Novel Method for DNA-Based Elliptic Curve Cryptography for IoT Devices

Harsh Durga Tiwari and Jae Hyung Kim

Elliptic curve cryptography (ECC) can achieve relatively good security with a smaller key length, making it suitable for Internet of Things (IoT) devices. DNA-based encryption has also been proven to have good security. To develop a more secure and stable cryptography technique, we propose a new hybrid DNA-encoded ECC scheme that provides multilevel security. The DNA sequence is selected, and using a sorting algorithm, a unique set of nucleotide groups is assigned. These are directly converted to binary sequence and then encrypted using the ECC; thus giving double-fold security. Using several examples, this paper shows how this complete method can be realized on IoT devices. To verify the performance, we implement the complete system on the embedded platform of a Raspberry Pi 3 board, and utilize an active sensor data input to calculate the time and energy required for different data vector sizes. Connectivity and resilience analysis prove that DNA-mapped ECC can provide better security compared to ECC alone. The proposed method shows good potential for upcoming IoT technologies that require a smaller but effective security system.

Keywords: Deoxyribonucleic acid (DNA), Elliptic curve cryptography (ECC), IoT security, Kobliz's algorithm, Nucleotide, Public key cryptography, Rivest, Shamir and Adleman (RSA).

I. Introduction

The Internet of Things (IoT) is one of the most recently developed and fastest emerging technologies. The IoT is a network of internet-connected objects that are able to collect and exchange data using embedded sensors. As more and more essential devices are connected, the issue of secure data transfer has been the focus of much research in this environment. Effective, low-complexity real-time encryption systems are essential to meet the growing demands of safe and secure data flow [1].

However, IoT devices have limited power and area of implementation [1]. Therefore, encryption schemes for IoT systems need to be less complicated, and should function with reduced memory usage, which is dependent on the key size. Compared to conventional systems, elliptical curve cryptography (ECC) can statistically provide stronger security with a smaller key size, thus making it a suitable alternative for IoT devices [1]–[13]. The strength of elliptical cryptography relies on the fact that the output of the decryption operation will be ambiguous without a proper authentication key. The focus of attacks on cryptosystems is to determine the authentication key by searching the meaningful or repetitive patterns in the decryption output using different keys. To counter these attacks, existing implementations use different mechanisms such that the outputs of encryption and decryption are nonrepetitive, even if the repetitive input is applied to encryption logic. In a brute-force attack, the attacker tries all of the possible key combinations and looks for meaningful messages in the decrypted text. In this way, the attacker tries to predict the encryption and decryption keys. However, the brute-force approach is a time-consuming process. To determine the encryption and decryption keys in smaller period, other computationally extensive methods have been used for attacks. As the technology evolved, quantum computers

Manuscript received Oct. 5, 2017; revised Feb. 9, 2018; accepted Mar. 27, 2018.

Harsh Durga Tiwari (corresponding author, harshdt@gmail.com) and Jae Hyung Kim (gmhyung@gmail.com) are with the College of Mechatronics, School of Electrical, Electronic & Control Engineering, Changwon National University, Rep. of Korea.

https://doi.org/10.4218/etrij.2017-0220

pISSN: 1225-6463, eISSN: 2233-7326

have been able to solve computation extensive problems in significantly less times. This has made it easier for security breaches to occur in traditional encryption architecture. A possible solution to this scenario is to increase the encryption key size. However, this will also increase the computation complexity and reduce the maximum achievable throughput when the processing power has increased. Various studies [2], [14]–[18] have shown that deoxyribonucleic acid (DNA) mapping-based cryptography is more resilient to crypto attacks in these situations. The data are encoded to form a random pattern prior to encryption in order to improve the encryption strength against attacks. Various schemes have been proposed to achieve data coding prior to encryption [2], [14]–[18].

DNA-based elliptical cryptography, or DNA-based data coding before encryption, has been proposed in existing research works. In these designs, DNA genome sequences are used to assign values to different character sets in a message. The mapping is made random so that a meaningful text sequence is converted to a pseudo-random sequence. This random sequence is then used for encryption. Existing designs have achieved improvements in the strengths of elliptical cryptosystems with DNA-based mapping [2]. Although the existing designs have presented the idea behind DNA mapping, to date, no implementation-friendly mapping procedure has been reported for DNA mapping. The real-time implementation of DNA mapping before encryption and DNA re-mapping after decryption is time consuming and requires significant energy. Hence, an implementation-efficient architecture is essential to realize DNA-based cryptosystems in mobile and IoT-based applications that have battery and processing power restrictions.

This paper explains a novel idea of sorting a selected DNA sequence in order to cipher the text and reassign the DNA encoded message for encryption using ECC. The proposed procedure for DNA mapping recognizes the need for implementation-friendly architecture. It is also important to note that using fixed rules for mapping will reduce the randomness between the text message and DNA mapped values. The rigidity in mapping will defeat the purpose of introducing randomness prior to encryption through DNA mapping. To provide mapping flexibility, the proposed design uses an architecture where the mapped values can be changed without the need for additional operational expense. The Joint Research Centre (JRC) has published a new database, JRC GMO-Amplicons, which contains more than 240,000 DNA sequences appearing in genetically modified organisms (GMOs) [19]. Other organizations have made many other DNA sequences public for research purposes. The length of the DNA sequence and not the actual sequence itself affects the computational complexity of the proposed scheme. Hence, the designated transmitter and receiver can choose any DNA sequence for mapping from among the pool of tens of millions of possible DNA sequences. With more than 106 possible DNA sequences being usable by the transmitter and receiver, there is a small probability that an attacker can predict which DNA sequence is used. Hence, the proposed scheme will be more resilient to attacks when compared with encryption techniques without DNA mapping. The proposed design was implemented and tested in an IoT environment. The simulation and performance results show that the proposed schemes improve the strength of standard elliptical cryptosystems using DNA mapping. The implementation and testing using real-time sensor data verify the applicability of the proposed design in future IoT applications.

The remainder of this paper is organized as follows: Section II outlines the principle behind ECCs, while Section III presents the elliptic curve descriptions of a prime field. Section IV describes the ElGamal ECC algorithm. Section V explains the proposed algorithm for modified implementation of the NAF and conditional Mod function. Then, the implementation methodology is explained in Section VI, and the simulation results are presented in Section VII. Finally, Section VIII concludes the paper.

II. Basics of Elliptic Curve Cryptography

To overcome the drawbacks of low speed, redundancy, and key size, ECC was proposed as an alternative to the established schemes such as the digital signature algorithm (DSA) and Rivest, Shamir and Adleman (RSA) algorithm [2], [8], [9]. ECC is an algebraic-curve-based system that uses elliptical curve points over a finite field. This scheme can be combined with the ElGamal encryption algorithm to secure new and emerging mobile technologies [20]. Depending on the application, elliptic curve cryptosystems are defined for both a prime field and a binary field.

1. Elliptic Curve Arithmetic

The equation for the elliptic curve on a prime field is given as:

$$(y^2) \bmod P = (x^3 + ax + b) \bmod P, \tag{1}$$

where $(4a^3 + 27b^2) \bmod P \neq 0$. All of the elements are integers between 0 and $(P - 1)$; further, all of the arithmetic operations will be performed within the same range, that is, 0 to $(P - 1)$. For improved security, the prime number P is selected from the range of 112–521 bits [21]. From (1) and the range of P, it can be seen that determining the y value on the elliptical curve based on the x value involves large data width addition, subtraction for modulo computation, as well as point-doubling operations based on values of x, a, and b. Even though the plots are not geometrically smooth, the curve arithmetic of point addition and doubling can be applied, as shown in the following sub-sections.

