Tommy Olney
Crypto
Hw 2.b

1.

a) $y_A = \alpha^{x_a} mod\ q\ = 7^5 mod\ 71\ =\ 51$

b) $y_B = \alpha^{x_b} mod\ q\ = 7^{12} mod\ 71\ =\ 4$

c) $shared\ key = {y_B}^{x_a} mod\ q\ = 4^5 mod\ 71\ =\ 30$

$shared\ key = {y_A}^{x_b} mod\ q\ = 51^{12} mod\ 71\ =\ 30$

d) If they send each other the modified version, then what they are really computing is $({x_a}^{\alpha})^{x_b}\ and\ ({x_b}^{\alpha})^{x_a}\ instead\ of\ (\alpha^{x_a})^{x_b}$. This means that the key that they are calculating is not guaranteed to be shared, and therefore cannot be used to encrypt information.

2.

a) The birthday attack:

The attacker generates $2^{\frac{n}{2}}$ variations of the original message, all of which are basically the same, and stores their hash value and message.

The attacker then generates their own false message, and the minor variations of it.

For each of these fraudulent variations, the attacker compares the hashed value of the fraud to each of the original hashes until a match is found.

After finding the same hash, the fraudulent hash is sent to the original author for verification, and is validated, as the fraudulent hash is identical to the original.

b) $2^{m/2}$ attempts, m = 64 so 2^{32} tries needed. As each attempt requires 64 bits of space, multiply that by 64 to get 274877906944 bits needed, or ~34.36 gigabytes

c)

$2^{32}\ attempts,\ processed\ at\ 2^{20} per\ second,\ leaving\ 2^{12}\ seconds,\ or\ 4096\ seconds\ (or\ 68\ minutes\ and\ 15\ seconds)$

d) $2^{64} * 128\ =\ 2.36 * 10^{21}\ bits$
$2^{64}/2^{20}\ =\ 2^{44}\ seconds,\ or\ 557\ 474.643\ years$

3.

a) 0+(9 mod 1999)+0+(45 mod 1999)+0+(215 mod 1999)+(450 mod 1999)+(946 mod 1999)+(1019 mod 1999)=1483

Decrypt: $c_0 = a^{-1} * 1483 \; mod \; 1999 \; = \; 1665$
Make the table with $c_{i-1} = c_i - s_i$
Result: [0,1,0,1,0,1,1,1]