4-Connected Passwords for Consoles

Andrew Glassner
The Imaginary Institute
Technical Note #12
www.glassner.com

Abstract

Contemporary password managers offer tools for generating new, complex passwords on demand. When we use a virtual keyboard, such as on a game or streaming console, we typically must enter these passwords using a slow process of scrolling and selecting.

We suggest that password managers should provide an option to generate 4-connected passwords, where each character is a direct neighbor of its predecessor on the virtual keyboard. We believe such passwords are easier and more pleasant to enter, and less error-prone.

To judge their security, we look informally at the entropy of 4-connected passwords. We show that passwords of about the same length as game download codes have an entropy that is competitive with "character soup" passwords of about half their length.

1. Scroll-and-Select Passwords

Passwords seem to be inescapable in today's computing environments, so it's important that we use systems that encourage their safe and convenient use.

Most password studies have focused on either protecting from "soft," or social attacks (such as "phishing"), or "hard," or cryptographic, attacks (such as brute-force dictionary searches).

For many years, the common advice for password structure followed guidelines like, "use no recognizable words, but use a mixture of upper and lower case, at least one number, and at least one symbol." This advice was due to a 2003 report from the US standards agency NIST [7]. In 2017 the lead author of that report, Bob Burr, said that those original guidelines were mostly based on guesswork, and turned out to not be very good [5]. Around the same time, NIST issued new guidelines [8].

We're interested here in adding another dimension to what makes a "good" password: the ability to enter it easily on a virtual keyboard that is controlled by a external device that does not offer random access to each key. These scenarios are frequent in the living room, where we use game controllers and TV remotes to navigate among a picture of a keyboard on a screen, as in Figure 1.

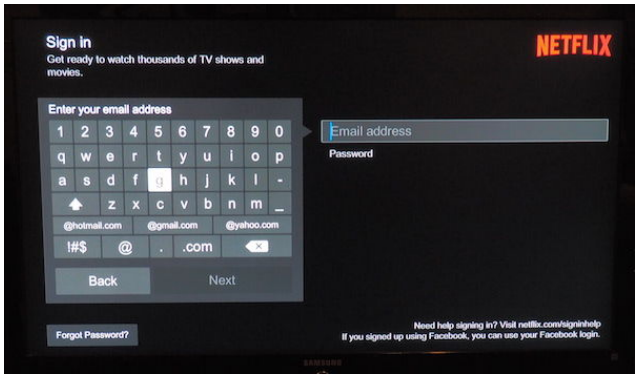

Figure 1. A NetFlix login screen. Image from [4].

We typically navigate these keyboards with a controller that provides buttons and joysticks. To enter a password, we scroll the cursor horizontally and vertically until it's over the character we want, and then we press a hardware button to enter it. If we make a mistake, there's typically a dedicated *erase* button. Most keyboards also offer a *shift* button, like the arrow in the lower-left of Figure 1. This lets us toggle between lower-case and upper-case characters, and the non-letter symbols associated with the keys. We call a password entered with this type of mechanism a *scroll-and-select* password.

Entering a typical "character soup" password, such as `jd%YHd7Bb*R`, can be a frustrating process of scrolling, overshooting, correcting, clicking, shifting, and unshifting. We propose that such systems would be more pleasant if our passwords used *4-connected sequences*: that is, each symbol is a direct east, west, north, or south neighbor of the previous symbol. We believe this would make entry easier and faster. To make sure such passwords are still secure, we briefly consider how resistant they are to being broken (or guessed) by an adversary.

Figure 2. A virtual keyboard from the Microsoft XBox. Image from [2].

2. Password Entropy

The *entropy* of a password, usually measured in bits, gives us an estimate of its complexity. The entropy is a concept from information theory designed to measure the amount of surprise, or unexpected content, in a message [9]. In our case, a higher entropy corresponds to a password that's more difficult to guess.

