1. You are an attacker trying to read messages between Alice and Bob. You happen to know that the software Alice uses picks keys 48 bits in size and you know the encryption algorithm. You have an array of computers capable of trying 1 million keys per second in a brute force attack against the message data you have accumulated.
   a. How long would it take (on average) for you to execute a successful brute force attack against this cipher?

   \[
   \text{Average (expected) time for brute force attack} = \frac{\text{size of keyspace}}{2} / \text{speed of key testing}
   \]

   \[
   \text{size of keyspace} = 2^{48} = 2.8 \times 10^{14} \text{ keys}
   \]

   \[
   \text{speed of testing} = 10^6 \text{ keys/sec}
   \]

   \[
   \text{expected time} = 1.4 \times 10^8 \text{ secs} / 3.15 \times 10^7 \text{ secs/year}
   \]

   About 4.5 years

   b. Suppose that you know that keys used by the algorithm are produced by calling the system's random number generator and that the operating system used by Alice has a weakness in its random number generator that causes it to pick only random numbers that have ones in the first four bits. How long would the brute force attack take if you take advantage of this information?

   If the random number generator always has ones in the first four bits, the key is effectively 4 bits shorter, since we don't need to search any of the keys with zeros in those positions.

   \[
   \text{size of keyspace} = 2^{44}
   \]

   Everything else in the computation is the same, so the answer is 1/16 times the previous one:

   About .28 years or 102 days

   c. What would your answer be to part (a) if the keys were 64 bits long? 128 bits long?

   64 bits gives us a keyspace \(2^{16}\) times as large as in question 1a. So the answer is 65000 times larger or approximately 290,000 years. 128 bits is \(2^{64}\) again as large. \(5.4 \times 10^{24}\) years.

   d. Suppose that the encryption key is 128 bits, but you know that the key is all zeros except for the last 64 bits, which are the user's password. (This is how Windows NT, 2000 and XP do it.) You also know that passwords are given out by the sysadmin, are never changed by users, are 8 characters long and consist of two lower-case English words with a single digit in between. For example, "you2shoe". You have a dictionary of all 10,000 English words with between 1 and 6 letters. How long will a brute force attack take under these conditions?

   \[
   \text{size of keyspace} = \text{choices for word1} * \text{choices for digit} * \text{choices for word2}
   \]

   we don't have the number of one-letter, two-letter words, etc. but we can form an upper bound.

   There are no more than 10000 words of any size.

   \[
   \text{size of keyspace} < 10000 * 10 * 10000 = 10^8
   \]

   Applying previous equation gives 8.33 minutes.