Working with cryptographic key information

Nurullaev Mirkhon Mukhammadovich[1], Aloev Rakhmatillo Djuraevich[2]
[1]Department of Information Communication Technology, Bukhara Engineering Technological Institute, Bukhara, Uzbekistan
[2]Department of Computational Mathematics and Information Systems, National University of Uzbekistan named after M. Ulugbek, Tashkent, Uzbekistan

Article Info

Article history:

Received Mar 28, 2022
Revised Jul 21, 2022
Accepted Aug 20, 2022

Keywords:

Biological random number
Cryptographic key generation
Key container
Key information
Random number generator

ABSTRACT

It is important to create a cryptographic system such that the encryption system does not depend on the secret storage of the algorithm that is part of it, but only on the private key that is kept secret. In practice, key management is a separate area of cryptography, which is considered a problematic area. This paper describes the main characteristics of working with cryptographic key information. In that, the formation of keys and working with cryptographic key information are stored on external media. The random-number generator for generating random numbers used for cryptographic key generation is elucidated. To initialize the sensor, a source of external entropy, mechanism "Electronic Roulette" (biological random number), is used. The generated random bits were checked on the basis of National Institute of Standards and Technology (NIST) statistical tests. As a result of the survey, the sequence of random bits was obtained from the tests at a value of $P \geq 0.01$. The value of P is between 0 and 1, and the closer the value of P is to 1, the more random the sequence of bits is generated. This means that random bits that are generated based on the proposed algorithm can be used in cryptography to generate crypto-resistant keys.

Corresponding Author:

Nurullaev Mirkhon Mukhammadovich
Department of Information Communication Technology, Bukhara Engineering Technological Institute
Bukhara, Uzbekistan
Email: nurullayevmirxon@gmail.com

1. INTRODUCTION

Cryptographic protection of information (CPI) is an effective way and a key component of the entire spectrum of issues of information security in information systems (IS) and information protection transmitted through communication channels. To ensure strong data security, keys have to be rotated frequently, and transported and stored securely, along with the high demand for strong data security [1]. In [1], the result of researching the question "How painful is key management?" is "Fifty-six percent of respondents rate key management as very painful, which suggests respondents view managing keys as a very challenging activity. The highest percentage pain threshold of 69% occurs in Spain. At 37%, the lowest pain level occurs in France".

It is important to create a cryptographic system in such a way that the encryption system does not depend on the secret storage of the algorithm that is part of it, but only on the private key that is kept secret. There are many international and national key management standards that differ in the variety of requirements covered. In practice, key management is a separate area of cryptography, which is considered one of the problem areas. Cryptanalysts usually carry out attacks on public-key cryptosystems through key management systems.

It is important to form a pseudo-random sequence of keys in order to protect them from a brute force attack to a certain extent. In addition to the difficulties associated with collecting real random events on computers, there are also a number of problems. The fact that random events do not always exist in computing devices. For example, when collecting incoming entropy from the computer keyboard, a system wait occurs when the user does not enter text. It is also possible to cite sources of random entropy, in particular, the development of physical random sources. For example, in the case of using physical devices on a computer, their failure. In addition, the fact that there is no exact calculation of the entropy coming from the device to the computer, and that computers cannot be used as a generator of real random numbers.

One more main concern in this study is that the functions in question are related to cryptographic keys used in the encryption algorithm “O’z DSt 1105:2009” [2] and in this algorithm 256-bit or 512-bit cryptographic keys were used, as well as functional, session and phase keys were also used to further enhance cryptographic security which make this work different from the previous.

We presented this article in four sections. Section 2 elaborate the method, containing the software random number generator, initialization states, algorithm operation, sensor session keys, and the sensor parcels synchronous. Section 3 presents results and discussion, containing the mechanism of electronic roulette and the testing result. Then, we conclude in section 4.

2. METHOD

The variety of key management systems is described available on the market, helps buyers understand important characteristics to consider when evaluating alternatives, and the document covers: key management concepts, criteria for comparing solutions, advantages of the key management system, building a business case, available solutions, market segmentation [3].

In study [4], design and implementation of a key-lifecycle management system are described. Besides, in [5], a secure and reliable bio-key generation approach is presented, where there are three main issues to be solved: accuracy issue, security issue and, privacy issue. They put forward ideas such as how variations in the biometric image, like pose, blur, and illumination; affect the generated bio-key. Due to the possibility of information leakage from stored helper data or auxiliary information, an attacker can recreate biometric data from the helper data in a database. An attacker can utilize the leaked bio-key to get authentication in other applications once the bio-key has been leaked. Furthermore, to deploy the application system, a new bio-key cannot be generated. In [6] a novel biometric crypto system involving a key binding mechanism is proposed. New objective functions have been introduced to create helper data by binding the secret key with the biometric data of the user. In the retrieval phase, the local minima of the objective functions act as anchors to get the secret key.

