Webology (ISSN: 1735-188X)
Volume 19, Number 2, 2022

Database meet Link generator based on linear feedback shift register and message digest algorithm

Ismael Abdul Sattar Jabbar 1st , Shaimaa Hameed Shaker 2nd , Mohammed Najm Abdullah 3rd

[1]University of Mustansiriyah Iraq, Baghdad

[2]University of technology Iraq, Baghdad

[3]University of technology Iraq, Baghdad

Abstract

The fast growing of data transferring over the internet for several services like social media interaction or e-learning requirements need a secure mechanism to generate a link for participating or joining these services. In this paper linear feedback shift register (LFSR) used to construct link generator system combined with message digest algorithm (MD5) as hash algorithm to get valid links with wide unique range. The proposed system simulates the google meet links generator as possible. The result shows the success of generated links output as well as verifying links validity.

Keywords: Links generators, Google meet links, linear feedback shift register (LFSR), message digest 5 (MD5)**.**

Introduction

Link generator one of the important requirement in the digital media now a day due to the wide use of the electronic learning as well as other activities of the human life through covid-19 pandemic. The links used to provide electronic meeting for a lot of purposes that's why become in need not only producing of a unique link but also secure link as much as possible to provide invitation for all the participants. The proposed system work on the combination of linear feedback shift register (LFSR) mechanisms as well as massages digest 5 (MD5) as hash function.

The important function for the hash functions S is taking input K with a variable length, such function producing a fixed length of hash code (hash value "v") as output[1]:

$$v = S(k)$$

Because of this property of tacking variable length and producing fixed length there are a wide uses of hash function as shown with figure (1).

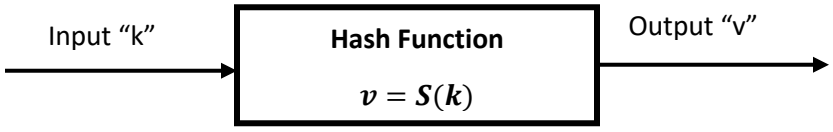

Figure 1. Hash Function.

There are two most common features available in the hash functions algorithms made such algorithm inter the cryptography as well as steganography doors, first feature hash functions S(k) is "one way function", on the other hand the second feature is "collusions free"[2,3].

"Hash functions" used for many kinds of security areas the main role is to achieve the integrity. Hash function used [4]:

1. Used Alone:
   1.1. File integrity verification.
   1.2. Public key fingerprint.
   1.3. Password storage.

Combined with encryption functions.

2. Information hiding.

The proposed system utilizing of message digest "MD5" as hash function used for two times, first combined with encryption and second achieve integrity and authentication. MD5 considered as enhanced form of MD4. Though extra complexity compared to the MD4 algorithm, it is similar in design and also gives a 128-bit as output [5]. The main loop of the MD5 algorithm can be shown with in figure. (2).

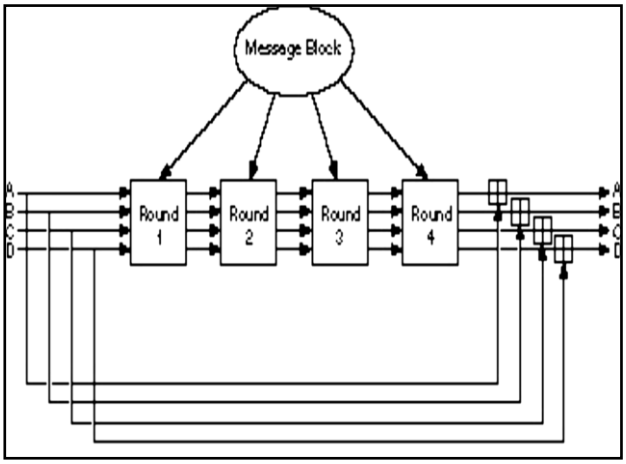

Figure 2.MD5 main loop.

