Analysis of the Security of Unicode Password Through Enumerative Combinatorics

Fatih Nararya Rashadyfa Ilhamsyah - 13521060
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13521060@std.stei.itb.ac.id

***Abstract*— This paper examines the theoretical security improvement of passwords by expanding the characters that they can use from ASCII to Unicode. The result is a reminder of the password's inherent flaws, the user of password.**

***Keywords*—password, security, ASCII, Unicode**

Nomenclature

S	A set of characters that can be used for a password
L	Length of password
N	Number of characters that can be used for a password
h	Random guesses required to guarantee the cracking of a password
H	Bits of entropy
P	Password

I. Introduction

Passwords as a tool of identity verification have existed since ancient times, long before computers were even invented. The first implementation of the password into computers was in 1961 in an operating system called Compatible Time-Sharing System (CTSS) created by MIT. The password's old age and the rapid emergence of technologies aiming to replace the password – such as biometrics and multi-factor authentication – in recent years make passwords seem obsolete. Yet, it is still the main method of user authentication in our modern age, from logging into websites to unlocking our devices.

Almost all passwords that are used today are encoded in ASCII or can only use ASCII characters. This gives those passwords a relatively limited number of characters that can be used. If we were to extend the set of characters that can be used in passwords by switching to Unicode (which has substantially more characters), we could theoretically increase the security of passwords by multiple orders of magnitude.

However, theoretical improvement, however massive, means nothing if it cannot be reflected into the real world. Thus, in determining whether such a change would actually increase the security of the password, one must also assess whether current real world conditions would allow the maximum utilization of the Unicode password, consequently tapping its hypothetical security benefits.

II. Theoretical Basis

A. Enumerative Combinatorics

Enumerative Combinatorics is the oldest subfield of combinatorics that has originated since ancient times. It is concerned with counting the possible ways to arrange a set of objects without enumerating each one of them [1]. It will be the main tool utilized in analysis of password security where the set of objects would be a set of characters and their arrangement is the password.

B. Rule of Sum

Rule of sum is one of the basic principles that combinatorics is founded upon. This principle describes the number of ways for how a number of actions could be done in conjunction with an arbitrary number of other actions.

Given n number of actions p_i each with a_i ways to do them, there are $a_1 \times a_2 \times \ldots \times a_n$ ways to do all of the actions in parallel (each action is not mutually exclusive) [2]. The last part is important to distinguish the rule of sum from the rule of product, another basic principle of combinatorics that is used when counting mutually exclusive actions.

C. Measuring the Security of a Password

To see how we can measure the security of a password P we first must see how to create P. Suppose that each characters in P is actually a "slot" into which we can insert a character from character set S which has N number of characters. If P has a length of L then there would be L number of slots. Assuming that all of the characters in S are as equally likely to be picked (an important criteria that will be brought up later) for any arbitrary slot, we can apply the rule of sum to see how many possible password combinations are possible. The number of possible password combinations is the number of guesses an attacker would have to make for any arbitrary password P to be

guaranteed to be cracked, called hereon as h. We can the derive the relationship of N, L, and h into the equation below.

$$h = N^L \quad (1)$$

A password P with a higher value of h would take a longer time to be cracked. This corresponds to P being a better password since it is more secure. Although h sufficiently describes the security of a password, it can be quite a lengthy quantity to express because of the tendencies of h to experience combinatorial explosion as seen in (1). So instead, there is another metric called bits of entropy that is more commonly used as a measure of password's security entropy because bits of entropy is more concise even in the face of combinatorial explosion [3]. Bits of entropy is a quantity derived from h with the relationship given by the equation below [4].

$$H = log_2(h) \quad (2)$$

Rearranging (2) can give us the number of guesses required by P given the bits of entropy it has.

$$h = 2^H \quad (3)$$

The convenience and brevity of using bits of entropy is apparent since for a password with, for example, 1,048,576 possible guesses, we could instead say the password has 20 bits of entropy.

From (3) we can see that the number of guesses grows exponentially in respect to bits of entropy, meaning that even a seemingly small change in H corresponds to a significantly higher h. This is an important aspect to remember later as we quantify securities of different kinds of password in bits of entropy instead of required guesses.

D. Matching Password Security With Different Character Set by Varying The Length of the Password

As we've seen in (1), there are two quantities that directly influence the security of a password : password length (L) and the number of characters that can be used (N). Intuitively, this means that two passwords with different size of N could be made equally secure by tweaking the length L of each password.