To achieve confidentiality, it uses a key hierarchy where every key except a root key itself is encrypted by the respective parent key [7]. The hierarchy also allows for key rotation and, ultimately, for secure deletion of data. The design permits key rotation to proceed concurrently with key-serving operations. In [8]–[10], key management in distributed systems is described. Simion [11] reviewed how mathematical entropy can be estimated and evaluated, of the construction mode (from technologies based on analogue procedures: thermal noise in a transistor to modern procedures: quantum devices), as well as to evaluate the security of binary sequence generators used for generating cryptographic keys or critical security parameters related to new technologies based on quantum principles. In [12], [13], a secret key exchange protocol that establishes a cryptographically secure connection is presented between two objects and presents a new scenario for symmetric key exchange for smart city applications, which protocol is based on the specific properties of Fuss-Catalan numbers and lattice path combinatorics.

Based on the above, the article to a certain extent touched upon issues such as the operation of cryptographic key information. In [13], the functionality of the cryptographic service provider (CSP) is given. CSP is used to generate encryption keys, public and private keys of electronic digital signature (EDS), and the creation of electronic signature authentication, hashing, encryption, and data simulation protection using the algorithms mentioned in [2], [14]–[20]. In [21], the overall structure of the software CSP was described. It is implemented as nine dynamic libraries and four interface programs. The present paper describes the main characteristics and capabilities of the CSP in “Working with the cryptographic key information”. The development is based on standards [22], [23]. CSP “Working with the cryptographic key information” is designed to create encryption keys, private and public keys of EDS. It can be used in telecommunication networks, public information systems, enterprise information systems through integration into application software provides storage, processing and transmission of information and the exchange of information and ensuring the legal value of electronic documents. CSP “Working with the cryptographic key information” performs two functions: formation of keys and work with the cryptographic key information stored on external media. CSP “Working with the cryptographic key information” is working with the cryptographic key information in the key container storage (key container) [14], [15].

2.1. The software random number generator

Second variant, a biometric template is used to generate the authentication key [24]. As a result, there is no need to be worried about the security of the key. Nevertheless, it usually has low discriminability, which can be measured in terms of key entropy and key stability. Key entropy refers to the number of possible keys that can be created, whereas key stability refers to how repeatable the key obtained from biometric data is. For instance, if a scheme produces the same key no matter what the input template is, it has high key stability but no entropy, resulting in a higher false accept rate. If, on the other hand, a scheme creates distinct keys for distinct templates of the same user, it has significant entropy but no stability, resulting in a higher false reject rate. As a result, the approach's limitation is that it is hard to create a key with high stability and entropy. Furthermore, the cancelability property of the key generation seems to be lost. True random number generator was also carried out [25]–[28]. The random number generator for generating random numbers is used for cryptographic key generation.

2.2. Initialization states

To initialize the sensor, we use a source of external entropy, mechanism "electronic roulette" (biological random number). For the implementation of the "electronic roulette", software module *Gui.dll*. The first call software random number generator with the "electronic roulette" removed 32 bytes software initialization vector of random numbers in making random checks the quality through statistical criteria.

2.3. Algorithm operation

The scheme of software random number is as follows.

a) Set the elliptic curve E of the order Q over a large prime field by equation $y^2 = x^3 + a * x + b \; mod \; P$, where a simple module P and the coefficients a and b are structural parameters of the software random number generator.
b) Define two points *A=(XA, YA)* and *B=(XB, YB)*, belonging to a curve are also structural parameters of the software random number generator. The internal state of the software random number generator is the point of an elliptic curve S.
c) Initialized the initial value S_0 of the internal state S by an external source of entropy.
d) Initialized the value S_0 by the rule: $S_0 = X * A$, where X is the external entropy source, presented in the form of an integer on modulo P, and the operation "*" is the multiplication of an integer on the point of the curve.

Each time a software random number generator for the next output value of the internal state is changed according to rule (1),

$$S_0 = (S_{i-1})_x * A, \; i = 1,2,.. \quad (1)$$

where i is indicating the number of treatments after initialization, and S_x is the x-coordinate of the point of an elliptic curve. The output at the i^{th} address to the random number is x coordinate of the point T of the elliptic curve E, calculated from (2).

$$T_i = (S_i)_x * B, \; i = 1,2, \ldots \quad (2)$$

The block diagram of a software random number generator is shown in Figure 1. A sensor session key is built based on the cryptographic algorithm GOST 28147-89, mode-XOR feedback. The sensor is used as the key initializes the sequence length of 32 bytes.

