Technology and Language Технологии в инфосфере.
2021. 2(1). 98-121
https://doi.org/10.48417/technolang.2021.01.08

*Special Topic: **In the Beginning was the Word – The Word as a Technical Artefact***

Non-linguistic Systems as a Way to Make a Password Secure but Memorable

Viktoria Vorotnikova[1](✉) and Sergey Karlin[2]
[1]Peter the Great St. Petersburg Polytechnic University (SPbPU), St. Petersburg, Polytechnicheskaya, 29, 195251, Russia
vorotnikova.vd@edu.spbstu.ru
[2]University of veterinary and pharmaceutical sciences Brno, Palackeho tr. 1946/1, 612 42 Brno, Czech republic
V20061@vfu.CZ

Abstract

This article is based on the study of ways to create a secure password by integrating symbols from non-linguistic sign systems, in order to combine cryptographic strength and ease of memorization. This is relevant, as the old ways of complicating the password become obsolete, in view of their triviality and, as a result, susceptibility to hacking. Our research is based on the use of a system of symbols from various fields of interest (chemistry, programming, music, etc.) in the password. We take into account the individual preferences of users, so that it would be easier for them to build an associative chain when remembering a password, and also consider the susceptibility of passwords obtained using the password techniques we proposed to the most common cyber-attacks. The respondents created one password on their own, and the second with the help of the proposed methods. The complexity and security of the password was estimated in terms of entropy, as well as using specialized programs. Using the proposed methods reduced the number of insecure keys.

Keywords: Password; Entropy; Reliability; Cyber Attacks; Cryptographic Strength; Symbol Systems; Passphrase; Keys; Hacking; Symbol

Аннотация

Данная статья базируется на исследовании способов создания надежного пароля с помощью интеграции символов из нелингвистических систем знаков, с целью совмещения криптостойкости и легкости запоминания. Старые способы усложнения пароля изживают себя, в виду их тривиальности, как следствие – подверженности взлому. Исследование основано на использовании системы символов из различных сфер интересов (химия, программирование, музыка и др.) в пароль. В работе принимались во внимание индивидуальные предпочтения пользователей, чтобы им было проще построить ассоциативную цепочку при запоминании пароля. Рассмотрена подверженность паролей, полученных с помощью предложенных техник, к наиболее распространенным кибератакам. Респонденты создавали один пароль самостоятельно, а второй с помощью предложенных способов. Сложность и надежность пароля оценивались в понятиях энтропии, а также с помощью специализированных программ. Использование предложенных способов сократило количество небезопасных ключей.

Non-linguistic Systems as a Way to Make a Password Secure but Memorable

INTRODUCTION

The Internet, electronic services and social networks have significantly changed the life of a modern person with a confident step (Bylieva et al., 2020). People, using these facilities, in most cases use their personal information, give sites access to their private communication within the network and personal data. When registering on a certain resource, the user sets a password, the purpose of which is to keep the confidentiality of information. Many sites that require a password to be entered, as well as a large daily flow of information, often push a person to create an easy password that he can easily and quickly remember. This leads to such negative consequences as: leakage of personal data, loss of user access to the personal information he needs, as well as loss of funds from accounts.

Concerns about his data are forcing the user to use a difficult password to enter the site. A complex password is considered to be a combination of different letters (uppercase and lowercase), numbers and symbols, in an amount of not less 12 signs, which does not fall into the list of most frequently used passwords (for example, Google, Apple account requirements). In most cases all the requirements will be met by a random set of characters, which is difficult to crack using a selection method, but at the same time it is difficult to remember. This is where the main problem lies when setting an account password. Linking cryptographic strength and ease of memorization is rather a difficult task. With the growing number of methods for cracking passwords, as well as with the expansion of the databases of merged and frequently used passwords, the use of words and standard structures of natural language (dates, addresses, etc.) is losing its relevance. Thus, modern passwords approach cherished doors, not only in terms of the functions performed, but also in terms of the dominant form of expression –meaningless for the uninitiated. That is why it is worth turning to alternative systems of symbols when creating a password, as this will preserve simplicity and ease of perception, without reducing its reliability.

LITERATURE REVIEW

As password-based authentication is widely used today, passwords are always susceptible to cyber-attacks, and it is difficult for users to create complex passwords (Haeussinger & Kranz, 2017; Ur et al., 2017). Let's take a quick look at the main ways to crack passwords because of the more complicated passwords.

The most common are: "Brute force" and dictionary attack. The brute force method is based on an automated sequential iteration of various combinations of signs. This method is certainly successful, as the number of signs in the password is determined. Therefore, Brute force allows you to crack any password. However, complex combinations can take years to determine. It is possible to counter this attack by increasing the number of characters in the password and the absence of repeated signs one after another. The use of personal information in a password, for example, date of birth, first or last name, phone number, also makes it more vulnerable to brute force attacks, as it is with the substitution of such basic combinations that the selection of a key from a user account or any other personal information begins (Bosnjak et al., 2018; Guo & Zhang, 2018; Narendar et al., 2020).

The dictionary method is another popular hacker method for cracking passwords and keys. Analysis of attacks on personal data shows that the percentage of hacked passwords using the dictionary method ranges from 17% to 24% of the total number of attacked passwords (Guo & Zhang, 2018; Singh & Pandey, 2019).The essence of this method lies in the fact that the key that must be matched is checked for matches in various bases of dictionary words and expressions that are often used by people as a password. The larger the quantitative base of the dictionary, the higher the probability of guessing the password. Some mechanisms are also tuned to look for typical sign substitutions in dictionary words, such as "a – @", as well as search for words letters of which are simply written in reverse order (Guo & Zhang, 2018; Singh & Pandey, 2019).

In general, the security and usability of entering passwords is opposite, as secure keys with a high degree of protection are difficult to remember, and passwords that are easy to remember can be cracked fairly quickly. More often than not, a user has many accounts that require passphrases to be protected, but it is quite difficult to create and remember multiple secure combinations (Li et al., 2018; Guo et al. 2020). Users tend to choose easy-to-remember passwords that include names, short words, dates, and patterns, resulting in a quick brute-force attack (Alomari et al., 2019; Bonneau, 2012; Veras et al., 2012; Yan et al., 2004). Many Internet services, including those offered by Microsoft and Google, have established lists of common, weak passwords that are prohibited from using on their servers (Habib et al., 2017).

Remembering a password is based solely on a person's memory, which has limited capabilities. Stanton & Greene (2014) studied how memorable complex character strings of varying lengths are memorable, which can be used as passwords with a higher degree of protection against cracking. Participants were shown a password once and asked to remember and recollect it. The results showed that the longer a character string, the longer it takes a person to remember, recall, and type it. Longer lines also increased the chance of errors. Another experiment aimed at studying the effect of password length on memorability found that the system-generated 4-digit combinations outperform 6-, 7-, and 8-digit combinations in long-term memorability at 48 hour intervals (Huh et al., 2015)

Ur et al. (2016) investigated how users rate the security and memorability of a range of passwords. They found that participants perceived the password to be significantly less memorable if it was longer or contained numbers. Some organizations and researchers on the topic have proposed some mnemonic techniques such as passphrases (Bonneau & Shutova, 2012) and mnemonic strategies (Kuo et al. 2006) to improve the ease of use and remembering of passwords.

Seitz (2017) argued that existing password generation methods do not account for individual differences between users and cannot effectively balance the usability and security of passwords relative to a specific person. Users with different personalities and interests have different preferences when creating a strong password (Bosnjak & Brumen, 2016; Guo et al., 2020; van Schaik et al., 2017).

Nowadays, some users use mnemonic hints to generate passwords. Passwords created with the help of these techniques are believed to be fairly easy to remember and more secure than simple passwords of similar difficulty to remember. For example, the authors of the article Bei et al. (2019) considered 4 techniques that can solve the problem of combining memorability and password complexity:

1) "SenSub: Sentence substitution" – replacing each word of a well-remembered sentence with the first letter of this word

2) KbCg: Keyboard change – a memorable word moves one by one across the keyboard button in any direction

3) UsForm: use of an arithmetic expression written in signs, partly, numbers, partly words. As passwords, the authors of the article propose to use formulas, for example, mathematical and write them down or describe it in words with a whole formula, or replace, for example, the signs in the formula with words.

4) SpIns. Special character insertion. The authors of the article suggest inserting easy-to-remember similar symbols into the base password, for example, a – @

Password complexity is estimated by different methods, but most often the concept of entropy is measured in bits. Password strength or entropy is the amount of information per elementary message from a source generating statistically independent messages. We can say that this is a conditional coefficient that shows the distribution of unique elements in the data array. Instead of the number of attempts required to accurately determine the password, the base 2 logarithm of this number is taken, which is called the number of "entropy bits" in the password. To crack a 40-bit password using brute force, you need to make 240 attempts, checking all possible options (Volokitin & Volokitina, 2020).

