International Journal of Advanced Smart Convergence Vol.13 No.2 35-43 (2024)
http://dx.doi.org/10.7236/IJASC.2024.13.2.35

IJASC 24-2-5

An Extension of Firmware-based LFSR One-Time Password Generators

HoonJae Lee[†*] and ByungGook Lee[††]

[†]*Professor, Dept. Information Security, Dongseo University, Korea*
E-mail hjlee@dongseo.ac.kr
[††]*Professor, Dept. Computer Engineering, Dongseo University, Korea*

Abstract

In this paper, we propose two 127-bit LFSR (Linear Feedback Shift Register)-based OTP (One-Time Password) generators. One is a 9-digit decimal OTP generator with thirty taps, while the other is a 12-digit OTP generator with forty taps. The 9-digit OTP generator includes only the positions of Fibonacci numbers to enhance randomness, whereas the 12-digit OTP generator includes the positions of prime numbers and odd numbers. Both proposed OTP generators are implemented on an Arduino module, and randomness evaluations indicate that the generators perform well across six criteria and are straightforward to implement with Arduino.

Keywords: *Authentication, OTP, One-Time Password, LFSR, Firmware, Information Security*

1. Introduction

Recent advancements in Artificial Intelligence(AI) technology have strengthened security authentication capabilities. Additionally, with the development of mobile banking, there is a growing demand for enhanced security technologies. Among the methods to reduce security incidents, user authentication methods are the most commonly used. Authentication is a technology required to guarantee the identity of a specific user when access is requested. Currently, the most widely used authentication relies on Identity/Password. These authentication methods include simple password authentication, media authentication where the user possesses the medium, and biometric information authentication such as fingerprints or iris recognition.

The user authentication methods commonly used in internet banking include methods using security cards and public certificates, as well as the OTP method [1, 2]. The OTP method, introduced recently to enhance security, involves generating a new password each time a user requests authentication. The newly generated password has a different value from previously generated ones, and once used, it is not reused. This reduces the possibility of security incidents even if a user loses their password or if it is eavesdropped by an attacker. Additionally, the OTP method offers advantages such as anonymity, portability, and scalability. Since personal information is not used in security based on OTP, it also prevents the leakage of personal information.

Manuscript Received: April. 10, 2024 / Revised: April. 16, 2024 / Accepted: April. 21, 2024
*Corresponding Author: hjlee@dongseo.ac.kr
Tel: +82-51-320-1730, Fax: +82-51-327-8955
Professor, Department of Information Security, Dongseo University, Busan, Korea)

This paper proposes an OTP generator based on LFSR, one of the most widely used methods in stream cipher algorithms, for the use of OTP authentication. The proposed generator consists of two variants, each incorporating a 127-bit LFSR to output 9-digit and 12-digit decimal values, respectively, considering randomness and usability of OTP. The proposed LFSR-based OTP generators are implemented using Arduino. The Arduino module, also known as a firmware cryptographic module, boasts high versatility and can be easily implemented at a low cost, allowing it to adequately fulfill the role of a cryptographic module. The structure of this paper follows an introduction in Section 1, followed by the OTP generator in Section 2, the proposed OTP generator in Section 3, performance verification and implementation using Arduino in Section 4, and concluding remarks in Section 5.

2. Usage of OTP generator

In OTP authentication methods, a new password, known as the OTP value, is generated each time a user requests authentication. The generation method can be classified into asynchronous and synchronous modes depending on the synchronization between the OTP device and the authentication server [1, 2].

2.1 Asynchronous mode

Asynchronous mode involves the user directly inputting the value received from the authentication server into the OTP device to obtain the OTP value, which is then transmitted to the server for authentication. A prominent example is the Challenge-Response method, currently used in electronic financial transactions. Figure 1 illustrates the asynchronous OTP authentication method using the Challenge-Response approach.

