

QR48 Solution Set 2

1. (a) Worst case, they are looking for a word that appears on the last page of the dictionary, which means it would take them $21730 \cdot 1.5 = 32595$ seconds.
(b) If they know how to use binary search, the worst case is $\lg(21730) = 14.4 \approx 15$ page lookups. Each page lookup takes 1.5 seconds, so a grand total of 22.5 seconds.
2. (a) It would take $2^{256-128} = 2^{128}$ times as long, or $2^{128} \cdot 6 = 2.0416942 \cdot 10^{39}$ months, or $1.70141183 \cdot 10^{38}$ years.
(b) We know that every two “time periods” (one “time period” is 2 years), the computer used to break these encryptions is twice as fast. So every time period, it will take one half as long to break them. If we assume 30 days per month, then 6 months is equal to $24 \cdot 60 \cdot 30 \cdot 6 = 259200$ minutes. This means that we want to solve $(1/2)^x \cdot 259200 = 1$, or $(1/2)^x = 1/259200$. Taking the logarithm to the base $1/2$ of both sides, we get $x = 17.98$, or 18 time periods. Since each time period is two years, we know that it will take 36 years. Similarly for Skype, we know that $2.0416942 \cdot 10^{39}$ months is $8.82011894 \cdot 10^{43}$ minutes. So we need to solve $(1/2)^x \cdot 8.82011894 \cdot 10^{43} = 1$. Solving in the same way, we get 146 time periods, which is 292 years.
(c) Since the speed of the computers will double every two years, we need the “difficulty” of the cracking to double every two years. This means we need to add one bit every two years, or $1/2$ bit per year.
3. (a) Assuming binary search, it will take $\lg(5000)$ (rounded up) accesses to primary storage to find the index entry, which is 13. Since each access takes one nanosecond, it takes 13 nanoseconds to find the index entry.
(b) Now the email client knows where to look in secondary storage for the file, so it only needs to access the hard drive once, which takes 3 milliseconds, and then it starts reading. The actual amount of time it takes to retrieve the contact depends on the size of the contact info and whether it is stored contiguously on the hard drive.
4. (a) Yes. You have computed $P_R \cdot S_Y \pmod{p} = S_R \cdot a \cdot S_Y \pmod{p}$. Your roommate has computed $P_Y \cdot S_Y \pmod{p} = S_Y \cdot a \cdot S_R \pmod{p}$ which is equal to $S_R \cdot a \cdot S_Y \pmod{p}$ because multiplication is commutative and associative. Thus you have computed the same value, and thus the same key.
(b) No. The problem with this method is that doing “division” (i.e., the opposite of multiplication) modulo p is not very hard. The reason Diffie-Hellman works is that raising a number to a power is easy, but the inverse (taking the discrete logarithm) is hard.