DOI: 10.24996/ijs.2023.64.2.38

ISSN: 0067-2904

A Secure Private Key Recovery Based on DNA Bio-Cryptography for Blockchain

Asia Ali Salman Al-karkhi[1], Nidaa Flaih Hassan[2]*, Raghad Abdulaali Azeez[3]

[1]*Computer Science Department, University of Technology, Iraq*
[2]*Computer Science Department, University of Technology, Iraq*
[3]*Computer Unit, Collage of Education for Human Science-Ibn Rushed, University of Baghdad, Iraq*

Received: 21/2/2022 Accepted: 28/6/2022 Published: 28/2/2023

Abstract

The existence of the Internet, networking, and cloud computing support a wide range of new technologies. Blockchain is one of these technologies; this increases the interest of researchers who are concerned with providing a safe environment for the circulation of important information via the Internet. Maintaining solidity and integrity of a blockchain's transactions is an important issue, which must always be borne in mind. Transactions in blockchain are based on use of public and private keys asymmetric cryptography. This work proposes usage of users' DNA as a supporting technology for storing and recovering their keys in case those keys are lost — as an effective bio-cryptographic recovery method. The RSA private key is responsible for maintaining the authenticity of the blocks' wallets throughout any transaction related to any block of the blockchain. This framework can be used for a wide range of applications such as student registration systems at universities: in order to prevent the forging of student graduation certificates. The experimental results demonstrated robustness of the proposed solution, using a number of key sizes. The effectiveness of our approach is compared to that of elliptic curve cryptography keys. Our approach shows that the security and authentication needed for blockchain technology can be accomplished using DNA combined with an RSA private key. On the other hand, the standard EC cryptography shows poor performance against our suggested method as demonstrated in the discussion section.

Keywords: blockchain, network security, blockchain key management, RSA encryption

تأمين استرداد المفتاح الخاص استناداً إلى تشفير الحمض النووي الحيوي لسلسة الكتلة

اسيا علي سلمان [1] , نداء فليح حس[2]* , رغد عبد العالي عزيز[3]

[1]قسم علوم الحاسوب, الجامعة التكنولوجية, بغداد, العراق
[2]قسم علوم الحاسوب, الجامعة التكنولوجية, بغداد, العراق
[3]وحدة الحاسبة, كلية التربية ابن رشد للعلوم الانسانية, جامعة بغداد, العراق

الخلاصة

وجود شبكة الإنترنت والشبكات والحوسبة السحابية تشكل دعم مجموعة واسعة من التكنولوجيات الجديدة Blockchain .هي واحدة من هذه التقنيات. وقد زاد ذلك من اهتمام الباحثين الذين يهتمون بتوفير بيئة

*Email: 110020@uotechnology.edu.iq

آمنة لتعميم المعلومات الهامة عبر الإنترنت.

ان الحفاظ على صلابة وسلامة معاملات blockchain هي قضية مهمة ، والتي يجب أن تؤخذ دائما في الاعتبار. تستند المعاملات في blockchain على استعمال التشفير غير المتماثل للمفاتيح العامة والخاصة. ان هذا البحث يقترح العمل استعمال الحمض النووي للمستخدم كتكنولوجيا داعمة لتخزين مفتاحه واستعادته في حالة فقدانه لهم – كوسيلة فعالة لاستعادة التشفير البيولوجي. المفتاح الخاص RSA هو المسؤول عن الحفاظ على صحة محافظ الكتل في جميع أنحاء أي معاملة تتعلق بأي كتلة من .blockchain ويمكن استعمال هذا الإطار في مجموعة واسعة من الطلبات مثل نظم تسجيل الطلبة في الجامعات: من أجل منع تزوير شهادات تخرج الطلبة. أظهرت النتائج التجريبية قوة الحل المقترح، باستعمال عدد من الأحجام الرئيسية. تتم مقارنة فعالية نهجنا بمفاتيح التشفير بالمنحنى البيضاوي كما هو موضح في قسم المناقشة.

1. Introduction

Blockchain is recently created a highly effective mechanism for instituting secure computing without allowing centralized privileges in an open network system; it has achieved rapid expansion and success. From a data management point of view, a blockchain is a network architecture supporting a distributed database[1][2][3].

In addition, blockchain has demonstrated its great potential for enhancing security and performance in IoT devices. Its popularity stems from its capacity to self-administer through distributed and consensus-driven behavior, as well as clarity, immutability, and strong cryptographic security[4].

Blockchain operates by 'scoring' a developing list of transaction records by arranging them into a structural chain of blocks. From a security point of view, each blockchain is originated and maintained using a peer-to-peer (P2P) overlay network and is secured through the smart and decentralized organization of cryptography via the application of gathered computing [5]. The solution provided by blockchain is a decentralized one; also, the costs connected with maintaining trust are reduced as the individuals are provided with more control of their assets. However, managing the individuals' keys is the responsibility of the network participants themselves [6][7], and this causes issues.

The use of bio-cryptography keys implies the use of biological characteristics to protect the data transferred through any communication channel provided by blockchain technology [8][9]. Such novel cryptographic characteristics, applied to existing blockchains, are likely lead to new challenges. These challenges become greater in relation to hierarchical network designs; and certain types of underlying asset mechanisms and cryptographic methods. In order to tackle these difficulties and find optimal solutions, many of the cryptographic foundations of blockchains, such as commitment protocols, signature schemes, and zero-knowledge proofs must be analyzed, as in [10]. To use a cryptocurrency wallet, for example, users must trust the program which is supposed to work properly, free of bugs, and/or malicious code that could steal their funds. To be secured in using crypto exchanges, people must also have faith in their proper functioning [11].

Blockchain technology has been deployed in IoT systems and for access control, privacy, and security across many diverse fields, such as healthcare, supply chain, and VANET enhancement [12][11][13]. Blockchain offers incredible potential for collaborating with IoT to improve confidentiality, clarity, and security. It has unique characteristics including distributed behavior, immutability and a consensus process[14] . Biometrics have been developed to play a role in user identification and access control[15]. In relation to this, three different components exist within biometric systems: a) sensors for acquiring the data; b) computerized devices for storing biometric data; and c) software for binding computer (hardware) devices to the sensor.