The basic formula to compute a password's entropy depends on two quantities. We imagine that the password is composed by picking *entries* from a *pool*. If the pool contains all the symbols on a keyboard, then the entries would be individual characters. For a pool size P, and a password made up of L entries, the entropy H is given by

$$H = \log_2(P^L)$$

3. Console Keyboards

Home game and streaming consoles are ubiquitious, and provide the virtual keyboards that we use to enter logins, passwords, and download codes.

We'll begin by considering a virtual keyboard for the Microsoft XBox One, shown in Figure 2.

Console virtual keyboards vary by region, language, and even release, but the images we show here are typical. Some virtual keyboards are toroidal (so scrolling off of any edge places the cursor at the opposite edge), but this is not a universal feature, so we'll ignore this feature.

Consider a password that is an arbitrary sequence of 4-connected characters. For simplicity, we won't treat the first character as a special case, and we'll disallow immediate duplication of characters (this remains a NIST recommendation). We will also assume that the hardware controller allows us to switch between character sets with little effort.

What is the average number of next character choices from any given key? Figure 2 shows that the keyboard is a grid of 4 by 10 cells. The 24 characters around the perimeter each have 3 neighbor keys, offering 6 characters, plus the character from shifting that key itself, for a total of 7 (e.g., the `1` key is adjacent to the `2`, `q`, and `w`, and by shifting we get access to the symbols associated with each, plus the symbol associated by shifting the `1`). The 16 keys inside the grid each have 4 neighbors, plus their shifted character, for a total of 9. The average number of neighbors per key is

Figure 3. A virtual keyboard from the Sony PlayStation 4. Image from [3].

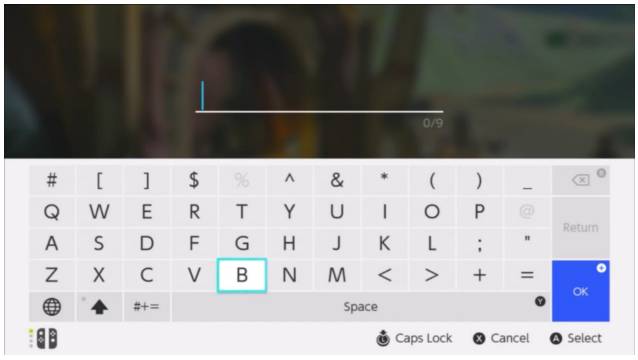

Figure 4. A virtual keyboard from the Nintendo Switch. Image from [6].

	P value
XBox	7.8
PS4	7.8
Switch	7.82

Table 1. P values for 3 consoles

.

then $[(24 \times 7) + (16 \times 9)]/40 = 7.8$.

The Sony Playstation 4 virtual keyboard, shown in Figure 3, has the same layout as the Xbox One virtual keyboard.

The Nintendo Switch, shown in Figure 4, uses a 4-by-11 grid. Its average neighbors is only slightly higher, at 7.82.

Table 1 summarizes the values of the pool size P for the three keyboards.

Using the entropy formula above, the entropy H for a 4-connected password of length L characters, assuming a P value of 7.8, is given by

$$H_{\text{4-connected}} = \log_2(7.8^L)$$

The result is given in bits. Figure 5 shows the entropy for a 4-connected password as a function of length L. We also show the entropy for a 4-connected password for a controller that doesn't provide hardware case shifts, and a

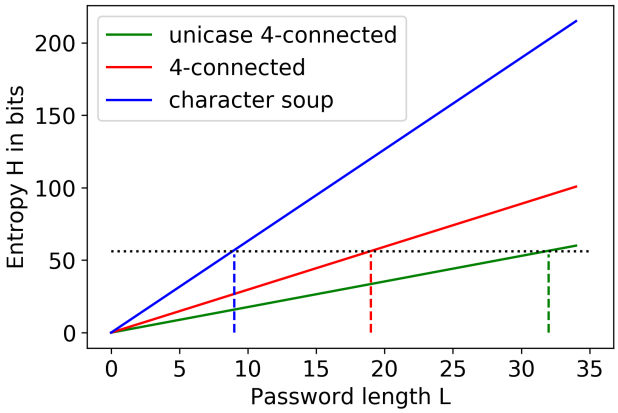

Figure 5. Entropy for 4-connected passwords with hardware shift (red, "4-connected") and without (green, "unicase 4-connected"), and "character soup" passwords (blue), by length L. We get about 56 bits of entropy from either 9 random characters, or 19 4-connected characters.