The N-bit entropy corresponds to the uncertainty of the choice of passwords (for example, generated by a random number generator), the entropy is calculated simply, it is equal to the logarithm based on two numbers of possible passwords for the given parameters. To calculate the entropy of a random password, use the formula [1]:

$$H = \log_2 N^L = L\log_2 N = L\frac{\log N}{\log 2} \qquad [1]$$

where N is the number of possible password characters; L – the number of characters used in the password; Entropy H is measured in bits. Increasing L or N will multiply the generated password. If the password is generated not by an impartial random number generator, but by a person, then calculating its entropy is much more difficult. The most common approach to calculating the entropy in this case is the approach proposed by the American Institute NIST: – the entropy of the first character of the password is 4 bits; – the entropy of the next seven signs of the password is 2 bits per sign; – entropy from 9 to 20 signs – 1.5 bits per sign; – all subsequent signs have entropy of 1 bit per sign.

If the password contains uppercase and non-alphanumeric signs, its entropy is increased by 6 bits. According to this approach, a human-selected eight-sign password with no uppercase letters or non-alphabetic signs is estimated to have an average of 18 bits of entropy (Volokitin & Volokitina, 2020)

There are also programs for assessing password strength. For example, programs "zxcvbn" is based on the analysis of 30 thousand common passwords, first and last names of US citizens from the census data, searching for classic combinations such as "123", "qwerty", phrases from movies and TV shows, as well as searching for pattern combinations: phone numbers, date of birth. Using "zxcvbn" allows for more flexibility in determining password strength, as it covers not only dictionaries, password length, different case and the presence of signs, but also various combinations of signs that are often used by users to complicate passwords (Wheeler, 2016).

USING NON-LINGUISTIC SYSTEMS TO CREATE A PASSWORD

Despite the proliferation of new methods of protection against unauthorized access, in particular those related to bio-identification (fingerprints, iris, voice, face, etc.), the need for strong passwords entered from the keyboard is very high today. The most common for natural languages of combinations of signs – words and dates – have a clear meaning, therefore, they are remembered, but are not suitable as passwords, as they are very easy to crack by guessing. Senseless combinations of letters, words, symbols and numbers are a reliable system of protection, but they are poorly stored in a person's memory, as he cannot associate this reliable combination with anything. It is possible, of course, to transfer the memorization function to technical devices – for example, a secure hardware password manager (Gupta et al., 2020). But if we consider only the capabilities of a person, then, the problem of memorizing a rather complex password is associated with giving it meaning. This can be solved either with the help of mnemonics(Al-Ameen et al., 2015), aimed at creating associations with a randomly generated obviously cryptographically strong password, or creating a password that makes sense for the user, but it is not a lexical unit. The first step in this direction was the idea of using a multilingual password (Bonneau & Shutova, 2012; Maoneke et al., 2020). However, in this study, we propose to express the meaning not by means of natural language and not only by letters, as such combinations of symbols are most vulnerable. Within the framework of the second approach, this paper explores the possibilities of using non-linguistic sign systems existing in human culture to create a password.

The purpose of this article is to propose a system of signs that is alien to natural languages and to test its reliability and ease of memorization. Let us examine this method using the combination of symbol systems from chemistry, music, mathematics, physics, programming, languages and drawing.

Chemical elements for creating passwords

Chemistry is known to be a system of elements, chemical formulas, and reactions. Those who know this science well can store in their memory a lot of standard symbolic combinations expressing basic knowledge. An entire chemical formula or reaction can be used as a password. The best way to create a long and complex password is to use complex compounds and reactions with them. For instance

[Cu (NH3) 4] SO4 = [Cu (NH3) 4] + SO4 – dissociation of tetraamminomed sulfate

This password without transformations can already be considered cryptographically strong, but it is quite long, which can cause accidental input errors. Therefore, as an option, it is worth considering simpler chemical formulas that can be combined with natural language, for example:

K3PO4 = 3K + PO4 – dissociation of potassium phosphate

When integrating, we can get: POTASSIUM3PO4 = 3POTASSIUM + PO4 and many other spellings, at the discretion of the user. It is worth noting that the length of the password turns out to be quite impressive, which makes it quite reliable, but at the same time, remembering this password for a chemist will not be difficult, as this is the simplest formula.

Another option available for quick memorization is to write the formula of a substance in the form in which we pronounce it. For instance:

Na2 SO 3 – NATRIYDVAESOTRI in Russian and SODIUMTWOSULFATEFOUR in English.

This is the simplest possible option. To complicate things, it is worth using a different register, which can also advise the dimensions in a chemical formula, for example: SODIUMtwoSULFATEfour or SODIUMtwoSULFATEfour. It is also possible to use transliteration (transferring text using someone else's alphabet), numbers and printing in another language layout: NATRIdvaES03, or any other options that will be understandable to the user and easy to remember.

Another way of using the sign system related to chemistry does not imply a serious basic knowledge of this science. In this case, the periodic table is used as an encryption key that allows you to translate numbers into letters denoting chemical elements. Using the data from the periodic table, the user can encrypt any information in memory containing numbers (date, phone number, address, etc.). Any numbers from 1 to 118 can be substituted for the corresponding chemical elements. Consider several examples:

1. Al08Ca05 (Date of birth 13.08.2005, in which several numbers are replaced by chemical elements)

In this case, the “Al” element has a serial number 13, which corresponds to the date of birth. "08" – month of birth and "2005" year of birth, if desired, can also be replaced by "8" --- "O" – oxygen (serial number 8), and the numbers from the year "20" and "5" can be converted into the corresponding "Ca" (calcium) and "B" (boron) in, that is, the result will be: Al0OCa0B This integration system is quite mobile and variable, so the user can choose the most easily remembered version for himself.

2. FeMgZn0F0Li (Numerical components of the address written down by chemical elements: apartment 26, house 12, post code 300903)

Here, the elements are replaced in a similar way, the digit in the address corresponds to the ordinal number of the element in the final password. Zeros can be left to increase the length of the password, as well as to complicate it.

Musical notation for creating passwords

Music is another vast area, with its own complete system of symbols. Music that users know by heart is a promising basis for creating a strong password. Moreover, to create a password, you can use both notes (for example, in letter notation) and tablature (a type of musical notation, a schematic recording of music for keyboards, some strings and wind instruments). The use of sharp and flat allows you to add special characters to the recording, uppercase and lowercase letters can indicate major and minor inclinations.

Let's look at the example of Ludwig van Beethoven's Moonlight Sonata and observe how you can make a password from a guitar tablature. The scheme of the whole piece is quite voluminous, so you can use the opening passage, or any other part that the user knows.

EADGBE – Letter designations of notes that correspond to 6 strings on a guitar. 0123 is the fret number on which you need to clamp the string. Thus, from the Moonlight Sonata we get the password: A0D2G2B1D2G2B1D2G2B1D2G2B1E3.
Chords can also be used to set a password for your account. Consider this method with the example of "Nirvana – About A Girl"

The first verse has 2 chords, the combination of which is repeated 4 times:
E5GE5Gx4
The first two lines of the chorus use 3 chords, which are repeated 2 times:
C # G # 5F # mx2
Combining the resulting lines, we get a password with a high degree of
protection, as it is composed of letters, numbers, symbols, and it also has sufficient length:
E5GE5Gx4C # G # 5F # mx2

Another possible way to integrate a musical symbol system is to represent the keyboard as a staff or guitar fretboard. In the first case, linear notation is used, in the second, the lines of letters and numbers correspond to the strings of the guitar, and the columns correspond to the frets. The keyboard has three full letter rows and one numeric row, which will correspond to 4 strings on a guitar. This amount is quite sufficient to record a simple melody as a password. Let us consider the example of the aforementioned work of Ludwig van Beethoven, Moonlight Sonata.

