ISSN No:-2456-2165

A Gamming of Captcha- The Confrontation Against Captcha with Implementation of the Random Dice with RSA

Dr. R. Rajesh
Assistant Professor and Head,
PG and Research Department of Computer Science,
Kaamadhenu Arts and Science College.

Abstract:- In the web there has been a play of undeniably significant jobs in our day to day life, with the openness of many web admins, likewise, web creeps and e-mail. In any case, these are frequently imperiled by assaults from PC AI-Bots Programs. To handle these type of issues, CAPTCHA [Completely Automated Public Turing Test to Tell Computers and Humans Apart] was created to recognize Client AI- Bots projects and human users. In spite of the fact that these frame work offers great security and cutoff programs enrolled to the web admins and some CAPTCHA's have a very few shortcomings which will permit these AI-Bot Programmers to penetrate the system of the CAPTCHA. This research work analysis the different CAPTCHA strategies and actualization with even bites the dust and RSA Classifications.

Keywords:- *Captcha, Gamming, Random, Dice.*

I. INTRODUCTION

The web contributes significantly an excessive number of fragments of user day to day life, for instance, interchanges, exercise, and online commercial practices etc.,. Some web administrations have online enlistment where the clients give data so as to interface and use administrations, for example, email in Yahoo, Gmail and Hotmail, be that as it may, numerous projects have been created by programmers which consequently complete site enrollment pages with mistaken data which can cause traffic clog, limiting the presentation of the framework and now and again, in any event, making it come up short, especially where a site has countless records.

In this manner, analysts built up a system to recognize human clients and PC programs such as Bot Programs on account of online enrollment. The standard component utilized right now to address this issue is CAPTCHA [Completely Automated Public Turing Test] to Tell Computers Bot programs and Humans Apart). The idea of CAPTCHA depends on the capacity of people to carry out specific responsibilities which PC programs can't, for example, requesting that clients type a contorted book picture or pick a specific picture from many showed pictures.

II. DEVELOP THE IMPLEMENTATION AND DEMONSTRATION OF RANDOM DICE ROLL

➢ *Different Dice Shapes:*

Most common shapes used in dices are listed below and shown in the image.

- Tetrahedron: Four Faces
- Cube: Six Faces
- Octahedron: Eight Faces

➢ *Non-Cubic Dies:*

- Trapezohedron Pentagonal: Ten Faces
- Dodecahedron Pentagonal: Twelve Faces
- Icosahedron: Twenty Faces

Fig. 1.Non-Cubic Dies

Though the above image shows some kind of more commonly used dice's shapes, there are many more polyhedral dices or dices with other shapes. There are also non-numeric die, dices which do not follow a counting some similar sequences that begins at one at a time, for instance spherical die.

➢ *Exactly how random is a dice?*

In light of likelihood, a bite the dust ought to have an equivalent likelihood of arriving on every one of its countenances. In any case, this isn't really the situation with accumulation delivered dice as they can't be genuinely arbitrary, subsequently it is hard to mass produce dice that are uniform and there might be contrasts in the evenness of the Frame. Each frame, specifically d20 [20-faces POLYHEDRAL Frame) & d8 [8- Faces polyhedral shakers] are frequently uneven, bound to roll certain numbers.

➢ *How to test a how random your dice is:*

Even though it may not be the most accurate way of test how random your dices are, one relatively quick test you can process these involves just a container, some water and some salt:

- A container that can fit the dice you want to test with water.
- Then add salt and the dice to the water – if the dice does not float, add more salt until the dice floats.
- Flick the dice and take note of the side which faces upward repeatedly and then record the result.

For a well-balanced dice, other than variety of numbers, if it is not well balanced it will be more likely to notice certain numbers occurring more often. Though, without these tests were performed many times, (or) the dice is heavily unbalanced, the client is not likely to notice a significant difference.

Here are various organizations that assembling die, certain increasingly thorough tests [than the one portrayed above] have been performed on dice made by various organizations with an end goal to decide how genuinely irregular the frame [generally 20 frame] are. These examinations affirmed that even shakers fabricated inside a similar organization under similar conditions can differ altogether from one another, and are not genuinely arbitrary. A few organizations delivered dice that were more irregular than others, yet and still, at the end of the day, were not seen as genuinely random.

Computer-generated dices are likely the one above, are almost always based on pseudo-random number generating algorithms, which are also not truly random. Though, a computer generated dices rolls are likely more close to true random dices than the most physical dices.