"character soup" password, where each entry can be any one of 80 different characters (that is, $P = 80$).

4. How Many Bits?

How many bits of entropy should we use for a secure password?

Opinions on this question rage online, and the recent NIST guidelines don't give us a direct answer (though their suggestion to limit "character soup" passwords to 64 characters suggests a massive upper limit of $\log_2(88^{64}) \approx 413$ bits [8]).

On the 1Password blog, Jeffrey Goldberg suggests that a password with 56 bits of entropy "would cost tens of millions of dollars to crack" [1]. This seems like an acceptable level of security for most home users protecting their streaming service passwords and online gaming IDs. As Figure 5 shows, that corresponds to about 9 random characters, or about 19 characters using the 4-connected scheme.

Is 19 characters a reasonable length for an infrequently-entered password? We can get some guidance from the console manufacturers, who use online codes to redeem purchases and bonuses. To redeem an online code on the XBox One, PS4, and Switch, users must enter a 24-character, 12-character, and 16-character sequence, respectively. Note that these are "character soup" sequences, potentially requiring a lot of horizontal and vertical scrolling. This suggests that a 19-character sequence, which is easier and faster to enter, is reasonable for occasional password entry.

If a controller does not offer a hardware shift feature, then $P = 3.4$, and we'd need about 32 characters for 56 bits of entropy. With both hardware shift and a toroidal keyboard, $P = 9$, so a 4-connected password would need only 18 characters for 56 bits of entropy.

5. Summary

We advocate that password managers provide an option for generating 4-connected passwords. These are easier and faster to enter on scroll-and-pick systems, and we suggest they are less error-prone as well. A casual look at the entropy of such passwords shows that the security of a 4-connected password of 19 characters is comparable to that of a "character soup" password of 9 characters.

References

[1] Jeffrey Goldberg, *How long should my passwords be?*, 1Password Blog, Oct 10, 2018. https://blog.1password.com/how-long-should-my-passwords-be/ 3

[2] Matthew Hartman, *Evolution from Xbox 360*, 2017. http://www.matth-design.com/xbox-vk/ 2

[3] JagAtPlay, March 2015. http://jagatplay.com/wp-content/uploads/2015/03/Playstation-4-Yukimura-jagatplay-21-600x338.jpg 2

[4] Manish, *NetFlix Sign In Screen*, July 27, 2015. https://forums.roku.com/viewtopic.php?t=88023 1

[5] Robert McMillan, *The Man Who Wrote Those Password Rules Has a New Tip: N3v$r M1d!*, The Wall Street Journal, August 2017. https://www.wsj.com/articles/the-man-who-wrote-those-password-rules-has-\a-new-tip-n3v-r-m1-d-1502124118 1

[6] modojo, *The Nintendo Switch Works With USB Keyboards*, March 7, 2017. https://modojo.com/article/23565/the-nintendo-switch-works-with-usb-keyboards 2

[7] NIST, *SP 800-63-2 Electronic Authentication Guideline*, August, 2013. https://csrc.nist.gov/publications/detail/sp/800-63/2/archive/2013-08-29 1

[8] NIST, *SP 800-63-3 Digital Identity Guidelines*, June 22, 2017. https://pages.nist.gov/800-63-3/ 1, 3

[9] C. E. Shannon, *A Mathematical Theory of Communication*, Bell System Technical Journal, v. 27, pp. 379-423, July 1948. http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html 2