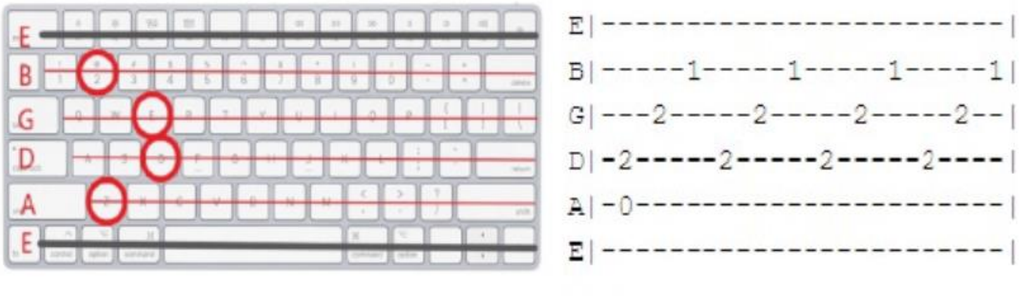
E
B
G
D
A
E
E|-----------------------|
B|-----1-----1-----1-----1|
G|---2-----2-----2-----2--|
D|-2-----2-----2-----2----|
A|-0---------------------|
E|-----------------------|

Figure 1. Transferring the tablature scheme to the keyboard

Figure 1 shows an easy-to-understand tablature for the beginning of a piece, as well as its integration into the keyboard. For A0 – we will have "Z" responsible, for D2 – "D", G2 corresponds to "E" on the keyboard, and B1 – "2"
Let's write down the result of our integration:

ZDE2DE2DE2DE2

In the above example, one of the simplest tablature options is disassembled; musicians can use much more complex options for recording a piece of music using keyboard tools.

Using formulas to create passwords

Next, we will analyze the use of mathematical and physical formulas as a password. This method is suitable even for people who are not professionals in the exact sciences, as basic knowledge from the school course can be used to create a strong password.

sin ^ 2 (A) + cos ^ 2 (A) = 1 is the basic trigonometric identity. It is easy to remember, but at the same time meets the requirements of a password with high cryptographic strength. This identity can also be written in the format:

sineAsquared + cosineAsquared = 1, which is similarly possible to use as a password.

Physical formulas, along with mathematical ones, can be used to protect your account. People who understand physics can use any formulas that they know well. We'll look at a trivial example:

I = q / t – current formula

When converting to a password, you can use the form:

Iequallyq / time

To complicate things, you can use formulas that have fixed numeric elements, for example, the formula for the area of a circle:

S = π * r2, where r is the radius, π is a constant that expresses the ratio of the circumference to the diameter, it is always 3.14

As a password, you can use:

Areaofthecircle = 3.14 * r2

This option is quite easy to remember, but it still retains a high degree of reliability, as it includes letters, numbers, symbols and contains more than 8 characters.

Dynamic and graphical languages for creating passwords

In addition to using alternative sign systems, languages that do not have a generally accepted sign expression can be used to create passwords. In this section, we will consider options for turning into a motion password. There are many sequences of movements that have the meaning for a person. A striking example is dance, but a person remembers the sequence of daily exercises, and the way to the place of study, and how to assemble a meat grinder.

However, the simplest use of a dynamic password is a picture. The easiest way to memorize a sequence of finger movements is to depict a familiar image. This technology is used to a limited extent as a "graphic password" of a smartphone, but if you consider the keyboard as a space for drawing, you can create a rather complex image. For example, a flower – figure 2

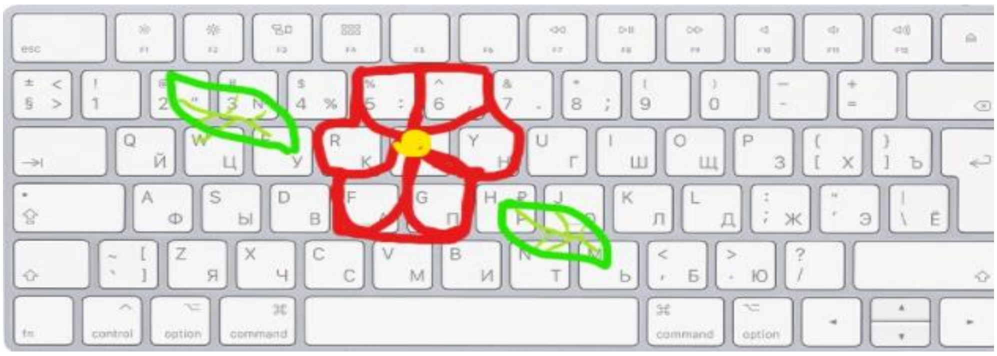

Figure 2. Image of a flower on a cleavage

This password will look like this:

23ew56ygfrthjmn – for complication, you can change the case of letters, at the discretion of the user.

A similar "drawing" option is suggested by Guo et al. (2019) and Schweitzer et al. (2011). However, the possibilities of using visual and "bodily" memory are very wide. Another easy option is – a sequential description of a drawing you are familiar with – figure 3.

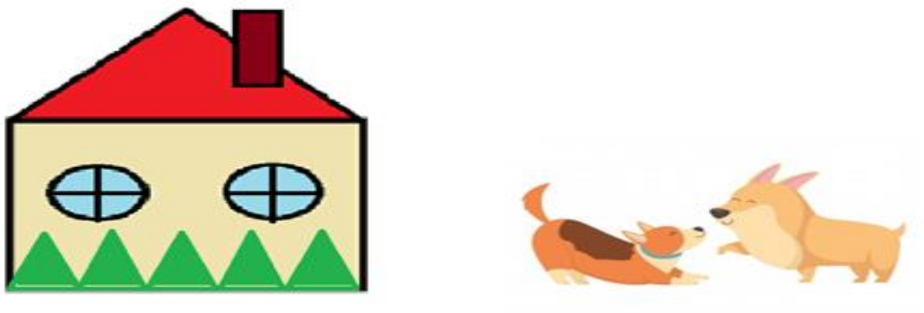

Figure 3. Possible variant of a familiar drawing

This image contains simple shapes, namely 6 triangles, 2 circles and 2 rectangles, as well as two dogs and a grate on the windows, which resembles a "+" sign. By integrating it into the password, we can get:

6triangle2circles ++ 2rectangle @@

The complexity of the picture, as well as the final password, depends on the user's imagination and memory.

Text smilies are a fun way to use symbols alternately. Those who are fond of the artistic expressiveness of ordinary signs are able to create a whole picture. The use of many special characters makes passwords consisting of emoticons especially cryptographically strong.

For instance:

(^_^) is a combination of symbols that denotes joy. There are several ways to write this emotion: (n_n) \ (^ _ ^) /
Sadness can also be depicted symbolically: (v_v) or (<_>)
Embarrassment: * ^ _ ^ *
Kiss: ^} {^
A meaning close to crying: (; _;) or (T_T)

The user can easily associate these symbols with human emotions. It is best to make a small combination of emotions (emoticons) and use it as a password. For example, "I am embarrassed, happy, kissing." We integrate into smiles:

* ^ _ ^ * \ (^ _ ^) / ^} {^

Or "The cat is sad":
(= ^ - ω - ^ =) (v_v)

Habitual actions or movements can serve as the basis for a password. Despite the lack of a generally accepted system of signs for describing dynamics, for example, a mixed system can be used that combines words of a natural language and arrows (^ – straight, <- – to the right, -> – to the left).

Such a record will allow you to describe the usual route, from the kitchen to the bedroom ◦ Bedroom ^ <- <--> ^ -> Kitchen or from home to the bus station ◦ Home ^ x3 -> x4 <----> ^ x3-- > x2BusStation

Programming languages for creating passwords

Programming has recently become an increasingly popular skill. This area is rich in combinations of characters that are suitable for use as a password. The most understandable and accessible option for ordinary users is to use the path to a folder or file known to the user, for example:
C: \ Windows \ System32

More advanced specialists can use commands created using programming languages as passwords.

Both specific commands related to projects important to users and standard commands can be used. For example, for the Java language, to enable the program, you need to write the required construction in the source code:

```
public static void main (String[] args) {
```

Programming languages are extremely promising, as they allow you to express by your own means belonging to other sign systems. For example, the mathematical formulas we are considering can be programmed. Calculating the area of a circle in Pascal will look like this:

```
const Pi = 3.1415;
begin
  var r: = ReadReal ('Enter the radius of the circle:');
  var S: = Pi * r * r;
Println ('The area of the circle is', S);
end.
```

As a password, you can use any line from the above program, for example "var S: = Pi * r * r"

Pictures and colors can be written using a combination of symbols, which is used for web programming and design. So you can write the sequence of colors of the Russian flag as # FFFFFF#0000FF#FF0000. The heart-eyed emoticon (😍) is written & # 128525, and the color MediumAquamarine is # 66CDAA. Combining the designation of the image and color together, we get the password: & # 128525 # 66CDAA.

There is a paradox here associated with the use of a language "understandable" by computers to protect against machine hacking that is tuned to human language.

METHODOLOGY

To study the ways users create a password, as well as to study their ability to compose and remember a new password, based on the techniques given above, a survey was compiled. 526 relevant responses were received using the Google Forms service. All participants consented for using the data provided by them in an anonymous form. Most of the respondents were in the age range of 14-18 years (63.8%) and 19-25 (29.2%). 54.9% of the respondents are women, 37.8% are men, and 7.3% refrained from answering.

