1. (a) Given any \(x < 123456\), we have \(H(x) = x\), so \(H\) is not preimage resistant. Also, \(H(123457) = H(1)\), so \(H\) is not strongly collision free.

(b) The line through \((1, 3)\) and \((-2, 0)\) has equation \(y = x + 2\). Intersecting with the curve yields \((x + 2)^2 = x^3 + 8\), so \(0 = x^3 - x^2 + \cdots\). The sum of the roots is \(1 + (-2) + x \equiv 1\), so \(x \equiv 2\). Then \(y \equiv x + 2 \equiv 4\). Reflecting yields the answer \((2, -4)\), or \((2, 7)\).

2. The remaining steps are:

4. Victor randomly chooses \(i = 1\) or \(i = 2\) and asks Peggy for \(r_i\).

5. Peggy sends \(r_i\).

6. Victor checks that \(r_i \equiv x_i\).

7. They repeat steps 1 through 6 seven times.

If Peggy does not know \(m\), the probability is \(1/2\) that Peggy can correctly supply \(r_i\) in a given round. Therefore, the probability is \((1/2)^k\) that Peggy can succeed for \(k\) rounds if she doesn’t know \(m\). When \(k = 7\), this probability is less than .01, so if Peggy succeeds for seven rounds then the probability is more than 99% that she knows \(m\).

3. Collisions can be found by a birthday attack. Make a list of \(H(x)\) for around \(2^{30}\) (maybe a little more) random values of \(x\). Since the length of the list is approximately \(\sqrt{2^{60}}\), there is a good chance that two hash values are the same. This yields a collision.

4. Bob switches \(C_0\) and \(C_1\), so he inputs \(C_1, C_0\) into the machine. It outputs \(C_0\) as the left half and \(C_1 \oplus f(C_0)\) as the right half. But \(C_0 = R\) and \(C_1 \oplus f(C_0) = (L \oplus f(R)) \oplus f(R) = L\). Therefore, the output is \(RL\). Switch the two halves to get \(LR\).

5. (a) Look for a match between the two lists. If there is, then \(\beta \equiv \alpha^{j+kN}\), so \(x = j + kN\) solves the discrete log problem. This always works because we can write \(x = x_0 + x_1 N\). When \(j = x_0\) and \(k = x_1\), we have a match. This means that there is a match between the two lists.

(b) Make two lists:

1. \(jA\) for \(0 \leq j < N\)

2. \(B - kNA\) for \(0 \leq k < N\). Look for a match between the two lists. When there is a match, we have \(B = (j + kN)A\).

6. (a) We need \(m\) so that \(\alpha^m \equiv \beta \cdot \alpha^r (\mod p)\). This simplifies to \(\alpha^m \equiv \beta \cdot (\alpha \beta)^{-r} \equiv \alpha^{-r} \pmod p\), so we can take \(m \equiv -r \pmod {p-1}\).

(b) We need \(H(m) \equiv -\alpha \beta\). But \(H(m)\) is assumed to be preimage resistant, so it is hard to find \(m\) satisfying this property.