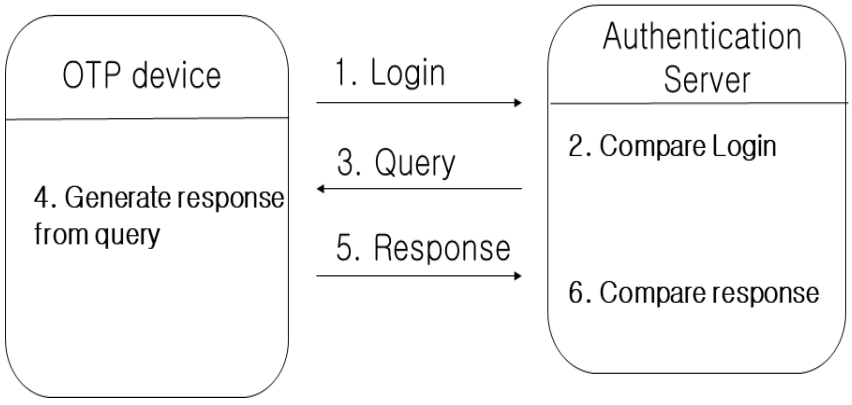

Figure 1. Asynchronous OTP.

In this method, when the user requests authentication from the authentication server, the server provides a random value, referred to as the challenge value. The user directly inputs this challenge into the OTP device, which generates the corresponding OTP value, known as the response value. The user then sends this response to the server for authentication. This method offers the advantage of not requiring synchronization information between the OTP device and the authentication server. However, it has drawbacks such as the inconvenience of users having to input the challenge value directly into the OTP device and transmit the response value to the server, as well as the burden on the authentication server to manage the user's challenge values separately.

2.2 Synchronous mode

Synchronous mode involves generating the OTP value, or password, using pre-agreed synchronization information between the authentication server and the OTP device when manufacturing or initializing the OTP device. Figure 2 illustrates the synchronous OTP method. This method alleviates the inconvenience of users having to directly input and transmit the challenge and response values, as seen in the asynchronous challenge-response method. However, a drawback of the synchronous mode is that accurate user authentication requires synchronization between the OTP device and the authentication server. Synchronous mode can be classified into time synchronous, event synchronous, and time-event synchronous based on the synchronization information used [3-5].

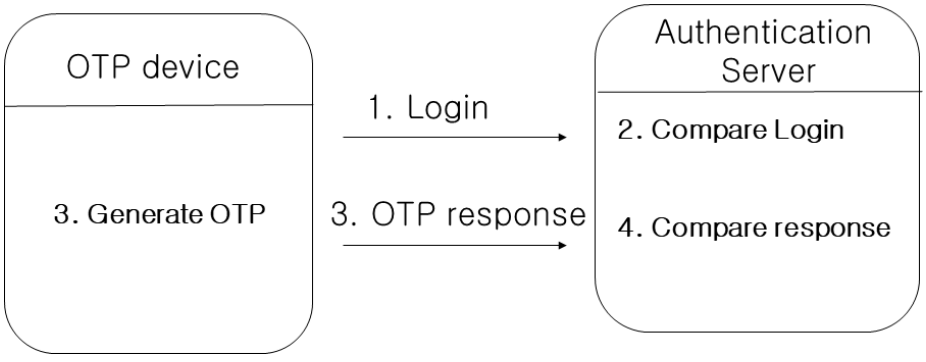

Figure 2. Synchronous OTP.

2.2.1 Time synchronous mode

In time synchronous mode, the OTP device generates a new password, known as the OTP value, at predefined intervals, and the user transmits this OTP value to the server for authentication within a certain timeframe. This method offers the advantage of relative ease of use as users do not need to input separate challenge values into the OTP device. However, there is a requirement for synchronization between the OTP device and the authentication server regarding time, and if the user fails to transmit the OTP value to the server within the specified time, they must wait until the next OTP value is generated. Additionally, a drawback is the risk that if an attacker obtains the OTP value through an attack between the user and the authentication server, they may use the obtained OTP value for a certain period, until the next OTP value is generated.

2.2.2 Event synchronous mode

In event synchronous mode, the OTP value is generated using the same counter between the OTP device and the authentication server. Initially, the user generates the OTP value using the counter as input and transmits it to the server for authentication. After the OTP value is generated, the counter is incremented and stored. The stored counter is used as the input when generating the next OTP value. By keeping the counter private between the OTP device and the authentication server, it becomes difficult for attackers to guess the value, which is an advantage. However, there may be authentication failures between the OTP device and the authentication server during multiple authentication requests, and in cases where the counter values between the OTP device and the authentication server differ due to communication errors, the counter values need to be reset to the same value between the OTP device and the authentication server, which is a disadvantage.