The respondent reported the number of cases of hacking of his page and their possible reasons. At the next stage, the respondent came up with any password that the user does not use anywhere – Password No. 1 (hereinafter in the text "First password", 1 password option, Password No. 1) Then the survey participants were asked to choose their area of interest. The choice was between chemistry, mathematics and physics, drawing, rare languages, programming, music. People who could not find their interests among the above could choose the direction "The Way Home".

After a person chose the area of his interests, he was asked to study a little help on how to create a password based on the direction he chose. After reading the creation technique and familiarizing himself with the signs that are used in this area, the user had to set a password based on the knowledge gained – Password No. 2 (hereinafter referred to as "Second password", second password option, Password No. 2)

The purpose of the study is to evaluate in practice how effective the proposed methodology is when creating a password and remembering it. For this, it was necessary to compare the complexity of passwords created spontaneously (Password # 1) with those that were created after reading the recommendations (Password # 2). The password complexity was estimated in terms of information entropy; the password analysis was also used using the demo version of the Zxcvbn program.

RESEARCH RESULT

41.63% of the respondents said that they had been hacked at least 1-2 times, this once again confirms the need to use strong passwords to protect data and, in general, to study the topic of the cryptographic strength of the password and how to remember it. 18.82% – have been hacked 3-4 times, but at the same time 30.04% state that they have never been hacked. 50 respondents were hacked more than 5 times, which is 9.51% of the respondents.

Because the National Institute of Standards and Technology (US) (NIST) recommends using a password with 80-bit entropy for the best security, we concluded that 332 (63.11%) survey respondents entered a password that was not strong enough. The average entropy value for Password 1 is 75.9509, which does not meet the requirements for reliability according to (NIST). The average number of characters in this attempt was 12.9.

Most of the first passwords are too easy to crack with the simplest methods. As an example, we give the most illustrative cases:
"Qwerty2", "password", "Qwertt1234" "nikname123" – standard combinations, break through in the first place when hacked. Sometimes users do not think about the password and enter the first combinations they come across, for example, 5 characters from the top

line of the keyboard "qwerty" or use the words "password" "login" – this is what the user sees first when registering an account. Adding combinations of numbers or their sequences to these data weakly increases the password strength, since they are just as predictable.

"88005553535" – numeric combination, pattern "phone number"
"Kamila2701" – personal data, word from a dictionary, pattern "number"

Let's analyze the above examples using the program "Zxcvbn" and calculate their entropy by the formula.

1. "qwerty2" – 7 characters.

The coincidence with the "Dictionary" pattern immediately drops out, entropy according to the formula = 32.9031 is a low indicator. As for the time of cracking, with one hundred attempts per hour the password will be cracked in 3 days (throttled online attack), with 10 attempts per second in 14 minutes (unthrottled online attack), 10 thousand attempts per second will lead to cracking this password in less than in 1 second (offline attack, slow hash, many cores).

2. "nikname123" – 10 characters

The element "nik" is quickly selected by brute force, "name" is a vocabulary word, "123" is a sequence pattern. Entropy calculated by the formula = 51.6993. At 100 attempts per hour, the password will be cracked for 100 years (throttled online attack), at 10 attempts per second – 5 months (unthrottled online attack), 10 thousand attempts per second will lead to password cracking in 4 hours (offline attack, slow hash, many cores), and 10 billion attempts per second will result in less than 1 second (offline attack, fast hash, many cores)

3. "Kamila2701" – 10 characters.

"Kamila" is a vocabulary word, "2701" is a date pattern, entropy = 59.5420. As for the time of cracking, with 100 attempts per hour the password will be cracked for 100 years (throttled online attack), with 10 attempts per second – 4 months (unthrottled online attack), 10 thousand attempts per second will lead to cracking the password in 3 hours (offline attack, slow hash, many cores), and 10 billion attempts per second will result in less than 1 second (offline attack, fast hash, many cores).

At the same time, 194 respondents (31.17%) have already coped well with the task, the entropy of their first invented password was above 80, several users immediately created a strong password option. This shows that a number of respondents are aware of the existing problems with creating passwords, and are thinking about what a secure password should be. At the same time, some of the respondents, it can be assumed, came up with a complex password at random, without implying that one day the password will have to be remembered. For example, "C1XQqcWkK8 ~ 9R2Z | O # hecWv5 $ 4SoEEHV" – this password is created using a 95-character alphabet (that is, a set of characters from ASCII) and contains 32 characters. This password meets all security requirements and belongs to strong ones, as it is resistant to brute force attacks and dictionary attacks. Password entropy = 210.2354 – high. The breaking time even with 10 billion attempts per second (offline attack, fast hash, many cores) is measured in centuries.

A smaller part of the respondents approached the issue more seriously, using the mnemonics they knew. For example, a fairly often described example is the use of changing a familiar line by adding signs – and a number, for example:

22 Forge-t-Withou-t-Burnin-g-. –30 characters

The element "With" is a dictionary word, "Burnin" is the same, but due to the fact that the rest of the password is subject only to brute-force attacks that will take a long time to find the necessary combination, the password can be considered highly reliable. Password entropy = 197.09. As for the time of cracking, it will be measured in centuries even with 10 billion attempts per second (offline attack, fast hash, many cores).
Let's consider how passwords were created after receiving information, how you can use the knowledge of sign systems to create a password.

137 respondents still created a password that does not reach 80 values of entropy, but the average value of entropy in Password 2 (that is, the password created after receiving recommendations) increased from 75.95 to 128.95, and the average number of characters in vein increased from 12.9 to 20.4.

Users managed to create strong passwords using translation from one character system to another, as an example worth considering:

1. "rjks, tkm --->cflbr --->irjkf --->eybdthcbntn ---> hf, jnf ---> --->byjqvbh" - "– 62 characters.

This password was obtained by entering Russian words in the English layout "lullaby ---> kindergarten ---> school ---> university ---> work ---> another world". Entropy calculated by the formula = 407.33 is an extremely high indicator.

"Eybdthcbntn" - an element subject to dictionary attack, "---" - repeating characters that reduce password strength. Due to the fact that this version of the key contains 62 characters, some dictionary matches do not greatly affect its reliability, which is confirmed by the calculation of the time of cracking in the program "zxcvbn"
The time to crack this password at 10 thousand attempts per second (offline attack, slow hash, many cores) and at 10 billion attempts per second (offline attack, fast hash, many cores) will be more than a century.

2. "LiNaKMgAlK [Fe (CN) 6] geksacianoferratkaliya2002 [] 7000r" – 52 characters.

"Aliya" is a vocabulary element, "2002" is a date pattern. Entropy calculated by the formula = 341, 63. The time to crack this password at 10 thousand attempts per second (offline attack, slow hash, many cores) and at 10 billion attempts per second (offline attack, fast hash, many cores) will be more than a century ...

3. «_0) | (X_X) | : v | (^ _ ^;) | " – 30 characters. Entropy of this password = 197.09

It does not contain a single vulnerability when evaluated using "Zxcvbn". The break-in time with such a high entropy is similarly measured in centuries, even with the most powerful attacks. At 10 thousand attempts per second (offline attack, slow hash, many cores) and at 10 billion attempts per second (offline attack, fast hash, many cores) it will be more than a century.

All of the above passwords, which were invented using the proposed techniques, were created using a 95-character alphabet (that is, a set of characters from ASCII) and contain at least 30 characters. These keys meet all security requirements and are reliable (according to NIST), since they are resistant to brute force attacks and dictionary attacks (according to the "zxcvbn" program)

Let us compare the graphs of the entropy of the first password entered by the respondents and the second password, which was compiled on the basis of the proposed symbol integration system (fig. 4).