Currently, there are two major problems related to management of keys: first is that when keys are lost, there is no effective retrieval technique; second is that there is no efficient and secure way to store the private keys of the users. This paper proposes an efficient technique that makes the blockchain wallet more secure and reliable; a new technique for keeping the private key saved. Other methods also have been studied such as the elliptic curve technique, which provides smaller key sizes than the RSA method as used in this work. However, the results from our approach, using RSA, demonstrate more efficient and reliable keys. Hence, to protect the blockchain wallet and its transactions, a user-friendly, secure store, and recovery method, which works by combining the private key of the wallet's owner with the owner's DNA bio-cryptography. Thus, a self-controlling technique represented by the characteristics of chromosomes has been employed to create a means of storage for private keys. The technique applies DNA biometrics and RSA encryption to create a recovery system for private keys, and so facilitates securing and protecting of the blockchain wallet and its transactions. In addition, this technique represents a robust approach when faced with non-secure communication channels in the peer-to-peer network.

2. Literature Review

Several related works that have proposed schemes for various blockchain-based applications such as smart home, smart grid, and industrial IoT are explained here.

Aydar et al. in [6] described a mechanism for key encryption and recovery whereby asset possessors are able to securely store their keys on their devices and recover the keys when they are lost. The researchers used a fingerprint-based method as a suggested technique for storing the private keys. This involves two systems: in the first, encryption and decryption were performed via private keys in an efficient manner using the possessor's biometric signature; in the second, an interactive recovery technique is employed which applies biometrics and a secrecy sharing method. Although fingerprints are relatively consistent over the course of a person's life, certain people may be disqualified from using a system based on these. Older people with a history of manual labor, for example, could find it difficult to register worn prints with the system, and people who have lost fingers or hands may be disqualified.

Bi et al, in[16], proposed a secure and efficient system based on blockchains operating using two elliptic curves which take into account the overlap between efficiency and security; they suggested an algorithm which applied multiple elliptic curves for digital signatures; the parameters of each curve could be edited as the number of elliptic curves employed in order to provide a system supporting the required secrecy which was at the same time practicable. One of the most noticeable drawbacks of ECC (Elliptical Curve Cryptography) is that it greatly increases the size of the encrypted message. Furthermore, because the ECC algorithm is complicated and difficult to implement, the probability of implementation errors is high, so reducing the level of protection which may be expected.

Hamer et al. in [7], described a protocol they had developed whereby facilities were provided for large institutions to offer basic services to people who had lost their identifying documents; this protocol could identify people and maintain appropriate control of their access. The cancelable biometrics is managed by the World Wide Web Consortium (W3C)

Verifiable Claims system which is incorporated within the protocol in order to maintain privacy and prevent any double enrolment with any system. This is achieved by the use of a transfer obscured biometric system within blockchain. This is achieved by giving each person at least one identity in each available identity domain; these identities cannot be linked between domains without a command to do so being issued by the individual themselves. However, this protocol suffers from a reliability issue; in biometry administration systems that use vein identification and fingerprints, the process of enrollment may be penetrative and this can lead to the modification of, or the tampering with, biometric information.

Murakami et al. in [17], implemented a new secure and practical signature method as a blockchain IoT system. Biometric information is used to create private keys, and on the basis of these, the security and practicality of their method are assessed. Fuzzy signature technology is employed for generating blockchain transactions. Via this, the creators of transactions can be proved to have correct biometric information, and so the transaction is verified. The biometric information of an individual is unique, and so it can be used to verify that the creator of a transaction is a proper user. In addition to this, the proposed signature method generates a short-term private key to be used for generating transactions. IoT systems can automatically create new transactions using this method. The application of fuzzy signature technology to create blockchain transactions accomplished strict confirmation of blockchain transaction creators; however, this process must be performed by the user or kept on a remote server. Also, a signing system must be available when generating a signature.

Yakubov et al. [2] implemented a blockchain technique for a new PGP key server management structure which resolves some of the problems of PGP key servers condensing (especially on the fast spread of certificate revocation among key servers) and removes the associated risks. These researchers also permitted user access rights control, whereby only the certificate holder can modify information related to the certificate. They implemented a prototype of the key server, designed on a permissioned Ethereum blockchain system. As well known, the Ethereum blockchain structure is still undergoing frequent changes; and the proposed methods had to be designed so that they were compatible with such changes. In addition, compatibility issues relating to PGP were not resolved.

Ajao et al. in [18] presented a novel decentralized secrecy ledger for the implementation of a database containing details of petroleum product distribution; this secrecy ledger manages records using a secure hash algorithm 1(SHA-1) based blockchain; the computation mechanism includes the hashing of every transaction created, depending on the previous transaction and the effective confirmation of the current one. The method is not vulnerable to user manipulation of the record, although the system will allow a way to modify when the user obtains 75% agreement over the chain (otherwise, permission is not granted). All the information relating to a particular transaction is kept on the distributed ledger, thus any user in the chain may readily acquire information or supply such, safe in the knowledge that this will remain secret and invulnerable against manipulation. As stated, Ajao L. A, et al, use the SHA-1 hash algorithm-based blockchain, encrypting a distributed ledger database, while our proposed method uses RSA encryption, which is more efficient and reliable by combining the private key of the wallet's owner with the owner's DNA-based bio-cryptography in order to construct a strongly secured blockchain.

Robles et al. in [19] proposed a method using smart contracts with blockchain technology. A cryptographic mechanism is utilized in order to enable employees to manage the data needed for anti-money laundering (AML) measures (consisting of the analysis of the distributed ledger). There are some limitations to this method. First, there are no restrictions on the reading of the data representing the contract; this is because of its exposure during processing. Second, the authorized parties list is also left within reach of outside actors. A cryptographic process could resolve these issues.

Thompson in [20], offered a model of authentication that does not require trust in an external source. An important characteristic of a distributed blockchain network is that the implementation of such prevents the problem of a single point of failure. Thus, the blockchain's hash functions suggest a plan for the use of signatures to protect the digital certificates; hashing provides the best security and privacy possible. When comparing Thompson's model with our approach, which depends on DNA biometrics and RSA encryption, it can be seen that the key sizes yielded by RSA encryption supports more reliable results but still leaves enough resources to store, recover keys, and so support trust in the third-party authority, resolving also the problem of a single point of failure in the blockchain.
The structure of the paper is as follows: Section 2 research methodology. Section 3, describes the roles of DNA in generating cryptography keys. Finally, Section 4 describes the results from our implementation; the conclusions follow.