A. Point Addition

Consider two distinct points A and B as

$$A = (x_A, y_A) \quad \& \quad B = (x_B, y_B). \tag{2}$$

Let C be the point addition of A and B, such that

$$C = A + B \ldots C = (x_C, y_C). \tag{3}$$

Coordinates of point C are given as

$$\begin{aligned} x_C &= (\lambda^2 - x_A - x_B) \bmod P; \\ y_C &= (-y_A + \lambda(x_A - x_C) \bmod P), \end{aligned} \tag{4}$$

$$\lambda = ((y_p - y_Q)/(x_p - x_Q)) \tag{5}$$

where λ is the slope of the line joining P and Q.

If $A = -B$, that is, $B = (x_B, -y_B) \bmod P$, then $A + B = O$ where O is a point at infinity.

B. Point Subtraction

Consider two distinct points A and B as

$$A = (x_A, y_A) \ \& \ B = (x_B, y_B). \tag{6}$$

Let S be the point subtraction of A and B, such that

$$S = A - B = A + (-B) \tag{7}$$

where $-B = (x_B, x_B + y_B)$.

C. Point Doubling

Consider a point A such that $A = (x_A, y_A)$, where $x_A \neq 0$. Let there be a point D such that

$$D = 2A \tag{8}$$

where $D = (x_D, y_D)$.

Then, the coordinates for D are given as:

$$\begin{aligned} x_D &= (\lambda^2 - 2x_A) \bmod P; \\ y_D &= (-x_A + \lambda(x_A - x_D)) \bmod P, \end{aligned} \tag{9}$$

$$\lambda = ((3x_A^2 + p_t)/2y_A) \bmod P, \tag{10}$$

where λ is the tangent at point A and p_t, which are the parameters chosen for the elliptic curve.

III. ElGamal ECC Algorithm

ElGamal ECC is a combination of generalized ElGamal encryption schemes with elliptic curve arithmetic. The complete procedure of encryption and decryption is described in Table 1 [22], [23], [24], [25].

IV. DNA Cryptography - Biological Background

DNA is the germ plasma of all life, and is a genetic polymer that is composed of many small nucleotides [18].

Table 1. Generalized ElGamal ECC scheme.

Process	Description
Domain parameter generation	Elliptic curve domain parameters over F_p are defined by the sextuple $T = (p, A, B, G_E, N_G, h)$
Key generation	Private key: V = Random number (1 to N_G) Public key: $\beta = V \cdot G_E$
Message representation on elliptic curve Point	Message as number, m such that $mr < p$ Representing X-coordinate, $x_j = m \cdot r + j$ such that $j \in [0, \omega - 1]$ Calculate $s_j = x^3_j + Ax_j + B$, such that $s_j^{(p-1)/2} = 1 (\bmod p)$ y_i = Quadratic residual of s_j Message will be represented as $P_M = (x_j, y_j)$
Encryption	Transmitter uses the public key, β Select k = Random number (1 to $N_G - 1$) Calculate $P_1 = k \cdot G_E$ and $P_2 = P_M + k_\beta$ Encrypted text $P_C = (P_1, P_2)$
Decryption	Calculate $M_1 = V \cdot P_1$, using receiver's private key V Calculate $P_M = P_2 - M_1$ Message, m, is represented by P_M
Representation of elliptic curve point into message	Calculate $m = ceil(x_j/r)$ Converting m back to message

https://doi.org/10.4218/etrij.2017-0220

Each nucleotide consists of three parts: nitrogenous bases; deoxyribose; and phosphate. DNA is like a biological macromolecule, and is composed of nucleotides. It is well known that DNA molecules consist of two biopolymer strands coiled around each other to form a double helix structure. These two DNA strands are known as polynucleotides as they are composed of simpler units called nucleotides. Each nucleotide is composed of a nitrogen-containing nucleobase: cytosine (C), guanine (G), adenine (A), or thymine (T) with a monosaccharide sugar called deoxyribose and a phosphate group.

V. DNA Cryptography Operation

Generally, three DNA cryptography methods are used as in [14]. They are as follows:

a. Insertion method.
b. Substitution method.
c. Complementary pair approach.

In all of the above approaches, a common method of encoding and decoding is used. The plaintext is converted to binary numbers. Then, these binary numbers are converted to an equivalent DNA nucleotide sequence. Now, one of the DNA-cryptographic methods of [14] is used for encryption or decryption. The encoding and decoding operations are based on the following facts: for DNA, there are four basic units that are encoded into binary in the following manner: DNA nucleotide base binary equivalent adenine (A): 00, thymine (T): 01, guanine (G): 10. cytosine (C): 11. A DNA sequence is taken from a publicly available sequence. Convert the DNA sequence into binary as described earlier. Divide the binary DNA sequence into segments, where each segment contains a randomly selected number of bits greater than 2. Each bit of binary plain text is then inserted at the beginning of a segmented binary DNA sequence. The inserted sequences are then concatenated to obtain an encoded binary sequence. A new fake (not found really, only obtained from operation) binary sequence is obtained by converting the encoded binary sequence into a nucleotide.

VI. Proposed DNA Encoded Elliptic Curve Cryptography Scheme

The procedure for the DNA-based encryption and decryption is shown in Figs. 1 and 2.

Step 1. Sequence Selection: First, the sender chooses a DNA stream from a known organism from the global database. This sequence is common to both the sender and receiver.

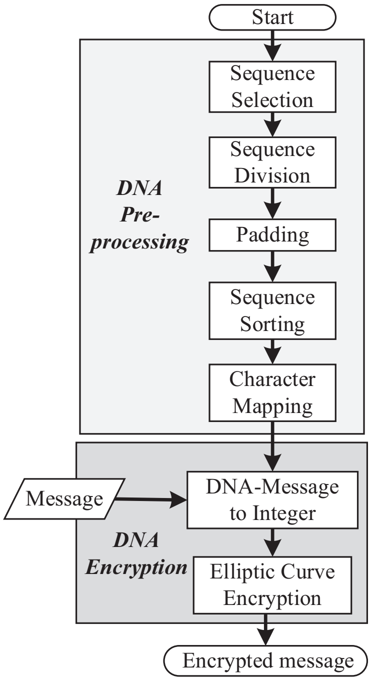

Fig. 1. DNA-based encryption.

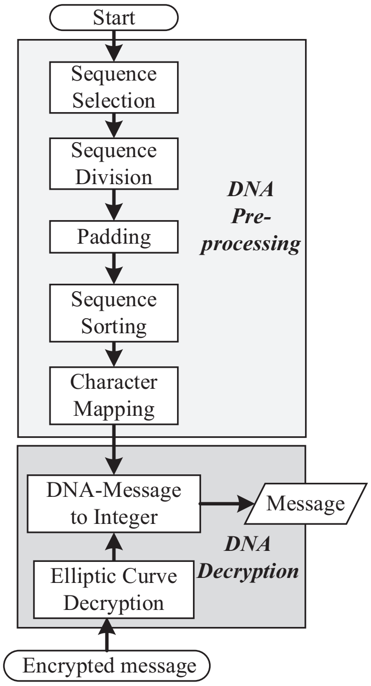

Fig. 2. DNA-based decryption.

