Comparasion of Password Generator between Coupled Linear Congruential Generator (CLCG) and Linear Congruential Generator (LCG)

Imamah1, A Djunaidy2, A Rachmad3, and F Damayanti4

1,4 Department of Information System, Faculty of Engineering, University of Trunojoyo Madura, Jl. Raya Telang, Kamal, Bangkalan, Madura 69162 Indonesia
2 Faculty of Information Technology, Institute of Technology Sepuluh Nopember, Surabaya, Indonesia
3 Department of Multimedia and Network Engineering, Faculty of Engineering, University of Trunojoyo Madura, Jl. Raya Telang, Kamal, Bangkalan, Madura 69162 Indonesia

Email: i2munix@gmail.com

Abstract. Password is needed to access the computing services. Text password is a combination between characters, numbers and symbols. One of issues is users will often choose guessable passwords, e.g. date of birth, name of pet, or anniversary date. To address this issue, we proposed password generator using Coupled Congruential method (CLCG). CLCG is a method to solve the weakness of Linear Congruential generator (LCG). In this research, we want to prove that CLCG is really good to generate random password compared to LCG method. The result of this research proves that the highest password strength is obtained by CLCG with score 77.4%. Besides of those things, we had proved that term of LCG is also applicable to CLCG.

1. Introduction

Text password is used to authenticate users to online services. Text password is a combination between characters, numbers and symbols. It has a weakness but text password is more widely used. Many attempts were made to replace simple password authentication, e.g. using biometrics, Tokens and multi-factor authentication. However, single factor password which is based on authentication remains very common [1]. Single-factor authentication is the simplest form of authentication methods[2]. A person uses Single-Factor authentication to verify himself online. The most popular example of single-factor authentication is a password. Most verification today uses this type of authentication method. In recent years the number of widely used password protection services has grown significantly, so the number of password users is expected to increase. There are various issues related to the use of text password. One of issues is users will often choose guessable passwords, e.g. date of birth, name of pet, or anniversary date [3]. Many people use passwords, for which most of them use a simple word such as “password” or numbers such as “1234.” Despite people already perceive that a simple password is not secure enough, they still use simple password because it is easy to use and to remember[4]. A malicious user who knows the user personally or able to find out certain things about user, such as the birthdate, favorite actor/actress or pet’s name, their password will be easily cracked. According to many service providers, awareness campaigns (US DHS), and government entities (US-CERT) stress two foundations for password security[5]:
A1: Passwords should be random and strong; and
A2: Passwords should not be re-used across accounts.
To address this issues, password generator have been proposed, which generate strong password that is
difficult to guess random-looking passwords and regenerate them whenever necessary. In this paper
we provide the first detailed specification of password generator using coupled linear congruential
generator, and also give a detailed analysis of its properties.

2. Related Work
The concept of Password Generator is not new and not too old. In 2016, Abderrahim et all [6]
proposed scheme for strengthening authentication in the cloud environment using the password
generator to surmount the security flaws of login/password scheme. Ani Ashish [7] presented A ‘Password Generator’ mobile app that can generate many passwords and store complex password for
the above mentioned e-mailing sites, social networking sites and for debit cards.

3. Proposed Work
Password generator is created in order to generate a strong and secure password. One example
of using this password generator is on any application that does not provide the user registration
feature. The right to register a user is on the admin, so it needs a password generator. In this study, the
coupled linear congruential generator method is used to generate strong passwords.
3.1. Coupled Linear Congruential Generator
The linear congruential generator (LCG) is a method to generates random sequences that are
linear, fast, simple and easy to implement, thus they have wide range of uses in simulation
applications. But recently there have been great demand for random generators by security
applications like automatic password generation, on-line gambling, digital signatures, encryption
algorithms etc [8]. All linear congruential generators use this formula:

\[ X_{i+1} = aX_i + b \mod m \]  \hspace{1cm} (1)

Where:
- \( X_0 \) is a seed.
- \( X_1, X_2, X_3, \ldots \) are the random numbers.
- \( a, b \) and \( m \) are constant.

If one chooses the values of \( a, b \) and \( m \) with care, then the generator produces a uniform
distribution of integers from 0 to \( m-1 \). LCG numbers have poor quality. \( X_n \) and \( X_{n+1} \) are not
independent, as true random numbers would be. Anyone who knows \( X_n \) can predict \( X_{n+1} \), therefore
LCG is not cryptographically secure. The weaknesses of a single LCG are removed by the coupling.
This is primarily because solving inequalities modulo \( m \) can only be done by searching through the
entire solution space[9]. Coupled LCG is defined as follows.

\[ X_{i+1} = a_1X_i + b_1 \mod m \] \hspace{1cm} (2)
\[ Y_{i+1} = a_2Y_i + b_2 \mod m \] \hspace{1cm} (3)
\[ B_{i+1} = \begin{cases} X_i + 1 > Y_i + 1 \\ 0 \text{ otherwise} \end{cases} \] \hspace{1cm} (4)