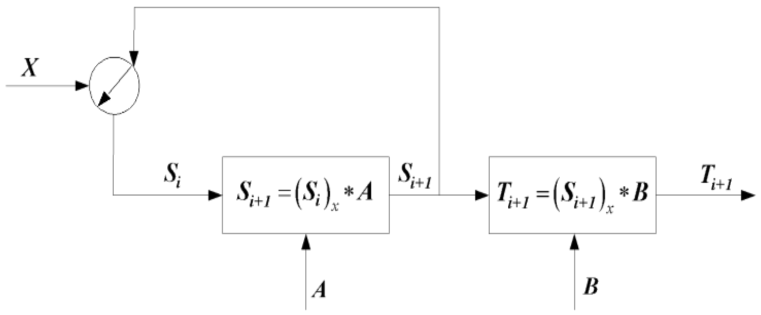


Figure 1. Block diagram of software random number generator

2.4. The sensor parcels synchronous

To develop the sensor parcels synchronous linear shift register *R* is used that contains 47-byte cells *R[0], ..., R[46]*. The initialization procedure is as follows.

a) Record a special clock vector, generated with the help of software random number, in the register cell *R[0], ..., R[31]*.
b) Write random bytes, emanating from the internal random number generator of the current time, in a cell registers *R[32], ..., R[39]*.
c) Write zeros into the cells of register *R[40], ..., R[46]*.
d) Register scrolling 47 beats during idling, i.e., output has not activated for any purpose.

Movement *R* register in the $j^{\text{-th}}$ beat of work occurs according to the following rules.

a) Calculate values:

$T = R[0] << 1$, if MSB of $R[0]$ is 0,
$T = R[0] << 1 \wedge 0xE7$, if the MSB of $R[0]$ is 1,
$T = T \wedge R[40]$;

b) Shift register cell R:

$R[i] = R[i+1], i = 0, ..., 45;$
$R[46] = T;$

c) Produce the register value *G(G[0], ..., G[7])*, with eight bytes long, starting with 48-th beat of work of the register *R*;
d) Compute the values of output bytes *G[0], ..., G[7]* by adding the values for the last eight bytes of cells *R[39], ..., R[46]* of the *R* register with the help of eight-byte mask *M* as follows:

$G[I] = R[39 + i] \wedge M[i], i = 0, ..., 7.$

the mask M is represented by an array of the following eight bytes:

```
const BYTE FixedMask [8]=
{
     0x19, 0x3B, 0x0E, 0xDC, 0xCF, 0x51, 0x6A, 0x33
}
```

Output byte *G[0], ..., G[7]* are used as the next portion of the random number generation at the sensor parcels synchronous.

3. RESULTS AND DISCUSSION

3.1. The mechanism of "electronic roulette"

Biometric cryptosystems were designed to secure a cryptographic key utilizing biometric data or to generate a cryptographic key directly from biometric data. They have, however, been used to protect biometric templates. In this method, certain public helper data are saved in the database. Whereas the helper data does not reveal any important information about the original biometric template, it is required to generate a cryptographic key from the query biometric data during matching [24]. To initialize the sensor, a source of external entropy, mechanism "electronic roulette" (biological random number) is used.

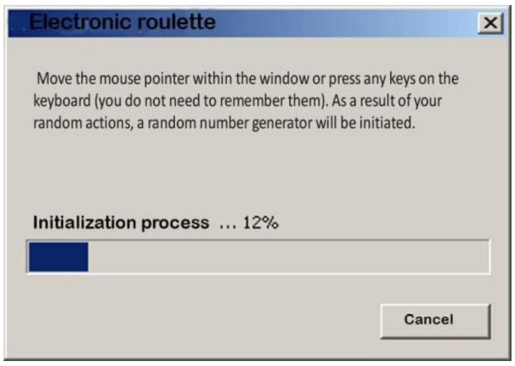
For the implementation of the "electronic roulette" is used software module *Gui.dll*. The first call to a software random number generator with the "electronic roulette" removed 32 bytes initialization vector software random number generator. In this implementation of the CSP, we used BioRandom function from the library *Gui.dll*. The source code for the procedure called "electronic roulette" is given below.

```
typedefint (* BioRandomFn)
(
 OUT LPBYTE lpbDest,
 IN DWORD dwNumberOfRandomBytes,
 IN HWND hParent
);

#define ROULETT_DLL_NAME "Gui.dll"
BOOL RAND_seed(constvoid *buf, int num)
```

```
{
 BioRandomFn MyBioRandom;
 HMODULE hRoulett = LoadLibraryEx (ROULETT_DLL_NAME, NULL, LOAD_WITH_ALTERED_SEARCH_PATH);
     if(hRoulett == NULL)
      return FALSE;
 MyBioRandom = (BioRandomFn)GetProcAddress(hRoulett, "BioRandom");
     if(MyBioRandom == NULL)
      return FALSE;
int retNum = MyBioRandom((LPBYTE)buf, num, NULL);
 FreeLibrary(hRoulett);
 return(retNum == 0);
}
```