Step 2. Sequence Division: Then, the sequence is divided into S size sub-groups.

Step 3. Padding: If the last element after division has fewer elements, then the starting part is padded onto the end.

Step 4. Sequence Sorting: Then, the sequence is sorted into non-repeated subsets. To do this, each subset is first converted to integer values. A, C, G, and T are assigned as 00, 01, 10, and 11, respectively. Assume that if the selected value of S is 3 and the first subset is ATC, then the equivalent integer will be $(001110)_2 = (14)_{10}$. In most of the sequences, the first elements are "*AGT*. . .." Thus, in order to prevent easy deciphering, the first subset is skipped. For sorting, two sets of arrays of length 4^S are made. The first array "*DNAseq*" stores the non-repeated subsets. Another array "*DNApos*" of the same length uses the equivalent subset value to save the position of that subset. If all of the elements are filled or the sorting reaches the last subset, sorting will end. Although this method requires more memory, it has been proven to be faster than conventional methods. In addition, the same algorithm can be used while deciphering the message.

Step 5. Character Mapping: Table 2 shows that depending on the selected value of S, the maximum possible combinations ($Comb_{\text{MAX}}$) can be used. Depending on this, the maximum number of characters ($Char_{\text{MAX}}$) that can be represented by each subset is shown. Table 2 also shows recommendations of possible characters that can be represented ($Char_{\text{MAP}}$). These choices of character sets are then mapped directly to the sorted array. For cryptology purposes, each character is represented by a *DNAseq* equivalent. For implementation, their binary equivalent is used. Then, each set of MESSAGE string is converted to an equivalent "*DNAmess*" using a *DNAseq* stream.

Step 6. DNA-Message to Integer for Encryption: The bit-stream is then divided into n bits. These n bits are then represented as integers and encrypted using ECC, as explained in Table 1.

Step 7. Decryption to DNA-Message: The received data are decrypted using ECC decryption, as described in Table 1.

Step 8. DNA-Message to Message: The receiver will select the same DNA-sequence and perform the sorting algorithm, as described above. The message bits are divided into DNA subsets of S, and converted to equivalent integers. These integer values will use the "*DNApos*" array to remap the characters and form the deciphered message string.

Table 2. Character mapping selection.

S	$Comb_{\text{MAX}}$	$Char_{\text{MAX}}$	$Char_{\text{MAP}}$
1	4	4	Multiple choices, A, B, C, and D
2	16	16	0–9, some special symbols
3	64	64	Plaintext (A–Z, a–z, 0–9, . . .)
4	256	256	All possible ASCII, pixels
5	1,024	1,024	All characters, high-definition data

Step 9. DNA-Message to Integer for Encryption: The bit-stream is then divided into n bits. These n bits are then represented as integers and encrypted using the ECC, as explained in Table 1.

Step 10. Decryption to DNA-Message: The received data are decrypted using ECC decryption, as described in Table 1.

Step 11. DNA-Message to Message: The receiver will select the same DNA-sequence and perform the sorting algorithm, as described above. The message bits are divided into DNA subsets of S, and converted to equivalent integers. These integer values will use the "*DNApos*" array to remap the characters and form the deciphered message string.

VII. Security Performance

In the proposed DNA elliptical cryptography, standard elliptical curve parameters and methods have been chosen. Existing works in elliptical cryptography have shown its resilience towards timing and simple power analysis (SPA) attacks [1]–[9]. The elliptical cryptography produces random data after decryption without a proper authentication key. The focus of attacks on cryptosystems is to determine the authentication key. These attacks succeed when meaningful data are obtained after decryption while trying a random key. However, if the data before encryption is itself encoded to nonrepeating patterns, the entire cryptosystem is more resilient to attacks. Hence, DNA mapping prior to encryption improves the security of the existing elliptical cryptography mechanism. In Section VI, the proposed DNA mapping procedure was explained. It was shown that each character is mapped into a six-bit sequence obtained by DNA mapping. For the proposed DNA mapping system, message characters are mapped using DNA subsets of length S. Every literal in the DNA mapping subset can assume four values (A, C, G, & T). Hence, the total number of permutations possible in a subset of length S and 4 values = 4^S.

During mapping, each value (A, C, G, & T) is mapped to a two-bit binary sequence. Therefore, every value can have 2^2 or 4 possible binary values. Hence, the *possible binary values that every message character can have* = $4 \times 4^S = 4^{S+1}$.

In the proposed system, the subset length, *S*, is 3. Each message character is mapped to a 6-bit binary value, as discussed in Section VI. Hence, for the proposed scheme, the *possible binary values that every message character can have* = 4^{3+1} = 256.

For simplicity of implementation, the DNA-mapped binary set is zero padded to obtain an 8-bit symbol. The location of 2-bit zero padding can be made random. Hence, the number of possible combinations where two zeros can be padded in an 8-bit symbol length = ${}_8C_2$ = 28.

Table 3 shows the possible positions where the zeros, Z_1 and Z_2, can be padded along with 6-bit DNA mapped data, $D_5D_4D_3D_2D_1D_0$, in order to obtain an 8-bit symbol for the DNA mapped data.

Owing to zero padding, the possible binary values that every message character can have increases by a factor of ${}_8C_2$.

Possible binary values that every message character can have with zero padding = $4^{3+1} \times {}_8C_2 = 256 \times 28 = 7{,}168$

With increased data-mapping possibilities, the randomness in data increases significantly. Therefore, using different mapping schemes, repetitive data will be converted to a pseudo-random data. Encrypting a random stream will produce a random stream after decryption. Hence, it is difficult to recover the authentication key by observing the repetitive patterns in decrypted data. Because the data stream obtained after DNA mapping does not resemble a meaningful text message, the decrypted data will also not resemble a meaningful text message. This further discourages the recovery of the authentication key by observing meaningful patterns in decrypted data. Hence, owing to a high degree of randomness, DNA mapping prior to encryption in the proposed DNA-based cryptosystem will be more resilient to attacks than standard elliptical cryptosystems without DNA mapping.

VIII. Simulation and Implementation Results

For simulation purposes, we selected two strands from "*Potato Genome.*" The different associated parameters related to two strands are given in Table 4. We selected *S* as 3, and depending on the selected parameters, the simulation results show a maximum number of unrepeated subsets by size of *DNAseq*. Table 4 also shows a possible character map for characters A–Z, 0–9, and some special characters. Note that for the two selected strands, the character maps are different. Table 4 shows that for the same message vector, the DNA-mapped message vector as per the character mapping. Consequently, it can be seen that if the gnome strand, even with the same organism, the character mapping may be completely different, and thus, the encrypted message will also be different. For the performance evaluation, P192 NIST recommended prime curves are used, while the DNA sequences presented in Table 4 are used. The complete system is implemented using "C," and to check the performance, the software is executed on Cortex-A8 running at 200 MHz. The implementation is based on the method explained in [25]. Figure 3 shows the time required for execution for different message-length vectors using different DNA sequences. Table 5 shows simulation parameters and results. Recently, IoT-based applications have gained significant interest. Data security has been a primary concern for device data sharing through cloud. The testing

Table 3. Possible data symbol patterns after zero padding Z_1 and Z_2 of DNA mapped data $D_5D_4D_3D_2D_1D_0$.