➢ *The Proposal Solution:*

To the occurrence factors let's consider dice1 and dice2 are announced to be open. These should be private, so that they are shielded from being changed from outside the class. Compose another form of the PairofDice class in which the algorithm factors dice1 and dice2 are private. In this class we will require techniques that can be utilized to discover the estimation of dice1 and dice2. [Alleged is to protect their qualities from being attacked from outside the class, but at the same time to authorize the qualities to be peruse.] Include different enhancements in the class, in the event that you can think about any.

To verify our class with a short program that counts how many times a pair of dice is rolled, before the total of the two dices are equal to two.

This part of the TwoDices class is different in how the dice are initialized. Adjusting the dice's to random values, so we will research with the following classes:

```
public class TwoDices
{

public int dies1;
public int dies2;

public PairOfDice() {
roll();
}
public void roll()
            {
dies1 = (int) (Math.random()*6)+1;
dies2 = (int) (Math.random()*6) +1;
            }
}
```

Now a TwoDices object for the above class is created, the number on each dice is definitely between 1 and 6 the number on each die is definitely between 1 and 6, like the number on a real dice, without the instance variables die1 and die2 are private, since they can not be changed from outside the class. So that there is to stop someone from hack them By making dies1 and dies2 private, we can guarantee, because the code that we write in the TwoDices class is the only code that will never affect the values of the variables.

So, we will make dies1 and dies2 private and add instance methods a) getDies1()) b) getDies2() to return the values of dies1 and dies2. If this is going to be done over, we can devolve method in the class to do so that to prevent duplicate codes, then also add a function getTotal() that returns the total value which shows in the two dice and the random roll of the TwoDices and start the RSA Encryption from this point onwards.

Note how the pair of dice object is employed. To test whether or not the entire the dice is 2, to implement in test "while (dice.getTotal() != 2)".

To show the numbers on the two dice, System.out.println("The dice come up "+dice.getDie1()+ " and " + dice.getDie2());.

Fig. 2.Two Dice Random Roll

III. PROPOSED SYSTEM: DEVELOP THE RSA [GCD] IMPLEMENTATION WITH PAIR OF DICE ALGORITHM IN CISCO IOS

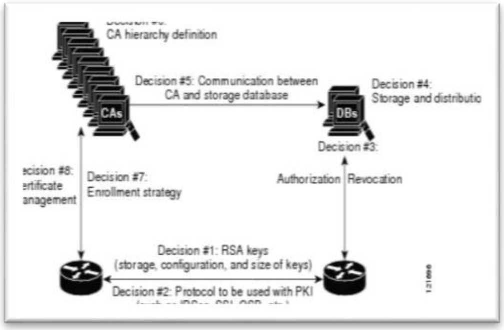

CA hierarchy definition
Decision #5: Communication between CA and storage database
Decision #4: Storage and distributio
CAs
DBs
Decision #3:
ecision #8: ertificate anagement
Decision #7: Enrollment strategy
Authorization
Revocation
Decision #1: RSA keys (storage, configuration, and size of keys)
Decision #2: Protocol to be used with PKI
121886

Fig. 3.PKI Simulation Diagram

The RSA key pair consist of a public key and a personal key. While setting up the PKI [Public Key Infrastructure], must include the pairofDice() algorithm to generate private key in the certificate enrollment request.

After pariofDice() added the certificate has been granted, the private key are going to be included within the certificate in order that peers can use it to encryption of data which is sent to the router.

Public key is kept on the router and used both the keys for decryption of the information sent by the host and to dogotally sing transaction when negotiating with the host.

➢ *Proposed Algorithm:*

- Certificate Enrollment with Dice Random Roll
- The end host generates the two dice random roll numbers (pairofDice())for RSA Key pair.
- Then the end host generates a certificate request and forwards it to the PKI[CA].

➢ *Then PKI [CA] receive the certificate enrollment request and on accord on the network configuration, the following options will occur:*

- The pairofdice() [Manual intervention] is required to approve the request.
- Then the end host is configured to automatically request a certificate from the PKI [CA].

Thus, the operator intervention is no longer required at the time the enrollment request is sent to the PKI server.

The RSA Key pair contains a pairofDice.pubkey module clause. The module does not determine the size of the RSA Key. In this we have used 2048 key size. Conversely, the Keys with large module values take longer to generate an encrypted data and decrypting the same data with larger Keys.