When you create a cryptographic key, you are prompted to enter the password for the key CSP. See Figure 2. Then, follow the standard instructions that appear on the screen during installation certification authority. It is known that random numbers and sequences of random numbers are used when implementing cryptographic substitution [29]. These random variables are the base point of cryptographic algorithms, that is, the consistency of cryptographic exchanges will depend on a random number or sequence of random numbers, the degree of their randomness. The difficulty of generalizing random numbers in computers is that computers are deterministic, that is, all processes are executed based on certain rules and the entire state is limited. We can say that the generalization sensors of all random numbers are periodic. All periodic cases are non-random. With the help of computers, it is possible to form pseudo-random rows. The period of such series is required to be such that a series of sufficient length must not be periodic. The non-periodic short-range series of a pseudo-random series must be close to the random series, that is, it must meet certain randomness criteria.


Electronic roulette

Move the mouse pointer within the window or press any keys on the keyboard (you do not need to remember them). As a result of your random actions, a random number generator will be initiated.

Initialization process ... 12%

Figure 2. Electronic roulette develops cryptographic key for the CA

Based on this, this study will consider the functions associated with cryptographic keys used in the "O‘z DSt 1105:2009" encryption algorithm. This algorithm used 256-bit or 512-bit cryptographic keys, as well as functional, session, and phase keys to further improve cryptographic security. The proposed CryptoAPI interface is a combination of basic cryptographic functions, as well as working with certificates (X.509), consisting of such functions as working with cryptographic information (PKCS#7) and working with keys as shown in Table 1.

Table 1. Cryptographic key management functions

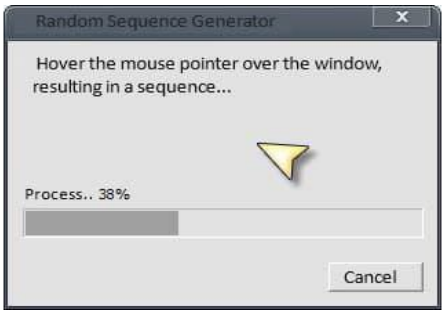
Function	Description
CryptDeriveKey	Create a session key using the specified parameters
CryptDestroyKey	Delete the key descriptor
CryptDuplicateKey	Create a copy of the key
CryptExportKey	Exporting a cryptographic key from a container
CryptGenKey	Generating a random session key and a key pair
CryptGenRandom	Generating random strings
CryptGetKeyParam	Return the key parameters
CryptGetUserKey	Returns a description of a pair of permanent keys
CryptImportKey	Keys are used when importing to the container
CryptSetKeyParam	Sets the key parameters

The key database consists of containers. Each key container will have its name, and the keys that will be used in it for a long time will be saved. Currently, the existing cryptographic encryption algorithms are based on a cryptographic key, that is, the cryptographic security of encryption algorithms is tied to the same keys. One of the methods of generalizing keys for encryption algorithms is to create a cryptographic key from these passwords (keywords). This password will be of variable length and will be shorter in length than the cryptographic key. However, a password is a resource that is not so good for creating a key, because users use a lot of words available in dictionaries that are remembered as passwords, and there is a high probability that this will be easily broken by dictionary attacks.

If a high level of security is required, in such cases the password must consist of random sequences. Such sequences can be obtained using the *CryptGenRandom()* function in the module or using other random number sensors [21]. There are several conditions for creating a key using passwords: i) the password can be specified of any length, while the key will be of a fixed length; ii) one-to-one and “close” passwords should be used to create “one-to-one” and “deleted keys” keys; and iii) the same keys should not be created using different passwords. It is considered very effective to use hash functions to create a key in such conditions. To further increase the cryptographic security of keys, you can add the “Basis” of the vector (salt) and the number of iterations to this mechanism. The “Basis” vector is a random sequence mixed with a password. The use of the “Basis” vector complicates the password dictionary, the recommended length of this vector is currently 8 bytes.

