End-to-End Security Scheme for E-Health Systems Using DNA-Based ECC

Sanaz Rahimi Moosavi(✉) and Arman Izadifar

California State University Dominguez Hills (CSUDH), Carson, CA 90747, USA
srahimimoosavi@csudh.edu

Abstract. Today, the amount of data produced and stored in computing Internet of Things (IoT) devices is growing. Massive volumes of sensitive information are exchanged between these devices making it critical to ensure the security of these data. Cryptography is a widely used method for ensuring data security. Many lightweight cryptographic algorithms have been developed to address the limitations of resources on the IoT devices. Such devices have limited processing capabilities in terms of memory, processing power, storage, etc. The primary goal of exploiting cryptographic technique is to send data from the sender to the receiver in the most secure way to prevent eavesdropping of the content of the original data. In this paper, we propose an end-to-end security scheme for IoT system. The proposed scheme consists of (i) a secure and efficient mutual authentication scheme based on the Elliptic Curve Cryptography (ECC) and the Quark lightweight hash design, and (ii) a secure end-to-end communication based on Deoxyribonucleic Acid (DNA) and ECC. DNA Cryptography is the cryptographic technique to encrypt and decrypt the original data using DNA sequences based on its biological processes. It is a novel technique to hide data from unauthorized access with the help of DNA. The security analysis of the proposed scheme reveals that it is secure against the relevant threat models and provides a higher security level than the existing related work in the literature.

Keywords: Deoxyribonucleic Acid (DNA) · Elliptic curve cryptography · E-health system · End-to-end security

1 Introduction

The Internet of Things (IoT) is a new paradigm for modern pervasive wireless communications that connects a wide range of physical devices via the Internet to collect and exchange data. A healthcare IoT network consists of smart sensors, wearable/implantable health devices, and medical instruments that can remotely monitor a patient's health. Among the major areas of concern in healthcare IoT are patient's security and privacy. In this regard, remote health caregiver (end-user) authentication and authorization, as well as end-to-end data protection, are critical requirements to prevent eavesdropping on sensitive medical data

S.-Y. Chang et al. (Eds.): SVCC 2021, CCIS 1536, pp. 77–89, 2022.
https://doi.org/10.1007/978-3-030-96057-5_6

or malicious triggering of specific tasks [1]. As humans are directly involved in healthcare IoT applications, robust and secure data communication among healthcare sensors, actuators, patients, and caregivers is critical. Cryptography is defined as the technique that applies logic and mathematics to keep and send information in the coding style and through a secured format so that only the intended receiver can read and translate the meaning. State-of-the-art security and protection mechanisms, such as existing cryptographic solutions, secure protocols, and privacy assurance, cannot be re-used in healthcare IoT systems due to resource constraints, security level requirements, and system architecture [2]. Strong network security infrastructures for short and long-range communication are required to mitigate the aforementioned risks.

Unlike symmetric ciphers, which uses the same secret key to encrypt and decrypt sensitive data, asymmetric ciphers, also known as public-key cryptography or public-key encryption, uses mathematically linked public- and private-key pairs to encrypt and decrypt sensitive data sent to and received from senders and recipients. An important advantage of asymmetric ciphers over symmetric ciphers is that no secret channel is necessary for the exchange of the public key. The receiver needs only to be assured of the authenticity of the public key. Due to its superiority in generating a powerful encryption mechanism with small key sizes, ECC is widely used in constrained environments for asymmetric cryptography. ECC improves device performance while decreasing power consumption, making it suitable for a wide range of applications, including healthcare IoT. On the other hand, Deoxyribonucleic Acid (DNA) cryptography can be defined as a technique of hiding data in terms of DNA sequence. In the cryptographic technique, each letter of the alphabet is converted into a different combination of the four bases which make up the human's DNA [4,5]. DNA cryptography is a rapidly developing technology that is based on DNA computing concepts. Beside of the huge parallelism, DNA molecules also have massive storage capacity. A gram of DNA molecules consist of 10^{21} DNA bases which is nearly about 10^8 tera-byte [6]. As a result, it can be concluded that a few grams of DNA can contain all of the world's data [7]. These benefits of DNA computation inspire the concept of DNA cryptography. Prof. Leonard Adleman, also known as the 'A' of the RSA algorithm, is regarded as the father of DNA computing [3].