2.2.3 Time-event synchronous mode

Time-event synchronous mode combines the advantages of both time synchronous and event synchronous modes, synchronizing the time between the OTP device and the authentication server and using the same counter to generate OTP values.

3. Proposed LFSR-based OTP generators

In stream cipher algorithms, key sequences are typically outputted at the bit or word level. These key sequences are then used to encrypt information, such as images, text, or communications, through XOR operations with related bit streams. Stream cipher algorithms generate repetitive key sequences with arbitrary periods and are widely used in synchronous authentication. This paper proposes OTP generators based on LFSR, a representative stream cipher algorithm, to output 9-digit and 12-digit decimal OTP values.

3.1 9-digit OTP generator

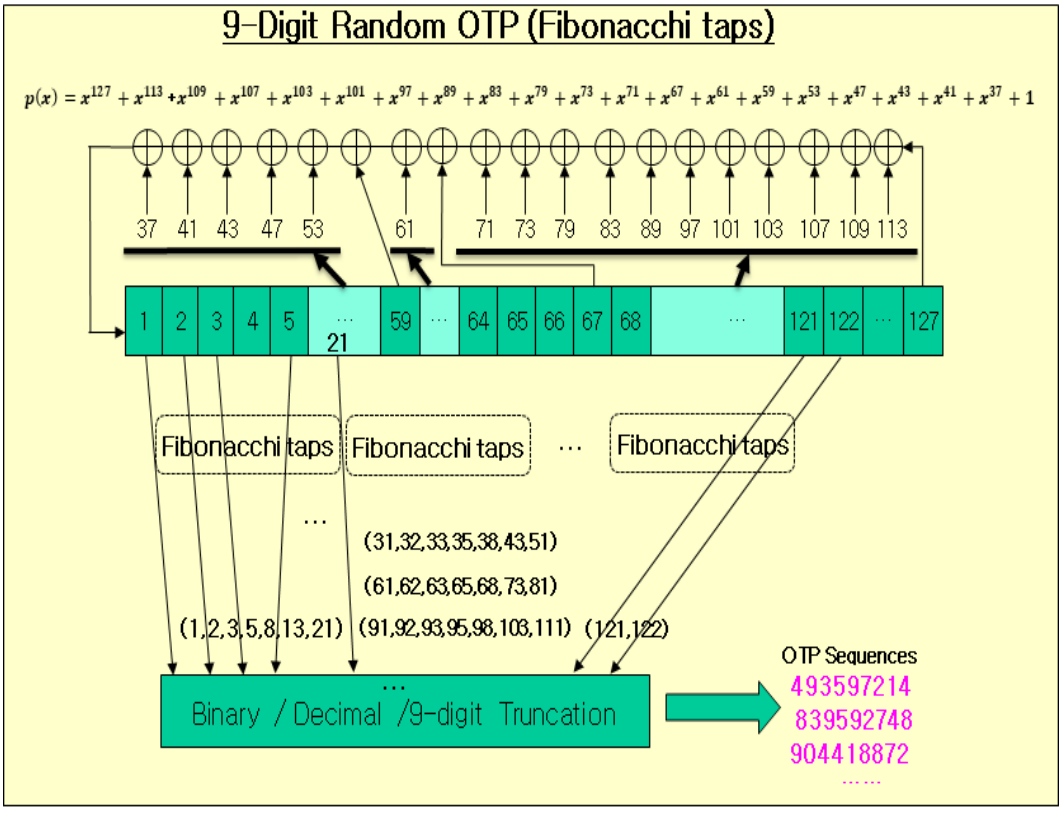

Figure 3. LFSR-based 9-digit OTP generator.