We assume that \( a_1, b_1, a_2, b_2 \) and \( m \) are known and the seed \((x_0, y_0)\) is secret. Coupled LCGs turn
out to be more secure than a single LCG [9,10]. According the result of equations, then we get a
random number series that will be converted into matrix order. Order of matrix is obtained from the
calculation modulus to number of lines, so that we obtain the equation (3):

\[ M(x,0)=X_{n+1} \mod i \] \hspace{1cm} (5)
Order of matrix obtained from the calculation modulus to number of column, so that we obtain the equation (4):

$$M(0, y) = Y_{n+1} \mod j$$

(6)

The mathematical model to get a result from random number using a new matrix can be calculated by using equation (5):

$$M_n = M[X_{n+1} \mod i][Y_{n+1} \mod j]$$

(7)

Where:
- $i$ is row of matrix
- $j$ is column of matrix

3.2. Algorithm

Password generator with Linear Congruential Generator works in the following ways. The determination maximum characters of the password $(m)$ is generated in the first stage. After determination $(m)$, a multiplier factor variable $(a)$ is performed. The third stage is determination variable $(b)$. The final stage is randomization Lower case, uppercase, number and symbol using equation (1).

Password generator using CLGC is different with LCG method. The first stage of CLGC is determination rows and columns of matrix $A$ where the number of elements corresponds to length of a password that is generated. Then, calculating variables $a$, $b$, and $m$ using similar ways as LCG must be repeated two times. The third stage is calculating random numbers $x$ and $y$ according to equation (2). Formation of $M_n$ matrix and $M_n$ matrix order adjustment with matrix $A$ according equation (3-5) is the end of the process randomization letters, numbers and symbols into a passwords.

4. Results and Discussion

Password generator is generated using LCG and CLCG. Password is generated by using combination of characters, numbers and symbols. The results of this study are shown in the Table 1. Password Strength is measured by using a password meter[11]. The combination of passwords is obtained from calculations using $a = 5$, $b = 5$ and $m = 8$, where value of $m$ is equal to the number of characters or length of password for LCG method. The values in the CLCG method we set $a_1 = 5$, $b_1 = 5$, $a_2 = 5$, $b_2 = 3$ and $m = 8$, the determination of this constant value are based on previous research conducted by Raj S Katti [6]. Table 1 shows the result of generate password using constant value in Raj S Katti research with average password strength for LCG method is 60.5% and using CLCG method is 77.4%.

| Table 1. Comparison password strength of LCG and CLCG. |
|-----------------------------------|------------------|------------------|------------------|------------------|
| LGC (a=5, b=5)                  | Password Strength (%) | CLCG (a_1=5, b_1=5, a_2=5, b_2=3) | Password Strength (%) |
|-----------------------------------|------------------|------------------|------------------|------------------|
| 1 PpJKoveQ                       | 34               | bj$Sjd$e         | 57               |
| 2 ZMXe3v4A                       | 68               | ZpPkjO#          | 72               |
| 3 qwAlJp@B                       | 60               | lzs%vTZ#         | 68               |
| 4 D41HI3pt                       | 74               | 6$kWdOR6         | 77               |
| 5 AD6Su8KF                       | 68               | &cUH2$h$e        | 100              |
| 6 MHR1%LX1                       | 70               | #mTp9ZUj         | 72               |
| 7 RLxBVlee                       | 30               | esb6Tw%i         | 72               |
| 8 Hkh28hJz                       | 67               | oxN!nme7         | 68               |
In our experiment, we change value \( a \) and \( b \) to compare and determine its effect to complexity of password. In first test, we set \( a=5 \) and \( b=5 \) for LCG method and the result is show in table 1. Second test, we set value of \( a=a-1 \) and \( b=b-1 \), so that with this reduction, the value of \( a=4 \) and \( b=4 \) for LCG method, and we set \( a_1=4, b_1=4, a_2=4, b_2=2 \) for CLCG. The result of the second experiment is shown in the table 2.

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
|   | LGC (\( a=4, b=4 \)) | Password Strength (%) | CLCG (\( a_1=4, b_1=4 \) \( a_2=4, b_2=2 \)) | Password Strength (%) |
| 1 | 6xTJvXpn | 58 | SZ4Ezv6u | 68 |
| 2 | 3jibQJbb | 49 | $jZPNJJv | 54 |
| 3 | QEZdmtW | 34 | %lWIOz9f | 70 |
| 4 | ^1Jkq7pc | 54 | FT49DYr | 68 |
| 5 | xK(opp)*g | 70 | @Dpt1pB | 70 |
| 6 | fBpEHNyq | 34 | jUJUYnwF | 55 |
| 7 | v9RHT&oO | 72 | lk1n1q7S | 70 |
| 8 | *lbcdrq | 55 | *CbJner^ | 68 |
| 9 | jQDK7GSD | 55 | %ubTE2O1 | 70 |
| 10 | VK9HK@7U | 61 | kv^C(5m5 | 92 |

| Average | 54.2 | 68.5 |

The second test shows that the reduction of \( a \) and \( b \) affects the average score of strength password. In the LCG method, it is decreased into 6.3%, and 8.9% for the CLCG method.