In this paper, an end-to-end security scheme for healthcare IoT systems is proposed. The main contributions of this work are twofold. First, we present our end-to-end security solution for e-health systems. In this regard, we exploit the DNA and ECC cryptography techniques. Second, we analyze the characteristics of the proposed scheme in terms of security. The security analysis of the scheme demonstrates that it is secure against the relevant threat models and offers a higher security level than the existing related work in the literature.

The remainder of this paper is organized as follows: Sect. 2 provides an overview of related work. The end-to-end security scheme for e-health systems using DNA-based ECC is presented in Sect. 3. Section 4 provides a comprehensive security analysis of our scheme. Finally, Sect. 5 concludes the paper.

2 Related Work

Roy *et al.* [8] devised a method based on DNA synthesis to improve key generation. The encryption and decryption processes are optimized by this system. A first level key and an encryption algorithm are used to convert plain text to primary cipher text. The concept of a second level key is introduced, enhancing the security of this technique. The second level private key strengthens the cipher text by adding primers and intron positions. Excellent results are obtained after analyzing the proposed method against brute force attacks. The hacker would need more than a half-year to decrypt the cipher text using a modern computer. This method has a high level of time and space complexity. Shinde *et al.* [9] proposed a new DNA-based cryptography technique. The method combines traditional cryptographic techniques with novel approaches to improve data security. The plaintext is first converted into an ASCII value, and then into binary strings. The binary strings are then converted to hexadecimal values, and a 128 bit key is generated using the MD5 algorithm. This key is converted into a 32-character hexadecimal string that is mapped to 16 dynamic values. The binary values are encoded using a mapping table. Following encoding, some mathematical and logical operations are carried out. This technique is both quick and efficient. However, he security offered in this algorithm is not suitable for healthcare IoT systems. Gogte *et al.* [10] presented a new type of DNA cryptography system for secure communication based on quantum cryptography. Quantum cryptography is a new security technique in which two parties communicate using a quantum channel. Its foundations are Heisenberg's uncertainty principle and the no-cloning theorem. Initially, a simulation of quantum key exchange and authentication is carried out. This is followed by the use of a DNA-based algorithm. The DNA encryption algorithm employs a symmetric block cipher with a 128 bit key as input. The method is secure against man-in-the-middle attacks, eavesdropping, replay attacks, packet sniffing, and spoofing. However, the technique is heavy-weight to be implemented for the resource-constrained e-health systems. A DNA cryptographic algorithm was proposed by Zhang *et al.* [11]. The method is based on the assembly of DNA fragments. The authors' algorithm incorporates DNA digital coding, DNA molecular keys, and some software techniques. This method is based on the concept of symmetric key cryptography. The encryption mechanism in this case is accomplished through the use of DNA digital coding. The main challenge of this algorithm is the implementation of the DNA molecular key. Ibrahim *et al.* [12] proposed using double DNA sequences to improve the security of data hiding. The main idea behind the scheme's design is to encrypt secret messages to ensure security and robustness. The encrypted message is tucked away in a different DNA reference sequence. Overall, a new data concealment algorithm based on DNA sequences has been suggested. The hiding of data in repeated characters in this scheme reduces the rate of modification. However, in this approach, if the attacker manages to obtain the secret message then the method is broken.

3 End-to-End Security Through DNA-Based ECC

In this section, we present our end-to-end security scheme for healthcare IoT systems. The proposed scheme consists of (i) ECC-based mutual authentication and authorization, (ii) DNA-ECC-based encryption. Our scheme offers the first-level of security through ECC algorithm requiring smaller key size and less computation overhead. The second level of security is provided by the use of a low computation DNA-ECC cryptosystem. The structure of a DNA-based cryptography techniques is shown in Fig. 1.

