On Passwords, Pornstars and Government Supercomputers

by John David Allsup (twitter: DoctrJohn4llsup)

Consider my last webshosting password. It was (and is *not* now, and never again shall be):

`IndiaI5FuckingGorgeous`

and whether the India in question is Ancient India, Modern India or even India Reynolds, to me this statement still makes sense, to a first approxmation is correct, and if a better approximation is needed, I shall reduce the possibility count from three to two by restricting the possible interpretations of India to either 'India Reynolds'[1] or 'Ancient India'[2] and of course anybody with access to the internet can easily verify that 'Ancient India Reynolds' is clearly nonsense by simply executing a few basic web searches. To me this is an example of a *practically effective* procedure. The Turing model of computation allows us to effectively ignore the issue of running out of space or time; Peano Arithmetic likewise permits us to assume that we will never run out of counting numbers, despite this contradicting any sensible notion of the laws of physics; modern set theory allows us to consider what happens if we could collect all those counting numbers, including the ones that the laws of physics effectively preclude ever actually counting to; nonstandard arithmetic allows us to pretend that numbers whose existence is proved by syntactically correct logic actually exist, despite not being able to count up to them even in principle (unless the geometry of time is replaced with a topological long line or something); and there are of course ways of having a theory of unordered collections of objects where we can rigorously play with unordered collections of objects that do not even exist within any model of our theory. It is important to test theories past the point of being ridiculous, and that is a practical necessity with life-critical systems, but mathematics has, for at least the past century or so, taken the liberty of pushing past the point where any sane person would normally remark something like: 'Quite frankly, now you really are taking the [expletive]!'.

When it comes to internet passwords and secret codes, it is not theoretical plausibility or theoretical efficiency that matter, since both these make underlying assumptions which violate the laws of nature, but grant us the ability to show convincing arguments for claims which, when these unrealistic assumptions are removed, still hold true. Examples are the Axiom of Infinity, the Successor Axiom of Peano Arithmetic and the order-completeness of the real line.

In practice, I want a sequence of characters which I can easily remember without difficulty, and that anybody who knows me well might have some chance of guessing, but that someone who does not know my character at all as a human being is effectively faced with a search space larger than the physical universe. There is an easy way, and we don't need to know the answer to P vs NP

[1] Google for 'india reynolds'

[2] Google for 'Buddhism', 'Dhammapada', 'Bhagavad Gita', 'Kalama Sutta', for example

to show it is easy. Indeed I conjecture, but in practice cannot prove, that all practical consequences of P vs NP are easier to prove by rephrasing in a different conceptual model than the one in which the P vs NP problem is stated. I offer the example of internet passwords as an example, and I am grateful to xkcd for making the logic of this straightforward and humorous, and including enough back-of-a-napkin calculations to see that the point makes sense[3].

Anyway, consider the following algorithm for making a secure password:

1. Make a short sentence involving a few of your favourite celebrities, for example Lucy Pinder and Tori Black, that is clearly true so far as you are concerned (other people's opinions do not matter so long as you are reasonably consistent as to your belief). Take

   `LucyPinderHasBiggerBreastsThanToriBlack`

2. Take one lowercase letter, out of $\{i, e, s, t\}$, and if your sentence doesn't contain at least three of these, go back to step 1. Replace this using the classic textspeak rules:

   $$i = 1, e = 3, s = 5, t = 7$$

   and/or replace an instance of 'Too' with 2, at your option. Pick one, and remember which it is. Pick the most obvious one to you so that your first guess as to which it is is likely to be correct.

3. Take your resulting string, perhaps save it in an encrypted file using this process to make the password, but ultimately have at least two or three master passwords which you could reliably reproduce, letter perfect, with a real loaded gun to your head (and do not be lazy an skip the step of ensuring you are reasonably confident here, else in a crisis you will forget) and actually make sure you can type it out into a simple text editor (and as a precaution, I open a file in /tmp, which is a flexible ramdisk, and have virtual memory disabled since a 6GiB laptop has enough RAM to run UbuntuStudio GNU/Linux perfectly fine without it,[4] and write it out a few times until it feels comfortable, then copy and paste it twice to set the password, then try t login without resorting to any written notes, and only then permit myself to peek at my encrypted notes.

