Jack Dates

① ⓐ $Y_A = \alpha^{X_A} \bmod q = 7^5 \bmod 71 = 51$

ⓑ $Y_B = \alpha^{X_B} \bmod q = 7^{12} \bmod 71 = 4$

ⓒ $\alpha^{X_A X_B} \bmod q = 7^{5 \cdot 12} \bmod 71 = 30$

ⓓ This breaks the protocol and won't result in a shared secret. It also can reveal the secret number X. In particular, if $a^{-1} \bmod q{-}1$ exists, $(x^a)^{a^{-1}} \equiv x \bmod q$ and the secret is revealed.

② ⓐ The attacker generates $2^{64/2} = 2^{32}$ variations of their desired fraudulent message (assuming the language/protocol allows for redundant meaning) along with their hashes. They then start generating valid messages and computing the hashes. If any hash matches the hash of one of the 2^{32} fraudulent messages, they have X sign the valid message, providing the signature for the fraudulent message of the same hash. If the attacker generates $\sim 2^{32}$ of the valid messages, the probability of a collision is about $1/2$.

ⓑ They generate 2^{32} valid and 2^{32} fraudulent messages, and for one of those groups they'll store the hashes (if they want to be efficient). That's $(2^{32} + 2^{32}) \cdot M + 2^{32} \cdot 64 = 2^{33} M + 2^{38}$ bits

ⓒ 2^{33} hashes $\cdot\, 2^{-20} \frac{s}{hash} = 2^{13}$ seconds $= 8192$ s

ⓓ number of messages (valid and fraudulent) is 2^{64} now,

so $(2^{64} + 2^{64}) \cdot M + 2^{64} \cdot 128 = 2^{65} M + 2^{71}$ bits

2^{65} hashes $\cdot\, 2^{-20} \frac{s}{hash} = 2^{45}$ s

③ This is the Merkle-Hellman Knapsack cryptosystem. We calculate the public key $\beta = (\dots a \cdot s_i \bmod p \dots) = (1097, 1175, 1409, 1877, 1009, 1194, 779, 456)$. To encrypt the bits of P we calculate $c = \sum \beta_i p_i =$

$1097 \cdot 0 + 1175 \cdot 1 + 1409 \cdot 0 + 1877 \cdot 1 + 1009 \cdot 0 + 1194 \cdot 1 + 779 \cdot 1 + 456 \cdot 1 = 5481$

To decrypt, we calculate $a^{-1} \bmod p = 1589$ and calculate $c' = a^{-1} c \bmod p = 1589 \cdot 5481 \bmod 1999 = 1665$. We now solve subset sum with the secret S. Since S is superincreasing, we greedily take the largest value of S that is less than c' and set that bit in the plaintext, subtracting from c' the element and continuing. $1665 \geq 946$ so the last bit is set, new $c' = 1665 - 946 = 719$. $719 \geq 450$ so the 2^{nd} last bit is set, and so on giving $P = 01010111$