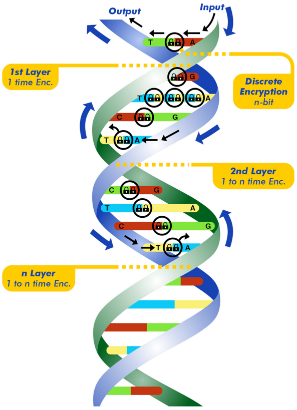

Fig. 1. DNA-based cryptography technique

3.1 ECC-Based Mutual Authentication and Authorization

This section describes our ECC-based mutual authentication and authorization scheme, which meets the security requirements of a healthcare IoT system. A mutual authentication scheme allows the communicating parties, the medical sensor device and the health care provider, to verify and ensure each other's identities. The scheme is divided into two phases: (i) health caregiver authentication and (ii) medical sensor device identification and verification. Elliptic curve cryptography (ECC) was firstly proposed by Victor Miller and Neal Koblitz [13]. It is a type of public key encryption system used to generate smaller, faster, and more efficient cryptographic keys. In contrast to the RSA algorithm, which is based on large prime numbers, keys in the ECC are generated using the elliptic curve equation's parameters. The encryption functionality provided by ECC requires fewer resources than RSA or other public key algorithms. In general, the longer the key, the better the protection for any system. However, when compared to RSA, ECC can provide comparable protection with a smaller key size. As a result, the ECC's resources must perform fewer mathematical computations. The security level of ECC can be achieved with a 164-bit key, whereas other systems require a 1024-bit key. Furthermore, the security of ECC is based on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP), and because the computation of DLP problems is not easy, it prevents an adversary from easily breaking the ECDLP. An ECC E [14] over a finite field $\mathbb{F}_p$ includes all points $(x,\ y)\ \in\ \mathbb{F}\ _p\ \times\ \mathbb{F}_p$ which fulfill an equation of the form $Y^2 + a_1XY\ + a_3Y\ =\ X^3\ +\ a_2X^2\ \ a_4X\ +\ a_6$ with $a_i \in \mathbb{F}_p$, whose discriminant is non-zero, accompanied by the point at infinity. Then, an Elliptic Curve $E\ (\mathbb{F}_p)$ over $\mathbb{F}_p$ defined by parameters $a,\ b\ \in\ \mathbb{F}_p$ made up of serious of points $P\ =\ (x,\ y)$ for $x,\ y\ \in\ \mathbb{F}_p$ to the equation:

$$y^2\ \equiv\ x^3\ +\ ax\ + b\quad (mod\ p) \tag{1}$$

The mentioned equation $y^2\ \equiv\ x^3\ +\ ax\ + b\quad (mod\ p)$ is called the description of the equation $E\ (\mathbb{F}_p)$ for a certain point $p\ =\ (xp,\ yp)$. Here, xp is entitled as the x-coordinate of P, and yp is called the y-coordinate of P. The number of point on $E\ (\mathbb{F}_p)$ represents as $\#\ E\ (\mathbb{F}_p)$ and:

$$p\ +\ 1\ -\ 2\sqrt{p}\ \le\ \#\ E\ (\mathbb{F}_p)\ \le\ p\ +\ 1\ \ +\ 2\sqrt{p} \tag{2}$$

The elliptic curve version of the Digital Signature Algorithm is known as the Elliptic Curve Digital Signature Algorithm (ECDSA) (DSA). The ECDSA is a modified version of the DSA and RSA that works with Elliptic Curve groups. The proposed ECDSA not only offers smaller key sizes for the same security level, but it also significantly improves ECC generation and authentication techniques. The diagram of elliptic curve is shown in Fig. 2. The proposed ECC-based mutual authentication scheme establishes a secure channel between the sensor and the caregiver, allowing them to communicate securely and efficiently. Before delving into the phases, we go over the parameters and notations used in the scheme.

