1. Suppose $p$ is a large prime, $\alpha$ is a primitive root, and $\beta \equiv \alpha^a \pmod{p}$. The numbers $p, \alpha, \beta$ are public. Peggy wants to prove to Victor that she knows $a$ without revealing it. They do the following:

   (1) Peggy chooses a random number $r \pmod{p-1}$.
   (2) Peggy computes $h_1 \equiv \alpha^r \pmod{p}$ and $h_2 \equiv x\alpha^{a-r} \pmod{p}$ and sends $h_1, h_2$ to Victor.
   (3) Victor chooses $i = 1$ or $i = 2$ asks Peggy to send either $r_1 = r$ or $r_2 = a - r \pmod{p-1}$.
   (4) Victor checks that $h_1 h_2 \equiv \beta \pmod{p}$ and that $h_i \equiv \alpha^{r_i} \pmod{p}$.
   (5) They repeat steps (1) through (4) one more time.

(a) Suppose Peggy does not know $a$ but she correctly guesses that Victor will ask for $r_1$ in the first round and $r_2$ in the second round. What strategy should Peggy use to be able to answer both of Victor’s questions correctly?
(b) Suppose Peggy does not know $a$. She knows the value of $r_1$ such that $\alpha^{r_1} \equiv h_2 \pmod{p}$, but Victor asks for $r_2$ in the first round. Why will it be difficult for Peggy to compute the value of $r_2$ quickly?

2. (a) Suppose Alice uses a budget hash function $h$ to sign her checks, so she signs a check $m$ by signing $h(m)$ where $h(m)$ is a string of 20 binary bits. This yields pairs $(m, \text{sig}(h(m)))$, which she stores on her computer. Suppose Eve has a set of $10^4$ fraudulent checks and she wants to put Alice’s signature on at least one of them. Eve breaks into Alice’s computer and obtains a list of $10^4$ signed checks $(m, \text{sig}(h(m)))$. Describe how Eve can (with very high probability) put Alice’s signature on some fraudulent check? (Note: $2^{20} \approx 10^6$)
(b) Suppose Alice upgrades to a better hash function $h_1$ such that $h_1(m)$ is a string of around 200 bits. Why is it unlikely that Eve will be able to use a birthday attack to put Alice’s signature on a fraudulent check.

3. Consider the following signature algorithm. Alice wants to sign a message $m$. She chooses a large prime $p$ and a primitive root $\alpha$. She chooses a secret integer $a$ and calculates $\beta \equiv \alpha^a \pmod{p}$. She publishes $(p, \alpha, \beta)$ but keeps the number $a$ secret. To sign the message, she does the following:

   (1) Chooses a random integer $k$ with $\gcd(k, p-1) = 1$.
   (2) Computes $r \equiv \alpha^k \pmod{p}$.
   (3) Computes $s \equiv am + kr \pmod{p-1}$.
   (4) The signed message is $(m, r, s)$.

Bob verifies the signature as follows:

   (1) Computes $u_1 \equiv \alpha^s \pmod{p}$.
   (2) Computes $u_2 \equiv \beta^m r^r \pmod{p}$.
   (3) Declares the signature valid if $u_1 \equiv u_2 \pmod{p}$.

(a) Show that if Alice signs the document correctly then the verification congruence holds.
(b) Suppose Eve finds out the value of $k$ that Alice used. Describe how Eve can figure out the value of $a$. (Note: she might at first have more than one possibility (but probably not very many possibilities) for $a$; you should include a description of how she determines which is the correct one.)

(c) If Eve chooses a value of $r$ for her own message $m$, why will she have a hard time finding a value of $s$ that makes the verification congruence hold?

4. Consider the following Feistel cryptosystem consisting of three rounds. The key $K$ is the same for each round and has 64 bits. The input for the $i$th round consists of 64 bits, divided into a left half and a right half: $L_{i-1}R_{i-1}$, where $L_i$ and $R_i$ each have 32 bits. The output is $L_iR_i$, where $L_i = R_{i-1}$ and $R_i = L_{i-1} \oplus f(K, R_{i-1})$. The function $f$ is given by $f(K, R) \equiv R^K \pmod{2^{64}}$, written as a 64-bit string.

If you receive the ciphertext $L_3R_3$, describe how you would decrypt it to obtain $L_0R_0$. Show that this decryption works. (You may not simply quote results about this type of decryption.)

5. Consider the following elliptic curve protocol: Alice wants to send a message $m$ to Bob. Alice and Bob publicly determine an elliptic curve $E$ mod a large prime $p$ and an integer $n$ such that $nP = \infty$ for all points $P$ on $E$. Alice represents $m$ as a point $P_0$ on $E$ by some publicly known procedure (the procedure is known, but not $P_0$ or $m$). They perform the following steps:

1. Alice chooses a secret integer $a$ with gcd($a, n$) = 1 and Bob chooses a secret integer $b$ with gcd($b, n$) = 1.
2. Alice computes $P_1 = aP_0$ and sends $P_1$ to Bob.
3. Bob computes $P_2 = bP_1$ and sends $P_2$ to Alice.
4. Alice computes $a_1 \equiv a^{-1} \pmod{n}$ and computes $P_3 = a_1P_2$. She sends $P_3$ to Bob.
5. Bob computes $b_1 \equiv b^{-1} \pmod{n}$ and computes $P_4 = b_1P_3$. It can be shown that $P_4 = P_0$, so Bob has received the message $m$ (that is, he can extract $m$ from $P_0$).

(a) Suppose Eve knows how to compute discrete logs for elliptic curves and she listens to the communications between Alice and Bob. How can she determine the secret integers $a$ and $b$? (This also allows Eve to determine $P_0$, and therefore $m$, but don’t show this.)

(b) Describe a classical version (that is, a non-elliptic curve version related to the classical discrete log problem) of the above protocol in which the message is now an integer $m$ mod a large prime $p$. 