Position of first zero bit, Z_2	Position of second zero bit, Z_1						
	1	2	3	4	5	6	7
2	$Z_1Z_2D_5D_4$ $D_3D_2D_1D_0$	-	-	-	-	-	-
3	$Z_1D_5Z_2D_4$ $D_3D_2D_1D_0$	$D_5Z_1Z_2D_4$ $D_3D_2D_1D_0$	-	-	-	-	-
4	$Z_1D_5D_4Z_2$ $D_3D_2D_1D_0$	$D_5Z_1D_4Z_2$ $D_3D_2D_1D_0$	$D_5D_4Z_1Z_2$ $D_3D_2D_1D_0$	-	-	-	-
5	$Z_1D_5D_4D_3$ $Z_2D_2D_1D_0$	$D_5Z_1D_4D_3$ $Z_2D_2D_1D_0$	$D_5D_4Z_1D_3$ $Z_2D_2D_1D_0$	$D_5D_4D_3Z_1$ $Z_2D_2D_1D_0$	-	-	-
6	$Z_1D_5D_4D_3$ $D_2Z_2D_1D_0$	$D_5Z_1D_4D_3$ $D_2Z_2D_1D_0$	$D_5D_4Z_1D_3$ $D_2Z_2D_1D_0$	$D_5D_4D_3Z_1$ $D_2Z_2D_1D_0$	$D_5D_4D_3D_2$ $Z_1Z_2D_1D_0$	-	-
7	$Z_1D_5D_4D_3$ $D_2D_1Z_2D_0$	$D_5Z_1D_4D_3$ $D_2D_1Z_2D_0$	$D_5D_4Z_1D_3$ $D_2D_1Z_2D_0$	$D_5D_4D_3Z_1$ $D_2D_1Z_2D_0$	$D_5D_4D_3D_2$ $Z_1D_1Z_2D_0$	$D_5D_4D_3D_2$ $D_1Z_1Z_2D_0$	-
8	$Z_1D_5D_4D_3$ $D_2D_1D_0Z_2$	$D_5Z_1D_4D_3$ $D_2D_1D_0Z_2$	$D_5D_4Z_1D_3$ $D_2D_1D_0Z_2$	$D_5D_4D_3Z_1$ $D_2D_1D_0Z_2$	$D_5D_4D_3D_2$ $Z_1D_1D_0Z_2$	$D_5D_4D_3D_2$ $D_1Z_1D_0Z_2$	$D_5D_4D_3D_2$ $D_1D_0Z_1Z_2$

Table 4. Simulation result for DNA-based encryption.

Gnome	StDT43 Glyceraldehyde 3-phosphate dehydrogenase	StDT15 Steroid dehydrogenase
Sequence	ATGGCTAAGGTTAAGATTGGAATTAACGGATTTGGAAGAATTGGCCGATTGGTCGCTCGGGTT GCTCTCCAAAGAGATGATGTTGAACTTGTCGCTGTTAACGACCCCTTCATCTCTGTTG ATTACATGACATATATGTTTAAGTATGATAGTGTACACGGCCAGTGGAAGCATCATGAGCT TAAGGTTAAGGATGAGAAAACCCTTCTCTTTGGTGAGAAGGCTGTTACTGTTTTTGGCT TTAGGAACCCAGAGGAGATTCCATGGGCACAGACTGGAGCTGATTACATTGTGGAGTCCAC GGGTGTTTTCACCGATAAGGACAAAGCTGCTGCTCATTTGAAGGGTGGTGCCAAGAAAGT TATCATTTCTGCTCCTAGCAAGGATGCTCCTATGTTTGTCGTTGGTGTCAATGAGAAGGA ATACAAGCCAGAGCTCAACATTGTTTCAAACGCTAGCTGTACCACAAATTGCCTTGCTCCCT TGGCCAAGGTTATAAATGACAGATTTGGAATTGTTGAGGGTCTCATGACCACAGTCCA CTCCATCACAGCCACTCAGAAGACTGTTGATGGACCATCAGCCAAGGATTGGAGAGGTGGAA GAGCTGCTTCGTTCAACATTATTCCCAGCAGTACTGGAGCGGCCAAGGCTGTTGGGAAAGTGC TACCAGCATTGAATGGAAAGTTAACTGGAATGGCTTTCCGAGTCCCAACTGTTGATGT GTCTGTGGTTGACCTCACAGTGAGGTTGGAAAAAGAAGCTACCTATGATGAAATCAAG GCTGCCATCAAGGAGGAGTCTGAAGGAAAACTGAAGGGAATTCTAGGTTACACTGAAGAT GATGTGGTTTCCACGGACTTTGTGGGAGACAATAGATCAAGCATATTTGATGCCAAGGCT GGAATTGCTTTGAGCAAGAATTTTGTCAAGCTTGTTGCATGGTACGACAATGAATGGGGTTACA GCACAAGAGTGGTGGACTTGATTATTCACATGTCATCAGTTCAGTAA	ATGCATCTGAGCGAAAACGAAGGGATAGAGAAGAAGAGTTTCGTGGTTACAGGTGGACTTGGCTTTATAGGCGCCGCACTTT GTCTTGAACTCGTCAGAAGGGGTGCTCGTCTTGTCAAAGCATTTGATCTTAGAACCCATTCTCCTTGGTCATCTCAACTCCGT CAATATGGGGTTCACCTCATTCAAGGGGATATAACAGACAAACAACATGTCCAGAAAGCACTACAAGGGTCAGATTGTGTTTT CCATCTTGCTTCTTATGGCATGTCGGGGAAAGAAATGCTTCAATATAGTCGTGTTGATGAGGTCAATATTAATGGAACCTGCCACAT TATTGATGCTTGCCTTGACCATCAGATTAAGAGGCTTGTTTATGTCAGCACACCTAATGTTGTTTATGGTGGCAAGGAAATTGT TAATGGCAATGAGAACCTACCTTATTTTCCAATAGATGACCATGTTGATCCATATGGACGAAGCAAATCAATTGCTGAACAATTGG TCCTAAAGAGTAACGGTCGTCCCTTCACTAAGAAGAATGGCAAATGCCTTTACACCTGCGCAATTCGTCCAGCTGCTATATA TGGTCCAGGTGAAGAAAGGCACCTGCCGAGGATCATAACTCTTACAAAGCTAGGTCTGTTTCCCTTTAAAATCGGTTCACCAAAT GTAAAATCAGATTGGGTGTACGTCGACAATCTTGTACTGGCGCTTTTATTGGCTAGCATGGGACTTTTGGATGACATCCCTGGCA GAGAAGGGCTTCCAATTGCTGCTGGTCAACCTTATTTTATATCAGATGGTTCTCCCATCAACAGTTTCGAGTTCCTTCTCCCTTTACT GAAAAGCCTGGATTATGATCTACCAAAGACATCACTAGCTGTCTCTCATGCTCTTCTTCTGGGAAAAATTTTCTGGGCTTTCTACT CCTTTCTGTATCCTTGGTTGAATAGTAGGTGGCTTCCTCAACCTCTGATCCTTCCTGCTGAAGTCTACAAGGTTGGTGTCACTCATTAC TTTTCTTTCCTAAAAGCAAAAGAGGAACTTGGGTACGTCCCAATGGTGAGTTCTAAGGAGGGCATGGCTGCAACTATTGCATATTGG CAAGAAAGAAAACGGAGGAGTTTGGACGGACCTACAATATGGGCATGGCTGTTTTGTGTGATTGGAATGTCGGCATTGTTTG CTGCTGCGTATCTGCCTGATTATGGACCCATTCCCTTTATTAGAGCAGTTCACCTCTTCTTCTTTCGCTCTATGTTGGCATTGAAAGTGA TTTTTGTTCTAGCAGCTGCTGCACACGTAGGTGAAGCTATATATGCATGGAATGTTGCAAAAACAATAGATCCTGCAAATG CAAGAGGGTGGTTTTGGCAGACATTTGCTTTGGGGATATTTTCCCTACGATTACTTCTGAAGAGAGCTAAAAAGTAG
Sequence Length	1,014	1,437
S	3	3
Size of *DNA seq*	62	63