3. Research Method

In this section, the theoretical background and the suggested algorithms which have been used to implement the suggested framework are explained in details.

a) Generating Cryptography Keys and Biometric DNA

DNA represents a very powerful resource in terms of cryptography. The DNA connecting characteristics (as between nucleotide bases (A-T, C-G)) provide the ability to create self-assembly structures, which leads to the creation of parallel molecular computations[21][22]. DNA-based cryptography was initiated by [23] who proposed procedures for two DNA onetime pad encryption projects: XOR and substitution. Conventional cryptography is based purely on mathematical computations methods, such as nondeterministic polynomial time completeness processes. These entirely computer-based operations, depending on schema, currently face unprecedented challenges because of the evolution of decryption methods and of computing power [24].

DNA biometrics uses DNA fragments as information storage resources and carriers. It utilizes the biochemical features of DNA; the quadruple permutations of ACTG nucleotides, which can be used to perform the same role as the binary encoding applied in conventional computing systems, employing modern fragment biotechnology as a means of recognition for maintaining data security. With recent progress related to the puzzle represented by DNA, researchers discovered the biological attributes of DNA molecules which suggested the multiple biological metrics-based encryption schemes [25][26]. DNA cryptography, in other words, hiding data in the form of DNA, which works on the concepts of DNA computing, is now one of the most widely known forms of cryptography across the globe. Research is ongoing into how to utilize DNA as an information carrier and how to leverage modern biotechnology as a mean to convert ciphertext into plaintext. DNA computing (biological computing) is a new mechanism for securing data confidentially by using biological texture. The characteristics of DNA computing include: a) least processing-power demanding, b) highest speed; and c) least demanding of storage of the available methods. One gram of DNA contains 1021 bases of DNA which can represent nearly 108 terabytes of data. In other words, a gram of DNA could store all the data currently held by electronic means worldwide[23] [27][26].

This shows the importance of implementing biological techniques for securing and authenticating the user and his/her personal information or commercial/other work. Figure 1 is a schematic of the blockchain transaction.

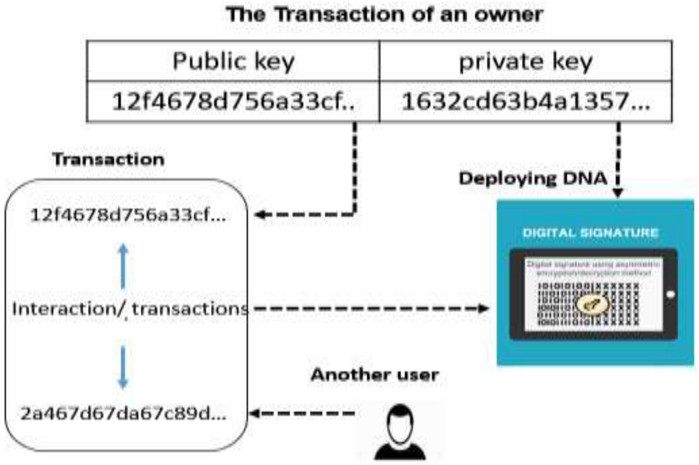


Figure 1: Blockchain transaction based on transaction explained in [6]

Generally, different researchers tend to support different approaches to the maintenance of the security of private keys. There are three general areas which together represent the domain of such methods. First, encrypting private keys with specific biometric data such as fingerprints, iris characteristics, signatures, and images. Second, adding another layer of security to the private keys when storing them onto a device. Third, using biometric data to generate asymmetric keys. In our approach here, a novel cryptography algorithm which is based on utilizing DNA biometric data sequence is employed. Each user in the proposed blockchain will depend on their own DNA data for their security. Figure 2 describes a blockchain system that uses DNA biometrics along with RSA encryption keys to create the private key, so the information inside any block in the chain cannot be accessed or changed because the private key cannot be acquired (except by biometric measurement of the user), this will decrease the risk that any corruption or falsification of the data belonging to any block can occur. In addition, this will guarantee that no transaction operation can be made unless the private key is known.

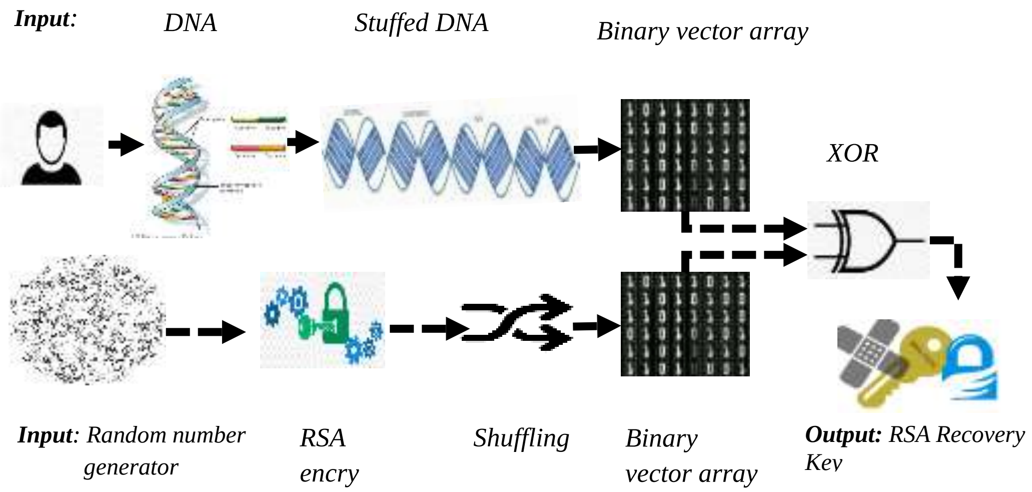


Figure 2: Key recovery process

b) Private Key Algorithm

Here, What is the best way to create asymmetric keys; with respect to this, a pseudo-random number generator for different key sizes (128, 256, 512, and 1024 bits) was studied and analyzed. Each key is checked to see whether it is a prime number or not, if it is not a prime number, then it is rejected and the process is started again. After that, the accepted private key will be represented as a string of numbers (but, just 0 or 1, so representing bits), each number being 8 bits long (i.e., a byte). Each of the bytes is then shuffled (in terms of sequence) with the other bytes; this makes it hard for an attacker to guess what has been done to create the encrypted data and so recover the private key held in a wallet, as shown in Algorithm 1. In this work, a standard data set for the DNA biometrics have been deployed, imported from (www.data.world.com). This data set is used, along with the RSA encryption algorithm, to generate a recovery private key for each block in the chain, as described in Algorithm 1. Thus, an important element is added to the blockchain: a bio-cryptography system whereby transactions can be maintained as private but readily retrieved by the legitimate user when she or he needs to start a new transaction.