Now we try to derive the intuition we have gained previously in a formal manner. Suppose there is a password P_1 with length L_1 that uses character set S_1 consisting of N_1 characters. P_1 now has bits of entropy H as formulated by (1) and (2). Then, there is another password P_2 with length of L_2 created out of character set S_2 that consists of N_2 characters. Now we'd like for P_2 to also have bits of entropy H by tweaking L_2. Knowing that both password has bits of entropy H, we can derive the formula of L_2 by equating H of the respective password and utilizing (1) and (2).

$$L_2 = L_1 \times log_{N_2}(N_1) \quad (4)$$

E. ASCII : Numeric, Alphanumeric, Printable

ASCII (American Standard Code for Information Interchange) is how letters, numbers, and other symbols we commonly use are encoded for use by a computer. The standard was first created by the ASA (American Standards Association) in 1963 and was last updated in 1986.. It assigns those characters by a number in the range 0-127 – because of its size of only one byte – that then can be decoded by computers. There are 4 subsets of ASCII characters that we are interested in because of their commonality for use in passwords : numeric, lowercase letters, letters, alphanumeric, and printable.

Numeric is quite self-explanatory, it is the set of numbers from 0-9. Lowercase letters contain letters in lowercase. Letters are all letters in uppercase or lowercase. Alphanumeric is a superset of numeric, consisting of numbers and all letters, uppercase and lowercase. Lastly, printable is a superset of both, containing numbers, letters, and symbols we commonly use such as *, ^, etc.. There is an important distinction between printable ASCII and the whole ASCII set of characters since certain ASCII characters are practically invisible and not meant to be shown, such as '\n' that represents newline. Below are the number of characters of the aforementioned character sets.

TABLE I
N OF EACH S

S	N
Numeric	10
Lowercase letters	26
Letters	52
Alphanumeric	62
Printable ASCII	95

Note that because there are as many lowercase characters as uppercase characters, the uppercase letters characters set are not used in this paper since it's sufficiently represented by lowercase letters character set.

F. The Unicode Standards

The Unicode Standard is, in principle, identical to ASCII, just with a (significantly) wider range of symbols. The Unicode Consortium is a non-profit organization responsible for maintaining and developing the Unicode Standard. It was first proposed in 1987 and continues to be in use today with its

latest version 15.0 released on the 13th of September 2022.

The importance of the standard is its aim of accommodating all symbols and characters used in every writing system. To quote from the Unicode Consortium [8], "Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language." This is the reason why Unicode is now the widely used standard used to represent characters in computing, from operating systems to web pages.

The Unicode Standard today consists of 149,186 characters. Its huge size is due to its aforementioned aim of representing all characters in all writing systems. Although the Unicode standard accommodates hundreds of alphabets, emojis, and even musical notation, its bulk is actually CJK (Chinese, Japanese, and Korean) characters. It is important to note that although 149,816 characters are all of the characters that are defined within the standard, it can actually accommodate many more characters. There are currently 825,279 reserved slots for future use in the standard [9].

III. Security of Passwords With Various Encodings

Using the method of measuring security of a password shown in (1) and (2), we can now calculate the security of a password given its length and the number of possible characters available for their creation, which depends on the characters set that it uses. We will examine the security of four distinct character sets that have been mentioned in the previous section : numeric, lowercase letters, letters, alphanumeric, printable ASCII, and Unicode. The length of the password will be kept as a variable in these calculations as shown below.

TABLE II
SECURITIES OF DIFFERENT S AS A FUNCTION OF L

S	Bits of entropy (H)
Numeric	$L \times log_2(10)$
Lowercase letters	$L \times log_2(26)$
Letters	$L \times log_2(52)$
Alphanumeric	$L \times log_2(62)$
Printable ASCII	$L \times log_2(95)$
Unicode	$L \times log_2(149,186)$

To contextualize and give more meaning to the values in Table II, here is what the bits of entropy of each respective character set would be given $L = 8$, which is the minimum length used in most passwords today.

TABLE III
H OF EACH S GIVEN A PASSWORD LENGTH OF 8

S	H
Numeric	~ 26.58
Lowercase letters	~ 37.60
Letters	~ 45.60
Alphanumeric	~ 47.63
Printable ASCII	~ 52.56
Unicode	~ 130.50

Using the values in Table II, we can use (2) to calculate the number of guesses for each character set. We can see how there are much more guesses required to crack a Unicode password ($2^{130.5}$) than an ASCII password ($2^{52.56}$) of 8 characters long.