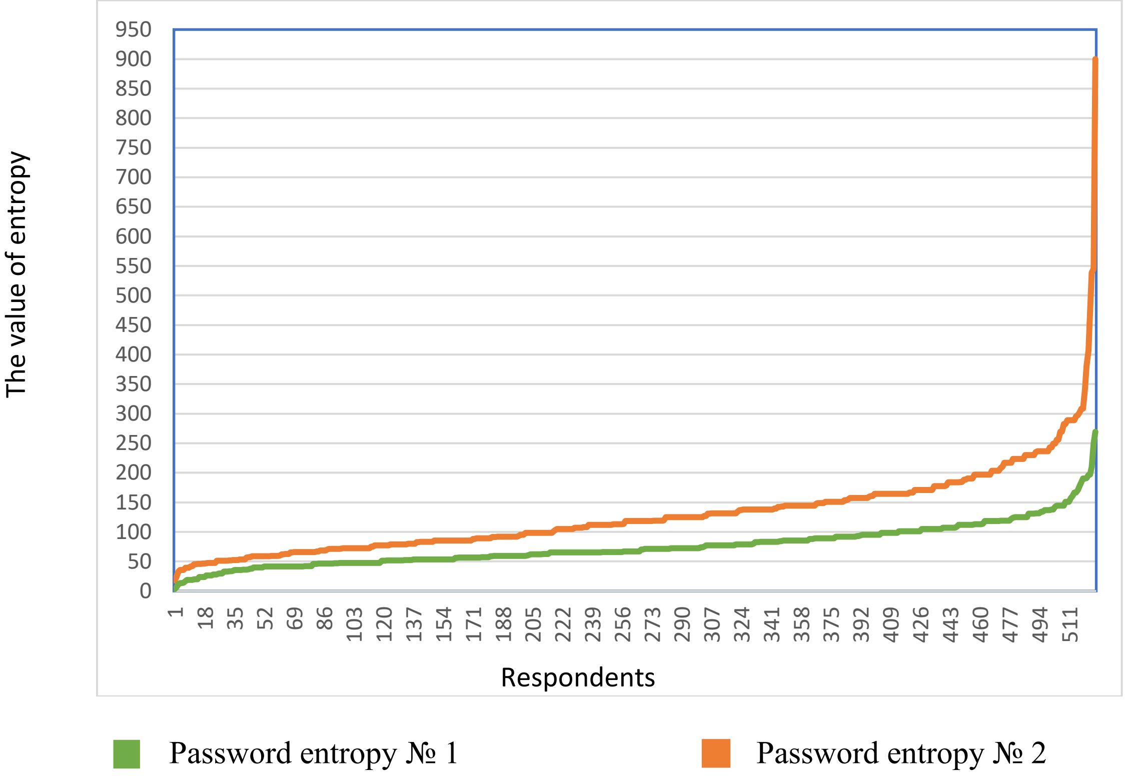

Figure 4. The difference between the entropy of the first password and the second

When comparing the entropy of password №1 and the entropy of password №2, we see a significant difference. The calculations were carried out according to formula (1).

Average entropy of first 526 passwords = 75.95

Average entropy of 526 second passwords = 128.96

In option № 2, the entropy and number of signs are significantly higher than in option №1. This was achieved by the fact that users began to actively use various symbols, numbers and letters, namely, 346 respondents in the second attempt applied exactly the 95-character alphabet (a set of characters from ASCII) That is, symbols from such spheres of life as "chemistry", " mathematics ",“ programming ”and others proposed, allow us to use the alphabet recommended by the National Institute of Standards and Technology (USA) (NIST), but at the same time make associations and are easier to memorize.

As for the perception of the password created by the proposed method, the respondents' answers are rather ambiguous.

In total, 214 people rated the created password at 4-5 on a five-point scale, which indicates the ease of remembering the received password for these respondents, but at the same time 112 people rated the received password as "3", and the remaining 200 chose "1" and

"2" , which suggests that more than half of users still experienced difficulties remembering the key, even using the techniques we proposed for creating it.

Consider the popularity and complexity of passwords relative to the areas of interest our respondents indicated.

Chemistry – 19.39%, drawing – 19.96% – the most popular hobbies of the people who passed the survey. The way home was chosen by 17.87% of respondents, 15.59% preferred mathematics and physics, 8.37% – programming – 8.37% – music and 6.65%
Let's consider how users' passwords changed after studying the techniques and symbols of a specific environment of human culture. Further in the text “№1” and “№2” will correspond to the first password entered by the respondent independently and the second password, which was created on the basis of the proposed techniques.

1. Chemistry – 102 respondents chose

Average value of the number of characters in the first password = 12,068.
After respondents got acquainted with the techniques for creating a password based on chemical characters, the average value of the number of characters in the password increased to 18,294
As for entropy, in the first case, the average value was 70.52. In the second case, it increased to 115.49
Let's look at examples of how users managed to improve the password strength by integrating it into the chemical language, using the example of specific passwords entered by the respondents.
1) №1 – Vola34, 6 characters, entropy – 35.72.
No. 2 – NaCl + AgNo3 = NaNo3 + AgCl, 21 symbols, entropy 137.96

2) № 1 – BubenetsOleg143010, 18 characters, entropy – 107.17
No. 2 – 1,3,5,5-tetrametil-2,2-dietilnonen-4, 36 characters, entropy – 236.51

3) №1 – Rfgh146ila, 10 characters, entropy – 59.54
№2 – NaKRdZn2356780SiFeG, 18 characters, entropy – 113,129
The dynamics of the average indicators can be seen on the fig. 6

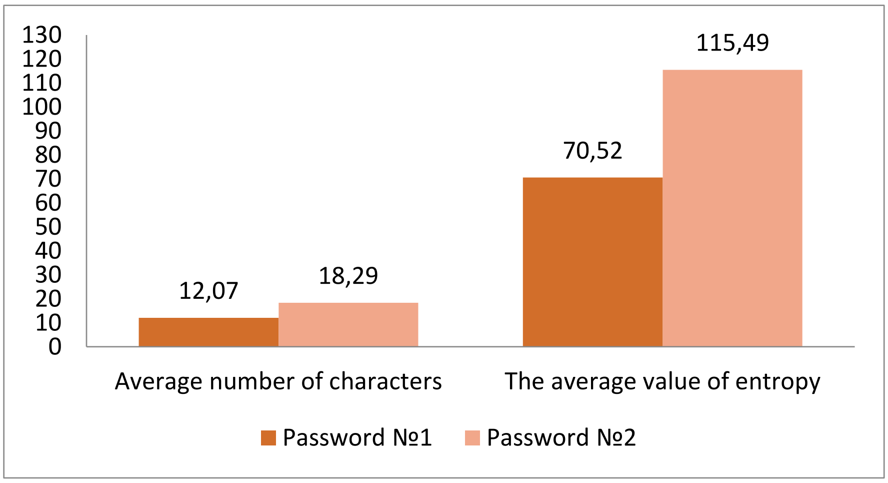

Figure 6. Dynamics of indicators in the "Chemistry" section.

2. Drawing – selected by 105 respondents

Average number of characters in the first password = 12,095.

After the respondents got acquainted with the techniques of creating a password by drawing, the average value of the number of characters in the password increased to 19,257 – as we can see, a little more than in chemistry.

As for the entropy, in the first case the average value was 69.96. In the second case, it increased to 118.56 – the increase in entropy is also more significant than in chemistry.

Let's take a look at examples of how users applied drawing techniques while creating a strong password.

1) № 1 – ghbdtn12, 8 characters, entropy – 41.35

№2 – '(0_0) (^ _ ^) () (@ _ @) (X_X) | : v | , 31 symbols, entropy 203.66

2) № 1 – Al = in896_nA564310, 17 characters, entropy – 111.68

№2 – 8xcircle11xsquare2xellipse, 26 symbols entropy – 154.80

3) № 1 – bonkbonk69420, 13 characters, entropy – 67.20

№ 2 – rdcvgfrdcvgfdcfvgrfctg, 22 characters, entropy 103, 40

The dynamics of the average indicators can be clearly seen on the fig. 7.

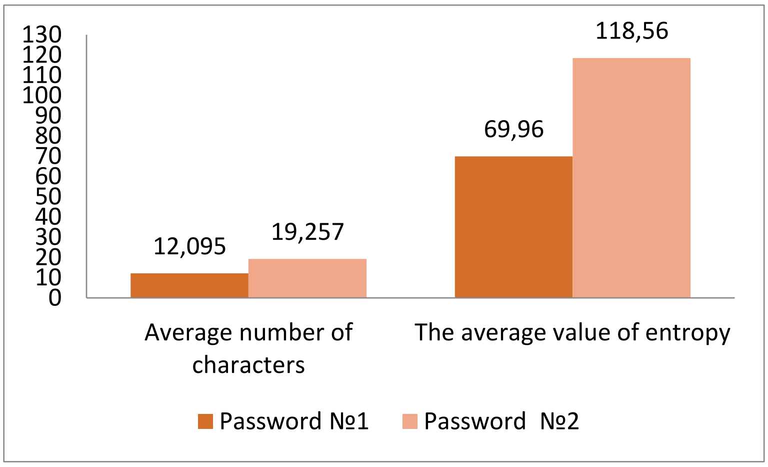

Figure 7. Dynamics of indicators in the section "Drawing".

3. “The Way Home” – 94 respondents chose.

The average number of characters in the first case is 13.7. After studying the recommendations for creating a password, based on their usual daily movements, the average number increased to 25.1.

