Nick Roper

1)
prime q=71, primitive root $\alpha = 7$
a)
Given that user A has private key $X_A = 5$, their public key Y_A must be $\alpha^{X_A} \mod q$, so $7^5 \mod 71$
$7^5 \equiv 343 * 7^2 \equiv 59 * 7 * 7 \equiv 413 * 7 \equiv 58 * 7 \equiv 406 \equiv 51 \mod 71$
so $Y_A = 51$

b)
Given that user B has private key $X_B = 12$, their public key Y_B must be $\alpha^{X_B} \mod q$, so $7^{12} \mod 71$
$7^{12} \equiv 7^{6^2} \equiv (7^5 * 7)^2 \equiv (51 * 7)^2 \equiv 357^2 \equiv 2^2 \equiv 4 \mod 71$
so $Y_B = 4$

c)
The shared secret key is ${Y_A}^{X_B} \equiv {Y_B}^{X_A} \mod q$
We will use ${Y_B}^{X_A}$
$4^5 \equiv 1024 \equiv 30 \mod 71$
so the shared secret key is 30

d)
For $x^\alpha \mod q$,
1) if $\gcd(q-1, \alpha) = 1$, an attacker can efficiently compute x.
2) if $\gcd(q-1, \alpha) \neq 1$, multiple x will map to the same value of $x^\alpha \mod q$ and thus x would not be recoverable from the message.

In either case, there is no way for the participants to get a shared secret that is not easily computable from the messages they have sent each other, because the security of Diffie-Hellman is based on the computational intractability of solving the discrete log problem, but one does not need to solve the discrete log problem to crack any secret resulting from these keys. In fact, one can possibly even determine the private key of the user that sent each message.

2)
a)
The attacker generates a large number (2^{32}) of valid messages with essentially the same meaning as well as a large number (2^{32}) of fraudulent messages, once again with essentially the same meaning.
Upon finding a matching hash among two messages, one valid and one fraudulent, the attacker can have the user sign the valid message, and then send the fraudulent message, which will have a valid signature due to having the same hash.

b)
The attacker needs 2^{32} valid + 2^{32} fraudulent = 2^{33} messages,
so if each of these is M bits, they will thus need $M * 2^{33}$ bits of memory, on the order of $M * 1.07$ gigabytes.

c)
One can sort each array of valid/fraudulent messages (by hash value) using about
$2^{32} \log_2(2^{32}) = 32 * 2^{32}$ hashes, so $64 * 2^{32}$ for both.
Then, comparing these arrays to find common values can be done in about $2^{32} + 2^{32}$ hashes, but the expected amount from this is halved due to the probability of there being a collision being 0.5, so the expected hashes to find a collision is 2^{32}.
This gives a total of $65 * 2^{32}$ hashes, which, processed at 2^{20} hashes per second, gives an expected time to find collisions as about $65 * 2^{12}$, which is just under 74 hours.

d)
The attacker needs 2^{64} valid + 2^{64} fraudulent = 2^{65} messages,
so if each of these is M bits, they will thus need $M * 2^{65}$ bits of memory, which is too big to give any reasonable units to.

Sorting each array of values takes about $64 * 2^{64}$ hashes, so $128 * 2^{64}$ for both, then comparing to find matches takes about $2^{64} + 2^{64}$ hashes, which is reduced to 2^{64} because the probability of finding a match is 0.5.
This give a total number of hashes as about $129 * 2^{64}$, which, processed, at 2^{20} hashes per second, gives an expected time to find collisions as $129 * 2^{44}$, which is over 71 million years.

3)
First, we construct each value of the public key $t_i = aS_i \mod p$:
$t_1 = 1019 * 5 = 5095 = 1097 \mod 1999$
$t_2 = 1019 * 9 = 9171 = 1175 \mod 1999$
$t_3 = 1019 * 21 = 21399 = 1409 \mod 1999$
$t_4 = 1019 * 45 = 45855 = 1877 \mod 1999$
$t_5 = 1019 * 103 = 104957 = 1009 \mod 1999$
$t_6 = 1019 * 215 = 219085 = 1194 \mod 1999$
$t_7 = 1019 * 450 = 458550 = 779 \mod 1999$
$t_8 = 1019 * 946 = 963974 = 456 \mod 1999$
so the public key is $t = (1097,\ 1175,\ 1409,\ 1877,\ 1009,\ 1194,\ 779,\ 456)$
Thus, we encrypt plaintext $P = 01010111$ as:

$$Y = E_k(P_1, ..., P_8) = \sum_{i=1}^{8} P_i t_i$$

$= 0(1097) + 1(1175) + 0(1409) + 1(1877) + 0(1009) + 1(1194) + 1(779) + 1(456)$
$= 5481$

For decryption, we must first calculate the value of $Z = a^{-1}Y \mod p$
$Z = 1019^{-1} * 5481 = 1589 * 5481 = 8709309 = 1665 \mod 1999$
Then, we can start to solve the subset sum problem:
$Z = 1665$ and $S_8 = 946$, so $Z \geq S_8$. Thus, $X_8 = 1$ and we subtract 946 from Z.
$Z = 719$ and $S_7 = 450$, so $Z \geq S_7$. Thus, $X_7 = 1$ and we subtract 450 from Z.
$Z = 269$ and $S_6 = 215$, so $Z \geq S_6$. Thus, $X_6 = 1$ and we subtract 215 from Z.
$Z = 54$ and $S_5 = 103$, so $Z < S_5$. Thus, $X_5 = 0$.
$Z = 54$ and $S_4 = 45$, so $Z \geq S_4$. Thus, $X_4 = 1$ and we subtract 45 from Z.
$Z = 9$ and $S_3 = 21$, so $Z < S_3$. Thus, $X_3 = 0$.
$Z = 9$ and $S_2 = 9$, so $Z \geq S_2$. Thus, $X_2 = 1$ and we subtract 9 from Z.
$Z = 0$ and $S_1 = 5$, so $Z < S_1$. Thus, $X_1 = 0$.
Thus, the recovered message $X = 01010111$, the same as the original plaintext.