Character Mapping

	StDT43	StDT15		StDT43	StDT15		StDT43	StDT15		StDT43	StDT15		StDT43	StDT15		StDT43	StDT15
	(GGC:41)	(GCA:36)	P	(TCC:53)	(ACA: 4)	!	(TAA:48)	(TCT:55)	Q	(ATG:14)	(TTA:60)	#	(GGT:43)	(GAG:34)	R	(AGC: 9)	(AAT: 3)
$	(GAT:35)	(CGA:24)	S	(GTC:45)	(GCT:39)	%	(TGG:58)	(AAA: 0)	T	(CAC:17)	(TAA:48)	&	(AAT: 3)	(AGG:10)	U	(GGG:42)	(CTG:30)
(	(CGG:26)	(GAT:35)	V	(CAA:16)	(CAC:17)	)	(ATT:15)	(AGA: 8)	W	(AGT:11)	(ACC: 5)	*	(AAG: 2)	(GAA:32)	X	(TTC:61)	(ACG: 6)
+	(CCG:22)	(TTT:63)	Y	(CGT:27)	(CGG:26)	,	(CGC:25)	(CGT:27)	Z	(GCC:37)	(ATG:14)	-	(TCG:54)	(GGT:43)	\	(AAA: 0)	(GCC:37)
.	(TGC:57)	(TAC:49)	^	(CTG:30)	(GTT:47)	/	(TCT:55)	(TGG:58)	_	(TTG:62)	(GTA:44)	0	(CCA:20)	(ACT: 7)	a	(TAT:51)	(AGC: 9)
1	(AGA: 8)	(CTT:31)	b	(GTT:47)	(ATT:15)	2	(TGA:56)	(TAT:51)	c	(GTA:44)	(TAG:50)	3	(TGT:59)	(CGC:25)	d	(TAG:50)	(AAC: 1)
4	(ACT: 7)	(TTG:62)	e	(ACA: 4)	(CCA:20)	5	(CGA:24)	(TGA:56)	f	(GTG:46)	(TTC:61)	6	(CCC:21)	(CAG:18)	g	(GAA:32)	(TCC:53)
7	(CTT:31)	(AAG: 2)	h	(GCA:36)	(GTC:45)	8	(CAT:19)	(GGG:42)	i	(TCA:52)	(ATC:13)	9	(CTC:29)	(TGC:57)	j	(GCT:39)	(TCA:52)
:	(TTA:60)	(TCG:54)	k	(GGA:40)	(CCG:22)	;	(GAC:33)	(TGT:59)	l	(AAC: 1)	(ATA:12)	@	(ATA:12)	(CAA:16)	m	(CCT:23)	(CCT:23)
A	(TAT:51)	(AGC: 9)	n	(TAC:49)	(CAT:19)	B	(GTT:47)	(ATT:15)	o	(TTT:63)	(GGA:40)	C	(GTA:44)	(TAG:50)	p	(TCC:53)	(ACA: 4)
D	(TAG:50)	(AAC: 1)	q	(ATG:14)	(TTA:60)	E	(ACA: 4)	(CCA:20)	r	(AGC: 9)	(AAT: 3)	F	(GTG:46)	(TTC:61)	s	(GTC:45)	(GCT:39)
G	(GAA:32)	(TCC:53)	t	(CAC:17)	(TAA:48)	H	(GCA:36)	(GTC:45)	u	(GGG:42)	(CTG:30)	I	(TCA:52)	(ATC:13)	v	(CAA:16)	(CAC:17)
J	(GCT:39)	(TCA:52)	w	(AGT:11)	(ACC: 5)	K	(GGA:40)	(CCG:22)	x	(TTC:61)	(ACG: 6)	L	(AAC: 1)	(ATA:12)	y	(CGT:27)	(CGG:26)
M	(CCT:23)	(CCT:23)	z	(GCC:37)	(ATG:14)	N	(TAC:49)	(CAT:19)	|	(CAG:18)	(GGC:41)	O	(TTT:63)	(GGA:40)	~	(ACC: 5)	(CCC:21)

Example text	The quick brown fox jumps over a lazy dog.	The quick brown fox jumps over a lazy dog.
DNA mapped	CACGCAACAGGCATGGGGTCAGTAGGAGGCGT TAGCTTTAGTTACGGCGTGTTTTTCGGCGCTG GGCCTTCCGTCGGCTTTCAAACAAGCGGCTAT GGCAACTATGCCCGTGGCTAGTTTGAATGC	TAAGTCCCAGCATTACTGATCTAGCCGGCAAT TAATGGAACCCATGCATTCGGAACGGCATCAC TGCCTACAGCTGCAGGACACCCAAATGCAAGC GCAATAAGCATGCGGGCAAACGGATCCTAC

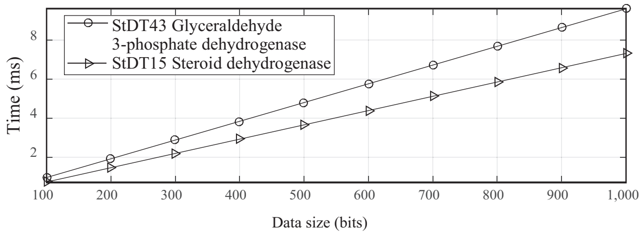

Fig. 3. Execution time for DNA-based encryption-based encryption for different data sizes.

Table 5. Parameters and outputs of encryption program.

Elliptic curve domain parameters	
Prime number (p)	3946183951
Coefficient of elliptic curve equation	$y^2 = x^3 + Ax + B$ A: 537680305 B: 1059676324
Generation point (G_E)	1152222263, 133703258
Order basic point (N_G)	3946206427
Private key (V)	2454757958
Public key (P_B):	3539395206, 802765602
Message (m):	The quick brown fox jumps over a lazy dog

Encrypted message			
Gnome: StDT43		Gnome: StDT15	
727351751	2680084257	727351751	2680084257
1888145626	615128714	3559534884	1169189895
2515192649	2786646526	3809650945	1778448443
2552423891	1777497663	2244436880	2848479238
2719751111	2745441300	2879999780	1331951813
54465015	2883926310	2889938808	3303653882
2649017278	597055342	552500634	606185578
3203311726	3369261069	948175606	2714550074
3230946071	3874467633	1621849346	2749817315

Table 6. Specification of test environment.