In algorithm (2), the byte-sized integer values of the private key are shuffled in order to increase its randomness. This will help to make the guessing of the private keys difficult. The shuffling algorithm based on that, described in [https://rosettacode.org/wiki/Perfect_shuffle] - in which the key, considered as a vector of integers (stored in bytes), is divided into two halves. After shuffling, the first value from the first half is followed by the first value from the right half, and so on (in other words, the values are interleaved). The returned values are the shuffled private key along with a counter indicating how many times this process was carried out. The counter value will be used in the de-shuffled function.

Algorithm 1: Recovery Key

```
Input: Block chain private key, standard DNA data frame of 60 bits
Output: Encrypted private key vector
 Step 1: using random number generator, generate RSA private and public keys size (ex: 1024 bit).
 Step 2: Check if the key is a prime or not if it is not, go back to step 1
 Step 3: Consider the private key as a binary sequence, Bin_Seq= 1011111100000……….etc)
 Step 4: Consider each byte of Bin_Seq as an integer value (ex: -34, 98, -50,…etc)
 Step 5: Call shuffling algorithm, algorithm (2) - Shuffling_ Private_KEY
 Step 6: Convert DNA data to a binary number with a specific frame size, considered as a vector of bytes for simplicity
 Step 7: Adjust the DNA frame to be the same size as the private key (1024 bit) - Go to DNA_Bit Stuffing Algorithm (3)
 Step 8: While (true)
 Step 9: Consider each of the (DNA, private key) vectors as binary.
 Step 10: XOR each bit of the shuffled array with paddind_DNA array
 Step 11: If Counter! =End_Data
 Step 12: Continue
 Step 13: End If
 Step 14: End While
```

c) DNA Bit Stuffing Process

The DNA vector (DDNA) must be filled (at the end) with non-information bits. Stuffing bits (SDNA) are non-informational bits that are necessary to form an array of size equals to the size of the RSA vector (Ksize). The size for the DNA is 60 bits, whereas the RSA key sizes which have been trialed are (Ksize) = (64, 128, 256, 512, and 1024) bits. In algorithm (3) the DNA vector is stuffed with values randomly selected from the original DNA vector. After the execution of algorithms 2 and 3, both the private key vector and the DNA vector are the same size. This means that the XOR operator can be performed, bitwise, across them. Whenever the user needs to apply his/her private key to access her/his wallet, a verification process will be executed on the created recovery key. The private key can be decrypted first using the XNOR logic gate operation, and then BY de-shuffled; this will result in the original private key being recovered.

Algorithm 2: Shuffling_Private_KEY

```
Input: Private key vector (Original_a), Vector_Size
Output:a new shuffled, private key vector (New_a), count returns the number of
       shuffle performed
Step 1: Newshuffled_Vector  New_a
Step 2: Half = Vector_Size / 2;
Step 3: count=0
Step 4: while (True)
Step 5: copy array from Original_a to New_a
Step 6: for loop starts from 0 to half of the original array
Step 7: copy to   Original_a [2 * i] = New_a [i];
Step 8: copy to    Original_a[2 * i + 1] = New_a[i + half];
Step 9: End for
Step 10: If (Original _a== New_a)
Step 11: go to step 13
Step 12: End if
Step 13: End while
  Step 14: Return (New_a, count )
  Step 15: End function
```

Algorithm 3: DNA_Bit Stuffing Algorithm

```
Input: DNA_vector[], Output: Stuffed_DNA vector []
Step 1: For (i) starts from  DDNA   to  Ksize
Step 2: R_Value=Choose random( 0 to 60)
Step 3: Update_R_Value= DNA_vector[R_Value]
Step 4: DNA_vector[i]= Update_Rand_Value
Step 5: End for
```

4. Results Discussion

After the blockchain user has, via the aforementioned process, kept his private key safe, using their DNA, the process of recovery can be initiated — that is, whenever the user needs to use their private key. This is done by applying the XNOR operation to split up the DNA data from the recovery private key. Then the shuffling process will be reversed and so the original private key is revealed.

The evaluation approach is regarding to private keys, especially in relation to security levels and key sizes, and taking into account the use of the shuffling algorithm. The type of keys that have been compared against those used in this work was that generated by the elliptic curve cryptography method. Number of ECC algorithms have been studied, each of which applied a different underlying elliptic curve (resulting in elliptic curve keys with different key lengths). The reason to compare with ECC keys is that, these keys are used in most applications such as OpenSSL, OpenSSH, and Bitcoin; in blockchain systems, usually, an EC key with a length of 256 bits is used. However, different curves support different standards of security (cryptographic strengths) different speeds of generation, and different key lengths. The popular cryptographic libraries and security standards adopt the ECC curve mechanisms. In most applications, the default key length for ECC private keys is 256 bits, but depending on the curve many different ECC key sizes are possible: 192-bit (curve secp192r1), 256-bit (curves secp256k1 and Curve25519), 283-bit (curve sect283k1), 409-bit (curve sect409r1), 521-bit (curve P-521), 571-bit (curve sect571k1), and many others.

In comparison/evaluation process, a statistical package suit [28], consisting of five statistical tests, was applied. This package can only be run on data presented explicitly as bit sequences. It was used for checking the validity of our work. The statistical tests it offers are: frequency test, runs test, poker test, serial test, and the auto-correlation test. When creating a random number generator (and a cryptographic algorithm is, essentially, a special kind of random number generator), it is essential to test its features. Uniformity and freedom are the two properties with which designers should be most concerned. The following illustrations show the results from using the statistical test suite in this regard.

In all the (five) identified statistical tests, and across all the relevant key sizes, the private keys generated by our approach were indicated as successful. This was, to a large extent, due to the fact all these keys were shuffled to increase the randomness of the bit distribution. This led to an enhanced performance in terms of the five statistical tests. On the other hand, moving to the elliptic curve different key sizes used in the statistical test show a very high fail values test with all the key sizes that has been used. The results for the various key sizes yielded by the elliptic curve method (the standard encryption method used with blockchain) show very high failure values across all of the five statistical tests.