The number of iterations is the number of replacement functions to create a key from a password. If the hash function is used as a replacement function, then the hash function is represented by a sum equal to the number of iterations. To date, the number of iterations should be at least 1,000.

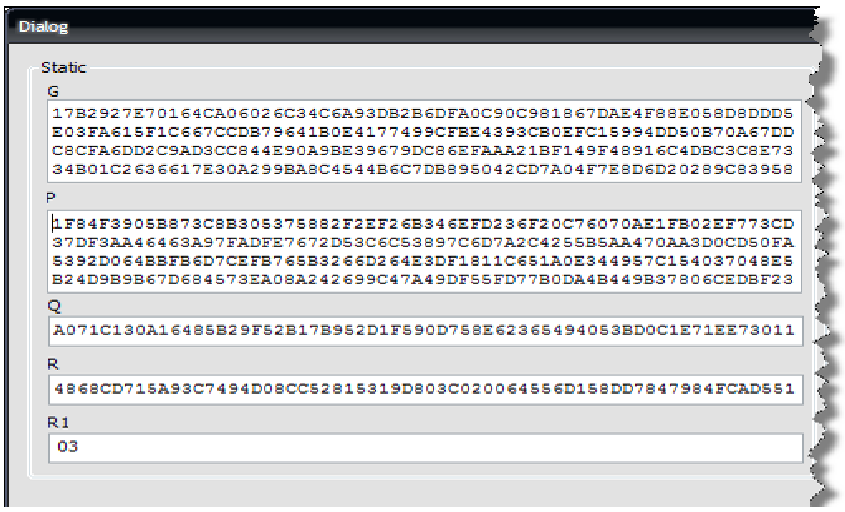
To calculate the EDS based on the standard “O‘z DSt 1092:2009”, it is necessary to initially formulate the user’s public and private keys. To generate keys, we will need the parameters described in the standard “O‘z DSt 1092:2009” *G, P, Q, R, R1[]*. With these parameters, the public and private keys of the EDS are formed using a special software module “random sequence generator”. See Figures 3 and 4.



Hover the mouse pointer over the window, resulting in a sequence...

Process.. 38%

Figure 3. Random sequence generator



G

```
17B2927E70164CA06026C34C6A93DB2B6DFA0C90C981867DAE4F88E058D8DDD5
E03FA615F1C667CCDB79641B0E4177499CFBE4393CB0EFC15994DD50B70A67DD
C8CFA6DD2C9AD3CC844E90A9BE39679DC86EFAAA21BF149F48916C4DBC3C8E73
34B01C2636617E30A299BA8C4544B6C7DB895042CD7A04F7E8D6D20289C83958
```

P

```
1F84F3905B873C8B305375882F2EF26B346EFD236F20C76070AE1FB02EF773CD
37DF3AA46463A97FADFE7672D53C6C53897C6D7A2C4255B5AA470AA3D0CD50FA
5392D064BBFB6D7CEFB765B3266D264E3DF1811C651A0E344957C154037048E5
B24D9B9B67D684573EA08A242699C47A49DF55FD77B0DA4B449B37806CEDBF23
```