Test board	Raspberry Pi 3 model B	Architecture	ARMv8-A (64/32-bit)
SoC	Broadcom BCM2837	CPU	1.2-GHz 64-bit quad-core ARM Cortex-A53
CPU SDRAM	800 M	GPU SDRAM	128 M
Core frequency	250 MHz	ARM frequency	600 MHz
On-board network	10/100 Mbit/s Ethernet, 802.11n wireless, Bluetooth 4.1	Core operating voltage	1.3062 V

of this proposed scheme in the IoT environment was done using Raspberry Pi 3. Table 6 shows the specifications of the test environment. To replicate a real scenario, the test board is powered by a constant-voltage battery source. The test board is connected to an ambient light measurement sensor for capturing the input data for DNA mapping. The mapped data are encrypted and sent to a cloud server. Then, a cloud application loops back the encrypted data to the testing board for decryption. The decrypted data are reverse DNA mapped to obtain the original message. The test is conducted using messages with various data lengths. The DNA mapping and encryption application is 638 KB in size. Additional memory is required for storing the input pattern. Table 7 shows the results obtained for the time consumption, CPU clock cycles, and energy consumption by DNA mapping and encryption with respect to data input size using a Raspberry Pi 3 model B running at 125 MHz. The StDT43 Glyceraldehyde 3-phosphate dehydrogenase sequence mentioned in Table 4 is used for testing in a Raspberry Pi 3 IoT environment.

The decryption and reverse DNA mapping application is 604 KB in size. Additional memory is required for storing the input pattern. Table 8 shows the results obtained for the time consumption, CPU clock cycles, and energy consumption by decryption and DNA remapping with respect to data input size. Figure 4 shows the time consumption for DNA mapping + encryption and decryption + DNA remapping with respect to the data length normalized with respect to the largest input data size. The results show that the time consumption increases linearly with respect to the input. Existing research has already verified that in both encryption and decryption, the point doubling, addition, and shifting operations can be done over time, which is linear to the input data length.

The linear relation between the time consumption and the input of the proposed schemes proves that DNA mapping before encryption and DNA remapping after decryption do not require complex quadratic computations. The energy consumption of the proposed scheme is slightly higher when compared to some of the existing designs [1] owing to DNA mapping and remapping in the proposed scheme. However, despite an increase in energy consumption, the proposed scheme significantly increases the resilience of the elliptical cryptosystem. Figure 5 shows the CPU core cycle consumption due to DNA mapping + encryption and decryption + DNA remapping with respect to the data length normalized with respect to the largest input data size.

Table 7. Performance result obtained for the proposed DNA mapping and encryption implementation.

A (bytes)	B	C (ms)	D	E (mA)	F (mJ)	G (mJ)
84	0.077	0.66716	759,621,740.00	52.494	43.572	32.65
168	0.154	1.274068	1,519,016,211.00	49.989	83.209	
252	0.231	1.910758	2,278,335,505.00	49.153	124.792	
336	0.308	2.547429	3,037,615,831.00	52.720	166.373	
420	0.385	3.191666	3,798,185,350.00	51.500	208.448	
504	0.462	3.824995	4,556,800,095.00	51.400	249.810	
588	0.538	4.472849	5,317,992,747.00	53.524	292.122	
672	0.615	5.112778	6,078,079,664.00	52.049	333.916	
756	0.692	5.751382	6,837,539,803.00	52.088	375.623	
840	0.769	6.415153	7,602,796,324.00	53.297	418.974	
924	0.846	7.047074	8,359,771,242.00	53.027	460.244	
1,008	0.923	7.670317	9,116,480,685.00	51.884	500.948	
1,092	1.000	8.298309	9874730922.00	49.915	541.963	

A: Input data length; B: Normalized input data length; C: Total execution time (DNA mapping + Encryption); D: CPU clock cycles; E: Core current; F: Core energy; G: Energy consumption reported in existing work [1].

Table 8. Performance result of proposed decryption and reverse DNA implementation.

A (bytes)	B	C (ms)	D	E (mA)	F (mJ)	G (mJ)
3,554	0.077	0.235619	280,971,661.00	50.200	15.388	32.65
7,108	0.154	0.471063	561,678,400.00	53.433	30.765	
10,662	0.231	0.706578	842,457,589.00	49.143	46.147	
14,216	0.308	0.94351	1,123,389,734.00	51.450	61.621	
17,770	0.385	1.179981	1,404,182,995.00	49.840	77.065	
21,324	0.462	1.416008	1,685,006,485.00	53.893	92.479	
24,878	0.538	1.650877	1,965,894,705.00	52.563	107.819	
28,432	0.615	1.889159	2,246,928,519.00	51.502	123.381	
31,986	0.692	2.134196	2,531,444,871.00	51.355	139.384	
35,540	0.769	2.36136	2,808,431,685.00	49.298	154.220	
39,094	0.846	2.603626	3,090,534,746.00	52.410	170.043	
42,648	0.923	2.831258	3,369,845,988.00	49.212	184.909	
46,202	1.000	3.067236	3,650,668,692.00	49.357	200.321	

A: Input data length; B: Normalized input data length; C: Total execution time (DNA remapping + Decryption); D: CPU clock cycles; E: Core current; F: Core energy; G: Energy consumption reported in existing work [1].

The linear relation between the time consumption and the input shows the implementation efficiency of the proposed scheme. By avoiding memory stalls, high CPU utilization can be achieved. High CPU utilization helps to reduce time consumption. A smaller time consumption and high CPU utilization makes the proposed schemes suitable for real-time IoT applications with restricted resources.

Figure 6 shows the energy consumption due to DNA mapping + encryption and decryption + D-A remapping

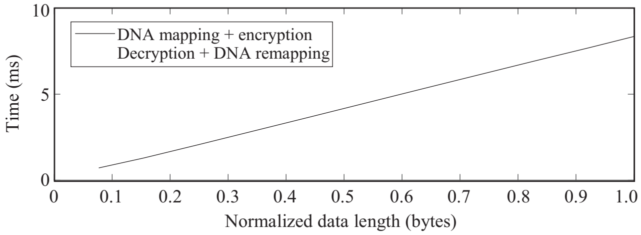

Fig. 4. Time consumption of proposed DNA mapping + encryption and decryption + DNA remapping.

https://doi.org/10.4218/etrij.2017-0220

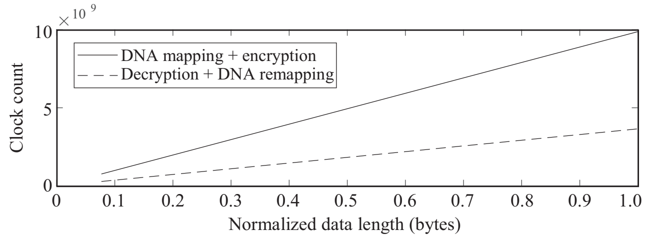

Fig. 5. CPU core cycle consumption due by DNA mapping + encryption and decryption + DNA remapping.

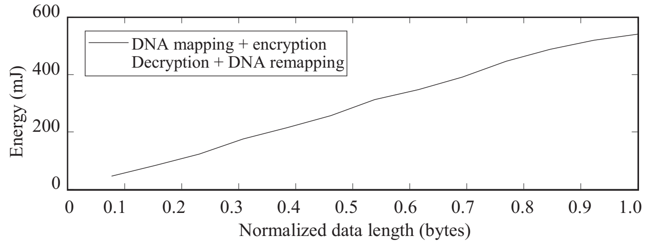

Fig. 6. Energy consumption due to DNA mapping + encryption and decryption + DNA remapping.

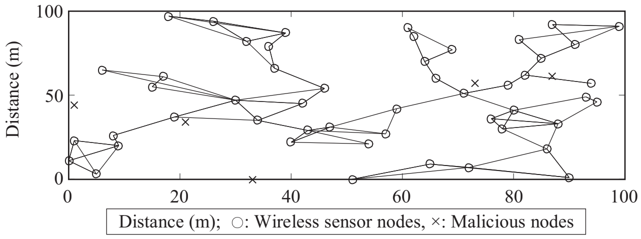

Fig. 7. Node interconnection with ECC-based key pre-distribution.

