УДК 004.9

N. Boyko, M. Stakhiv
Lviv Polytechnic National University
ISM department

USING CAPTCHA TECHNOLOGY: PROS AND CONS. WAYS TO SECURE FROM SPAM-BOTS ATTACKS

Розглянуто технологію CAPTCHA. Подано алгоритм створення та роботи капчі. Наведено схему роботи капчі. Проаналізовано способи кодування та "ускладнення" зашифрованого тексту. Розглянуто основні способи автоматизованого розпізнавання CAPTCHA. Наведено варіанти визначення тестів CAPTCHA, методи програмних засобів та методи, які не потребують дій від користувача. Проаналізовано переваги та недоліки тестування капчі, а також способи вирішення проблем, що можуть виникати із застосуванням капчі.

Ключові слова: CAPTCHA, алгоритм роботи, інтернет-сервіс, спам-боти, захист, скрипт, кукі, нейронні мережі.

This paper is devoted to CAPTCHA technology. Algorithms of creation and work of captcha are described. The scheme of how captcha works is also included. The means of encoding and 'complication' of decoded text are described. The main ways of automatic captcha's recognition are described. Also, it is given in examples options of CAPTCHA's tests. The programing means that do not require user's actions are considered. By the way, pros and cons of testing captcha are analyzed. In addition, the main problems that may occur during developing captcha are analyzed.

Key words: CAPTCHA, algorythm of work, Internet service, spam-bots, protection, script, cookies, neural networks.

Introduction

Nowadays a CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) is present on almost web sites on their register forms. This is a type of challenge-response test used in computing to determine whether the user is human. Actually, it is so-called protection from spam-bots (computer program or a package with one main purpose – advertising messages and their automatic sending), because web-resources with big amount of data and lot of ads are not attractive now. Therefore, users will not visit such web sites next time.

However, is user guesses the CAPTCHA text from first attempt? In addition, how should elder people who are new to PC know what "reCAPTCHA" is and what to do with it?

ReCAPTCHA – is a system designed at Carnegie Mellon University to protect web sites from Internet bots and supporting digitization books. It is CAPTCHA-like system, so-called continuation.

Unlike traditional recognition systems of user as human by inputting, certain set of symbols and dagits, reCAPTCHA system offers user to input two words. One of them is recognized and system knows that, and second is unknown for system and it cannot be recognized as well. Second word is picked from source, which requires recognition (e.g. recognition of book). Checking and pass through are carried by word which system knows. Second word is unnecessary: it is inputted by user, then stored in system and used as possible recognition option. Final recognition is performed by selecting a most used word for input. ReCAPTCHA system provides user with an image to recognize and collects results. After that, it gives them to sponsors digitization of content.

Nonetheless, captcha should be both readable to users and resistant to recognition. In addition, CAPTHCA generator should have regular expression (e. g. “a\.?” stands for “a.” or “a,” in other words regular expression is a set of allowed symbols). So, it is required to combine those two cases to ensure readability and protection from bots. It is known that human “reads” complete picture, and bot – picture’s pixels, so it is necessary to “encode” pixels, so bot will not read them correctly.

Last researches and publications analysis

CAPTCHA began to grow in popularity on many web sites for the last 7 years [1–5]. However, for last 4 years it began to be uncomfortable for users [7]. Unfortunately, this is true, but also new means to deal with this “barrier” appeared. Nonetheless, for most popular web-resources it is required to keep their reputation and security on a level [1–5]. Therefore, in this article authors will describe the so-called question of CAPTCHA existence: “to use or not to use?”

In addition, authors will describe alternative ways to protect web sites from spam-bots and will give an advice how to create CAPTCHA and use them correctly [6].

The aim of the study is to determine the characteristics of using one of the methods of functionality of computer tests CAPTCHA with regard to their advantages and disadvantages and their possible effective use from spam-bots.

Main Plot

Main idea of CAPTCHA text is to offer user such type of task, which he will solve, but computer will not or it will be too complicated. In this article mathematical tasks (arithmetical operations), text tasks, object recognition etc. will be examined.

Nowadays the most used networks are Internet and Local Area Network. The first allows to get access to World Wide Web and contains various contents (social networks, blogs, formulas, help resources, etc.), when the second allows to connect different amount of PC users into one group for data sharing. Those people who know and trust each other work In Local Area Network, as usual [1].

