Passphrases That You Can Memorize — But That Even the NSA Can't Guess

Micah Lee, The Intercept

https://theintercept.com/2015/03/26/passphrases-can-memorize-attackers-cant-guess/

Mar. 26 2015, 4:29 p.m.

IT'S GETTING EASIER to secure your digital privacy. iPhones now encrypt a great deal of personal information; hard drives on Mac and Windows 8.1 computers are now automatically locked down; even Facebook, which made a fortune on open sharing, is providing end-to-end encryption in the chat tool WhatsApp. But none of this technology offers as much protection as you may think if you don't know how to come up with a good passphrase.

A passphrase is like a password, but longer and more secure. In essence, it's an encryption key that you memorize. Once you start caring more deeply about your privacy and improving your computer security habits, one of the first roadblocks you'll run into is having to create a passphrase. You can't secure much without one.

For example, when you encrypt your hard drive, a USB stick, or a document on your computer, the disk encryption is often only as strong as your passphrase. If you use a password database, or the password-saving feature in your web browser, you'll want to set a strong master passphrase to protect them. If you want to encrypt your email with PGP, you protect your private key with a passphrase. In his first email to Laura Poitras, Edward Snowden wrote, "Please confirm that no one has ever had a copy of your private key and that it uses a strong passphrase. Assume your adversary is capable of one trillion guesses per second."

In this post, I outline a simple way to come up with easy-to-memorize but very secure passphrases. It's the latest entry in an ongoing series of stories offering solutions — partial and imperfect but useful solutions — to the many surveillance-related problems we aggressively report about here at *The Intercept*.

It turns out, coming up with a good passphrase by just thinking of one is incredibly hard, and if your adversary really is capable of one trillion guesses per second, you'll probably do a bad job of it. If you use an entirely random sequence of characters it might be very secure, but it's also agonizing to memorize (and honestly, a waste of brain power).

But luckily this usability/security trade-off doesn't have to exist. There is a method for generating passphrases that are both *impossible* for even the most powerful attackers to guess, yet very possible for humans to memorize. The method is called Diceware, and it's based on some simple math.

Your secret password trick probably isn't very clever

People often pick some phrase from pop culture — favorite lyrics from a song or a favorite line from a movie or book — and slightly mangle it by changing some capitalization or adding some punctuation or using the first letter of each word from this phrase. Some of these passphrases might seem good and entirely unguessable, but it's easy to underestimate the capabilities of those invested in guessing passphrases.

Imagine your adversary has taken the lyrics from every song ever written, the scripts from every movie and TV show, the text from every book ever digitized and every page on Wikipedia, in every language, and used that as a basis for their guess list. Will your passphrase still survive?

If you created your passphrase by just trying to think of a good one, there's a pretty high chance that it's not good enough to stand up against the might of a spy agency. For example, you might come up with "To be or not to be/ THAT is the Question?" If so, I can guarantee that you are not the first person to use this slightly mangled classic Shakespeare quote as your passphrase, and attackers know this.

The reason the Shakespeare quote sucks as a passphrase is that it lacks something called *entropy*. You can think of entropy as randomness, and it's one of the most important concepts in cryptography. It turns out humans are a species of patterns, and they are incapable of doing anything in a truly random fashion.

Even if you don't use a quote, but instead make up a phrase off the top of your head, your phrase will still be far from random because language is predictable. As one research paper on the topic states, "users aren't able to choose phrases made of completely random words, but are influenced by the probability of a phrase occurring in natural language," meaning that user-chosen passphrases don't contain as much entropy as you think they might. Your brain tends to continue using common idioms and rules of grammar that reduce randomness. For example, it disproportionately decides to follow an adverb with a verb and vice versa, or, to cite one actual case from the aforementioned research paper, to put the word "fest" after the word "sausage."

Passphrases that come from pop culture, facts about your life, or *anything* that comes directly from your mind are much weaker than passphrases that are imbued with actual entropy, collected from nature.

A short but enlightening video from Khan Academy's free online cryptography class illustrates the point well: https://youtu.be/vVXbgbMp0oY

Make a secure passphrase with Diceware

Once you've admitted that your old passphrases aren't as secure as you imagined them to be, you're ready for the Diceware technique.

First, grab a copy of the Diceware word list, which contains 7,776 English words — 37 pages for those of you printing at home. You'll notice that next to each word is a five-digit number, with each digit between 1 and 6. Here's a small excerpt from the word list:

> 24456 eo
> 24461 ep
> 24462 epa
> 24463 epic
> 24464 epoch

Now grab some six-sided dice (yes, actual real physical dice) and roll them several times, writing down the numbers that you get. You'll need a total of five dice rolls to come up with the first word in your passphrase. What you're doing here is *generating entropy*, extracting true randomness from nature and turning it into numbers.

If you roll the number two, then four, then four again, then six, then three, and then look up in the Diceware word list 24463, you'll see the word "epic." That will be the first word in your passphrase. Now repeat. You want to come up with a seven-word passphrase if you're worried about the NSA or Chinese spies someday trying to guess it (more on the logic behind this number below).

Using Diceware, you end up with passphrases that look like “cap liz donna demon self,” “bang vivo thread duct knob train,” and “brig alert rope welsh foss rang orb.” If you want a stronger passphrase you can use more words; if a weaker passphrase is OK for your purpose you can use less words.

How strong are Diceware passphrases?