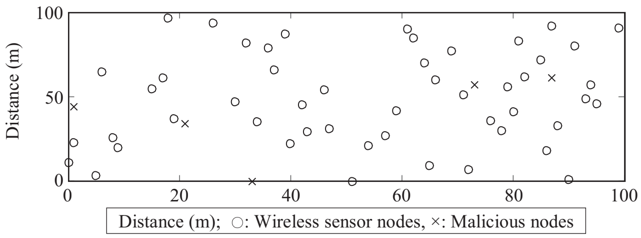

Fig. 8. Node interconnection with ECC-based key pre-distribution with proposed DNA mapping.

with respect to the data length normalized with respect to the largest input data size. The current measured includes the current consumption by peripherals, such as network adapters and display drivers. Various video processing peripherals were disabled to reduce the current consumption. Despite a small deviation in the current, the energy consumption in DNA mapping + encryption and decryption + DNA remapping increases linearly with respect to the data length. The linear relation between the energy consumption and data length is due to the linear relation between the time consumption and data length. The implementation results show that the proposed scheme of DNA mapping and the elliptical cryptosystem was successfully implemented in the IoT environment. The proposed DNA mapping and elliptical cryptosystem require a small amount of memory for storing the application. Hence, the proposed DNA mapping + encryption and decryption + DNA remapping can be efficiently used in IoT applications with limited resources and battery power availability.

IX. Connectivity and Resilience Analysis

1. Connectivity Analysis

In IoT applications, the connectivity between nodes and the resilience of networks to attacks improve by using ECC [26]. DNA mapping along with ECC-based key-pre-distribution in wireless sensor networks (WSNs) further improve the robustness of systems to network-wide attacks. Inter-connectivity graphs show the ability of sensors to form communication links based on pre-distributed keys, as explained in [26]. Using identical sensor location and seed keys assigned to each node, the inter-connectivity in WSNs using ECC-based key pre-distribution with and without the proposed DNA mapping was considered. Figure 7 shows the inter-connectivity graph based on the ECC-based key pre-distribution in WSNs with 50 sensors and five randomly placed malicious sensors.

Figure 8 shows the inter-connectivity graph based on the ECC-based key pre-distribution with proposed DNA mapping in WSNs with 50 sensors and five randomly placed malicious sensors. All 50 sensors use identical DNA sequences for DNA mapping.

Considering the scheme presented in [26], the DNA mapping does not play a role in forming links between sensors. Hence, under identical conditions, the proposed DNA mapping with elliptical cryptography can achieve the same inter-connectivity as a system without DNA mapping-based elliptical cryptography. The connectivity between two nodes depends on the probability that they have at least one common private key [26]. When all the sensors use the same DNA sequence for DNA mapping, the probability that the two nodes share a common key is given as [26].

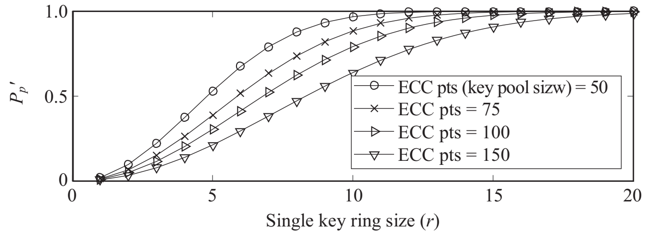

Fig. 9. Inter-connection probability when all sensors use the same DNA sequence for mapping.

$$P_p' = 1 - \{(p - r)!/[(p - 2 \times r)! \times p^r]\}, \qquad (11)$$

where r represents the private keys (key ring size) for each sensor node that is generated, and p is approximately the total number of points in the elliptic curve. Figure 9 shows the probability that each node shares a common key with one other sensor. As the pool size of private keys assigned to each sensor is increased, the probability that each node will share a key with at least one other sensor increases significantly.

If all of the sensor nodes use different DNA sequences for mapping, the connectivity between two nodes depends on the probability that they have at least one common DNA mapping sequence. In this situation, each sensor uses a set of s DNA sequences from a pool of d DNA sequences. The probability that two nodes will use a common sequence for DNA mapping is given as

$$P_d' = 1 - \{(d - s)!/[(d - 2 \times s)! \times d^r]\}. \qquad (12)$$

Figure 10 shows the probability that each node shares a common DNA sequence with one other sensor for mapping. As the pool size of DNA sequences assigned to each sensor is increased, the probability that each node will share a DNA sequence with at least one other sensor increases significantly.

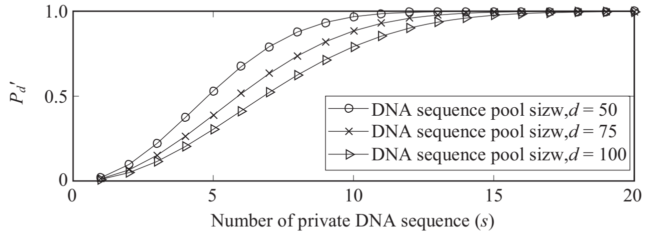

Fig. 10. Inter-connection probability when all sensors use different DNA sequences for mapping.

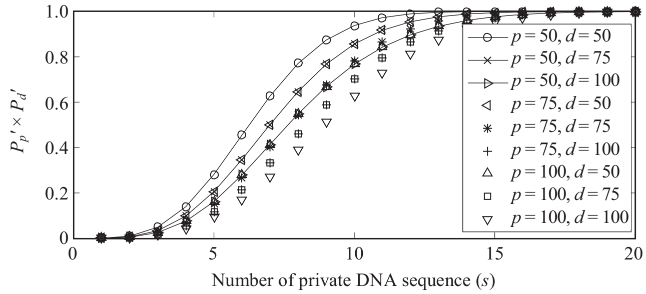

Fig. 11. Inter-connection probability when all sensors use a different DNA sequence to map a different key ring size.

If all of the sensor nodes use a different DNA sequence for mapping and a different private key, the connectivity between two nodes depends on the probability that they have at least one common DNA mapping sequence, and that they have at least one common private key. In this situation, each sensor uses a set of s DNA sequences from a pool of d DNA sequences. The probability that two nodes will use a common sequence for DNA mapping and a common private key is given as

$$\begin{aligned} P' = P_p' P_q' &= [1 - \{(p - r)!/[(p - 2 \times r)! \times p^r]\}] \\ &\times [1 - \{(d - s)!/[(d - 2 \times s)! \times d]\}]. \end{aligned} \qquad (13)$$

Figure 11 shows the probability that each node shares a common DNA sequence for mapping and a private key for encryption with one other sensor. As the pool size with the number of DNA sequences and key is increased, the probability that each node will share a DNA sequence with at least one other sensor increases significantly.

The analysis presented above assumes that the set of DNA sequences assigned to each sensor is chosen randomly from a pool of DNA sequences. Further research on the systematic distribution of DNA sequences to each sensor may simplify the distribution process. At this point, independent analysis of the systematic DNA sequence distribution is beyond the scope of this study.