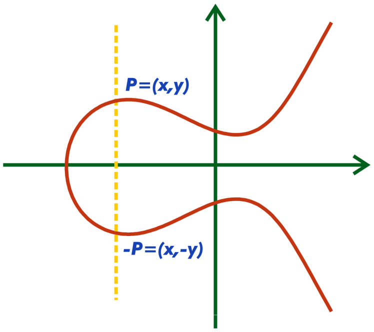

Fig. 2. Elliptic curve diagram

- G: a group of order q on an elliptic curve having the order n,
- P: a primitive element or the base point of G,
- $sp_1,\ sp_2$: each sensor keeps two secret points $sp_1,\ sp_2\ \in$ E(F_g), which will change over time. These secret points will be varied each time the sensor is successfully identified,

- ID_n: the sensor's identification number or ID,
- sp_3: each end-user keeps a secret point $sp_3 \in Z_n$, which will change over time. This secret point will be varied each time the end-user is successfully authenticated,
- $ID_k = sp_3.P$: the end-user's public key,
- r_n, j_1, j_2: random numbers in Z_n,
- h: a lightweight hash function,
- (x, y): a signature generated by the sensor in its identification phase.

3.2 Health Caregiver Authentication Phase

The Health Caregiver authentication phase of our scheme is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP) [13]. In this phase, the caregiver is assigned with a random number $rs_1 \in Z_n$ and its public key is computed as $R_1 = rs_1.P$. Next, the caregiver initializes its counter value j_1 to one and sends both R_1 and j_1 to the medical sensor. It then increments the value j_1 by rn_1. Upon receiving the message, the sensor checks whether j_2 (which is initialized to zero) is greater than j_1. If the condition holds, it replaces j_2 by j_1 and selects a random number $rn_2 \in Z_n$. Then, the sensor computes:

$$rn_3 = X(rn_2.P) * Y(R_1) \tag{3}$$

where * is a non-algebraic operation over the abscissa of $(rn_2.P)$ and the ordinate of R_1 and it sends the value rn_3 to the caregiver. After receiving rn_3, the caregiver computes R_2 and sends this value to the sensor. Finally, if the following equation holds, the sensor verifies that the caregiver is authentic.

$$R_2 = rn_1.ID_n + rn_3.sp_3 \tag{4}$$

$$(R_2 - rn_1.ID_n)rn_3^{-1}.P = ID_k \tag{5}$$

3.3 Medical Sensor Authentication and Verification Phase

Our scheme's medical sensor identification and verification phase is based on the Elliptic Curve Digital Signature Algorithm (ECDSA) using Quark lightweight hash design. Quark is one of the most efficient lightweight hash designs and it was first proposed by Aumasson *et al.* [14]. Quark lightweight hash is based on non-linear Boolean functions and bit shift registers. As a result, not only is its implementation feasible, but the circuit area requirements of this hash design are ideal for implantable medical devices. A digital signature also provides identification, integrity, and non-repudiation. Because of resource constraints and the delicate use cases of healthcare IoT systems, lightweight cryptographic hash designs must be carefully considered. As a result, we use the D-Quark in our proposed ECC-based medical sensor authentication (i.e., one of the flavors of Quark) lightweight hash design rather than the general purpose hash designs. In the sensor identification phase of our scheme, the sensor's initial secret point

Table 1. DNA nucleotide to binary and decimal conversion

DNA nucleotide base	Binary equivalent	Decimal equivalent
Adenine (A)	00	10
Thymine(T)	01	20
Guanine(G)	10	30
Cytosine(C)	11	40

is $sp_1 \in E(F_g)$ from which the next secret point sp_2 and ID_n will be computed. To generate the second secret point, the sensor computes:

$$s_2 = f(X(s_1)).P \tag{6}$$

Obtaining the first secret point from the second is difficult, as it requires the computation of an ECDLP. Since the second key is generated from the second key, our scheme provides forward security. For the sake of efficiency, the function f should be selected in a manner that avoids large hamming weights for sp_2, assuring that the computation of $sp_2.P$ will be fast without compromising security [12]. Once the generation of the second secret point sp_2 is done, the sensor selects a random integer $k \in Z_g$ and computes a curve point $(d, c) = k.G$. To send its digital signed message (x, y) to the back-end system, the sensor computes $d = x \bmod n$. If $x = 0$, the sensor starts to select a another random number $k \in Z_g$ and computes the next curve point and its ID as:

