1. (20 points = 10+10) (a) Let \( ax + b \) be the encryption function. Then \( h = 7 \) encrypts to \( N = 13 \), so \( 7a + b \equiv 13 \pmod{26} \). Also, \( a = 0 \) encrypts to \( O = 14 \), so \( b = 14 \). This yields \( 7a + 14 \equiv 13 \), so \( 7a \equiv -1 \). Therefore, \( a = 11 \). The encryption function is \( 11x + 14 \).

(b) Displacement by 1 gives 2 matches, by 2 gives 6 matches (8 if we wrap around), and by 3 gives 2 matches. Therefore, the key length is probably 2. The 1st, 3rd, 5th, 7th, 9th letters are \( \text{BBBAB} \). Since \( b \) is the most common letter in the language, there is probably no shift. The 2nd, 4th, 6th, 8th, 10th letters are \( \text{AAAAA} \). These are probably shifted by 1. The key is \( ab \) (or 0,1). The plaintext is \( \text{bbbbbbabbb} \).

2. (30 points = 10+10+10) (a) Eve needs \( ed \equiv 1 \pmod{p-1} \), since Fermat’s theorem replaces Euler’s theorem here. This means that she needs to solve \( 361d \equiv 1 \pmod{1093} \). The extended Euclidean algorithm yields \( (-36)(1093) + (109)(361) = 1 \), which means \( 109 \times 361 \equiv 1 \pmod{1093} \). Therefore, \( d = 109 \) works.

(b) Use the Chinese Remainder Theorem to find \( x \) satisfying \( x \equiv 7 \pmod{p} \) and \( x \equiv -7 \pmod{q} \).

(c) Compute \( \gcd(x-7, n) \). This will be \( p \) or \( q \).

3. (25 points = 9+8+8) (a) \( s^e \equiv H(m)^{ed} \equiv H(m) \pmod{n} \), since this is RSA encryption/decryption.

(b) Eve needs to find \( s \) satisfying \( s^e \equiv H(m) \pmod{n} \). This is the same as decrypting the RSA “ciphertext” \( H(m) \), which is hard to do.

(c) Eve needs to find \( m \) satisfying \( H(m) \equiv s^e \pmod{n} \). Since \( H \) is preimage resistant, this is hard to do.

4. (15 points = 5+10) (a) Eve switches left and right to get \( R_{16}L_{16} \). She puts this into the machine. In a Feistel system, using the keys in reverse order is needed, but here the keys are all the same. The output is \( R_0L_0 \). She switches left-right to get \( L_0R_0 \).

(b) Bob’s method is weaker. If Eve has a plaintext/ciphertext pair \( (m, c) \), she can make two lists: \( E_K(E_K(m)) \) for all possible \( K \) and \( D_L(D_L(c)) \) for all possible \( L \). The desired pair \( (K_1, K_2) \) is among the pairs \( (L, K) \) that yield matches. Trying another pair \( (m, c) \) eliminates many of the pairs that yielded matches in the first round. A few more iterations should yield the key. There does not seem to be a way to do a meet-in-the-middle attack on Alice’s method.

5. (10 points) The remaining steps:
2. Victor checks that \( X_1 + X_2 = Q \).
3. Victor asks for an \( r_i \) and Peggy sends it.
4. Victor checks that \( r_iP = X_i \).
5. They repeat 6 more times (since \( 1/2^7 < .01 \)) with different \( r_i \)’s.

6. (20 points = 10+10) (a) Eve makes around \( 2^{35} \) versions of each document by adding and removing spaces, commas, etc., and computes the hashes of these modified documents. She thus obtains two lists of length \( \sqrt{2^{70}} = 2^{35} \), one being hashes of good documents and the other being hashes of bad documents. She expects a match. She gets Bob to sign the hash of the good version, which is also the hash of a bad version.

(b) \( H \) is preimage resistant: given \( y \), solving \( a^2 \equiv y \pmod{p} \) is a discrete log problem,
which should be difficult. $H$ is not strongly collision-free: $H(x) = H(x + p - 1)$ for every $x$.

7. (20 points = 6+6+8) (a) Let $x = 0, 1, 2, 3, 4$ and solve for $y$. We get $(0, 2), (0, 3), (1, 2), (1, 3), (2, 0), (4, 2), (4, 3), \infty$.

(b) The slope of the line through the two points is $3/2 \equiv 4$. The line is $y = 4(x - 2)$.

Intersecting with the curve, we get $(4x - 8)^2 = x^3 - x + 4$, so $0 = x^3 - 16x^2 + \cdots \equiv x^3 - x^2 + \cdots$. The sum of the roots is $1 \equiv 2 + 4 + x$, so $x \equiv 0$. Then $y = 4(x - 2) \equiv 2$. Reflect across the $x$-axis to get $(0, 3)$.

(c) They first use RSA (Diffie-Hellman or ElGamal are also possible) to establish a key, which is then used in DES or AES to transmit the data.