The strength of a Diceword passphrase depends on how many words it contains. If you choose one word (out of a list of 7,776 words), an attacker has a one in 7,776 chance of guessing your word on the first try. To guess your word it will take an attacker at least one try, at most 7,776 tries, and on average 3,888 tries (because there’s a 50 percent chance that an attacker will guess your word by the time they are halfway through the word list).

But if you choose two words for your passphrase, the size of the list of possible passphrases increases exponentially. There’s still a one in 7,776 chance of guessing your first word correctly, but *for each first word* there’s also a one in 7,776 chance of guessing the second word correctly, and the attacker won’t know if the first word is correct without guessing the entire passphrase.

This means that with two words, there are $7{,}776^2$, or 60,466,176 different potential passphrases. On average, a two-word Diceware passphrase could be guessed after the first 30 million tries.

And a five-word passphrase, which would have $7{,}776^5$ possible passphrases, could be guessed after an average of 14 quintillion tries (a 14 with 18 zeroes).

The amount of uncertainty in a passphrase (or in an encryption key, or in any other type of information) is measured in *bits of entropy*. You can measure how secure your random passphrase is by how many bits of entropy it contains. Each word from the Diceware list is worth about 12.92 bits of entropy (because $2^{12.92}$ is about 7,776). So if you choose seven words you’ll end up with a passphrase with about 90.5 bits of entropy (because 12.92 times seven is about 90.5).

In other words, if an attacker knows that you are using a seven-word Diceware passphrase, and they pick seven random words from the Diceware word list to guess, there is a one in 1,719,070,799,748,422,591,028,658,176 chance that they’ll pick your passphrase each try.

At one trillion guesses per second — per Edward Snowden’s January 2013 warning — it would take an average of 27 million years to guess this passphrase.

Not too bad for a passphrase like “bolt vat frisky fob land hazy rigid,” which is entirely possible for most people to memorize. Compare that to “d07;oj7MgLz’%v,” a random password that contains slightly less entropy than the seven-word Diceware passphrase but is significantly more difficult to memorize.

A five-word passphrase, in contrast, would be cracked in just under six months and a six-word passphrase would take 3,505 years, on average, at a trillion guesses a second. Keeping Moore’s Law in mind, computers are constantly getting more powerful, and before long 1 trillion guesses a second might start looking slow, so it’s good to give your passphrases some security breathing room.

With a system like this, it doesn’t matter at all that the word list you’re choosing from is public. It doesn’t even matter what the words in the list are (two-letter words are just as secure as six-letter words). All that matters is how long the list of words is and that each word on the list is

unique. The probability of guessing a passphrase made of these randomly chosen words gets exponentially smaller with each word you add, and using this fact it's possible to make passphrases that can never be guessed.

Do I really have to use dice?

This is a longer discussion, but the short answer is: using physical dice will give you a much stronger guarantee that nothing went wrong. But it's time consuming and tedious, and using a computer to generate these random numbers is almost always good enough.

Unfortunately there doesn't appear to be user-friendly software available to help people generate Diceware passphrases, only various command-line-only Diceware projects on GitHub, which power users can check out. Stay tuned for a future post about this.

How to memorize your crazy passphrase (without going crazy)

After you've generated your passphrase, the next step is to commit it to memory.

I recommend that you write your new passphrase down on a piece of paper and carry it with you for as long as you need. Each time you need to type it, try typing it from memory first, but look at the paper if you need to. Assuming you type it a couple of times a day, it shouldn't take more than two or three days before you no longer need the paper, at which point you should destroy it.

Typing your passphrase on a regular basis allows you to memorize it through a process known as spaced repetition, according to promising research into high-entropy passphrases.

Now that you know passphrases, here's when to avoid them

Diceware passphrases are great for when you're typing them into your computer to decrypt something locally, like your hard drive, your PGP secret key, or your password database.

You don't so much need them for logging into a website or something else on the internet. In those situations, you get less benefit from using a high-entropy passphrase. Attackers will never be able to guess a trillion times per second if each guess requires communicating with a server on the internet. In some cases, attackers will own or take over the remote server — in which case they can grab the passphrase as soon you log in and send it, regardless of how strong or weak it is cryptographically.

For logging in to websites and other servers, use a password database. I like KeePassX because it's free, open source, cross-platform, and it never stores anything in the cloud. Then lock up all your passwords behind a master passphrase that you generate with Diceware. Use your password manager to generate and store a different random password for each website you log in to.

How we use Diceware to protect our sources

At *The Intercept* we run a SecureDrop server, an open source whistleblower submission system, to make it simpler and more secure for anonymous sources to get in touch with us.

When a new source visits our SecureDrop website, they get assigned a code name made up of seven random words. After submitting messages or documents, they can use this code name to log back in and check for responses from our journalists.

Under the hood, this code name not only acts as the source's encryption passphrase, but it's also really just a passphrase generated using the Diceware method, but with a digital cryptographically secure random number generator, rather than rolling dice. SecureDrop's dictionary is only 6,800 words long (the developers removed some words from the original word list that could be considered offensive), making each word worth about 12.73 bits of entropy. But this is still plenty enough to make it impossible for anyone to ever simply guess a source's code name, unless they happen to have massive computational resources and several million years.

Simple, random passphrases, in other words, are just as good at protecting the next whistleblowing spy as they are at securing your laptop. It's a shame that we live in a world where ordinary citizens need that level of protection, but as long as we do, the Diceware system makes it possible to get CIA-level protection without going through black ops training.

Thanks to Garrett Robinson for double-checking my math and preventing me from making stupid mistakes.

Top photo: Getty Images

Contact the author:

Micah Lee✉micah.lee@theintercept.com