$$ID_n = Mb(X(sp_1)) * Mb(X(sp_2)).P \tag{7}$$

where Mb will output some middle bits of the input values. The operand * is a non-algebraic operation $\in F_g$ done over the abscissa of the first and the second secret points. Then, the sensor computes the following equation:

$$l = k(hash(ID_n) + X(sp_1).x) \tag{8}$$

If the computed $y = 0$, the sensor will start the algorithm by selecting another random integer e. Finally, the sensor sends the computed values (x, y) and (ID_n) to the back-end system. Algorithm 2 shows the pseudocode of the sensor identification phase of the proposed scheme. To verify the sensor is authentic the beck-end system selects a random integer $rn_s \in Z_n$ and it computes its public key $p_r = rn_s.P$ for $j \in [1, n-1]$, the back-end system checks whether $x, y \in Z_n$. If the result is valid, the back-end system calculates $h = Hash(ID_n)$, where Hash is the same Quark lightweight hash function that is used in the previous phase to generate the sensor's signature. Once the hash value of (ID_n) is computed, the back-end selects the leftmost bit of h and denotes it as z. Then, the back-end calculates the values U, m_1, m_2. Based on the calculated values, the back-end system computes the curve point as:

$$(x, y) = m_1.P + p_r \tag{9}$$

Finally, the back-end system will accept the sensor's signature as a valid one if the equation $r = x \bmod n$ holds.

3.4 DNA-Based ECC Cryptography

DNA cryptography is a method of concealing data in terms of DNA sequence. Each letter of the alphabet is converted into a different combination of the four bases that make up human DNA in the cryptographic technique. DNA cryptography is a rapidly developing technology that is based on DNA computing concepts. Inside the tiny nuclei of living cells, DNA stores a massive amount of information. It contains all of the instructions required to create every living creature on the planet. The main advantages of DNA computation are miniaturization and parallelism, which are not available in conventional silicon-based machines. With its unique data structure and ability to perform many parallel operations, DNA allows one to view a computational problem from a new perspective. The following are the benefits of using DNA cryptography:

Algorithm 1. DNA-ECC Cryptographic Algorithm

Input: Plaintext (P), number of bits of DNA sequence segment (k), known DNA sequence (D).
Output: Ciphertext (C)
Global Variables: ECC points, which are denoted as $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, an auxiliary base parameter k for which both entities need to agree upon this; **Global Constants:** $Tasks$: Vector of ECC **Body:**

```
Input $Plaintext$
Convert $Plaintext$ into Binary $P'$
Convert $D$ into Binary $D'$
Segment $D'$ with $k$ bit in a segment
Insert each bits of $P'$ into the beginning of each segment of $D'$
Concatenate segments of $D'$
Convert each character of the $DNA$ Nucleotide into Numbers as $10(00), T = 20(01), G = 30(10), C = 40(11)$;
Koblitz Method: Pick an elliptic curve $ECC = (a, b)$.
Each number $mk$, takes $x = mk + 1$ and tries to solve for $y$.
while $y \neq solved$ do
    for each $x \in Tasks$ do
        $x \leftarrow mk + k - 1$;
        Take the point $(x, y)$ and covert $m$ into a point on the $ECC$;
    end for
end while
```

1. *Power Requirements:* While the computation is taking place, no power is required for DNA computing. Chemical bonds, which are the building blocks of DNA, form without the assistance of an outside source of energy. The power requirements of traditional computers are incomparable.

2. *Speed:* Conventional computers have a peak performance of about 100 MIPS (millions of instruction per second). Combining DNA strands as demonstrated by Adleman made computations equivalent to 10^9 or better, arguably over 100 times faster than the fastest computer.
3. *Storage Requirements:* Memory is stored in DNA at a density of about 1 bit per cubic nanometer, where conventional storage media requires 10^{12} cubic nanometers to store 1 bit.