IV. Matching the Security of Unicode Password

It has been demonstrated in the previous section that Unicode passwords are, at least in theory, dozens of orders of magnitude more secure than ASCII passwords, as shown in Table II. However, as also have been demonstrated by (4), we can match the security of any password P_1 that uses character set S_1 with another password P_2 that uses character set S_2. Suppose there's a Unicode password P_U of length L_U, we can always match its security with a non-Unicode password P_B of length L_B, no matter what character set that P_B uses as long as we pick L_B wisely. Using (4) and applying it to the 3 aforementioned character sets other than Unicode, we can see the relationship between L_B and L_U presented below.

TABLE IV
REQUIRED LENGTH FOR DIFFERENT S TO MATCH SECURITY OF UNICODE PASSWORD WITH ARBITRARY LENGTH

S	L_B (in L_U)
Numeric	$\sim 5.17 \times L_U$
Lowercase letters	$\sim 3.67 \times L_U$
Letters	$\sim 3.02 \times L_U$
Alphanumeric	$\sim 2.88 \times L_U$
Printable ASCII	$\sim 2.62 \times L_U$

Table IV has shown that one can always use a longer password to achieve the same security as the Unicode password instead of actually using the Unicode character set. To contextualize the values in table III, below are the L_B of each respective character sets given L_U of 8, which is the commonly used minimum length for password. It is important to highlight that since we'd like P_B to be as secure as P_U, the numbers have been rounded up during calculation to prevent overestimation of security. The results are shown below.

TABLE V
REQUIRED LENGTH FOR DIFFERENT S TO MATCH THE SECURITY OF UNICODE PASSWORD 8 CHARACTERS LONG

S	L_B
Numeric	42
Lowercase letters	30
Letters	25
Alphanumeric	24
Printable ASCII	21

V. High Likelihood of Underutilization

All of the security improvement of the password – by extending the characters set from ASCII to Unicode – that have been mentioned previously rests on the yet-to-be-proven assumption that all 149,186 characters are equally likely to be picked by a user as a character in a Unicode password. If Unicode passwords were to be implemented but most users still only use ASCII characters then any potential attacker could limit the scope of their brute-force attacks to ASCII characters, nullifying all of the theoretical security improvement of those user's passwords.

Proving the correctness of the aforementioned assumption is not easy, but we can start by simply looking at the current state of the ASCII password and then try to make an interpolation for the Unicode password. To that end, we will be looking at the 200 most common passwords of three consecutive years : 2019, 2020, and 2021, according to the data gathered from NordPass [6].

TABLE VI
CHARACTER COMPOSITION OF THE 200 MOST COMMON PASSWORD OF 2019 FROM NORDPASS DATABASE

Character Composition	Number of Password	Percentage (%)
Numeric	30	15
Lowercase letters	140	70
Letters	1	0.5
Alphanumeric	26	13
Alphanumeric & Special Characters	3	1.5

TABLE VII
CHARACTER COMPOSITION OF THE 200 MOST COMMON PASSWORD OF 2020 FROM NORDPASS DATABASE

Character Composition	Number of Password	Percentage (%)
Numeric	56	28
Lowercase letters	90	45
Letters	0	0
Alphanumeric	54	27
Alphanumeric & Special Characters	0	0

TABLE VIII
CHARACTER COMPOSITION OF THE 200 MOST COMMON PASSWORD OF 2021 FROM NORDPASS DATABASE

Character Composition	Number of Password	Percentage (%)
Numeric	64	32
Lowercase letters	78	39
Letters	1	0.5
Alphanumeric	57	28.5
Alphanumeric & Special Characters	0	0

Aggregating the data from Table VI, VII, and VIII we will see the relative proportions of each character composition.

TABLE IX
CHARACTER COMPOSITION OF THE 200 MOST COMMON PASSWORD OF 2019-2021 FROM NORDPASS DATABASE

Character Composition	Number of Password	Percentage (%)
Numeric	150	25
Lowercase letters	308	51.34
Letters	2	0.33
Alphanumeric	137	22.83
Alphanumeric & Special Characters	3	0.5

Despite having the option to use 33 characters in ASCII that are not alphanumeric characters such as %, ^, &, etc. almost none of the most popular passwords uses those characters. Even worse, a substantial number of passwords do not even use the full alphanumeric set of characters, opting to use only numbers or letters instead. The data in Table VI, VII, VIII shows that even the current ASCII password which has multiple orders of magnitude less characters compared to Unicode password is not utilized to its full potential, handicapping its security immensely.