Such that the first step is add the expanded in process to make it 64-bit with multiple of 512-bit length. This process done by adding 1-bit to the end of the messages followed by set of zeros as required. Four variables of 32 bits used for initialization called chaining variables [6]:

1. A = 0x01234567
2. B = 0x89abcdef
3. C = 0xfedcba98
4. D = 0x76543210

The main loop of the MD5 algorithm contains 4 round these lops continue for running 512-bit length of the message as required. The four variable copied into other variables like (a copied A, and b copied B and so on). At each round the different operation applied in 16 times in such a way that each operation process nonlinear operation on variables a, b, c, and d. the functions can be applied at each operation as follow [7] [8]:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ denote the XOR, AND, OR and NOT operations respectively.

These functions well planned so that condition the equivalent bits of "X, Y, and Z "are autonomous and impartial, such that, the bit of the output will even have considered autonomous and impartial. The function F is the bitwise restricted: "If X Then Y Else Z". The function H is the bit-wise parity operator**.** One MD5 operation can illustrated using Figure. 3

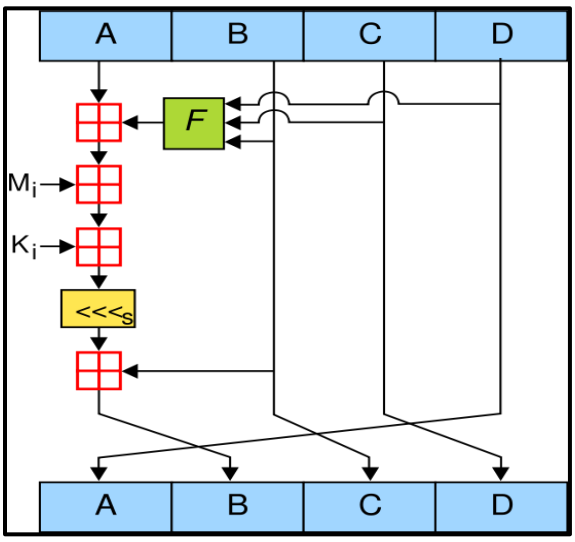

Figure. 3 One MD5 operation [9]

Proposed system

The email ID hashed using MD5 hash algorithm to produce 128 bits that will be unique values used as seed key fed into 5 Linear feedback shift registers each register

with capacity of 25 bits seed key totally will use 125 bits out of 128 hash bits as shown in Figure (4)

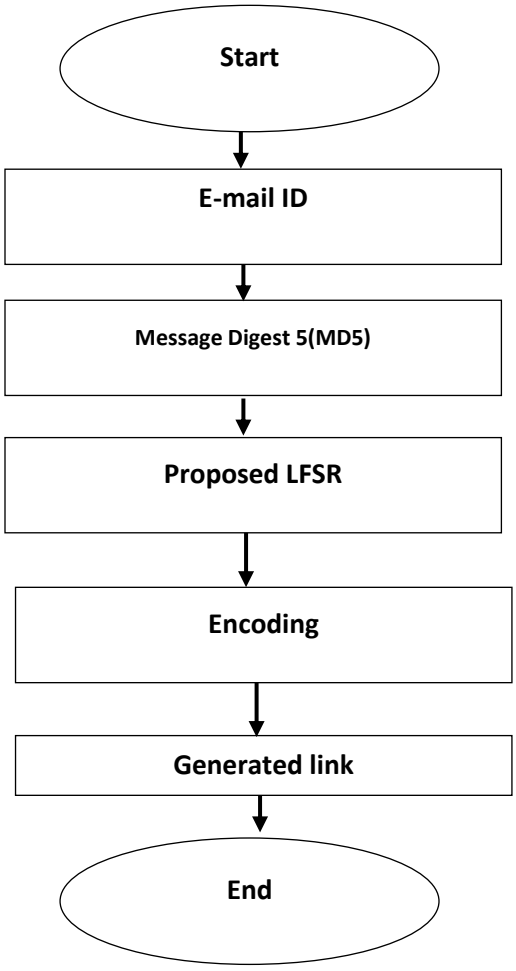

Figure (4) flowcharts of the proposed system.

The structure of the linear Feedback shift registers used in the proposed system can be shown with the figure 5.