Entropy also increased significantly in the second case. If in the first variant of the password it was 80.51, then the entropy of the second variant in its average value reached 169.2 (Figure 8)

1) № 1 – Polikika543, 11 characters, entropy – 65.49

№2 – Dom | -> | -> | -> ○ -> | -> Uni, 21 symbols, entropy – 137, 96

2) № 1 – FD43mixyil, 10 characters, entropy – 59.54

№2 – Home ^ <----> ^^^> ^ -> Kitchen 26 characters, entropy – 170.81

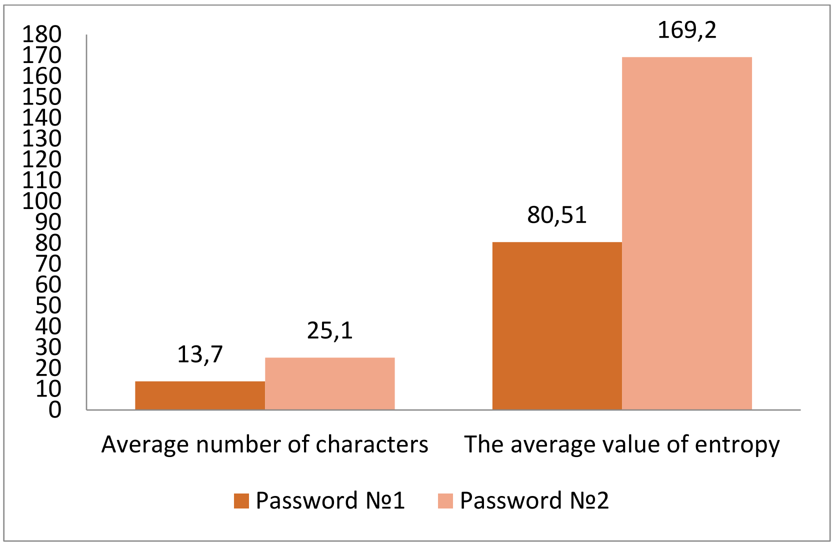

Figure 8. Dynamics of average indicators "Way home"

4. Mathematics and physics – 83 people

The average number of characters in the first password is 11.91, in the second 16.63. The average value of entropy in the first case is 70.63. After studying the ways of using various formulas as a password, the entropy increased to 106.84 (figure 9)

1) № 1 – wjbswjksnsnsksk, 15 characters, entropy- 70, 5
№2 – cosinus (65) * pi * 2718281, 22 symbols, entropy – 144, 53

2) № 1 – Brido4277 # UU, 12 characters, entropy – 78.83
№ 2 – 'Mendeleev-Clapeyron: pV = vRT, 26 characters, entropy – 170.81

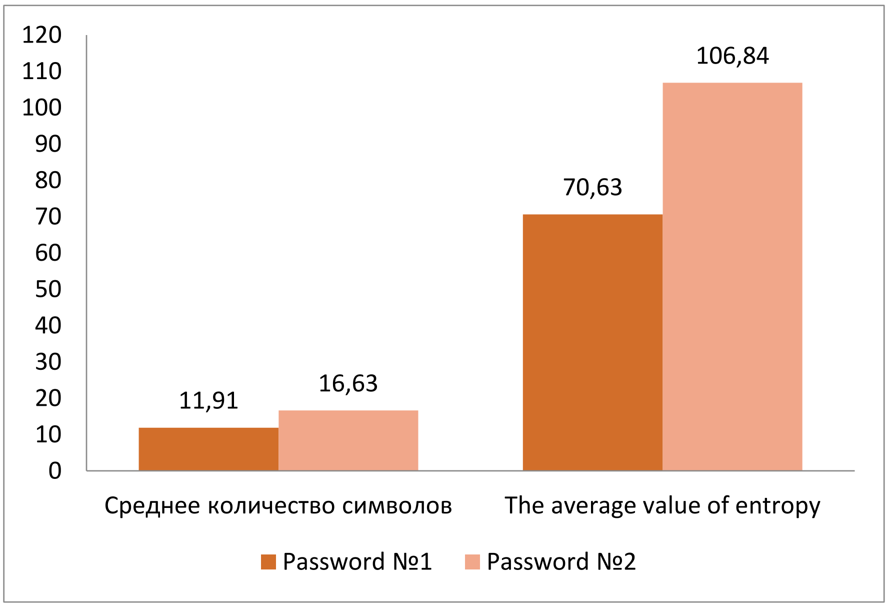

Figure 9. Dynamics of average indicators in the section "Mathematics Physics"

5. Music – this area was selected by 64 respondents

If we consider the average number of characters in the first password, then we get a value equal to – 14.01.

After learning how to integrate music and its typical symbols into the password creation process, users created passwords with an average number of characters equal to 19.51 (figure 10).

As for the entropy, it also increased in the second case, although not as much as, for example, in "The Way Home". The average value of the entropy of passwords in the first case is 82, 21. The average value of the entropy of passwords in the second case is 116.9 (figure 10).

Consider how the respondents applied the suggested methods:

1) №1 – SuperJack154 (, 13 characters, entropy –85, 4
   №2 – AmEmG7CDmAmA7EmCG7F #, 20 characters, entropy 131.39
2) № 1 – Traueifk4551; / 82, 16 characters, entropy – 105, 11
   №2 – mollT53T6T64S53SFHSKSLD6D64, 27 characters, entropy – 160, 76

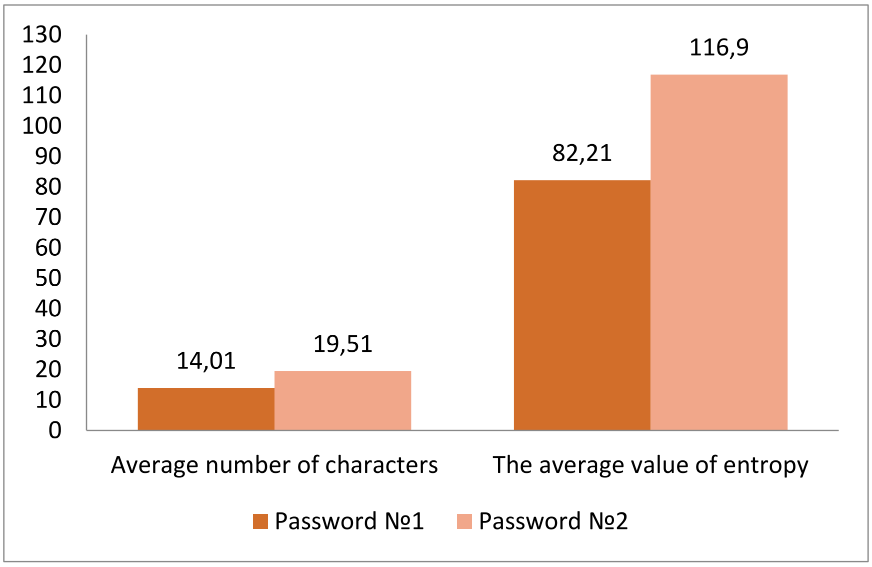

Figure 10. Dynamics of average indicators in the "Music" section

6. Programming – selection of 44 respondents

The average number of characters in the first password is 13.79, in the second it has increased to 26.65.

The increase in entropy in programming is especially remarkable, since it is maximal relative to other interests to choose from.The initial password had an average entropy value of – 84, 30. After the recommendations on integrating the system of programming symbols into the password, the respondents created keys, the average entropy of which reached 170.37 (figure 11).

Let's illustrate it with the examples:

1) No. 1- Hpetr3794 characters 9, entropy –53, 58
   №2 – System.out.println ("Poka"), characters 9, entropy – 170.81
2) № 1- Swft24671dF characters 11, entropy –65, 4953
   №2 – W: // document / copy2 / team25, characters 25, entropy – 164.24

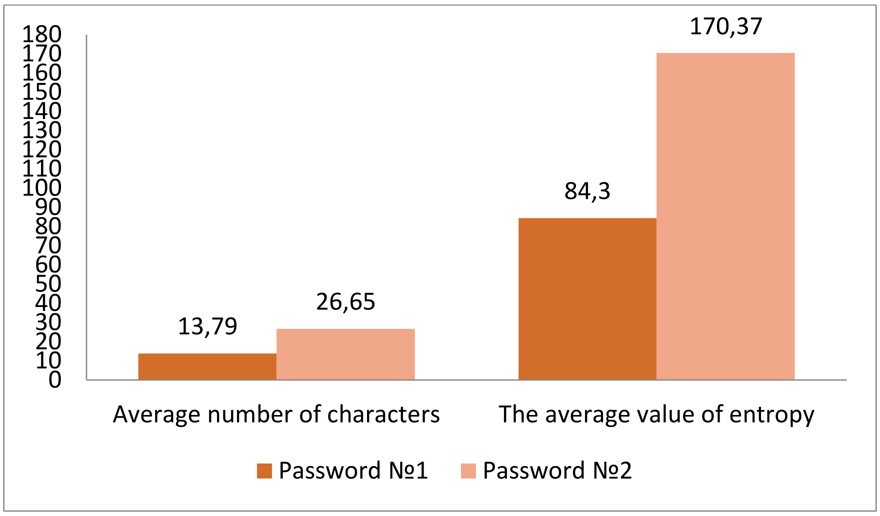

Figure 11. Dynamics of indicators in programming

Entropy scores "Programming" and "The Way Home" are significantly higher than in the other sections. As for programming, then, most likely, this is due to the fact that the address, for example, to a folder will always contain characters like: ": //" and have a large number of characters. Commands in programming languages also include characters like ")." And most often consist of phrases that are easy to remember. It is also worth considering that people who chose this interest most likely already had information about the methods creating a strong password, which could affect the results of entropy.
For clarity, let's compare how the average indicators increased in each of the proposed categories. Consider Figure 12.

