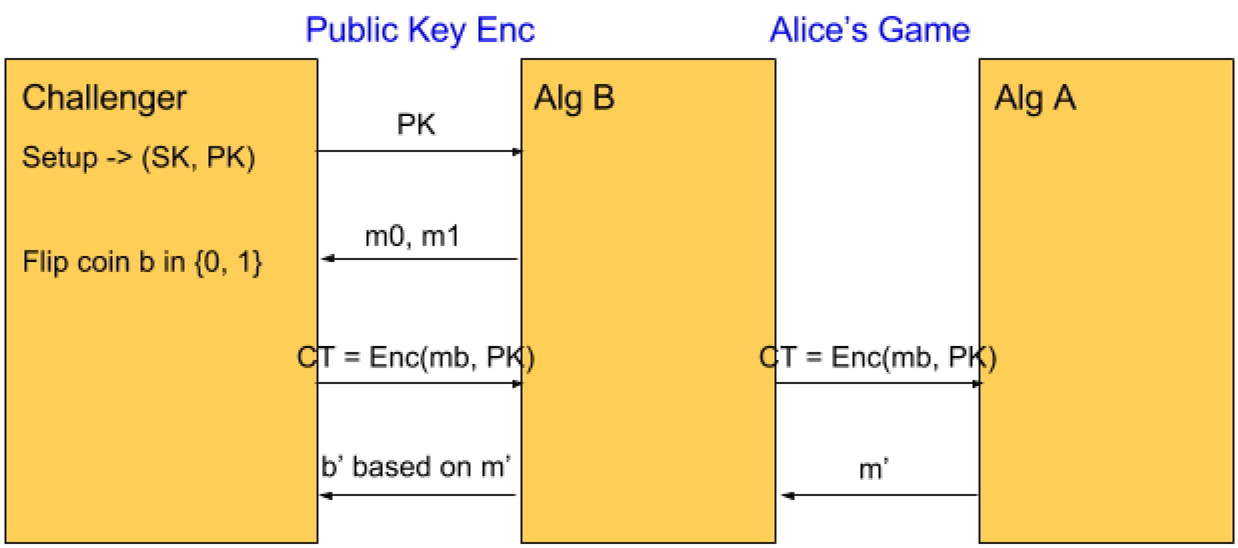


Figure 2 ***(Problem 3)***



1. **One-Way Function**

   We design our function to be
   $$f(r) = PK$$
   where r is the random bits fed to *Setup* and PK is the public key generated by *Setup*. The function domain is $r \in \{0,1\}^n$ and the range is the public key space in *Setup*.

   We prove the function is one-way using the reduction shown in Figure 1. Suppose there is an algorithm A that has advantage telling the input x of our function $f(x)$. Then, we construct an algorithm B to show that the public key encryption scheme would no longer be secure either. In the reduction, the challenger runs *Setup* and sends out public key PK through B to the attacker A. B then sends two messages, m_0 and m_1, in exchange to a ciphertext that is encrypted from one of them depending on the result of flipping a fair coin. The returned r' from A of guess random bits can be used to recover a secret key SK^* to decrypt the cipertext from the challenger. If the cipertext is decrypted to either m_0 or m_1, B sends 0 or 1 to the challenger, respectively; otherwise, B sends back a random bit. From the reduction, because we know that the public key encryption is secure, we are ensured that our function is one-way.

2. **Attack on** $(m+r)^d$

   First, we can observe that this variant of RSA does not satisfy the typical security definition for signatures. Because r is a known part of the signature, when we query for $(m+r)^d$, we know $(m+r)$ as well, which makes the scheme as venerable as the one concerning m^d.

   Here is one way to attack this scheme in which we can forge a signature for a message that we have not queried before. First, we query $\sigma = (m+r)^d$. Then, for message $m' = m^2$, we set $r' = 2mr + r^2$. Because $\sigma^* = (m'+r')^d = (m^2+2mr+r^2)^d = (m+r)^{2d} = \sigma^2$, if we square the queried result σ, we get a valid signature σ^* for the message m' with some r'.