Figure 3 shows a 9-digit OTP generator based on the 127-bit LFSR proposed in this paper. The 30 numbers located below the 127-bit LFSR in the figure represent the positions of the taps in the LFSR. The OTP output value is obtained by converting the random 30-bit value (in binary) indicated by these 30 taps into decimal. The positions of the taps are constructed iteratively by adding 30/60/90/120 to the base Fibonacci sequence of 7 elements (1, 2, 3, 5, 8, 13, 21) to adjust the randomness of the output sequence. In other words, the selected

30 taps have positions at 1, 2, 3, 5, 8, 13, 21, 31, 32, 33, 35, 38, 43, 51, 61, 62, 63, 65, 68, 73, 81, 91, 92, 93, 95, 98, 103, 111, 121, 122.

3.2 12-digit OTP generator

We propose new 12-digit OTP generator based on the 127-bit LFSR. As seen in Figure 4, the proposed 12-digit OTP generator selects the positions of its taps by combining prime and odd numbers to enhance randomness. The selected random tap positions consist of 20 prime values (2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71) and 20 odd values (73, 77, 79, 81, 83, 87, 89, 91, 93, 97, 99, 101, 103, 107, 109, 111, 113, 117, 119, 121). The reason for selecting tap positions in this manner, using both prime and odd numbers, is to increase the randomness of the output sequence, making it difficult to predict the next OTP value. The LFSR-based OTP generators proposed in Figures 3 and 4 serve almost identical purposes as OTP generators used in banking today and are easily implementable on Arduino boards.

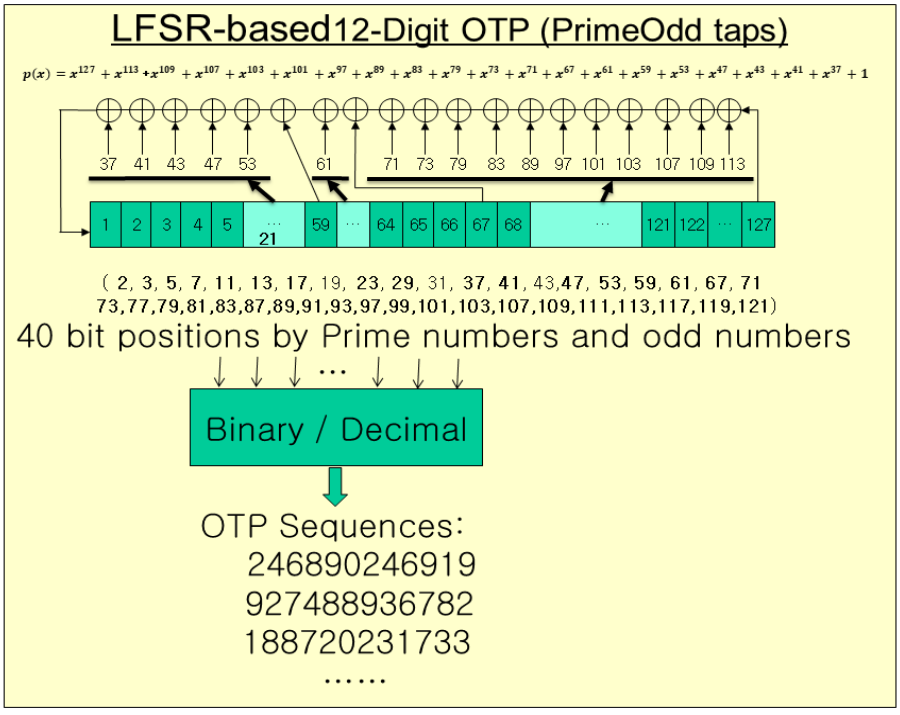

Figure 4. LFSR-based 12-digit OTP generator.

4. Validation and implementation

4.1 Randomness verification

To verify the randomness of the LFSR-based OTP generator proposed in this paper, random tests such as frequency test, serial test, generalized serial test, poker test, and autocorrelation test [6] were applied to the sequence of 160,000 consecutive output bits.

Table 1 presents the experimental results for each verification item. Two random samples (sample-1, sample-2) of 160,000 bits each were extracted from the output binary sequence of the LFSR, and randomness was verified according to the Menezes random verification criteria [6]. The experimental results for the two

samples in the third and fourth columns indicate good randomness (pass) when they fall within the criteria specified in the second column. As shown in Table 1, the results of all verification items are satisfactory.

Table 1. Test results of randomness.