In that case, the application of CAPTCHA occurs when it is need to alert about usage of Internet services by spam-bots, for example, to prevent automatic messages post, downloading files, registrations, adding comments, sending text messages etc.

Web-form of captcha is displayed on Fig. 1, which is used on web site of The Free Encyclopedia [1].

Fig. 1. Captcha web-form example

As you can see from Fig. 1, a text is “smum” and it is decoded: some letters inside are curved, their position is very dense, and angle of letters are not the same.

Classic scheme of how captcha works is given below, which contains input fields and, obviously, captcha, so user can submit that he is a human

Classic scheme of how captcha works:

1. Script generates random (“secret”) text and it saves on a server via session. Then, using methods of programming language captcha is displayed with that text.

About session – this mechanism allows to create and use variables, which save their value during whole session. These variables have different value and can be executed on any site page before user exists from system. By the way, each time user visits site, he gets new variable values, which help to identify during session. Therefore, that is from where the mechanism’s name comes from.

2. When filling in a form and submitting it user has to input that “secret” text into respective field
3. Script that operates web-form checks if submitted value is equal to session’s one. When it is true, script continues to work and new user is registered.

For example, when it is a need to check equality between submitted field and session’s, PHP acts like this:

With a help of POST method from submitted form data “extracts” from input field and checks for equality with a set value on a server. When values are equal user registers. This process cannot be seen by anyone, except server, who makes captcha security more resistant [5].

Classification of captcha’s complications:

Different techniques and secrets are used to provide protection for both web site and “firewall”. For example, the usage of constant matrix (those, which built with constant numbers) lets to generate cascade image, which then can be modified with complications (make it less readable). There are many ways to code and “complicate” encoded text. The following are most popular

- Different colors usage on the image:

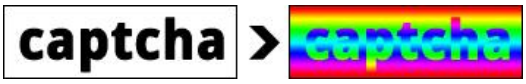

Fig. 2. Example of different colors usage for CAPTCHA’s web-form.

This image above is an example of how this method works. In real situation, this is not reliable, because text and background are contrasting, so not only background, but also text is colorful. When border for letters is absent, user could not recognize a text

- Presence of so-called “noise”:

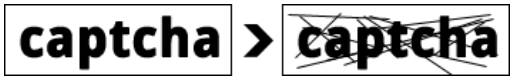

Fig. 3. Example of noise usage for CAPTCHA

Usually each captcha contains such type of complication, which represented by many crossing lines on text with various angles and lengths.

- The width between letters should be tight:

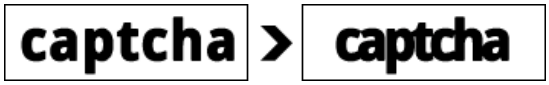

Fig. 4. Example of tight width between letters in CAPTCHA.

It is required to keep balance, because excessive approximation can make text less readable. On Fig. 3, you can see that letters are on each other; that is so-called obstacle for bots during image segmentation.

This process includes input image division on regions (segments), which have common criteria (e.g. same brightness, height etc.). Therefore, if symbols are located tight enough, then harmful program will not output the correct result

- Different size of symbols

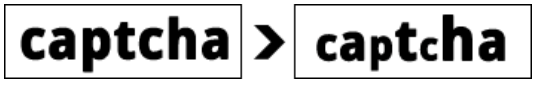

Fig. 5. Example of different size of symbols usage in CAPTCHA.

While using this technic, it is recommended to remember that this obstacle will not let bot to use constant matrix during image segmentation. So, if it is going to use different sizes of symbols, then it is required to set dynamic size for each symbol.

- Distorted font:

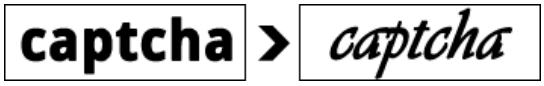

Fig. 6. Example of distorted font usage in CAPTCHA

Good and useful technic: turn ups, italics and style – great obstacles for bot. By the way, when combining this with the use of 'noise' it is required to set thick font. Amazing combo describes various fonts for each symbol.

• Different angle position for symbols:

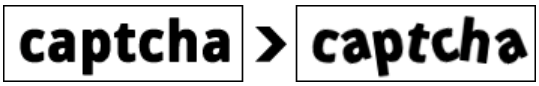

Fig. 7. Example of different position for symbols usage in CAPTCHA

This technic is useful enough. To say again, segmentation will be complicated but not enough. It is recommended to choose a small angle, otherwise text will not be readable (letters will be on each other).