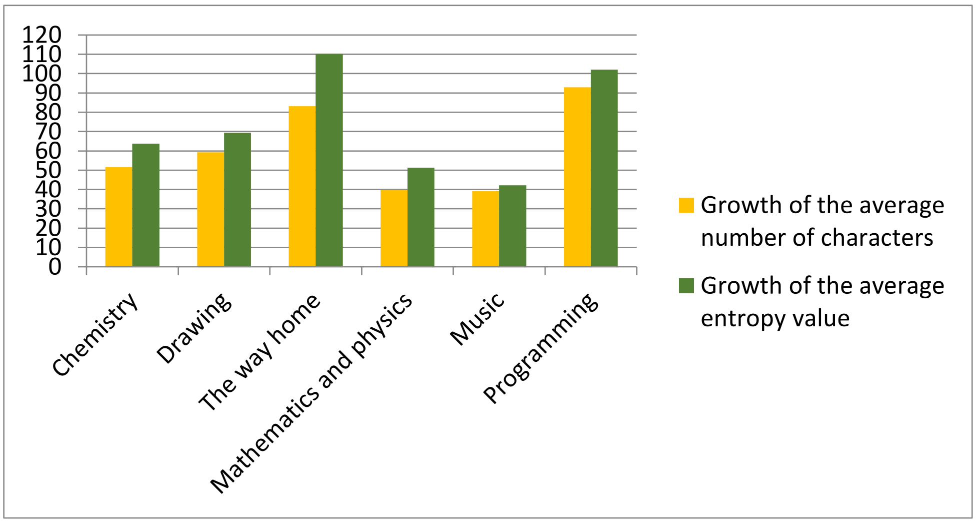

Figure 12. Percentage of growth in the average number of characters in a password and growth in entropy after respondents read the recommendations.

In the graph above, we can see that coding and driving home yielded the largest percentage gains. The use of the "Way Home" technique made it possible to increase the average number of password characters by 83.21%, and increase the entropy by 110.2%. Programming the average number of characters gave an increase of as much as 93%, and the average value of entropy increased by 102.02%. At the same time, "Mathematics / Physics" and "Music" showed the lowest results. Growth in "Mathematics/physics "on average was 39.63%, and in entropy 51.26%. "Music" gives similar results, namely an increase of 39.26% and 51.26%. "Chemistry" and "Drawing" – turned out to be average in terms of values. The growth in chemistry was 51.59% and 63.76%, respectively, and in painting – 59.21% and 69.47%.

It is also worth considering the number of respondents in each category and the average value of the entropy of the second password (figure 13).

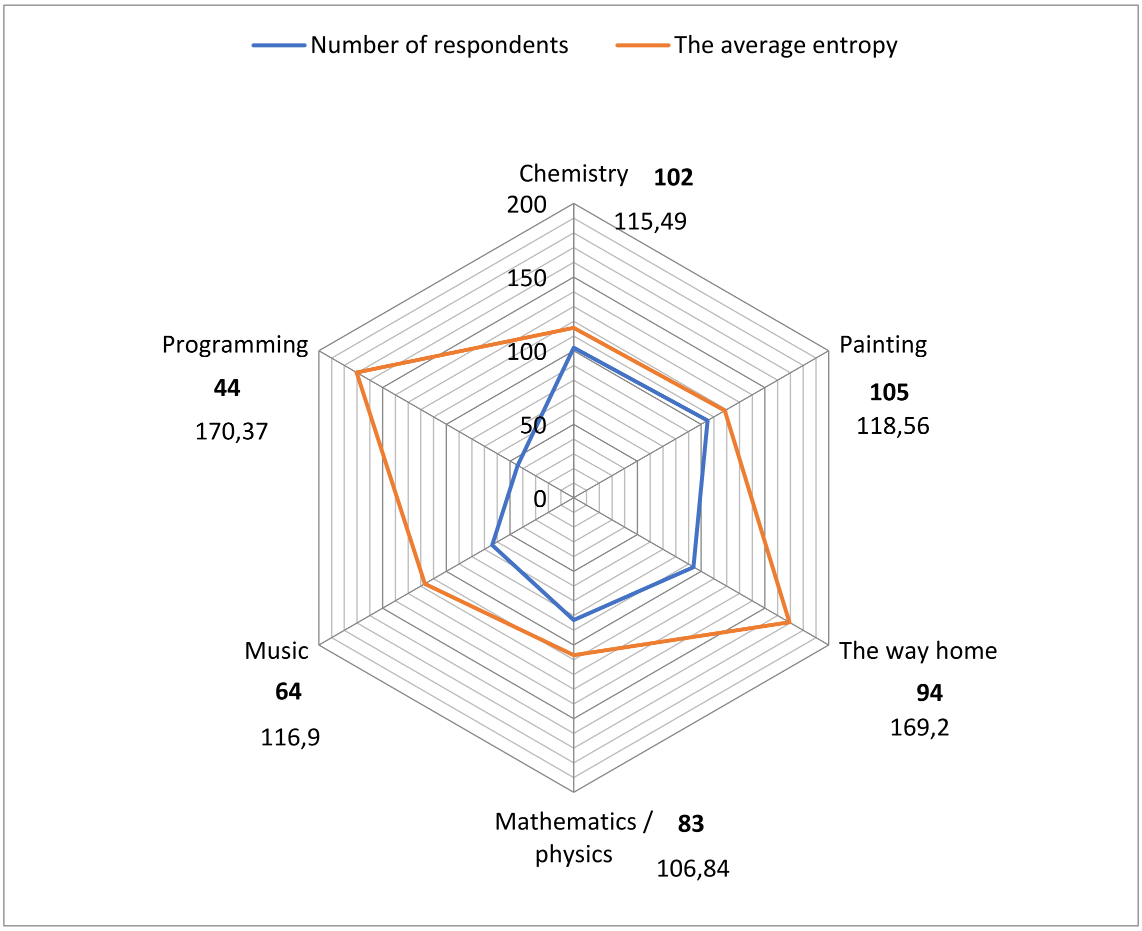

Figure 13. The number of respondents in each category and the average entropy of the second password

CONCLUSION AND DISCUSSION

As a result of our research, we found that 71.70% of our surveyed respondents are not able to create a strong password by themselves. Indeed, in most cases, they entered passwords that were easy to remember and yet completely unreliable in (Guo et al., 2020; Li et al., 2018). Some users have taken serious attempts to create their first password, and several respondents have already known some of the techniques from the Bei et al. (2019) study.

Our recommendations and schemes for composing reliable, but at the same time memorable passphrases using a non-linguistic sign system have reduced the number of unsafe keys from 71.70% to 26.2%, which can be called a good result.

The increase in the entropy of passwords from the first option to the second is associated with several reasons:

1. Respondents in the first case were not reminded of the need to use a strong password, while in the second attempt, users were clearly assigned the task of creating a strong password using the proposed techniques.

2. The selected areas of interest had their own system of symbols, which, when integrated into a password, in most cases met all modern requirements regarding password security, such as: more than 8 characters in length, different case, numbers and symbols.

As for the memorability of the password, according to the survey results, difficulties arise even when using the techniques we proposed, but since they have direct associative chains with music or chemistry, which can facilitate the process of remembering a more reliable password. Memorization problems may indeed be related to long strings, as has been argued in research Stanton & Greene (2014) and Huh et al. (2015)

Users have successfully applied the symbol systems from the areas of interest that they themselves have chosen. The resulting password, in most cases, remained reliable, but at the same time was understandable to the user. This is confirmed by studies (Guo et al., 2020) and (Seitz, 2017) that users tend to choose passwords based on their personal characteristics and interests.