Although the data from NordPass is certainly an eye-opener on the underutilization of ASCII password, it only looks at a very small portion of all ASCII passwords because it only looks at the most popular password and only 200 of them. Fortunately, there has been a lot of research regarding user behavior in choosing passwords that relies on much bigger data samples.

Unfortunately, cross checking the finding from examining the NordPass data against bigger data samples still points to the same conclusion that users gravely undermine the security of their own passwords. Guven, Boyaci, and Aydin [7, Table 5] have found from 10 million passwords gathered from data breaches that the vast majority of passwords (>80%) are composed of only numeric characters and less than 1% utilizes special characters and uppercase letters. This only confirms the insight gained by analyzing NordPass data.

Through the presented data, we can safely conclude that ASCII passwords are currently severely underutilized. It then becomes obvious that Unicode password, even if implemented, would not achieve any improvement to security by the simple fact that people would stick to passwords that purely consist of numeric or letter characters as they already do now with ASCII password.

VI. Conclusion

On paper, Unicode passwords are multiple orders of magnitude more secure than ASCII passwords. However, not only that such security improvements, although certainly impressive, are achievable with the existing ASCII password by merely multiplying the length of the password by a constant factor, those security improvements will only remain on paper as long as users only use a small subset of characters available to them.

Although the data and research used in the previous section to show how users severely self-sabotaged the security of their own password is relatively recent, this problem is not a new phenomenon. Thompson [10] has already mentioned the issue and its severity back in 1979 in the context of security systems for the Unix. This tricky and hard problem then seems to be prevalent and ever-present, without any possible technical fixes due to its origin of not the computer, but the human. It only adds to the long-standing and increasingly common opinion held by experts and laymen alike that the password is an antiquated technology in need of replacement or if not, at least it shows fixing the problem of the password is not as simple as stuffing more characters into it.

VII. Appendix

This simple program written in Python was used to process the common password data from NordPass for Table VI, VII, and VIII. The raw data in txt format is included in the Github repository.

VIII. Acknowledgment

First and foremost, this paper would not have been possible without the abundant resources provided by IF2120 Discrete Mathematics class, especially the lectures of Dr. Nur Ulfa Maulidevi, S.T., M.Sc., the lecturer for my class.

The author would also like to give many thanks to authors and companies who have conducted copious research on the user's trend of password creation without which it's not possible to assess the effectiveness of Unicode password and draw the conclusion of this paper.

Lastly, the author would like to thank everyone of his peers who has assisted the author during the writing of this paper by giving advice, feedback, criticism, and insights that the author has missed or lacked.

References

[1] R. P. Stanley, *Enumerative Combinatorics*. Belmont, CA: Wadsworth, 1986.

[2] R. Munir, *Kombinatorial (Bagian 1)*. Bandung, West Java, 2022.

[3] bk2204 (2021, Aug. 6). bk2204 Answer to "Why is entropy measured in bits?" [Forum]. Available: https://security.stackexchange.com/a/254049 [Accessed: Dec. 9, 2022].

[4] J. Munroe (2019, Jun. 24). Jordan Munroe's Answer to "Are special characters in passwords even effective?" [Forum]. Available: https://qr.ae/pr1wsA [Accessed: Dec. 9, 2022].

[5] *The Unicode Standard*, 15.0.0, The Unicode Consortium, Mountain View, CA, 2022.

[6] "Top 200 most common passwords." NordPass. https://nordpass.com/most-common-passwords-list/ [Accessed: Dec 9, 2022].

[7] E. Y. Guven, A. Boyaci, and M. A. Aydin, "A Novel Password Policy Focusing on Altering User Password Selection Habits: A Statistical Analysis on Breached Data," in *Computers and Security*, vol. 113. February 2022, https://doi.org/10.1016/j.cose.2021.102560.

[8] "What is Unicode?." Unicode. https://www.unicode.org/standard/WhatIsUnicode.html. [Accessed: Dec. 11, 2022].

[9] "Unicode® Version 15.0 Character Counts." Unicode. https://www.unicode.org/versions/stats/charcountv15_0.html. [Accessed: Dec. 11, 2022].

[10] R. Morris, and K. Thompson, “Communications of the ACM,” in *Computers and Security*, vol. 113, no. 11, pp. 594-597 November 1979, https://doi.org/10.1016/j.cose.2021.102560.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2022

Fatih Nararya Rashadyfa Ilhamsyah (13521060)