In the third test, we set value of \( a=a+1 \) and \( b=b+1 \), so that with this addition, the value of \( a=6 \) and \( b=6 \) for LCG method, and we set \( a_1=6, b_1=6, a_2=6, b_2=4 \) for CLCG. The result of the third experiment is shown in the table 3.

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
|   | LGC (\( a=6, b=6 \)) | Password Strength (%) | CLCG (\( a_1=6, b_1=6 \) \( a_2=6, b_2=4 \)) | Password Strength (%) |
| 1 | WMnfdpNo | 32 | ag%^*rmmw | 44 |
| 2 | MSGgsqH | 30 | uCEdZFb# | 60 |
| 3 | ZzNpnD^r | 64 | iU8F2OoP | 72 |
| 4 | CwighG6P | 58 | jwQ*Rge5 | 80 |
| 5 | y4Fhoyd | 54 | ZPN&b& | 69 |
| 6 | THDmQi5P | 60 | k6wtWqvk | 57 |
| 7 | 9AHsCLzN | 58 | GrkS1Jd | 76 |
| 8 | 3x%KCAo | 67 | *El9t8!A | 90 |
| 9 | UavZ01Ab | 60 | AP16fBH6 | 77 |
| 10 | tpsi#NZe | 58 | iSeYay(g | 72 |

| Average | 54.1 | 69.7 |

The third test shows that the addition of \( a \) and \( b \) affect the average score of strength password. In the LCG method, it decreased into 6.4%, and 7.7% for the CLCG method.

5. Conclusion
Based on the tests that we had been done, it indicates that changes values of a and b on LCG or on CLCG influences the results of randomization generated. The determination values of a and b determines the passwords strength that is generated. Besides those things, password strength that is generated by using Coupled linear congruential generator (CLCG) method is also determined by the terms applicable to the LCG method. One of the term is a value of b which should have relatively prime to m. In the first test we set b=5, it means that b is relatively prime to m=8. The average score of password strength in the first test is highest than other, it proves that b should have relatively prime to m to get highest password strength. On the other terms, a-1 is a multiple of 4 if m is a multiple of 4, which is also proven to be applicable to CLCG. Meanwhile, the highest score is obtained at a = 5 or a-1 = 4 because m = 8.

6. Acknowledgment

We thank the Indonesian government RISTEKDIKTI, University of Trunojoyo Madura, and Institute Technology Sepuluh Nopember which has funded this research through research programs for lecturers.

References

[1] Fatma Al Maqbali, Chris J Mitchell. AutoPass: An Automatic Password Generator. 2017. Researchgate.
[2] Gunson, N, Marshall, D, Morton, H & Jack, M. 2011, User perceptions of security and usability of singlefactor and two-factor authentication in automated telephone banking. Computers & Security, vol 30, no. 4, pp. 208-220. DOI: 10.1016/j.cose.2010.12.001.
[3] Dinei Florencio, Cormac Herley, and Paul C van Oorschot. Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts. In Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014, pages 575-590. USENIX Association.
[4] Junho jeong, Jung-Sook Kim. Automatic Fortified Password Generator System Using Special Character. International Journal of Fuzzy Logic and Intelligent Systems. 2015. Vol. 15, No 4, pp 295-299.
[5] Dinei Florencio, Cormac Herley et all. Password Portfolios and Finite-Effort User: Sustainably Managing Large Numbers of Accounts. Proceedings of the 23rd USENIX Security Symposium. 2014, pp. 575-590.
[6] Abderrahim Abellaoui, et all. A Novel Strong Password Generator for Improving Cloud Authentication. International Conference on Computational Modelling and Security (CMS). 2016. Elsevier, pp 293-300.
[7] Ani Ashish, Kumari Priti. Password Generator: A Utility Mobile Application for Android. International Journal of Science, Engineering and Technology. 2015. Vol. 2, Issue: 1, pp 53-58.
[8] Matthew N. Anyanwu, Lih-Yuan Deng, Dipankar Dasgupta. Design of Cryptographically Strong Generator By Transforming Linearly Generated Sequences. International Journal of Computer Science and Security (IJCSS), Vol.3, Issue 3. 2015, Pp 186-200.
[9] Raj S Katti, Rajesh G Kavasseri. Secure Pseudo-Random Bit Sequence Generation using Coupled Linear Congruential Generators. IEEE. 2008. Pp 2929-2932.
[10] Raj S Katti, Rajesh G Kavasseri. Pseudorendom Bit Generation Using Coupled Congruential Generators. IEEE:Express Briefs, Vol.57, No 3. 2010. Pp 203-207.
[11] Steven Van Acker, Daniel Hausknecht, Andrei Sabelfeld. Password meters and Generators on The Web: From Large-Scale Empirical Study to Getting It Right. Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. 2015. Pages 253-262.