This reasonably maximises the chance that I will reliably remember my password, that if someone cracks it it will most likely be a friend I trust enough to tell who my favourite glamour models and pornstars actually are, and that in

[3] See http://xkcd.com/936/

[4] And in the event of an out of memory error, I'd rather move over to my > 16GB (second-hand) dual Xeon workstation than enable vitual memory, since then I do not have to be paranoid about what is going to get written to my machine's hard drive, since source code availability and the fact that I can compile it myself and even step through critical parts of both assembly and machine code are suffient to ensure to my satisfaction that a critical password will never be written to disk without my say-so.

the absence of key information like that, given only a billion dollar, million core supercomputer and a thousand computer science PhDs, I can be sure to have changed the password and effectively invalidated any infomation that may have been gained through cracking the password, regardless of whether a cryptographer considers the approach used to be brute force or not, and regardless of whether P actually equals NP. The basic thing is, if it takes the CIA a week to change my password and I change it every day, and likewise any other passkeys, such that my information is arranged so that the passkeys are all essentially English sentences in a funny language so that all that matters is that at the end of pass-the-parcel decryption exercise, the encrypted plaintext is, in fact, some known statement, and this can be as simple as 'yes' or 'no', provided there is enough redundancy to effectively eliminate any possible chance of a false positive (so basically the last password is not a password at all, and the final plaintext is just a fullstop). Then it comes down to grouping English words (or whatever language) into phrases and doing trivial substitutions, and changing your password as necessary.

Consider if I take

"."

and encrypt it (maybe using AES, backdoored or not backdoored) using

"`SonOfGod`"

as the password, encrypt again and again so that the final password is

"`InTheBeginning`"

adhering letter perfectly to the word order in, say, the English Standard Version of the Bible, then the eavesdropper has to crack AES approximately a hundred times in succession just to get the plaintext matching `/. {19}$/` out, and then all he has got for his effort is a 'well done' message. Of course if I put logic into the testing algorithm so as to alert me with every successful try, I can simply replace an element of the chain with another Bible verse, suitably chunked, and re-encrypt, provided of course that I thoroughly know the relevant parts of my Bible.

Whether that makes me Christian, whether I believe in God, whether I trust in God, and whether or not I do in fact take Jesus Christ as my Lord and Saviour are matters I fix by modifying how I interpret English, only then making my usage reasonably compatible with what they speak in the local pub. An eavesdropper with a supercomputer then effectively has to simulate my brain, cannot even try out one guess until he has cracked all the passwords, and all without even having met me. I consider this unlikely given the current state of computer software.

Finally, consider that even the most efficient algorithm for searching a million item list is going to struggle to win a race against a trivial brute-force algorithm that only has to crack a 3-bit password. There will be, for every problem, a number G for the problem such that it reduces to nothing easier than a brute

force search of a list of G totally random numbers. If I can use my nervous system to make an easy way to randomly generate such problems, such that my nervous habits, love of naked women and severe case of bipolar disorder make it easy to know the answer, I can most likely find problems for which the U.S. Government's best solution is to use their supercomputer to simulate my brain's behaviour. The trouble is that winning the lottery will massively change my behaviour, so they need to simulate that if I simply buy a few tickets and sometimes bother to check them. In practice, that means they must simulate the entirely planet pefectly, and the question then is: where exactly are they going to put their supercomputer so that it doesn't disrupt the calculations by using physical resources. Most likely putting it in the same Solar system as our lovely Earth isn't going to work, since NASA probes will most likely pick up some kind of EM radiation, or at least might. Basically their only route is to stick their supercomputer outside the observable universe, and then try and find a way to get the results back to Earth before I bother to change my passwords again. And cracking after the fact isn't going to happen since, like in Kung Fu Panda, the message on the scroll effectively only carries a few bits of information used to error-check the decryption process: the passwords are the data, the plaintext is the punctuation mark at the end and I reserve the right to change my mind. If I can't have that, please just take me out and shoot me!