3. **Bob's Imposter**

   The imposter can only fool Alice if the message space M is small. In that case, the imposter performs a brute force search for all $m \in M$ and finds the one that encrypts to the ciphertext. Otherwise, we provide a reduction that proves the security of Alice's game given the IND-CPA security of *Setup*, *Encrypt*, and *Decrypt*.

   In the reduction shown in Figure 2, we suppose that there exists an attacker A who can break Alice's game. The challenger first sends the public key, and B sends two messages m_0 and m_1. When the challenger sends out the encrypted message, A returns the claimed original message as m'. If m' is the same as either m_0 or m_1, B sends 0 or 1 to the challenger, respectively; otherwise, B sends back a random bit. We see that, if Alice's game is broken, then the public key scheme would also be insecure, which is a contradiction.

4. **Three-Party Shared Key**

   First, Alice and Bob get the shared key between them two, g^{ab}, just like the regular two-party Diffie-Hellman. Then, both of them obtain $x = H(g^{ab})$ using the publicly known hash function $H: G \to Z_q$. Receiving g^x, Charlie raises it to $(g^x)^c$ with his secret key c. Alice and Bob receive g^c from Charlie, and compute $(g^c)^x$ with $x = H(g^{ab})$.

   First of all, the three-party shared key scheme is correct. Because
   $$(g^x)^c = (g^c)^x = g^{xc},$$
   each party ends up having the same key g^{xc}.

   In addition, we claim that the three-party shared key scheme is secure. An eavesdropper in the middle of their communications can obtain g, g^a, g^b, g^c, g^x, and g^{xc}. Therefore, the scheme is secure if an attacker cannot distinguish between the two following distributions:

$$D_1 : g, g^a, g^b, g^c, g^x, g^{xc};$$

$$D_2 : g, g^a, g^b, g^c, g^{t_1}, g^{t_2}.$$

We find a third distribution to help bridge the gap between the two distributions:

$$D_3 : g, g^a, g^b, g^c, g^{t_1}, g^{t_1 c}.$$

First, the listener is not able to distinguish D_3 from D_1 under the Diffie-Hellman assumption. Because the hash function is assumed to randomly map to an element in Z_q, in the following proof, we consider $x = H(g^{ab})$ as a random element from Z_q. We construct a reduction as follows. Suppose there exists an attacker A who gains advantage distinguishing D_3 from D_1, and an algorithm B between A and the challenger. The challenger flips a fair coin and sends g, g^s, g^{t_1} and g^T, where $s * t_1 = x$ and T is either x or a random t_1 depending on the result of the coin. B will pick a random $c \in Z_p$ and sends

$$g, g^a, g^b, g^c, g^T, g^{Tc}.$$

Consequently, the sent distribution is either D_1 or D_3. The attacker then sends back the guess b' back to B and the challenger. $x = H(g^{ab})$ should not be distinguishable from t_1 and $Pr[A \to 1 \mid D_1] - Pr[A \to 1 \mid D_3] = negl(n)$.

Second, the listener is not able to distinguish D_3 from D_2 either. In our second reduction, we suppose there exists an attacker A who gains advantage distinguishing D_3 from D_2, and an algorithm B between A and the challenger. The challenger flips a fair coin and sends g, g^{t_1}, g^c, and g^T, where T is either a random t_2 or $t_1 c$ depending on the result of the coin. B will pick a random $c \in Z_p$ and a random $t_1 \in Z_p$. B then sends

$$g, g^a, g^b, g^c, g^{t_1}, g^T$$

Consequently, the sent distribution is either D_2 or D_3. The attacker then sends back the guess b' back to B and the challenger. t_2 should not be distinguishable from $t_1 c$, and $Pr[A \to 1 \mid D_3] - Pr[A \to 1 \mid D_2] = negl(n)$.

Therefore, it must be the case that $Pr[A \to 1 \mid D_1] - Pr[A \to 1 \mid D_2] = negl(n)$ when neither D_1 nor D_2 is distinguishable from D_3. The eavesdropper is not able to infer much information from what is listened so the protocol is secure.