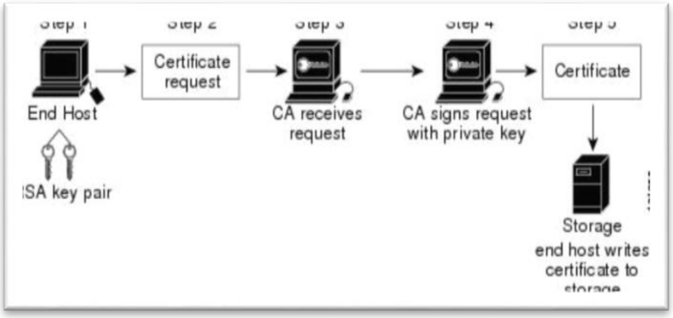
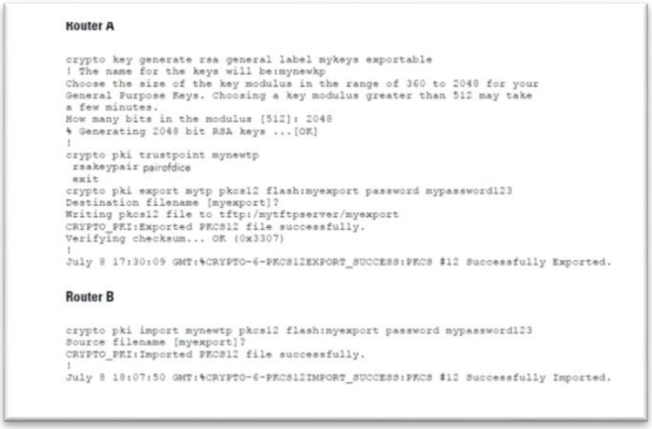


Fig. 4.PKI Simulation Diagram



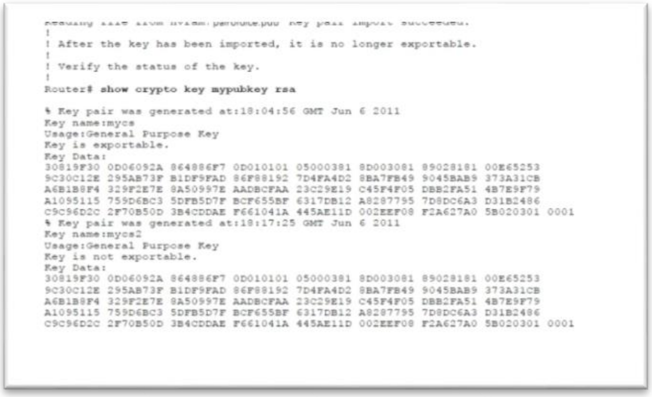
Router A

```
crypto key generate rsa general label mykeys exportable
! The name for the keys will be:mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys ...[OK]
!
crypto pki trustpoint mynewtp
 rsakeypair pairofdice
 exit
crypto pki export mytp pkcs12 flash:myexport password mypassword123
Destination filename [myexport]?
Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
!
July 8 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

Router B

```
crypto pki import mynewtp pkcs12 flash:myexport password mypassword123
Source filename [myexport]?
CRYPTO_PKI:Imported PKCS12 file successfully.
!
July 8 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```

Fig. 5.RSA Simulation Model



```
Reading file from nvram:pairofdice.pub Key pair import succeeded.
!
! After the key has been imported, it is no longer exportable.
!
! Verify the status of the key.
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:18:04:56 GMT Jun 6 2011
Key name:mycs
Usage:General Purpose Key
Key is exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
% Key pair was generated at:18:17:25 GMT Jun 6 2011
Key name:mycs2
Usage:General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
```

Fig. 6.RSA with pairofdice Key generation

Table 1: Tabulation for Total Packets Dropped

Time(ms)	**Before Encryption Packets dropped**	**After Encryption Packets dropped**
0	0	0
5	90	25
10	68	39
15	70	52
20	80	35
25	76	25
30	65	31
35	59	38
40	53	22
45	48	26

IV. CONCLUSION

The instance variables are declared to be public Key in RSA. The number in die1 and the number in die2 are the private Key, so that they are protected from being hacked or attacked from the outside network. To create PairOfDice class, in which the instance keys die1 and die2 are private, the key will need the decryption methods that can be used to find out the values of die1 and die2. We can include other improvements in the Pair of dice with Un-balanced dice and also with RSA algorithm, if we can think of any so that there are no attacks can occur.