• Dynamic disfiguration text:

One of the worst decisions. Usually Disfiguration of CAPCTHA highly reduces readability of text both for spam-bot and for human (so-called two-side protection). The main requirement is that those disfigurations should be minimal.

Nevertheless, it is required to produce security of encoded text that no one could gain access to transferring data. Therefore, an element that contains image (captcha) should not have a value of image in code when sending registration form to user's browser; otherwise, it will cause less work for spam-bot.

Nowadays there are three main ways of automatic CAPTCHA recognition:

1. Using errors in security algorithm:

This approach is set for searching logical errors (susceptibility; it is all about algorithm of program, that works not correctly, though it can be run for other purposes), that allows to submit form without CAPTCHA recognition. This is the easiest way to omit protection, but it can be used only in simple decisions (low-budget projects).

The most popular error is sending code-check of CAPTCHA via form fields (text input fields) or via *cookie* (in computer terminology this is a concept, used to describe the information as text or binary data, which are received from web-site on web-server, which is stored in the client browser, and then sent to the same site, if the site re-visit) in open format, in *base64* (a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation) or in *md5*-hash (128-bit hash algorithm). It will not be hard for attacker to get it, even if it is required to generate *rainbow*-table (special tables search option to access cryptographic hash function that uses the mechanism of reasonable compromise between the time of search for tables and used memory) according to the captcha's alphabet (e.g. five-letter Cyrillic or six-figure numbers) and make a comparison.

On Fig. 8, there is a *rainbow*-table with chain length of three.

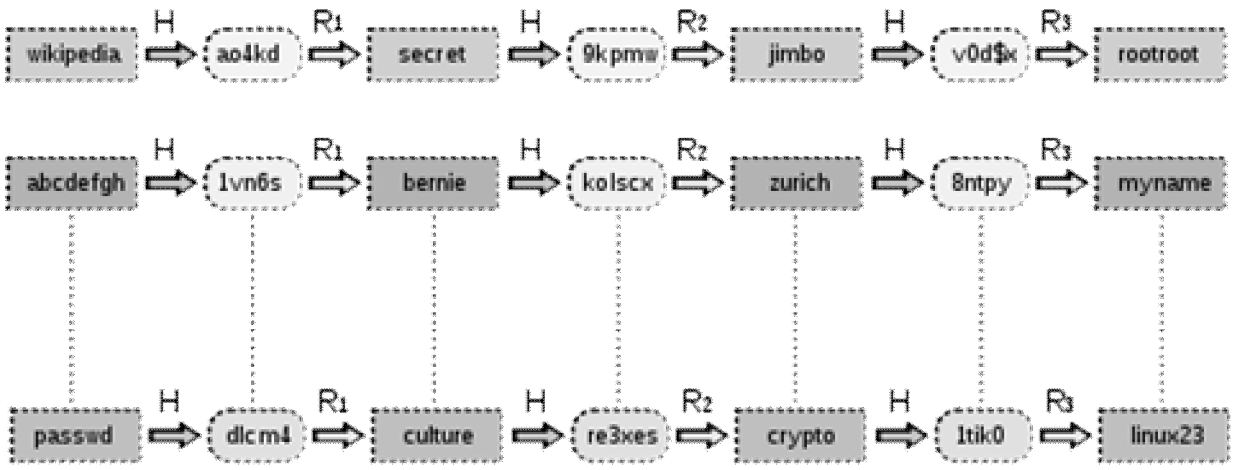

Fig. 8. Scheme of shortened rainbow-table

As you can see from Fig. 8, *R1, R2, R3* are functions of reduction (process or action that causes decrease, weakening or shortening of something; sometimes it causes total loss of some objects, features), *H* – hash function (transforms input data array of any length into output bit string of fixed length)

If random text generates when sending page with form, rather than when sending image, then danger will occur, because bot sends a couple of requests for image's script though to get a couple of options of the same text, and that makes his job easier.

The most common mistake is that when checking input text for accuracy, it has taken from session and compared with user answer: it was said above about form fields, which contain also CAPTCHA, that they should not contain encoded values. The problem is that attacker can send a number of session that does not exist and input blank text into field. That text equals to number of non-existent session – obstacle is dodged.

