Crypto 1 HW 2.B

1.
    1. $\alpha^{X_A} \mod q \Rightarrow Y_A$, $7^5 \equiv 51 \mod 71$
    2. $7^12 \equiv 4 \mod 71$
    3. $7^{5 \cdot 12} = (7^5)^{12} = 30 \mod 71$
    4. Finding the exponent of an equation is relatively difficult, due to the complexity of the discrete log problem. Finding the root based on the answer and exponent is easy, by dividing multiple times.
2.
    1. A birthday attack is when any two messages collide with the same 64 bit hash and an adversary replaces a message with a completely different message that happens to have exactly the same hash.

    2. Depending on the approach, different amounts of memory can be attached. The following diagram shows the first approach to comparison and memory:

    $$2^{\frac{m}{2}} \begin{cases} 64 \quad [-m_{11}-] & 64 \quad [-m_{12}-] \\ 64 \quad [-m_{21}-] & 64 \quad [-m_{22}-] \\ \qquad \vdots & \qquad \vdots \\ 64 \quad [-m_{\dots 1}-] & 64 \quad [-m_{\dots 2}-] \end{cases}$$

    Each line is a comparison between two messages. This diagram shows that the resulting memory usage is $\left(2^{\frac{64}{2}}\right)(2m + 2 \cdot 64)$ where m is the length of a message.

    $$K \begin{cases} 64 \quad [-m_1-] \\ \qquad \vdots \\ 64 \quad [-m_{\dots}-] \end{cases}$$

    where K is the value for which $\binom{K}{2} = 2^{\frac{64}{2}}$. This means that a smaller amount of bits needs to be stored in memory, but now every message needs to be compared to each other message to find a possible collision. This results in only $K(64 + m)$ bits in memory. K is found in the following way:

    $$\frac{K \cdot (K-1)}{2} = 2^{\frac{m}{2}}$$
    $$K \cdot (K-1) = 2^{\frac{m}{2}+1}$$
    $$K^2 - K + \frac{1}{2} = 2^{\frac{m}{2}+1} + \frac{1}{2}$$
    $$\left(K - \frac{1}{4}\right)^2 = 2^{\frac{m}{2}+1} + \frac{1}{2}$$
    $$K - \frac{1}{4} = \sqrt{2^{\frac{m}{2}+1} + \frac{1}{2}}$$
    $$K = \sqrt{2^{\frac{m}{2}+1} + \frac{1}{2}} + \frac{1}{4}$$

    3. $2^{\frac{m}{2}}$ pairs of messages must be analyzed to find a collision with a certainty of more than 50%. Therefore, $2 \cdot 2^{\frac{m}{2}} = 2^{\frac{m}{2}+1}$ individual messages to hash. because our computer can hash 2^{20} messages per second, $\frac{2^{\frac{m}{2}+1}}{2^{20}} = 2^{\frac{m}{2}+1-20}$. For $m = 64$, this results in 2^{13}.

    4.

   1. Using the same approach as above, for 128 bits, the amount of bits stored to find collisions is either $2^{\frac{128}{2}} \cdot (2m + 2 \cdot 128)$ or $K(128 + m)$, where K is defined above with $m = 128$
   2. Using the same approach as previously, for $m = 128$, the answer becomes 2^{45}.

3. for $P =' 01010111'$, $a = 1019$, $q = 1999$, $a^{-1} \equiv 1589 \mod 1999$, $S = \{5, 9, 21, 45, 103, 215, 450, 946\}$, we can create the ciphertext C:

$$\begin{gathered}
\beta_i = S_i a \mod q \\
\beta = \{1097, 1175, 1409, 1877, 1009, 1194, 779, 456\} \\
\text{We sum the values of the public key set that match one's in the plaintext} \\
1175 + 1877 + 1194 + 779 + 456 = 5481 = C \\
C' = C \cdot a^{-1} = 5481 \cdot 1589 \mod 1999 \\
C' = 1665
\end{gathered}$$

   now we can use 1665 and a greedy approach to the secret super-increasing set to get our plaintext back.