A simple mechanism of transmitting two related messages while concealing the message is insufficient to prevent an attacker from breaking the code. DNA cryptography has a unique advantage for secure data storage, authentication, digital signatures, steganography, and other applications. DNA strands are long polymers made up of millions of linked nucleotides. As Algorithm 1 indicates, these nucleotides are made up of four nitrogen bases, a five-carbon sugar, and a phosphate group. The nucleotides that make up these polymers are named after the nitrogen base in which they are composed: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). This means we can utilize this 4 letter alphabet $\Sigma = \{A, G, C, T\}$ to encode information, which is more than enough considering that an electronic computer needs only two digits, 1 and 0, for the same purpose. Three DNA cryptography methods are used in this cryptosystem. They are (i) the insertion method, (ii) the substitution method, and (iii) the complementary pair approach. A common method of encoding and decoding is used in all of these approaches. Binary numbers are generated from the plaintext. The binary numbers are then converted to a DNA nucleotide sequence.

Algorithm 2. DNA-ECC Cryptographic Example

Input: *Plaintext Message (P)*: ”m”
ASCII Message: 109
Binary Message(P’): 01101101
DNA Sequence (D): TCGCAATTCGCGCTGAGTCACAATTCGCGCTGAGTCACA ATTCGCGCTGAGTCACAATTGTGACTCAGCCGCGAATTCCTGCAGCCCCGA ATTCCGCATTGCAGAGATAATTGTATTTAAGTGCCTAGCT.
Output: *Ciphertext (C)* **Body:** *converting plaintext to ciphertext*

- Binary DNA Sequence (D’): 01 11 10 11 10 11 01 01 00 00 10 11 01 00
- Segmented Binary DNA sequence (where k = 3): 110 100 111 010 000 101 111 110 110 1
- Insert each bits of P’ into beginning of each segments of D’:1-010—1-111—0-000—1-111—1-100—0-000—0-101—-110—-1
- Concatenate the segments of D’:000011111100000010100101010111 01
- Convert D’ 10-00- 11-11-11- 00- 00- 00-10-10- 01-11-01- 01-11-01 to DNA nucleotide A- C- A- A- G- G- C- C- C- A- T- T- C- A- G- T
- Convert DNA nucleotide to ASCII A C A A G G C C C A T T C A G T 10 40 10 10 30 30 40 40 40 10 20 20 40 10 30 20
- Convert the ASCII of DNA nucleotide to ECC point

The following facts underpin the encoding and decoding operations. As shown in Table 1, there are four basic units in DNA that are encoded into binary in the following manner. Binary equivalent of a DNA nucleotide base Adenine (A) is 00, Thymine (T) is 01, Guanine (G) is 10, and Cytosine (C) is 11. The DNA sequences in this work are taken from a publicly available database and converted into a binary sequence. The binary DNA sequences are divided into segments, each of which contains a random number of bits greater than two. Then, each bit of binary plain text is inserted at the start of a segmented binary DNA sequence. The inserted sequences are concatenated to obtain an encoded binary sequence. The segments are needed to be concatenated again and converted to Nucleotide letter. For encryption, we use the Koblitz method to convert decimal numbers into elliptic curve points. The plaintext is represented by ECC curve points. The ECC encryption algorithm is used to encrypt these points (Algorithm 2) [16].

$$\{kG,\ Pm\ +\ k\ PB\} \tag{10}$$

where, G is the generated points, Pm is the plaintext points, k is a random number being selected by the user, and PB is the public key of the user. The ECC decryption algorithm is used to decipher the ciphertext points. The Koblitz method is used to convert deciphered points into numbers. These numbers are decoded using DNA nucleotides, and the required plaintext is obtained.

$$Pm + kPB - nB(kG) = Pm + k(nB)G - nB(kG) \tag{11}$$

4 Security Analysis of the Proposed Scheme

In this section, we analyze the security of the proposed scheme in order to verify whether the essential requirements have been satisfied.