For all the above tests, it is required that each sequence to be evaluated has at least 100 bits (i.e., M =100). These tests work by examining the distribution of zeroes and ones across a number of aspects; they use spectral analysis methods to examine the harmonics of the bits stream. The tests are detailed as follows:

The frequency test: this test determines whether the frequency of ones in an M-bit block is roughly M/2, as would be expected under a randomness assumption. Figure 3 shows that in this test, all the RSA/shuffling/biometrics generated keys, of all the relevant key sizes (64, 128, 256, 1024 bits) yield an accepted value, within the range of values as determined by the observed degree of freedom, 3.84

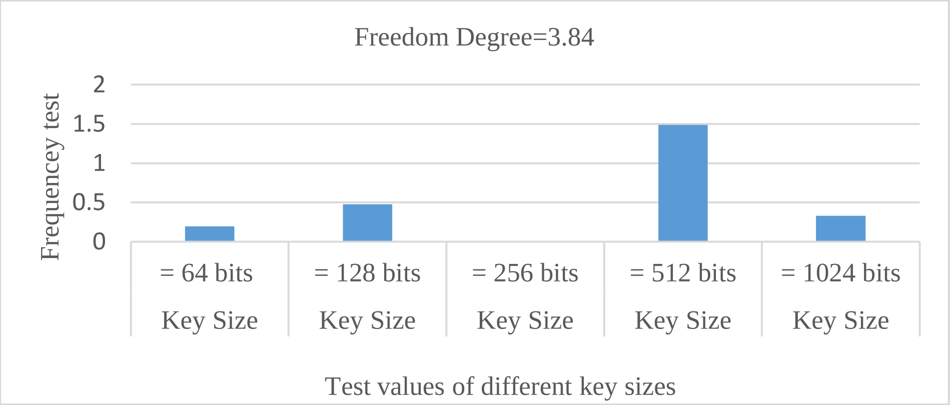


Figure 3*:* Frequency test over different key sizes

Figure 4 shows the results of the serial test. This test examines the frequencies of all the potential overlapping m-bit patterns throughout the entire series. The purpose of this test is to determine whether the number of occurrences of the 2mm-bit overlapping patterns is roughly equal to that which would be expected from a random series.

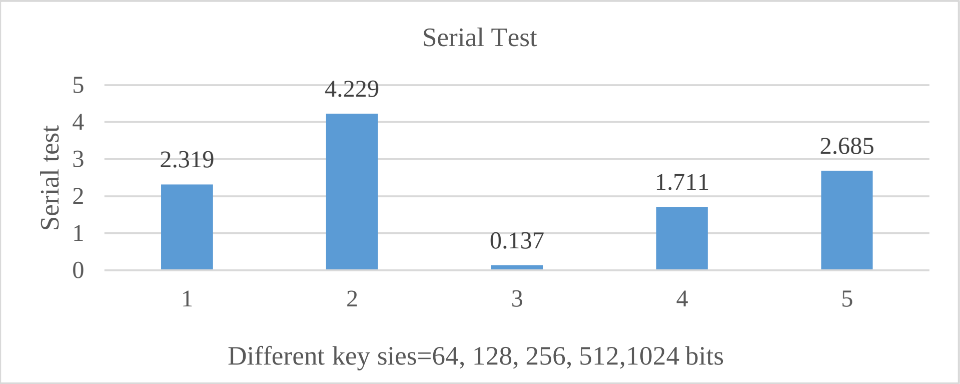


Figure 4: Serial test with different key sizes

In the poker test, the frequency at which digits (here, bits) are repeated in a sequence of numbers is used as a measure of randomness. For instance, assuming decimal digits, numbers such as 0.255, 0.577, 0.331, 0.414, 0.828, 0.909, 0.303, 0.001, etc., all contain a pair of identical digits. the degree of freedom for the test must be <=11.1, as shown in Figure 5.

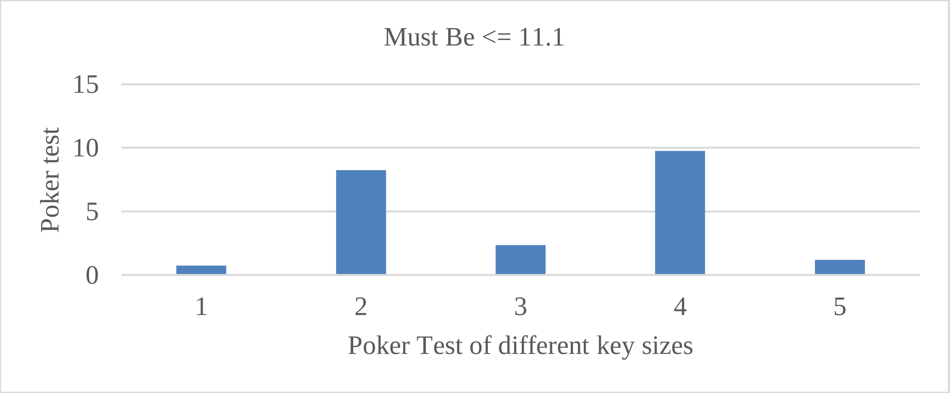


Figure 5: Poker test with different key sizes

To test the hypothesis of independence, the runs analysis examines the order of numbers in a sequence. This test evaluates, in particular, whether the oscillation between zeros and ones is too fast or too slow. The results here show that the RSA keys passed in relation to the threshold value in run T0 (runs up and down) and run T1 (runs above and below the mean). The latter was used because the T0 results are insufficient to demonstrate that the series is random, as shown in Figure 6.