It is also important that session should be cleaned after each check (no matter if it was successful or not). It concerns cases where generated identifier of captcha (unique captcha's number that is not repeated) is valid during 5-10 minutes, but it does not have any limitation for checking. In that case, attacker that knows the answer gets a chance for multiple usage of guessed captcha's identifier. In addition, it is possible to use brute force (a solution of cryptographic task, which checks all possible key options. Difficulty of checking depends on all possible solutions of the task. If solution space is too large, then a result will not come even after several years or centuries) with same identifier, that provide desired solution.

2. Automatic recognition:

There are three main ways of successful automatic captcha's recognition:

I. Using ready-made means of optical character recognition (*OCR*):

That is the easiest approach that does not require any programming skills. Such type of applications are spread free, and there is a lot of them: *ocropus* [2], *cuneiform* [3], *tesseract* [4], *gocr* [5], *orcad* [6] and others.

Attacker has only to send image to such application, and output will provide him a recognized text. Usually, such type of applications have a number of settings, which allow to make recognition process more effective (e.g. number of search algorithm loops, number of image divisions, that will increase probability of successful recognition). So, some tricks are used to avoid recognition: usage of different kinds of disfigurement, different fonts, "noise" etc.

In this case, a possibility of recognition can be low (about 10 %; 10 of 100 captchas are recognized correctly, other are not completely), but attacker will gain success: having a database with recognized captchas will help to find similar records and get the right result.

II. Own written scripts using *GD* (Graphics Draw), *ImageMagick* or other libraries:

Such type of scripts allow to clean image of debris (mass text crossing), clean background, align text vertically, crop image and leaving just text, clean colors etc. Using such scripts in real situation makes recognition difficult enough.

It is more useful to use such script for beforehand image cleaning and leaving main recognition process to other means (e.g. using XRunner application).

III. Neural Network:

Neural networks are most popular now. For someone this is something supernatural. Attackers began to use neural networks with aim to provide automatic recognition of any captchas. They teach neural network that can recognize even the most difficult captchas with high probability.

There are many libraries for various programming language, which are free. One of them is

Fast Artificial Neural Network [7], which journal "Hacker" [5] remembered previously.

3. Half-automatic recognition using cheap human resources:

There are many web sites such as *antigate.com*, *rucaptcha.com*, *captchabot.com* etc., which provide their customers beneficial service. They accept an image of captcha from customer automatically and after a few seconds (10 or 30) send back a correct result as they see it. The percentage of correct recognition is very high: about 90-95 %. It should be noted that a fresh user would have a lower percentage.

A price of such service is about $1-3 for 1000 correctly recognized captchas. You probably would ask who is going to do manual recognition for such low profit. These are citizens of the poorest countries of the world such as India, China, Vietnam, Pakistan, Cambodia etc.

Usually the first think that comes to mind is using Cyrillic alphabet and omit Latin. Someone even try to omit digits. Therefore, as you can see, even such means of protection will not provide a great security, and so soon owners of services will redirect traffic (that means, when Cyrillic captcha comes on, then case of such option fires and chooses co-workers from queue that works with Cyrillic captchas) of such captchas to those, who specializes in Cyrillic.

Others try to complicate captchas using different filters, disfigurement, and "noise" etc., thinking that bot will not recognize them. In this case, they complicate readability and recognition both for users and those, who work in this sphere.

Main customers of such services are big SEO [9] (search engine optimization) companies, that receive statistic data automatically from key words searches about position in output search, about output search etc. In addition, many of spammers use those services that send messages in social networks, which register in automatic mode accounts in mail services on forums etc.

As said above, it is possible to make a list of pros of most captchas:

- Modern bots now "learned" how to recognize even curved symbols, so they can dodge such obstacles

Ben Boyter, Bachelor of Information Technology at the Charles Sturt University, chose theme "Reading the text from the image" (o-called "captcha decoding") for thesis, in a course of which created a neural network. Its algorithm consists of the following steps:

1. Receiving input image and converting into GIF (that makes work much easier, because GIF has only 255 colors).
2. Histogram colors print (a graphical way to introduce table data).
3. Creating table with numbers with number of repeats, that equals a particular color.
4. Finding a number, that equals text color on an image (this neural network works with captchas that have different symbols colors).
5. Creating a new output image that will contain black text to be decoded and white background.
6. Checking if new image has text (symbols and digits).
7. Then it goes horizontal search of symbols
8. After that, it goes image recognition (it accepts two dictionaries and outputs number: "0" – dictionaries are not related or "1" - dictionaries are related).
9. Then it goes neural network teaching to improve captcha's decoding.