Test items	Criterion	Sample-1	Sample-2
Frequency test	3.841	0.482	0.245
Serial test	5.991	2.634	0.258
Generalized serial test t=3 t=4 t=5	 9.488 15.507 26.296	 2.949 6.278 18.137	 5.848 13.263 16.311
Poker test m=3 m=4 m=5	 14.067 24.996 44.654	 3.800 7.937 42.692	 9.554 14.885 28.596
Autocorrelation test	max≤0.05	0.007	0.007

4.2 Implementation using Arduino

To verify the proposed module, we implemented a simple Arduino-based cryptographic module, which can be easily implemented. The Arduino module, with its high versatility and low cost, is capable of sufficiently performing the role of a cryptographic module.

For the experiment, the Arduino module was connected to a Windows PC USB port, and the environment was set up so that the one-time password was displayed on a serial monitor. Figure 5 illustrates the implementation of new OTP generators using Arduino.

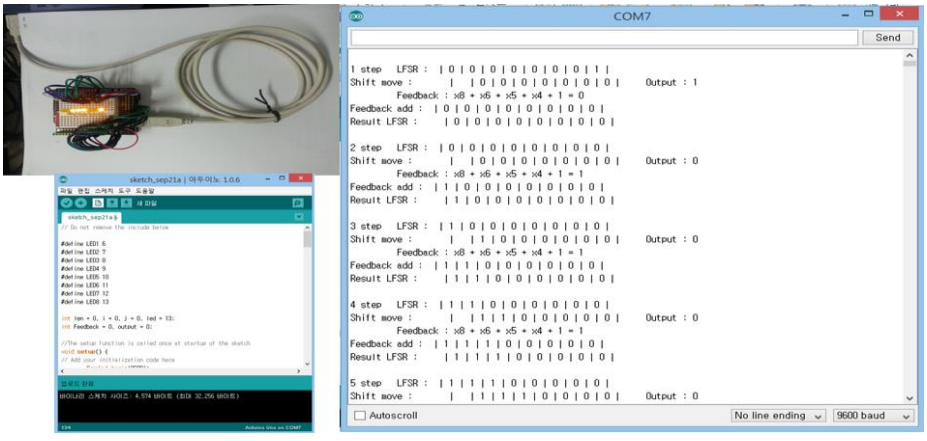

Figure 5. OTP generator implemented using Arduino

Figure 6 shows a partial code snippet from the OTP generator implemented on Arduino cryptographic module. In the figure, the 127-stage LFSR can generate basic coding with good randomness by performing 20 linear feedback XOR operations.