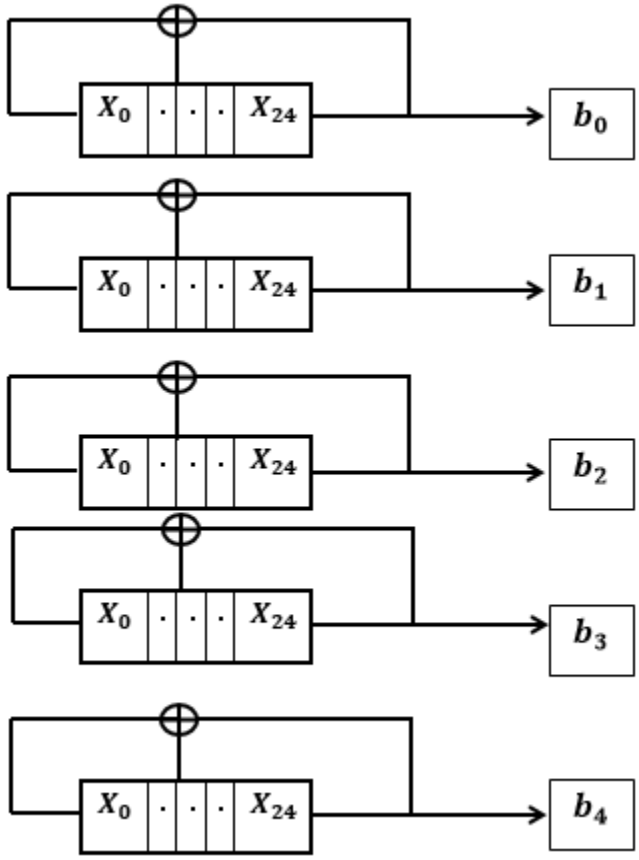

Figure 5. Proposed LFSR Generators utilizing in link generators.

The join function in LFSR will be determine by the following steps

1. Convert first 5 bits out of 25 into decimal value D.
2. Fund the target cell using $j = D \bmod 25$. Because the range of D is [0…31]
3. Join function of the LFSR is $f = j \oplus X_{24}$

The LFSRs will work fast because achieving the process in the parallel way means that's each registers can work separately not depended on the other registers.

The maximum period for each registers when the value of k is the number of cells in the register $2^k - 1 = 2^{25} - 1 = 33{,}554{,}431$, Thus, such value will give a wide verity of differences to serve links generators.

The above generators will run 10 time to produce (50 bits) represent 10 alphabetic characters based on the following equation:

$$\mathbf{X} = (\mathbf{1} * \mathbf{b_0}) + (\mathbf{2} * \mathbf{b_1}) + (\mathbf{4} * \mathbf{b_2}) + (\mathbf{8} * \mathbf{b_3}) + \quad (\mathbf{10} * \mathbf{b_4})$$

Such that, the value of X will be in the range of [0...25] representing English alphabets as shown in the figure (6)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Figure 6. English alphabets

Each 5 bits will encode to one character

Results

The proposed system of LFSR combinations at each step produce 5 bits encoded to one character the google meet consist of 10 characters thus the system will run 10 steps get 50 bits to represent the generated link. Figure (7) shows that output results of 4 run to generate 4 links each run produce 50 bits.