- CAPTCHA causes discomfort for user (sometimes he needs to wink and read carefully encoded text, and those will not guarantee correct result), who inputs an appropriate field.
- It is hard to recognize captcha for visually impaired people and daltonian
- It happens that image is encoded too hard; therefore, people with great eyesight cannot recognize the text. Situation becomes more conflict when web form is almost complete (e.g. form with 10 fields and profile photo attachment) and the last one filed is captcha's text, that was not recognized. Sometimes there are captcha fields with reCAPTCHA function, which provides user with new image, though he will not lose input data.

In this way, captcha is not safe protection, moreover comfortable. Therefore, there are many alternatives of classic CAPTCHA: audio captcha, mathematical tasks, text tasks, object recognition etc.

It is a good alternative to use mathematical tasks (e.g. "three plus four equals?"). This solution can provide much better protection only because it is new and less popular. However, you should keep in mind, that mathematical operations are much easier for bot than for human, who can be exhausted while doing calculations

Text tasks are better than classic captcha, because checksum is available for visually impaired people. Therefore, there is no need to create captchas, which will be not safe for such kind of people. However, there is disadvantage of such method: user should know the language, which is used in question. In addition, set of questions is countable, so attacker can create a database of questions and answers.

Nowadays the perfect option is CAPTCHA tests, which use graphical images of different objects. The purpose of such test is before submitting a web form with data user should check images from list, that contain nature, cats or something else. By the way, the order could be required or not. As bots cannot recognize images (because they work with text, not with content), this captcha test is effective protection enough from spam. This security realization from spam is promising enough, because causes less discomfort for user. This type of captcha could be seen, for example, on cell operator web site of “Kyivstar” [4] company when sending SMS (Fig. 9).

Fig. 9. Example of CAPTCHA text

As you can see from Fig. 9, to pass the test and send a message user has to check all images, which contain shop sign.

CAPTCHA 3D [1]: this mean of security represents image generation with volume images, that easy to read for human, than “flat” images. Therefore, a computer has to process 3D objects to recognize a text, and that takes time and causes probability of error.

Fig. 10. Example of 3D captcha

Advantages of CAPTCHA 3D:

- Easy to integrate on site.
- Flexibility (you can gain a particular security level).
- There is no need to follow CAPTCHA images updates.

KCAPTCHA [2]: is a complete solution, that is written on PHP, and can be downloaded and installed for free on your web site to protect it from spam (mass dispatch of ads to people, who did not want it) and flood (vacuity, message on Internet forums or chats, that are large and contains nothing useful). There is set of bitmap fonts in package, perhaps script will work from the start and will not require installation of additional components. It is possible to customize colors and set of symbols, which will be used when creating an image.

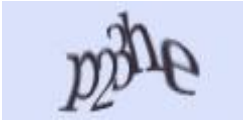

Fig. 11. Example of work of KCAPTHCA

Despite provided applications of captcha, there are means that do not require actions from user. That means are friendly for web site's user, than any captchas, so when using them ease of site use increases. System could limit a number of requests (level of limit is set) and could try to distinguish user from bot by indirect features in behavior ("indirect" features do not provide a warranty, but increase a probability that bot "will no pass").

The main difference between bot application and human is the following: bot sees a web page as a sequence of text and tags, while human sees only a complete result (visual image), and page code is less important. In other words, user sees only fields of registration form and its customization, while bot "reads" that fields: name, value by default, input data type.

Most of means that do not require active actions from user when sending a web from are based on this: usual bot is not taught to interpret CSS (Cascading Style Sheets – special language that is used for page formatting), *JavaScript* (dynamic object oriented language), *Flash*, etc.

The following methods do not require any actions from user:

• Message frequency limit:

In terms of mathematics count of sent messages is provided, which then compared with constant limit. Therefore, user becomes blocked when exceeding that value. Comparison could be provided within established period: for example, each five minutes.