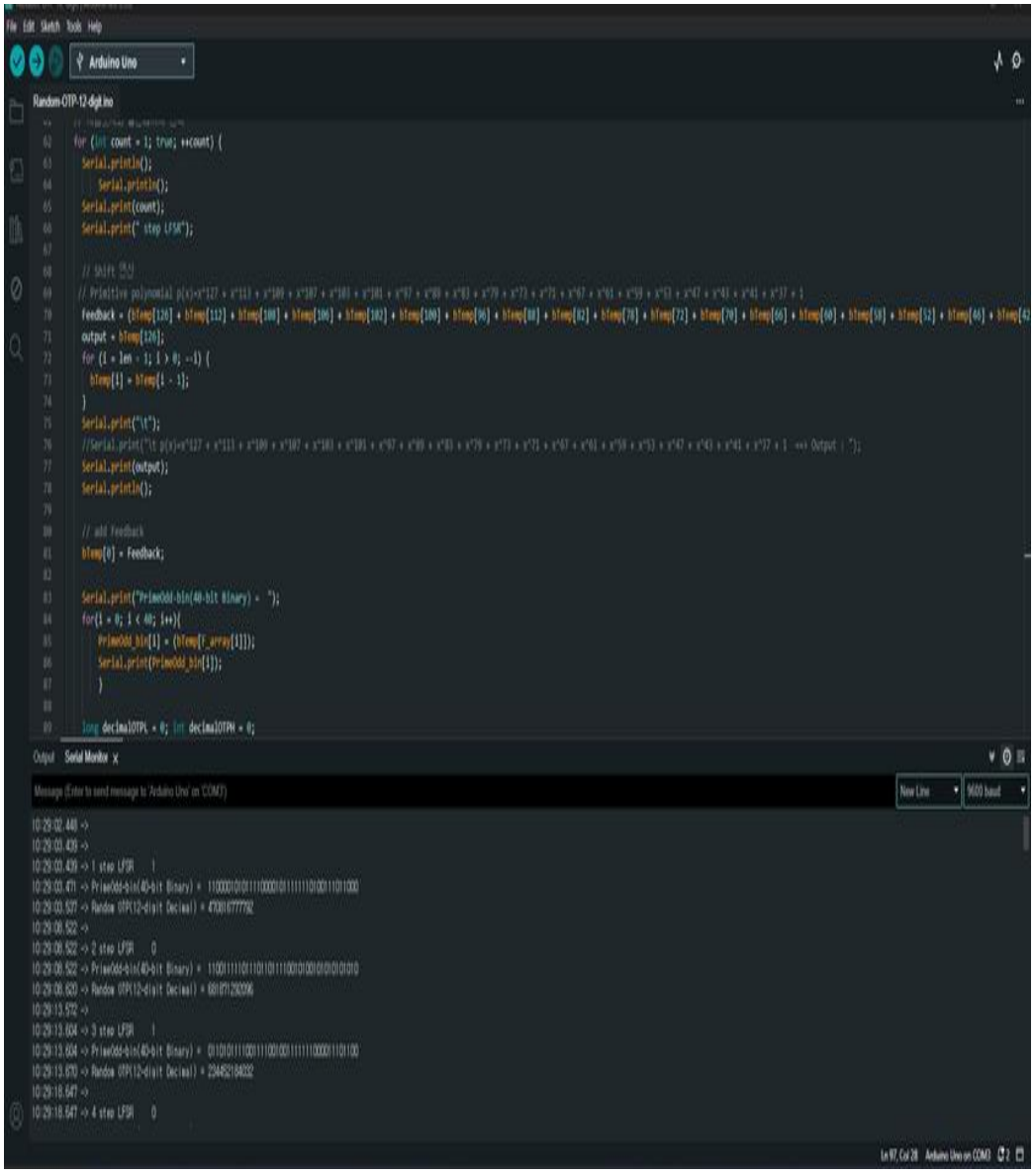

Figure 6. Some codes of LFSR-based OTP generator using Arduino.

Figure 7 displays some examples for 12-digit OTP values generated by Arduino crypto-module. The proposed method is expected to be extendable by increasing the number of bits and taps of the LFSR when larger OTP values are required in the future [9-10].