Mutual Authentication: In the end-user authentication phase of our scheme, to verify that the end-user is legitimate, the medical sensor computes whether $(R_2 - rn_1.ID_n)rn_3^{-1}.P = ID_k$ holds or not. Similarly, to verify whether the medical sensor is authentic (based on its transmitted (ID_n) and the digital signed message), the end-user needs to checks if $r = x \bmod n$ holds. This is how our proposed scheme achieves mutual authentication.

Availability: In our scheme, since the sensor and the end-user change their secret points sp_1, sp_2, and sp_3 once they are successfully authenticated, it is not possible that an adversary performs a denial of service attack.

Forward Security: Here, if an adversary attempts to decrypt some of the information he has intercepted, for example the sensor's second secret key s_2, he/she cannot benefit from the gained information. Obtaining the first secret key from the second will necessitate a solution to the ECDSA, which will be difficult.

Impersonation Attack: Concerning this type of attack, we consider two different scenarios: (i) *Impersonation of the end-user*: Here, if an adversary tries to impersonate the end-user, he/she will fail. This is because if the attacker tries to impersonate as a fake health caregiver to the medical sensor, he/she must compute R_1 and at the same time try to guess the value rn_2 (which is not easily feasible). Nevertheless, without the end-user's computed value $R_2 = rn_1.ID_n + rn_3.sp_1$, the adversary cannot compute $(R_2 - rn_1.ID_n)rn_3^{-1}.P = ID_k$ to verify whether the end-user is authentic. (ii) *Impersonation of the medical sensor*: In order to impersonate the sensor in our proposed scheme, an adversary needs to have an access to the sensor's secrets sp_1 and sp_2 and as it was presented earlier in this section, the values of the secret keys cannot be acquired from the public information of the system ID_n.

Brute-Force Attack: The DNA sequences in the proposed scheme are chosen randomly from a pool of available DNA sequences. Hence, it is impossible to predict the DNA sequence used in this study. In other words, no predictive model can be used by an attacker to determine the used DNA sequence. Without knowledge of the DNA sequence, the attacker will be unable to capture the network. When each sensor is assigned multiple DNA sequences, the DNA sequence pool is formed by randomly selecting DNA sequences from a pool of thousands. Each DNA sequence in the pool is distinct from the other DNA sequences in the pool. There are currently no methods for predicting which DNA sequences are present in the pool. Using any predictive model, an attacker cannot determine the entire DNA sequence pool. As a result, without knowledge of the DNA sequence, an attacker cannot easily capture the network.

Eavesdropping: In our scheme, (i) in the sensor identification phase, if an adversary tries to guess the sensor's secrets sp_1 and sp_2, the only public information concerning it is *ID*. As it was discussed earlier, the bits of the sensor's *ID* result from a non-algebraic operation done over some middle bits of the abscissa of two different secret points sp_1 and sp_2. Thus, it is computationally unfeasible to obtain the secret from its *ID*. (ii) In the digital signature generation section, if an adversary could guess the value x, it cannot obtain the value y effortlessly. This value is also generated from a non-algebraic operation done over the abscissa of the secret point sp_1 and the value x. The gained result will be added to the hash value of ID_n and multiplied by a random number k. Such an operation cannot be easily computed by an adversary as it requires to compute the discrete logarithm problem that is not computationally feasible. For the same reason, in the end-user authentication phase, even if an adversary could guess one of the values R_1 or R_2 or rn_3, he/she still cannot easily obtain other secure information related to the end-user. Based on the discussion above, the adversary also cannot implement any *Replay Attack*.

Unauthorized Tracking of the Sensor: Here, the only public information concerning the sensor is its *ID*. In the sensor identification phase, it was shown that the value of the sensor's *ID* results from the product of a non-algebraic operation done over some middle bits of the abscissa of the first and second secret keys of the sensor. Hence, it is impossible to compute and obtain the sensor's secret keys from its current *ID*. The main reason for this is that obtaining the secret points necessitates solving the elliptic curve discrete logarithm problem. Solving the discrete logarithm problem is as difficult as solving the integer factorization problem, this problem cannot be solved easily. Thus far, there has not been any polynomial time algorithm proposed to solve discrete logarithm problems.