Q

```
A071C130A16485B29F52B17B952D1F590D758E62365494053BD0C1E71EE73011
```

R

```
4868CD715A93C7494D08CC52815319D803C020064556D158DD7847984FCAD551
```

R1

```
03
```

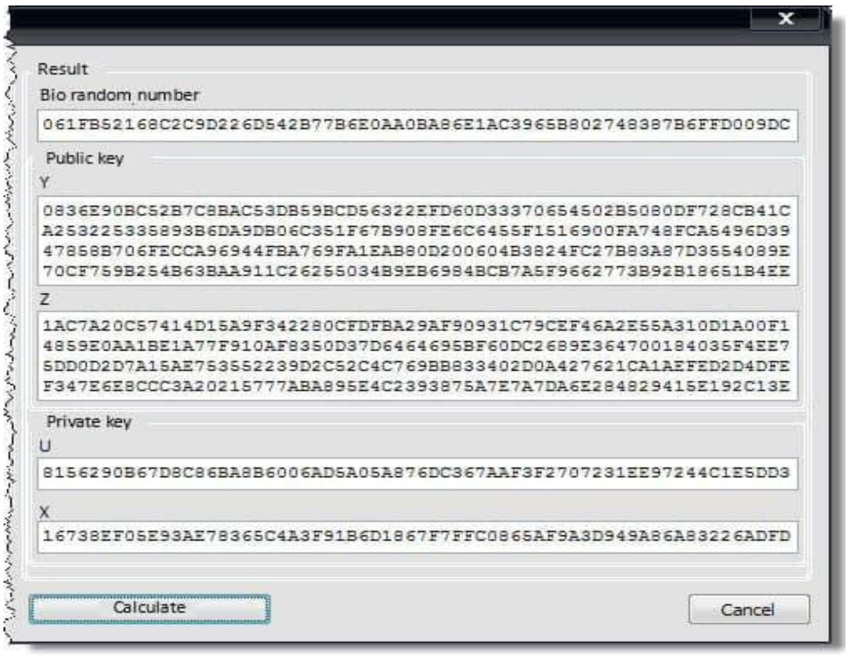
Figure 4. Parameters form

After entering the main parameters in this window, the switch proceeds to the key generation stage. A random sequence is used to obtain an initialization vector, if the user moves the mouse pointer over the window, a random sequence is formed, depending on this action, this generator can be called *BioRandom*, since random values in this are generated by random user movement. This program produces a random sequence of 256 bits as shown in Figure 5.

The random sequence is located in the "biorandom value" field. Public keys are 1024 bits long and are formed based on a random value. The private keys are formed using (3) and (4), respectively.

$$y \equiv g^{\backslash x}(mod\, p) \tag{3}$$

$$z \equiv g^{\backslash u}(mod\, p) \tag{4}$$



```
Bio random number
061FB52168C2C9D226D542B77B6E0AA0BA86E1AC3965B802748387B6FFD009DC
Public key
Y
0836E90BC52B7C8BAC53DB59BCD56322EFD60D33370654502B5080DF728CB41C
A253225335893B6DA9DB06C351F67B908FE6C6455F1516900FA748FCA5496D39
47858B706FECCA96944FBA769FA1EAB80D200604B3824FC27B83A87D3554089E
70CF759B254B63BAA911C26255034B9EB6984BCB7A5F9662773B92B18651B4EE
Z
1AC7A20C57414D15A9F342280CFDFBA29AF90931C79CEF46A2E55A310D1A00F1
4859E0AA1BE1A77F910AF8350D37D6464695BF60DC2689E36470018403 5F4EE7
5DD0D2D7A15AE753552239D2C52C4C769BB833402D0A427621CA1AEFED2D4DFE
F347E6E8CCC3A20215777ABA895E4C2393875A7E7A7DA6E284829415E192C13E
Private key
U
8156290B67D8C86BA8B6006AD5A05A876DC367AAF3F2707231EE97244C1E5DD3
X
16738EF05E93AE78365C4A3F91B6D1867F7FFC0865AF9A3D949A86A83226ADFD
```

Figure 5. Result of parameters

The data encryption algorithm uses two keys. This is an encryption key and a function key, each of which is a 256-bit sequence. The original function key is generalized in one of two ways: i) as an encryption key function and ii) without taking into account the encryption key, that is, in the form of a pseudo-random number.

The data encryption algorithm, which uses two-dimensional arrays of session and phase keys to process data in *CryptoText* and converts it into open text, is formed based on the encryption key and the functional key. To work with these keys, a software module is created that performs cryptographic functions and manages these functions via CryptoAPI. The private keys (encryption keys and private electronic signature keys) available in the container are protected with a security key, which is the PIN code of the user's token. Cryptographic procedures are called in the cryptographic data protection system using the *PKCS#11* interface. The same or different *PKCS#11* slots can be used for *UZ KEYEXCHANGE* and *UZ SIGNATURE* [20]. The module can be used in telecommunications networks, information systems, programs that provide storage, processing, and transmission of information without access to information constituting a state secret, as well as in ensuring the legal significance of the exchange of information and electronic documents. The software module is currently applicable only in the Windows operating system and has not been investigated in other operating systems such as Linux, Mac, and Unix.

3.2. Testing result

During the experiments, 100 sequences of random bits with a length of 12,800 bits were generated. The generated random bits were tested using the statistical tests given in [30]. The P-value of all generated random bits is bigger than 0.01 Table 2.

As a result of the survey, the sequence of random bits was obtained from the tests at a value of P≥0.01. The value of P is between 0 and 1, and the closer the value of P is to 1, the more random the sequence of bits is generated. This means that random bits that are generated based on the proposed algorithm can be used in cryptography to generate strong keys.