Formula that calculates message frequency (1):

$$n = \frac{1}{\mathrm{T}} \le const, \tag{1}$$

υ – Frequency; T – period, in our example it equals five minutes, and constant value – particular value that is set by default.

Main purpose of method is: it is required to track the same IP address and check if sent messages are not higher, for example, than ten. It is hard to believe that user can send messages with such frequency. It should be noted, that this trick does not protect you from spam, but only reduces it. It has place when using captcha is not an option, but protection is required. This solution is simple, and well-prepared attacker could dodge it.

• Blocking via time of uploading form:

Security could be provided with blocking while filling in a form: that means that human have to have more than one or two seconds to make a message, while bot can handle it within 2 seconds.

• Blocking messages by key words:

Usual filter that works in this way: messages that contain obscene words, offensive and abusive phrases, are blocked automatically and are not published.

For example, user inputs in his message text "I like BAD_WORD". Then, system scans from submitted form that value and with words comparison checks if they are in a blacklist. If that search is successful, then message blocks. If no match found – then message passes through. Therefore, there is a time delay of one or two seconds in real systems

• Replacing fields name that are involved in data transferring:

Most of spam-bots are finding standard name of fields on page, such as *"name"*, *"email"*, *"mail"* etc. It is better to name those fields specially, though to complicate a task for bot. Alternatively, for example, name field "email" with "name", and "name" with "email", expecting that bot will make decision by *name* attribute and its <input> tag.

• Creating a lure fields:

As bot seeks fields *"name"*, *"email"* etc., then why do not give them him? We have to create hidden field (but not with attribute *hidden*) using means of CSS. Regular user will not see it and essentially will not fill in it. In addition, spam-bot acts vice versa – it will find it and fill in. Therefore, the web form is going to be passed when those "hidden" field are empty. In addition, there should be lure fields visible for user, but with non-regular names.

• Blocking messages with display size:

This method of protection works with simple scheme: if software identifies that visitor does not have screen dimensions (width and height), then bot is trying to send a form, therefore it should not be processed.

To add to said above alternatives we can include user register and comments moderation. The main purpose of this method is, for example, comments in Guest room can leave only particular members (registered). Registration is so-called obstacle for spam spreading. In addition, moderation is a process, which checks comments and leaves only those, which are approved by administrator of Web resource: he gets lists of comments left by users, which he has to check and analyze for words that are in blacklist. After result, there are two options: administrator leaves or not the comment.

Disadvantages of such approach are: administrator has to pay attention and use his time to check users, therefore to delete or edit incorrect messages or approve comments etc. By the way, there is discomfort for users: not everyone will desire to register with aim to leave a comment about published article or to wait for approval of his post in Guest room

Conclusion

These subject is topical nowadays, therefore the question about protection and anti-spam is still popular, and probably will be require clear answer. Because each solid company and a trainee web developer are interested in high attendance of their web sites and without spam and bot users. In this case, people in such field of activity have to think not only of web site security, but also about means how to dodge this though to prevent unpredictable situations.

Those described means of protection and ways of web resource solution will not guarantee exactly that bots will not spam site or forum. There are always presence of bottlenecks even in most popular and perfect system. Here is an example when on web sites all over the world in SQL (Structured Query Language) requests was found boner, that was getting the name of SQL Injection – one of the most widespread ways to hack a site or applications, that works with databases, based on inserting any SQL-query code.

As a rule, professional attackers always follow a rule: the price of captcha dodging should not be higher of a possible profit. From here is a conclusion: if site has a big attendance, then spam attacks grows faster.

Nevertheless, you should keep in mind that honest users (they are 99.99 % of all users) should not be harmed because of spammers.

1. CAPTCHA [Electronic resource] – Access mode: https://uk.wikipedia.org/wiki/CAPTCHA. 2. Web services security from spam [Electronic resource] – Access mode: http://captcha.opti-mail.net/. 3. Checking captcha algorithm [Electronic resource] – Access mode: http://www.captcha.ru/articles/algorithm/. 4. Building a security image [Electronic resource] – Access mode: http://www.captcha.ru/articles/visual/. 5. Alternative ways of security [Electronic resource] – Access mode: http://www.captcha.ru/articles/alternatives/. 6. CAPTCHA [Electronic resource] – Access mode: http://www.captcha.ru/. 7. CAPTCHA: humans and computers [Electronic resource] – Access mode: http://www.captcha.net/