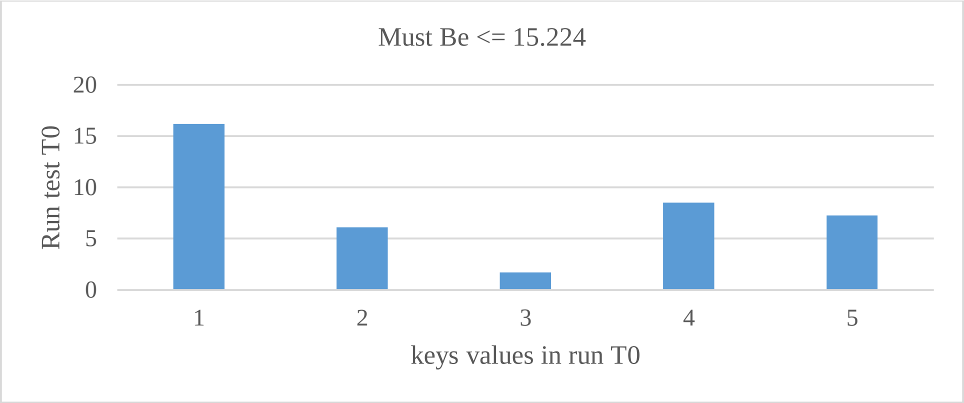
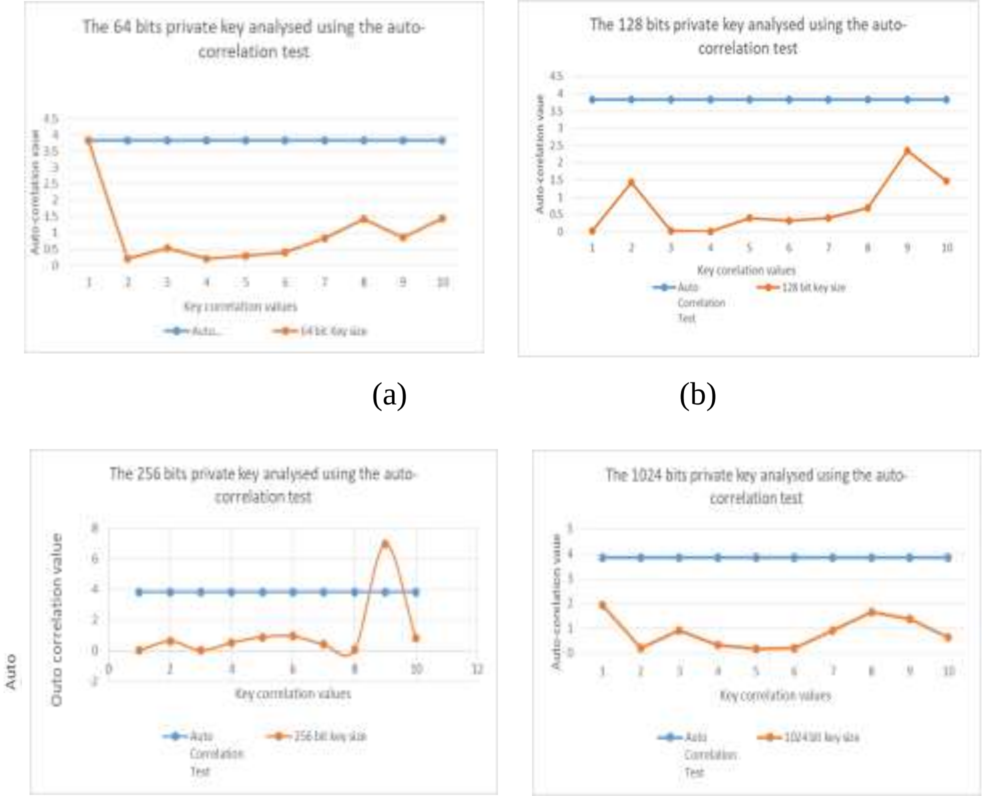


Figure 6: Run T0 and T1 with various key sizes as generated by RSA

The autocorrelation tests investigate the relationships which exist between numbers in a sequence. Starting with the i_{th} number, the test computes the autocorrelation between any m other numbers (I is termed the index, and m is the lag) as shown in Figure 7 below.



(a) (b)

(c) (d)

Figure 7: Auto-correlation test applied to (a) 64, (b) 128, (c) 256, and (d) 1024 bits private key

The same (five) tests, as implemented in the statistical package, were applied to the various key sizes as generated by the EC algorithm. Every one of these tests (to establish the randomness of the generated sequences) yielded what was considered as fail values — across all the different EC key sizes (192, 256, 409, 517 bits). Figures 8, 9, 10, and 11 show the frequency, serial, and poker and run tests as related to all the EC key sizes have; all results exceeded the observation values of the random number tests.