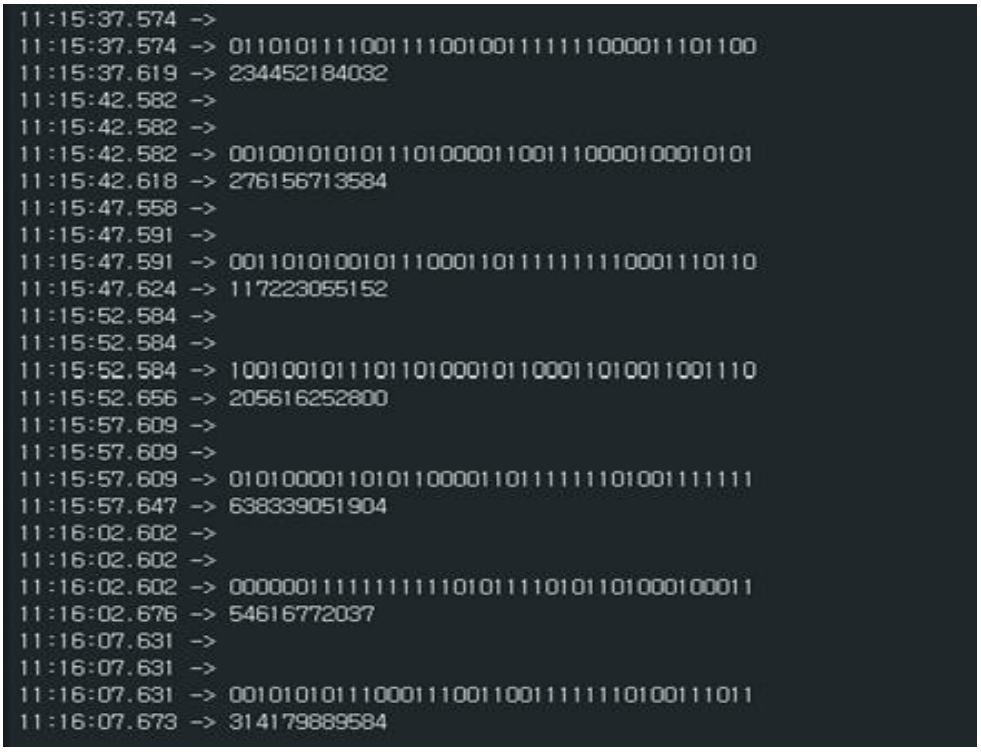

Figure 7. Results of OTP generation using Arduino.

5. Conclusion

In this paper, we proposed various OTP generators applicable not only to internet banking services but also to mobile applications and various user authentications. The proposed OTP generators were experimented with based on Arduino cryptographic modules, implementing 9-digit and 12-digit OTP generators using a 127-bit LFSR. The Arduino module, with its high versatility and low cost, can adequately perform the role of a cryptographic module. The proposed 9-digit generator increased randomness by using Fibonacci numbers for the tap positions of the LFSR, while the 12-digit generator enhanced difficulty in prediction by selecting prime numbers from 1 to 71 and odd numbers from 73 to 121. Moreover, the randomness of the generating LFSR was validated, showing satisfactory results in six randomness verification criteria. The proposed method is expected to be extendable by increasing the number of bits and taps of the LFSR when larger OTP values are required in the future.

Acknowledgement

This work was supported by Dongseo University, "Dongseo Cluster Project" Research Fund of 2023 (DSU-20230004).

References

[1] Baek, Mi Yeon, "Security Enhancement Measures for Electronic Financial Transactions and Current Status of OTP (One Time Password) Usage," *Payment and Information Technology*, pp. 71-100, Apr. 2006.

[2] T. Tsuji, T. Kamioka, and A. Shirnizu, “Simple and secure password authentication protocol,” ver. 2(SAS-2), *IEICE Tech. Rep., OIS 2003-30*, vol. 102, no. 314, Sep. 2002.

[3] Kim, Woo Bin, and Kim, Ki Cheon, "Encryption Key Transformation Using OTP Algorithm," *Proceedings of the Korea Institute*

of Communication Sciences Academic Conference, pp. 155-156, 2015.

[4] M. L. Das, A. Saxena, V. P. Gulati, "A dynamic ID-based remote user authentication scheme", *IEEE Trans. Consumer Electron.*, vol. 50, no.2, pp. 629-631, Feb. 2004. https://doi.org/10.1109/TCE.2004.1309441

[5] H. Y. Chien, C. H. Chen, "A remote authentication scheme preserving user anonymity," *IEEE AINA'05*, vol. 2, pp. 245-248, 2005.

[6] A. Menezes, *Handbook of Applied Cryptography*, CRC Press, 2001. https://doi.org/10.1109/AINA.2005.54

[7] G Goos, J Hartmanis and J van Leeuwen, 2017 lecture on data security, *Modern Cryptology, in Theory a nd Practice*, vol 3 (Springer).

[8] Arisman, Mahyuddin K M Nasution, and Syahril Efendi, "Enhancement of OTP stream cipher algorithm b ased on bit separation," *Journal of Physics: Conference Series*, Vol. 1339 (2019). doi:10.1088/1742-6596/1 339/1/012010

[9] Soonduck Yoo, " Comparative analysis of blockchain trilemma," *International Journal of Advanced Smart Convergence*, Vol. 12, No. 1, Mar. 2023. http://dx.doi.org/10.7236/IJASC.2023.12.1.41

[10] Mohammed Abdulhakim Al-Absi, Ahmed Abdulhakim Al-Absi, Ki-Hwan Kim, Young-Sil Lee,Hoon Jae L ee, " Summary of Maritime Cyber Attacks and Risk Management," *International Journal of Advanced Sm art Convergence*, Vol. 11, No. 3, Sep. 2022. http://doi.org/10.7236/IJASC.2022.11.3.7