REFRENCES

[1]. Walid hasan, al jabal, "a survey of current research on captcha" (ijcses vol.7, no.3, june 2016

[2]. Sushma yalamanchili & kameswara rao "a framework for devanagari script-based captcha", (ijait) vol. 1, no. 4, august 2011

[3]. albert b.jeng,chien-chentseng, der-feng tseng,et.al.,"a study of captcha and its application to user authentication", technologies and applications. Iccci 2010. Lecture notes in computer science, vol 6422. Springer, berlin, heidelberg

[4]. Human aspects of information security, privacy, and trust. Has 2013. Lecture notes in computer science, vol 8030. Springer, berlin, heidelberg

[5]. Chih hsu ch., lee yl. (2011) effects of age groups and distortion types on text-based captcha tasks. In: jacko j.a. (eds) human-computer interaction. Users and applications. Hci 2011. Lecture notes in computer science, vol 6764. Springer, berlin, heidelberg

[6]. Kim, j., chung, w. & cho, h. A new image-based captcha using the orientation of the polygonally cropped sub-images. Vis comput 26, 1135–1143 (2010).

[7]. Ikeya y., fujita m., kani j., yoneyama y., nishigaki m. (2014) an image-based captcha using sophisticated mental rotation. In: tryfonas t., askoxylakis i. (eds) human aspects of information security, privacy, and trust. Has 2014. Lecture notes in computer science, vol 8533. Springer, cham

[8]. Crocce, f., & mordecki, e. (2016). A finite exact algorithm to solve a dice game. Journal of applied probability, 53(1), 91-105. Doi:10.1017/jpr.2015.11

[9]. Smith, d. R., & scott, r. (2018). Golden arm: a probabilistic study of dice control in craps. Unlv gaming research & review journal, 22(1). Retrieved from https:// digitalscholarship. Unlv. Edu/grrj/vol22/iss1/1

[10]. R. Rajesh, m. Ramakrishnan and b. Sugumar, "a modest approach on manet using certificateless cryptography," 2017 international conference on intelligent sustainable systems (iciss), palladam, 2017,pp.11971204.doi:10.1109/iss1.2017.8389376 url: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8389376&isnumber=8389225

[11]. V.perumal, dr.k.meenakshisundaram "cluster based secured data transmission using hybrid cryptography techniques in wireless sensor network", international journal of information engineering and science (ijies) scopus indexed journal – communicated.

[12]. International journal of scientific & engineering research volume 11, issue 12, december-2020 96 issn 2229-5518 jser © 2020 http://www.ijser.org deep learning of attack detection and low rate tests on ddos attacks dr. R. Rajesh, dr. M. Ramakrishnan, p. Ganesan.

[13]. Modelling and simulation of ddos attack using simeventsabubakarbala*1 and yahya osais2

[14]. The top 10 ddos attack trends, discover the latest ddos attacks and their implications, www.imperva.com.

[15]. A deep learning based intelligent framework to mitigate ddos attack infogenvironmentrojalinapriyadarshini⇑, rabindra kumar barikkiit university, bhubaneswar, india article info article history: received 28 september 2018 revised 16 april 2019accepted 17 april 2019. Journal of king saud university computer and information sciences journal homepage:www.sciencedirect.com

[16]. Diro, a.a., chilamkurti, n., 2018. Distributed attack detection scheme using deeplearning approach for internet of things. Future generation comput. Syst. 82,761–768.

[17]. Erolgelenbe, michael gellman, george loukas, an autonomic approach todenial of service defence, in: sixth ieee international symposium on a world ofwireless mobile and multimedia networks, 2005. Wowmom 2005, june 2005,pp. 537–541.

[18]. Hasan, walid. (2016). A survey of current research on captcha. International journal of computer science & engineering survey. 7. 1-21. 10.5121/ijcses.2016.7301.

[19]. Ning zhang, mohammadreza ebrahimi, weifeng li, hsinchun chen, "a generative adversarial learning framework for breaking text-based captcha in the dark web", intelligence and security informatics (isi) 2020 ieee international conference on, pp. 1-6, 2020.

[20]. Jeng a.b., tseng cc., tseng df., wang jc. (2010) a study of captcha and its application to user authentication. In: pan js., chen sm., nguyen n.t. (eds) computational collective intelligence. Technologies and applications. Iccci 2010. Lecture notes in computer science, vol 6422. Springer, berlin, heidelberg. Https://doi.org/10.1007/978-3-642-16732-4_46