5 Conclusion and Future Work

In this paper, we presented a novel end-to-end security scheme for healthcare IoT systems using ECC and DNA cryptography techniques. To the best of our knowledge, previously proposed end-to-end security schemes, concerning e-health systems in general, cannot fully fulfill the essential security requirements of health-care IoT systems. The majority of the previously proposed solutions were not secure against most common attacks on healthcare IoT systems. The proposed scheme was specified and designed by employing (i) ECC and the Quark lightweight hash design to mutually authenticate and authorize medical sensors and end-users (i.e. health caregivers), and (ii) the DNA-based ECC cryptographic technique to encrypt and decrypt the health data using DNA sequences of the patients. We demonstrated that our proposed scheme is secure against the relevant attacks and provides a higher level of security than related work found in the literature. Based on the security analyses presented in this paper, we conclude that the proposed scheme has the appropriate features for use in e-health systems. We believe that our scheme is not just limited to healthcare IoT systems and can also be applied to any application of IoT that requires secure and efficient end-to-end communication. Our future work will focus on performance analysis of the proposed scheme in terms of terms of communication overhead, latency, and memory footprint.

References

1. Hummen, R., Shafagh, H., Raza, S., Voig, T., Wehrle, K.: Delegation-based authentication and authorization for IP-based Internet of Things. In: IEEE International Conference on Sensing, Communication, and Networking, pp. 284–292 (2014)
2. Hung, X., Khalid, M., Sankar, R., Lee, S.: An efficient mutual authentication and access control scheme for WSN in healthcare. J. Netw. **6**(3), 355–364 (2011)
3. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science **266**(5187), 1021–1025 (1994)
4. Akiwate, B., Parthiban, L.: A dynamic DNA for key-based cryptography. In: IEEE International Conference on Computational Techniques, Electronics and Mechanical Systems, pp. 223–227 (2018)
5. Rafiul, M., Rokibul, K., Akber, A., Morimoto, Y.: A DNA cryptographic technique based on dynamic DNA encoding and asymmetric cryptosystem. In: International Conference on Networking, Systems and Security, pp. 1–8 (2017)

6. Pradeeksha, A., Sathyapriya, S.: Design and implementation of DNA based cryptographic algorithm. In: IEEE International Conference on Devices, Circuits and Systems, pp. 299–302 (2020)
7. Zebari, D., Haron, H., Zeebaree, S., Zeebaree, D.: Multi-level of DNA encryption technique based on DNA arithmetic and biological operations. In: IEEE International Conference on Advanced Science and Engineering, pp. 312–317 (2018)
8. Chakraborty, R., Rakshit, G., Roy, B.: Enhanced key generation scheme based on cryptography with DNA logic. Int. J. Inf. Commun. Technol. Res. **1**(8), 370–374 (2011)
9. Gehlot, L., Shinde, R.: A survey on DNA-based cryptography. Int. J. Adv. Res. Comput. Eng. Technol. **5**(1), 107–110 (2016)
10. Gogte, S., Nemade, T., Nalawade, P., Pawar, S.: Simulation of quantum cryptography and use of DNA based algorithm for secure communication. J. Comput. Eng. **11**(2), 64–71 (2013)
11. Fu, B., Zhang, Y., Zhang, X.: DNA cryptography based on DNA fragment assembly. IEEE Int. Conf. Inf. Sci. Digital Content Technol. **1**, 179–182 (2012)
12. Abdelkader, H., Ibrahim, F., Moussa, M.: Enhancing the security of data hiding using double DNA sequences. In: Industry Academia Collaboration Conference (2015)
13. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **A8**, 203–209 (1987)
14. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-39799-X_31
15. Aumasson, J., Henzen, L., Meier, W.: QUARK: a lightweight hash. J. Crypt. **26**(2), 313–339 (2013)
16. Vijayakumar, P., Vijayalakshmi, V., Zayaraz, G.: DNA computing-based elliptic curve cryptography. J. Comput. Appl. **36**(4), 1–4 (2011)