In general, we can say that the proposed techniques justify themselves, since they allow you to create a password that is composed using a 95-character alphabet (that is, a set of characters from ASCII) and contains at least 30 characters. The resulting passwords will meet all security requirements (according to NIST), but at the same time they are easy to associate with, for example, chemistry or programming, which should make them easier to remember. It should be noted that if the proposed techniques become popular, then the databases of dictionaries used by hackers will certainly be replenished with chemical and mathematical formulas, for example. This is obvious from the example of the previously popular method of complicating the password – the use of transliteration, when the words of the Russian language were written in Latin letters, or case switching when the user presses a sequence of keys that would form a Russian word, but the case is at the same time adventurous in Latin letters, so it turns out meaningless set of letters. However, the variety of sign systems suggests that even the introduction of some of the above-mentioned elements into dictionaries will not make these techniques irrelevant in the future.

As for the prospects for researching this topic, it is worth paying attention to the quality of memorizing the password obtained using the techniques given in the article, since the ideal password should ultimately remain easy to remember with its maximum reliability.

ACKNOWLEDGEMENT

We would like to express our gratitude to Daria S. Bylieva, who was our mentor and the most honest critic of this work, as well as to Filip Yermilov and Valeria

Vorobyova, who made a contribution to the collection of statistical data. We would also like to thank all the students who took part in the surveys.

REFERENCES

Al-Ameen, M. N., Wright, M., & Scielzo, S. (2015). Towards making random passwords memorable: Leveraging users' cognitive ability through multiple cues. In *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 2315–2324). ACM. https://doi.org/10.1145/2702123.2702241

Alomari, R., Martin, M. V., MacDonald, S., Maraj A., Liscano, R., & Bellman, C. (2019). Inside out – A study of users' perceptions of password memorability and recall. *Journal of Information Security and Applications*, *47*, 223-234. https://doi.org/10.1016/j.jisa.2019.05.009

Bei, Y., Yajun, G., Lei, Z., & Xiaowei, G. (2019). An empirical study of mnemonic password creation tips.*Computers & Security*, *85*, 41-50. https://doi.org/10.1016/j.cose.2019.04.009

Bonneau, J. (2012). The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *IEEE symposium on security and privacy* (p. 538–52). IEEE. https://doi.org/10.1109/SP.2012.49

Bonneau, J., & Shutova, E. (2012). Linguistic properties of multi-word passphrases. In J. Blyth, S. Dietrich, & L. J.Camp (Eds.), *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Lecture Notes in Computer Science*, *7398*, (pp. 1–12). Springer.https://doi.org/10.1007/978-3-642-34638-5_1

Bosnjak, L., & Brumen, B. (2016). What do students do with their assigned default passwords? In*39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016 – Proceedings* (pp. 1430–1435). IEEE. https://doi.org/10.1109/MIPRO.2016.7522364

Bosnjak, L., Sres, J., & Brumen, B. (2018). Brute-force and dictionary attack on hashed real-world passwords. In *41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 – Proceedings* (pp. 1161–1166). IEEE.https://doi.org/10.23919/MIPRO.2018.8400211

Bylieva, D., Bekirogullari, Z., Kuznetsov, D., Almazova, N., Lobatyuk, V., & Rubtsova, A. (2020). Online Group Student Peer-Communication as an Element of Open Education. *Future Internet*, *12*(9), 143. https://doi.org/10.3390/fi12090143

Guo, Y., & Zhang, Z. (2018). LPSE: Lightweight password-strength estimation for passwordmeters. *Computers & Security*, *73*, 507-518. https://doi.org/10.1016/j.cose.2017.07.012

Guo, Y., Zhang, Z., & Guo, Y. (2019). Optiwords: A new password policy for creating memorable and strong passwords. *Computers and Security*, *85*, 423–435. https://doi.org/10.1016/j.cose.2019.05.015

Guo, Y., Zhang, Z., Guo, Y., & Guo, X. (2020). Nudging personalized password policies by understanding users' personality. *Computers & Security*, *94*, 101801. https://doi.org/10.1016/j.cose.2020.101801

Gupta, P., Marur, D. R., Kalisetty, H., & Khanna, A. (2020). A novel secure and high-entropy hardware password manager. *Materials Today: Proceedings* (in press). https://doi.org/10.1016/j.matpr.2020.09.524

Habib, H., Colnago, J., Melicher, W., Ur, B., Segreti, S., Bauer, L., Christin, N. & Cranor, L. (2017). Password creation in the presence of blacklists. In *Workshop on Usable Security (USEC '17)*. Internet Society.

Haeussinger, F., & Kranz, J. (2017). Antecedents of employees' information security awareness – review, synthesis, and directions for future research. In *Proceedings of the 25th European Conference on Information Systems (ECIS)* (12). https://aisel.aisnet.org/ecis2017_rp/12

Huh, J. H., Kim, H., Bobba, R.B., Bashir, M.N., & Beznosov, K. (2015). On the memorability of system-generated pins: can chunking help? In *SOUPS 2015 – Proceedings of the 11th Symposium on Usable Privacy and Security* (pp. 197–209). USENIX Association.

Kuo, C., Romanosky, S., & Cranor, L. F. (2006). Human selection of mnemonic phrase-based passwords. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security* (pp. 67–78). ACM. https://doi.org/10.1145/1143120.1143129

Li, J, Stecker, L., Zeigler, E., Holland, T., & Liang, D. (2018). Scramble the password before you type it. In A. Rocha, H. Adeli, L. Reis, & S. Costanzo (Eds), *Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing, 746* (pp. 1097–1107). Springer. https://doi.org/10.1007/978-3-319-77712-2_105

Maoneke, P. B., Flowerday, S., & Isabirye, N. (2020). Evaluating the strength of a multilingual passphrase policy. *Computers and Security, 92*, 101746. https://doi.org/10.1016/j.cose.2020.101746

Narendar, D., Guddeti, P., & Rao, C. (2020). Analysis of password protected document. *An International Journal of Advanced Computer Technology*, *9*(7), 3762–3767. https://ijact.joae.org/index.php/ijact/article/view/1174

Schweitzer, D., Boleng, J., Hughes, C., & Murphy, L. (2011). Visualizing keyboard pattern passwords. *Information Visualization*, *10*(2), 127–133. https://doi.org/10.1057/ivs.2010.12

Seitz, T. (2017). Personalizing password policies and strength feedback. In *Proceedings of the Second International Workshop on Personalization in Persuasive Technology co-located with the 12th International Conference on Persuasive Technology* (pp. 64–69). CEUR.

Singh, V., & Pandey, S. K. (2019). Revisiting Cloud Security Threat: Dictionary Attack. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3444792

Stanton, B. C., & Greene, K. K. (2014). Character strings, memory and passwords: what a recall study can tell us. In T. Tryfonas, & I. Askoxylakis (Eds.), *Human Aspects of Information Security, Privacy, and Trust. HAS 2014. Lecture Notes in Computer Science, 8533* (pp. 195–206). Springer. https://doi.org/10.1007/978-3-319-07620-1_18

Ur, B., Bees, J., Segreti, S. M., Bauer, L., Christin, N, & Cranor, L. F. (2016). Do users' perceptions of password security match reality? In *Proceedings of the 2016 CHI conference on human factors in computing systems* (pp. 3748– 3760). ACM https://doi.org/10.1145/2858036.2858546

Ur, B., Noma, F., Bees, J., Segreti, S., Shay, R., Bauer, L., Christin, N., & Cranor, L.

(2017). “I added ‘!’ at the end to make it secure”: observing password creation in the lab. In*SOUPS '15: Proceedings of the Eleventh USENIX Conference on Usable Privacy and Security* (pp. 123–140). ACM.

Van Schaik, P., Jeske, D., Onibokun, J., Coventry, L., Jansen, J., & Kusev, P. (2017). Risk perceptions of cyber-security and precautionary behaviour. *Computers in Human Behavior*, *75*(2017), 547–559. https://doi.org/10.1016/j.chb.2017.05.038

Veras, R., Thorpe, J., & Collins, C. (2012). Visualizing semantics in passwords: the role of dates In *VizSec '12: Proceedings of the Ninth International Symposium on Visualization for Cyber Security* (pp. 88–95). ACM. https://doi.org/10.1145/2379690.2379702

Volokitin, T., & Volokitina A. (2020). Sravnitel'nyy analiz bezopasnosti paroley. [Comparative analysis of password security]. In *Youth and systemic modernization of the country*(pp. 69-72). Southwestern State University.

Wheeler, D. L. (2016). Zxcvbn: Low-Budget Password Strength Estimation In *SEC'16: Proceedings of the 25th USENIX Conference on Security Symposium* (pp. 157-173). ACM.

Yan, J, Blackwell, A., Anderson, R., & Grant, A. (2004). Password memorability and security: empirical results *IEEE Security & Privacy*, *2*(5) 25–31. https://doi.org/10.1109/MSP.2004.81