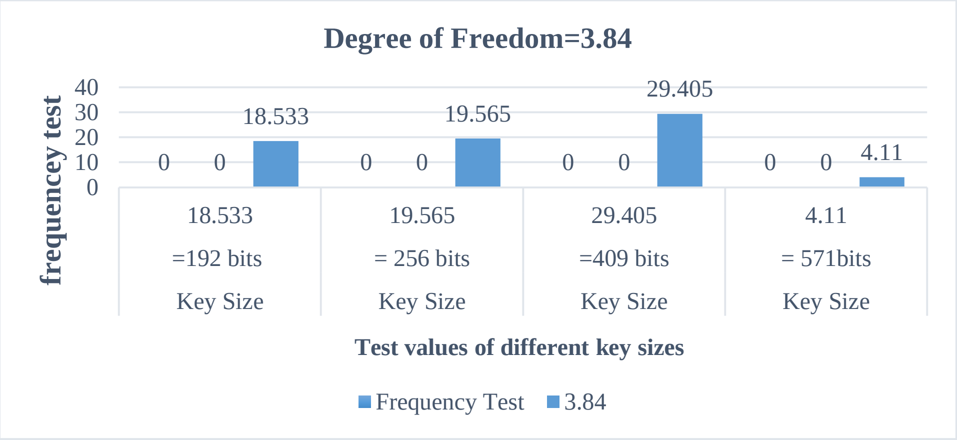


Figure 8: Frequency test applied to different sizes of EC generated keys

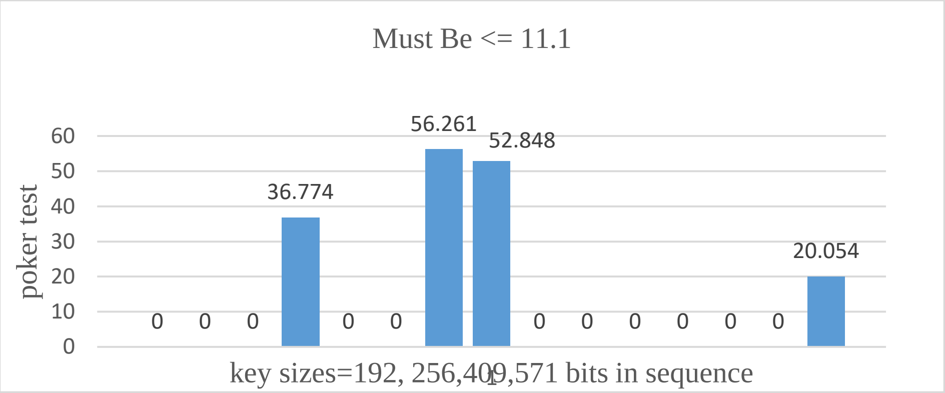


Figure 9: Serial test applied to EC generated keys of various different sizes

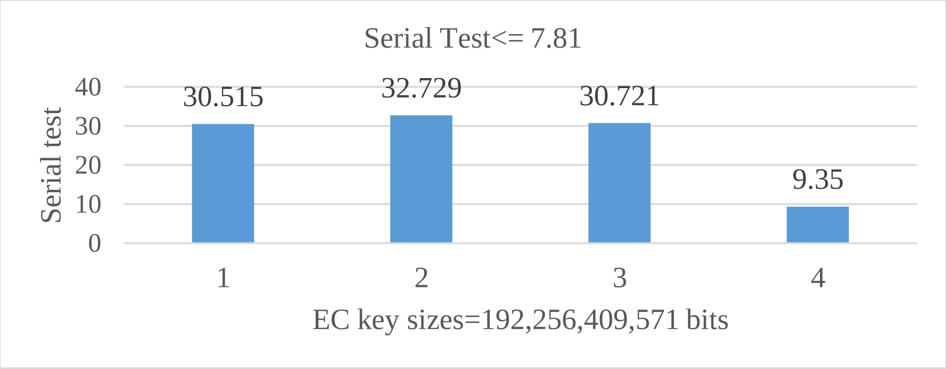


Figure 10: Poker test applied to EC generated keys of various different sizes

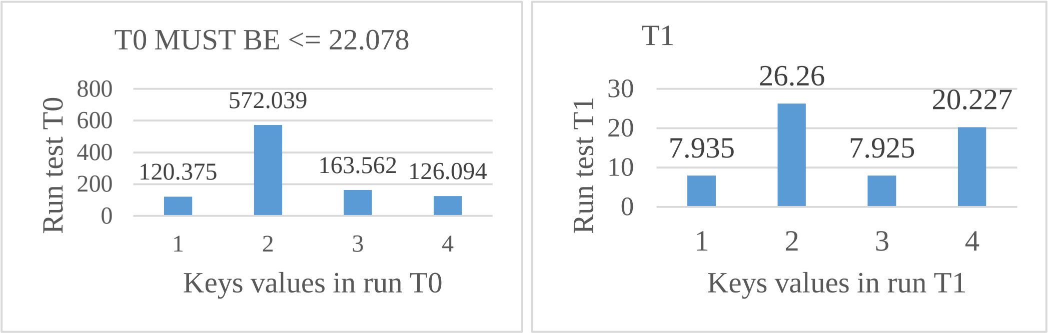
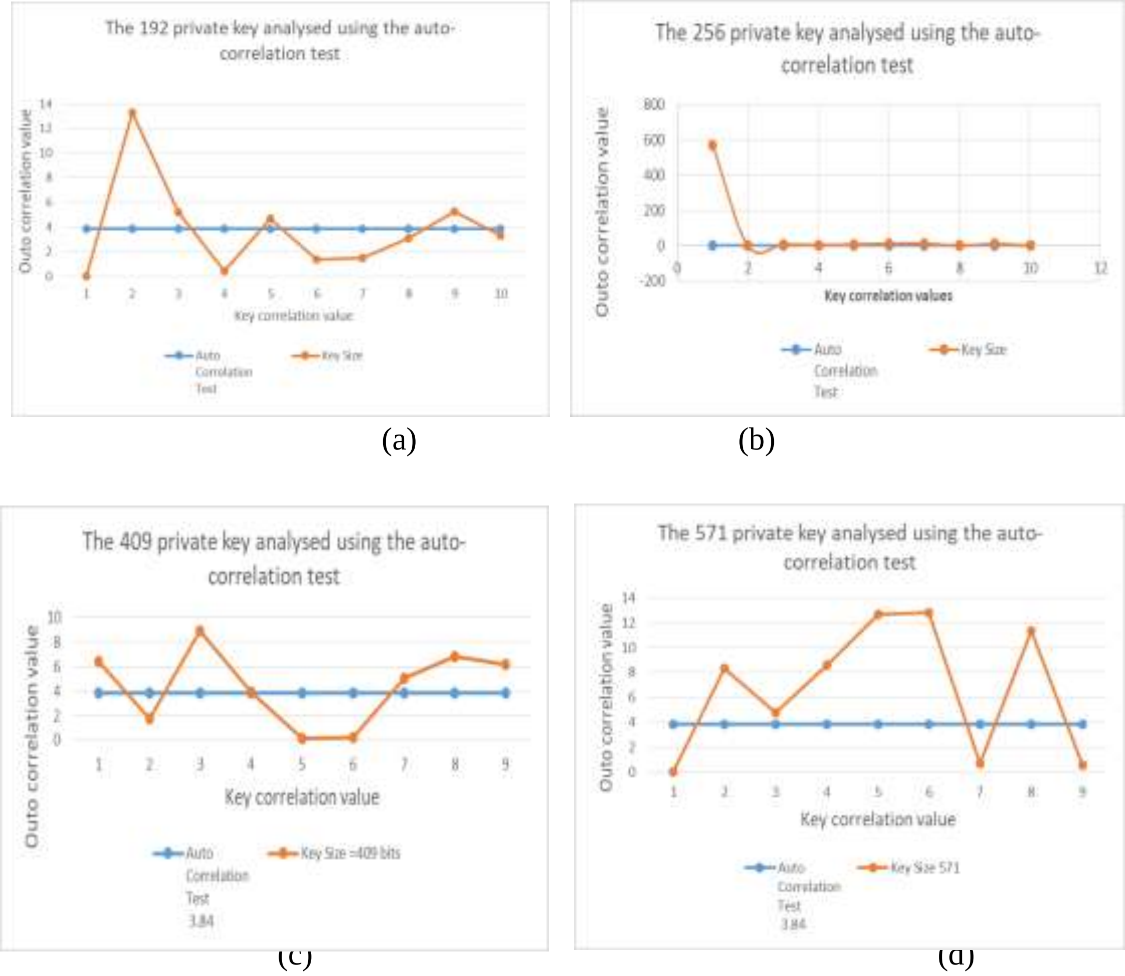


Figure 1: Run tests, T0 and T1, applied to EC generated keys of various different sizes

Moving to Figure 12, this figure shows the autocorrelation tests in relation to the various sizes of EC generated keys. All the results represent failure in terms of the randomness of the bits. Hence, keys generated in this way cannot be considered adequate in terms of their randomness.



(a) (b)

(c) (d)

Figure 12: Auto-correlation test applied to (a) 192, (b) 256, (c) 409, (d) 571 bits EC private key.