Table 2. The test result of random bits

P-Value	Proportion	Statistical Test
0.236810	98/100	Frequency
0.816527	98/100	Block Frequency
0.191637	98/100	Cumulative Sums
0.289658	100/100	Cumulative Sums
0.883117	100/100	Runs
0.383874	100/100	Longest Runs

4. CONCLUSION

In this research work, the main characteristics of working with cryptographic key information are described. In that, the formation of keys and work with the cryptographic key information stored on external media. The random number generator for generating random numbers used for cryptographic key generation is elucidated. To initialize the sensor, a source of external entropy, mechanism "electronic roulette" (biological random number) is used. It is important to form a pseudo-random sequence of keys in order to protect them from a brute force attack to a certain extent. In making random checks the quality through statistical criteria. Generating random numbers is suitable for many specific applications, such as cryptographic information protection systems, gambling, lotteries, and simulation. Accordingly, the future work could refer specifically to the application that can be used to develop random number generators in mobile applications and IoT systems.

REFERENCES

[1] Ponemon Institute, "2021 Global encryption trends study." Ponemon Institute Research Report, *Entrust*, 2021. Accessed: Mar. 26, 2022 [Online]. Available: https://www.entrust.com/-/media/documentation/reports/2021-global-encryption-trends-exec-summary-re.pdf

[2] "Information technology. Cryptographic protection of information. The data encryption algorithm." O'zDSt 1105:2009, 2009. Accessed: Mar. 16, 2022 [Online]. Available: https://lex.uz/files/5011556.zip

[3] Cryptomathic, "Selecting the right key management system." *Cryptomathic*, 2019. https://www.cryptomathic.com/news-events/blog/selecting-the-right-key-management-system (accessed Mar. 19, 2022).

[4] M. Björkqvist *et al.*, "Design and implementation of a key-lifecycle management system," in *Financial Cryptography and Data Security*, 2010, pp. 160–174. doi: 10.1007/978-3-642-14577-3_14.

[5] Y. Wang, B. Li, Y. Zhang, J. Wu, and Q. Ma, "A secure biometric key generation mechanism via deep learning and its application," *Applied Sciences*, vol. 11, no. 18, Sep. 2021, doi: 10.3390/app11188497.

[6] R. Asthana, G. S. Walia, and A. Gupta, "A novel biometric crypto system based on cryptographic key binding with user biometrics," *Multimedia Systems*, vol. 27, no. 5, pp. 877–891, Oct. 2021, doi: 10.1007/s00530-021-00768-8.

[7] M. Bjorkqvist, C. Cachin, F. Engelmann, and A. Sorniotti, "Scalable key management for distributed cloud storage," in *2018 IEEE International Conference on Cloud Engineering (IC2E)*, Apr. 2018, pp. 250–256. doi: 10.1109/IC2E.2018.00051.

[8] T. Acar, M. Belenkiy, C. Ellison, and L. Nguyen, "Key management in distributed systems," *Microsoft Research*, pp. 1–14, 2010.

[9] I. Kuzminykh, B. Ghita, and S. Shiaeles, "Comparative analysis of cryptographic key management systems," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, 2020, pp. 80–94, doi: 10.1007/978-3-030-65729-1_8.

[10] T. S. Algaradi and B. Rama, "An authenticated key management scheme for securing big data environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, pp. 3238–3248, Jun. 2022, doi: 10.11591/ijece.v12i3.pp3238-3248.

[11] E. Simion, "Entropy and randomness: from analogic to quantum world," *IEEE Access*, vol. 8, pp. 74553–74561, 2020, doi: 10.1109/ACCESS.2020.2988658.

[12] M. Saračević, S. Adamović, N. Maček, M. Elhoseny, and S. Sarhan, "Cryptographic keys exchange model for smart city applications," *IET Intelligent Transport Systems*, vol. 14, no. 11, pp. 1456–1464, Nov. 2020, doi: 10.1049/iet-its.2019.0855.

[13] M. Saračević, S. Adamović, N. Maček, A. Selimi, and S. Pepić, "Source and channel models for secret-key agreement based on catalan numbers and the lattice path combinatorial approach," *Journal of Information Science and Engineering*, vol. 37, no. 2, pp. 469–482, 2021, doi: 10.6688/JISE.202103_37(2).0012.

[14] R. D. Aloev and M. M. Nurullaev, "Development of the software cryptographic service provider on the basis of national standards," *Journal of Systemics, Cybernetics and Informatics*, vol. 17, no. 1, pp. 260–272, 2019, doi: 10.6688/JISE.20200X_36(X).00XX.

[15] M. Nurullaev and R. D. Aloev, "Software, algorithms and methods of data encryption based on national standards," *IIUM Engineering Journal*, vol. 21, no. 1, pp. 142–166, Jan. 2020, doi: 10.31436/iiumej.v21i1.1179.

[16] M. M. Nurullaev, "Modeling of information processes in integrated security systems," *Journal Molodoy uchoniy*, vol. 17, no. 203, pp. 26–27, 2018.