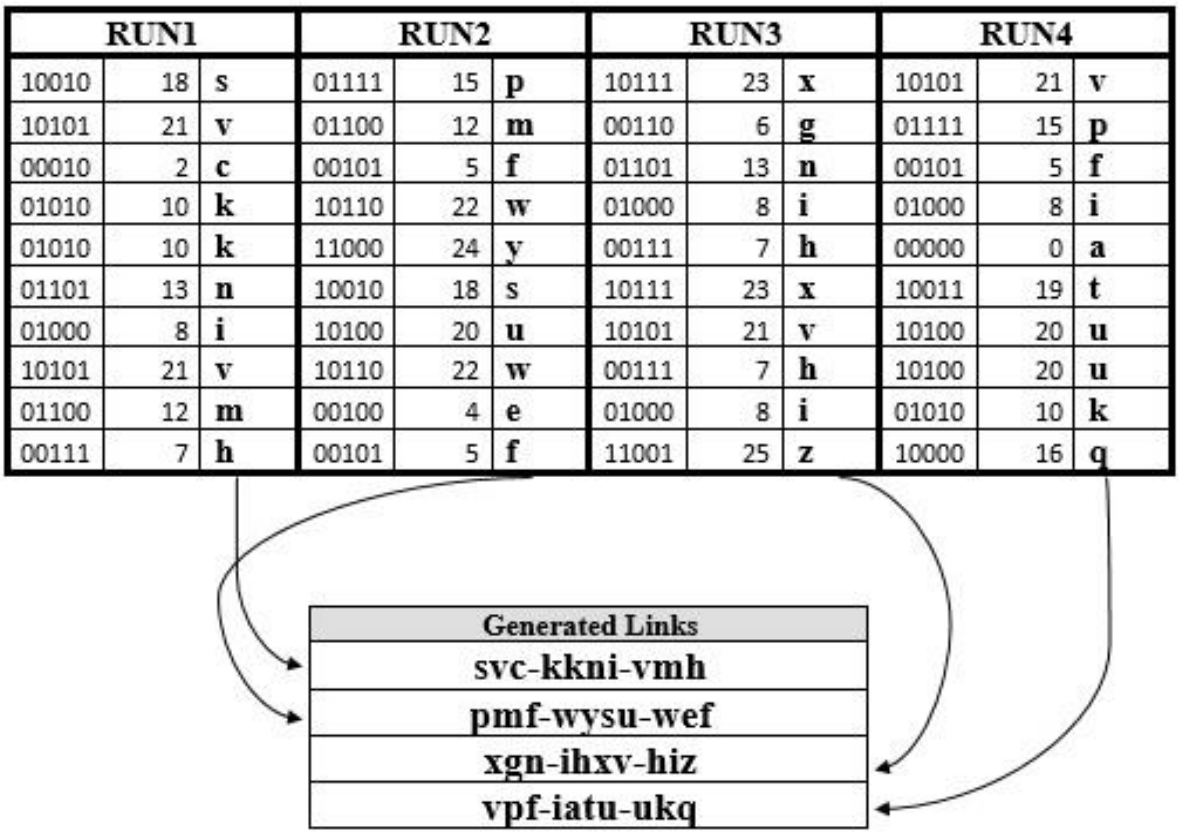

Figure 7. four Runs for the proposed system

The generated links will be unique assigned to the user because the seed key that the LFSR depend on the message digest (MD5) which is one-way function used as hash function used the

E-mail ID (unique) as input to MD5 algorithm. The generated links probability will produce a verity of 10 character that's give $26^{10} = 141{,}167{,}095{,}653{,}376$, such value will be a good range to serve a links for individual users and organizations.

Conclusion

The implemented system can work in real time environment to support generating links for the serving users not only to join meeting but also provide invitation system at proper level of security. The link generator will be used from multiple organization to provide secure meeting as well as fast serving due to the proposed system work on linear registers as beside one-way hash function. More security level added to the linear feedback shift register LFSR by the algorithm that determine the join function in each register. Since the registers length are 25 cell will provide wide verity for producing different links based on the hashed Email ID.

References

[1] Rawat, A. and Agrawal, D., 2015, An Enhanced Message Digest Hash Algorithm for Information Security.

[2] Kundu, R. and Dutta, A., 2020. Cryptographic Hash Functions and Attacks-A Detailed Study. International Journal of Advanced Research in Computer Science, 11(2).

[3] Easttom, W., 2021. Cryptographic Hashes. In Modern Cryptography (pp. 205-224). Springer, Cham.

[4] Al-Awawdeh, M.H.I., 2019. Strengthening the MD5 File Integrity Algorithm with User Fingerprint (Doctoral dissertation, Middle East University).

[5] Kishore, N. and Raina, P., 2019. Parallel cryptographic hashing: Developments in the last 25 years. Cryptologia, 43(6), pp.504-535.

[6] Long, S., 2019, October. A Comparative Analysis of the Application of Hashing Encryption Algorithms for MD5, SHA-1, and SHA-512. In Journal of Physics: Conference Series (Vol. 1314, No. 1, p. 012210). IOP Publishing.

[7] Gillela, M., Prenosil, V. and Ginjala, V.R., 2019, January. Parallelization of brute-force attack on MD5 hash algorithm on FPGA. In 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID) (pp. 88-93). IEEE.

[8] Sagar, F.A., 2016. Cryptographic Hashing Functions—MD5. no. September, pp.1-9.

[9] Gupta, P. and Kumar, S., 2014. A comparative analysis of SHA and MD5 algorithm. architecture, 1, p.5.