5. Conclusion

Deploying the blockchain technology in p2p networks increases the network participants' responsibility for key management. In the blockchain, assets are locked to the asset owner's public key, and this can only be activated for spending with the associated private key (which is asymmetric). In terms of security and privacy, the current solutions which have been mooted in this domain and described the issues that remain in conventional private key storage and recovery mechanisms have been examined. For key encryption, DNA cryptography has been used, and by doing so, this work has focused on increasing the security and authenticity of the blockchain. Creating methods for the efficient and secure recovery of the private keys contained in blockchain wallets is the goal of this work. Here, it is important to control the private and public keys creation, using RSA, as compared to elliptic curve keys. The error rate could prove to be a hurdle to the practical use of our system. These errors are due to a combination of error sources inherent to biometric identification systems. However, our approach shows that the security and authentication needed for blockchain technology can be accomplished using DNA combined with an RSA private key. On the other hand, the standard EC cryptography shows poor performance against our suggested method as shown in the figures of the result in the discussion section. In addition, conducting an in-depth investigation is planned to look at the possibility of relying on DNA biometrics to generate the prime numbers for the blockchain private key.

References

[1] A. K. Kibet, D. G. Bayyou, and R. Esquivel, "BLOCKCHAIN : IT'S STRUCTURE, PRINCIPLES, APPLICATIONS AND FORESEEN ISSUES," no. May, 2019.

[2] A. Yakubov, W. Shbair, and R. State, "BlockPGP: A Blockchain-based Framework for PGP Key Servers," in *2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW)*, 2018, pp. 316–322.

[3] T. T. A. Lin, Qian; Pingcheng Ruan, "Blockchains vs. distributed database : dichotomy and fusion," p. pre-print paper, 2021.

[4] A. Rahman, M. S. Hossain, Z. Rahman, and S. A. Shezan, "Performance enhancement of the internet of things with the integrated blockchain technology using RSK sidechain," *Int. J. Adv. Technol. Eng. Explor.*, vol. 6, no. 61, pp. 257–266, 2019, doi: 10.19101/ijatee.2019.650071.

[5] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *arXiv*, vol. 1, no. 1, 2019.

[6] M. Aydar and S. C. C, "Private Key Encryption and Recovery in Blockchain."

[7] T. Hamer, K. Taylor, K. S. Ng, and A. Tiu, "Private digital identity on blockchain," *CEUR Workshop Proc.*, vol. 2599, pp. 1–7, 2019.

[8] Z. Khalid, M. Rizwan, A. Shabbir, M. Shabbir, F. Ahmad, and J. Manzoor, "Cloud Server Security using Bio-Cryptography," vol. 10, no. 3, pp. 166–172, 2019.

[9] Y. Liu *et al.*, "Blockchain-Based Identity Management Systems : A," *J. Netw. Comput. Appl.*, p. 102731, 2020, doi: 10.1016/j.jnca.2020.102731.

[10] M. Raikwar, D. Gligoroski, and K. Kralevska, "SoK of Used Cryptography in Blockchain," *IEEE Access*, vol. 7, pp. 148550–148575, 2019, doi: 10.1109/ACCESS.2019.2946983.

[11] N. O. T. To and B. E. Cited, "Issues Paper on Harnessing blockchain for sustainable development : prospects and challenges," no. January, 2021.

[12] P. Patil, M. Sangeetha, and V. Bhaskar, "Blockchain for IoT Access Control , Security and Privacy," *Wirel. Pers. Commun.*, no. 0123456789, 2020, doi: 10.1007/s11277-020-07947-2.

[13] D. C. Nguyen, M. Ding, Q. Pham, P. N. Pathirana, and L. B. Le, "Federated Learning Meets Blockchain in Edge Computing : Opportunities and Challenges," pp. 1–19, 2021.

[14] S. Hossain, S. Waheed, Z. Rahman, S. K. A. Shezan, and M. Hossain, "Blockchain for the Security of Internet of Things: A Smart Home use Case using Ethereum," *Int. J. Recent Technol. Eng.*, vol. 8, no. 5, pp. 4601–4608, 2020, doi: 10.35940/ijrte.e6861.018520.

[15] H. Zhu and Z. Li, "An Efficient Biometric Authenticated Protocol for Arbitrary-domain-server with Blockchain Technology," vol. 23, no. 3, pp. 386–394, 2021, doi: 10.6633/IJNS.202105.

[16] W. Bi, X. Jia, and M. Zheng, "A secure multiple elliptic curves digital signature algorithm for blockchain," *arXiv*, 2018.

[17] Y. Kaga, M. Fujio, K. Takahashi, and T. Murakami, "A Secure and Practical Signature Scheme for Blockchain Based on Biometrics," no. December 2018, 2017, doi: 10.1007/978-3-319-72359-4.

[18] L. A. Ajao, J. Agajo, E. A. Adedokun, and L. Karngong, "Crypto Hash Algorithm-Based Blockchain Technology for Managing Decentralized Ledger Database in Oil and Gas Industry," *J*, vol. 2, no. 3, pp. 300–325, 2019, doi: 10.3390/j2030021.

[19] T. Robles, B. Bordel, R. Alcarria, and D. Sánchez-de-Rivera, "Blockchain Technologies for Private Data Management in AmI Environments," *Proceedings*, vol. 2, no. 19, p. 1230, 2018, doi: 10.3390/proceedings2191230.

[20] S. Thompson, "The preservation of digital signatures on the blockchain," vol. 3, no. Spring, 2017.

[21] O. Tornea and M. E. Borda, "Algorithm steps," pp. 223–226, 2009.

[22] S. M. Hussain and H. Al-bahadili, "A DNA-Based Cryptographic Key Generation Algorithm A

[23] A. Gahlaut, A. Bharti, Y. Dogra, and P. Singh, "DNA based cryptography," *Commun. Comput. Inf. Sci.*, vol. 750, no. C, pp. 205–215, 2017, doi: 10.1007/978-981-10-6544-6_20.

[24] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions How to Break MD5 and Other Hash Functions," no. May 2005, 2014, doi: 10.1007/11426639.

[25] Y. Zhang, X. Liu, and M. Sun, "DNA based Random Key Generation and Management for OTP Encryption," *BioSystems*, no. October, 2018, doi: 10.1016/j.biosystems.2017.07.002.

[26] G. Gürsoy, C. M. Brannon, S. Wagner, and M. Gerstein, "Storing and analyzing a genome on a blockchain," *bioRxiv*, pp. 1–24, 2020, doi: 10.1101/2020.03.03.975334.

[27] M. Rathi, S. Bhaskare, T. Kale, N. Shah, and N. Vaswani, "Data Security Using DNA Cryptography," vol. 5, no. 10, pp. 123–129, 2016.

[28] A. Rukhin, J. Soto, and J. Nechvatal, "Nistspecialpublication800-22R1a.Pdf," no. April, 2010.