[17] "Unified software documentation. The terms of reference, the requirements for content and design. Re-release (November 1987). With Amendment number 1." (in Russian), G. 19.201-78, 1981. Accessed: Mar. 21, 2022 [Online]. Available: https://files.stroyinf.ru/Data/318/31884.pdf

[18] "Cryptographic protection of information. The formation and verify digital signatures," (in Russian), O'z DSt 1092:2009. 2009. Accessed: Mar. 16, 2022 [Online]. Available: https://tace.uz/ru/docs/OzDSt_1092.pdf
[19] "Information technology. Cryptographic protection of information. Hash function," (in Russian), O'z DSt 1106:2009. 2009. Accessed: Mar. 16, 2022 [Online]. Available: https://lex.uz/files/5011555.zip
[20] "Information processing systems. Cryptographic protection. Cryptographic transformation algorithm," (in Russian), GOST R 28147-89. 1989. Accessed: Mar. 21, 2022 [Online]. Available: https://files.stroyinf.ru/Data2/1/4294826/4294826631.pdf
[21] R. D. Aloev and M. M. Nurullaev, "Cryptography service provider – data encryption," in *Proceedings Conference on Complexity, Informatics and Cybernetics, Orlando*, 2019, pp. 127–131.
[22] IETF, "Additional cryptographic algorithms for use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms." RFC 4357, Accessed: Mar. 21, 2022 [Online]. Available: https://datatracker.ietf.org/doc/rfc4357/
[23] B. Kaliski, "PKCS #5: password-based cryptography specification v2.0," RFC 2898, RSA Laboratories, September 2000. Accessed: Mar. 26, 2022 [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2898
[24] T. A. T. Nguyen, T. K. Dang, and D. T. Nguyen, "A new biometric template protection using random orthonormal projection and fuzzy commitment," in *IMCOM 2019: Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (IMCOM) 2019*, 2019, pp. 723–733. doi: 10.1007/978-3-030-19063-7_58.
[25] P. Nannipieri *et al.*, "True random number generator based on fibonacci-galois ring oscillators for FPGA," *Applied Sciences*, vol. 11, no. 8, Apr. 2021, doi: 10.3390/app11083330.
[26] X. Wang *et al.*, "High-throughput portable true random number generator based on jitter-latch structure," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 2, pp. 741–750, Feb. 2021, doi: 10.1109/TCSI.2020.3037173.
[27] J. Lin, Y. Wang, Z. Zhao, C. Hui, and Z. Song, "A new method of true random number generation based on galois ring oscillator with event sampling architecture in FPGA," in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, May 2020, pp. 1–6. doi: 10.1109/I2MTC43012.2020.9129357.
[28] N. Fujieda, "On the feasibility of TERO-based true random number generator on xilinx FPGAs," in *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*, 2020, pp. 103–108. doi: 10.1109/FPL50879.2020.00027.
[29] L. Y. Shcherbakov and A. V Domashen, *Applied cryptography. Use and synthesis of cryptographic interfaces*. Moscow: Russian Edition, 2003.
[30] A. Rukhin *et al.*, "A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic application," *NIST Special Publication 800-22.*, 2010.

BIOGRAPHIES OF AUTHORS

Nurullaev Mirkhon Mukhammadovich is a research trainee in the Department of Information communication technology, Bukhara Engineering Technological Institute, Uzbekistan. He obtained his bachelor's degree at Bukhara State University in 2000, in the field of "The Applied Mathematics and Computer Science". He obtained his master's diploma with honors at Bukhara Technology Institute in 2004, in the field of "Mathematical and software support of computers, complexes, systems and networks". He finished his postgraduate studies at National University of Uzbekistan in 2009, at Mechanics-Mathematical Faculty. He is engaged in new areas of science in the fields of "information security" and "cryptography". He can be contacted at nurullayevmirxon@gmail.com.

Aloev Rakhmatillo Djuraevich is a university professor at the Department of Computational Mathematics and Information Systems, National University of Uzbekistan named after M. Ulugbek, Uzbekistan. He is a (candidate) Ph.D. of Physical Mathematical Sciences obtained from Council of the Institute of Mathematics, Siberian Branch of the USSR Academy of Sciences in 1984. He obtained his Doctor of Sciences Physics and Mathematics from The Supreme Attestation Committee of the Russian Federation in 1994, in the field of Computing Mathematics. The fields of research are computational mathematics, applied mathematics and numerical analysis, difference schemes, stability, differential equation, hyperbolic systems, existence and uniqueness of the solutions, information security, cryptology, and cryptographic service provider. He can be contacted at aloevr@mail.ru.