2. Resilience Analysis

Resilience analysis shows the rate at which the network gets captured as the number of malicious nodes increase. Attacks such as Sybil and brute-force attacks are used by malicious nodes to capture networks. As shown in Fig. 7 and Fig. 8, the set of 50 sensor nodes is simulated when the number of malicious nodes is increased for different attack schemes. In five simulation scenarios, a common DNA sequence was allocated to all of the sensors. When a common DNA sequence is used by the entire network, in four attack scenarios, the attacker knows the DNA sequence, while in one case, the attacker does not know the DNA sequence assigned to the sensors. In one simulation scenario, each sensor is allocated a set of DNA

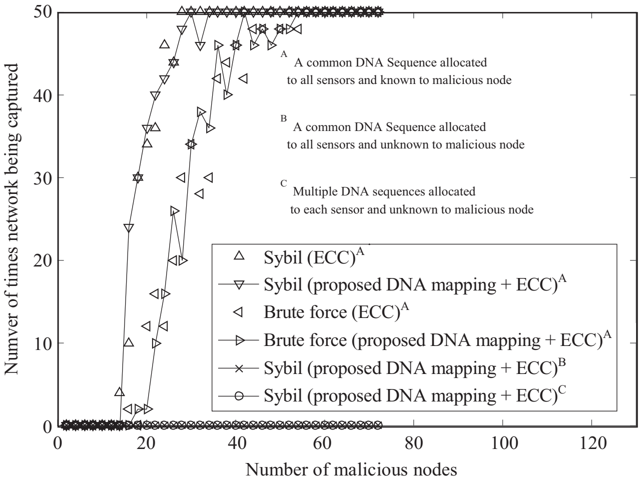

Fig. 12. Resilience analysis (number of nodes = 50) ring size.

sequences that are chosen randomly from a pool of DNA sequences. When a common DNA sequence is allocated to all of the sensors and the attacker knows the DNA sequence used for mapping, the number of times that an entire network is captured depends on the resilience of the key pre-distribution scheme towards different types of attacks. Figure 12 shows the number of times that the network was captured when the number of malicious nodes was increased. In the proposed scheme, the DNA sequences are chosen randomly from several thousand DNA sequences that have been made public for research purposes. There is no method for predicting the DNA sequence used for mapping. In other words, an attacker cannot determine the DNA sequence that is used for mapping by employing any predictive model. Hence, it is not practically possible for the attacker to capture the network without a knowledge of the DNA sequence. When each sensor is assigned multiple DNA mapping sequences, the DNA sequence pool is created by randomly choosing DNA sequences from several thousand DNA sequences that have been made public for research. Each of the DNA sequences in the pool is entirely different from other DNA sequences in the pool. There are currently no methods that can be used to predict which DNA sequences are present in the pool. An attacker cannot determine the entire DNA sequence pool using any predictive model. Hence, it is not practically possible for the attacker to capture the network without the knowledge of DNA sequence. Figure 12 shows that malicious nodes are never able to capture the entire network without the knowledge of the DNA mapping sequence. As previously shown, DNA sequences are text strings that comprise “A,” “C,” “G,” and “T” characters. In IoT applications, the inclusion of the DNA sequence will marginally increase the memory requirement. However, the resilience analysis shows that the proposed DNA mapping significantly improves the resistance of elliptical cryptography-based application towards unwarranted attacks.

X. Conclusion

In this paper, a novel DNA mapping procedure is presented for ECC. The DNA sequences are sorted, and non-repeated subsets are assigned to characters. The mapped characters are used for encryption and decryption. The security analysis of the proposed scheme shows that using the proposed DNA mapping, existing elliptical cryptosystems are more resilient to timing and SPA attacks. The proposed system was successfully implemented and ported on the real-time IoT device. The time required for the proposed DNA mapping + encryption and decryption + DNA remapping is linear with respect to the input data length. The energy consumption of the proposed scheme is near to the energy consumption of existing systems without DNA mapping and remapping. Hence, the proposed DNA mapping and remapping system improves the strength of existing elliptical cryptosystems without excessive energy and time consumption. Owing to the small energy consumption and verified implementation in an IoT environment, the proposed system has good potential for use in mobile and cloud-based applications.

Acknowledgements

This work was financially supported by the Changwon National University in 2017–2018.

References

[1] Z. Liu, X. Huang, Z. Hu, M.K. Khan, H. Seo, and L. Zhou, “On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age,” *IEEE Trans. Depend. Sec. Comput.*, vol. 14, no. 3, May–June, 2017, pp. 237–248.

[2] N.H. UbaidurRahman, C. Balamurugan, and R. Mariappan, “A Novel DNA Computing Based Encryption and Decryption Algorithm,” *Procedia Comput. Sci.*, vol. 46, 2015, pp. 463–475.

[3] M. Luo, X. Zhou, L. Li, K.K.R. Choo, and D. He, “Security Analysis of Two Password-Authenticated Multi-key Exchange Protocols,” *IEEE Access*, vol. 5, Apr. 2017, pp. 8017–8024.

[4] F. Al-Turjman, Y.K. Ever, E. Ever, H.X. Nguyen, and D.B. David, “Seamless Key Agreement Framework for Mobile-Sink in IoT Based Cloud-Centric Secured Public Safety

Sensor Networks," *IEEE Access*, vol. 5, Apr. 2017, pp. 24617–24631.

[5] A.G. Reddy, E.J. Yoon, A.K. Das, V. Odelu, and K.Y. Yoo, "Design of Mutually Authenticated Key Agreement Protocol Resistant to Impersonation Attacks for Multi-server Environment," *IEEE Access*, vol. 5, Apr. 2017, pp. 3622–3639.

[6] C.S. Park, "A Secure and Efficient ECQV Implicit Certificate Issuance Protocol for the Internet of Things Applications," *IEEE Sens. J.*, vol. 17, no. 7, Apr. 2017, pp. 2215–2223.

[7] N. Li, D. Liu, and S. Nepal, "Lightweight Mutual Authentication for IoT and Its Applications," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 4, Oct. – Dec. 2017, pp. 359–370.

[8] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi, "Securing the Internet of Things in a Quantum World," *IEEE Commun. Mag.*, vol. 55, no. 2, Feb. 2017, pp. 116–120.

[9] V. Odelu, A.K. Das, K.K.R. Choo, N. Kumar, and Y. Park, "Efficient and Secure Time-Key Based Single Sign-On Authentication for Mobile Devices," *IEEE Access*, vol. 5, Apr. 2017, pp. 27707–27721.

[10] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "IoT Applications on Secure Smart Shopping System," *IEEE Internet Things J.*, vol. 4, no. 6, Dec. 2017, pp. 1945–1954.

[11] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption," *IEEE mbedded Syst. Lett.*, vol. 9, no. 1, Mar. 2017, pp. 1–4.

[12] A.G. Reddy, A.K. Das, E.J. Yoon, and K.Y. Yoo, "A Secure Anonymous Authentication Protocol for Mobile Services on Elliptic Curve Cryptography," *IEEE Access*, vol. 4, 2016, pp. 4394–4407.

[13] M.S. Hossain, G. Muhammad, S.M.M. Rahman, W. Abdul, A. Alelaiwi, and A. Alamri, "Toward End-to-End Biometrics-Based Security For IoT Infrastructure," *IEEE Wireless Commun.*, vol. 23, no. 5, Oct. 2016, pp. 44–51.

[14] K. Kainth and G. Singh, "A Review to an Invincible Cryptographic Approach: DNA Cryptography," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 1, Jan. 2015, pp. 327–331.

[15] P. Barman and B. Saha, "An Efficient Hybrid Elliptic Curve Cryptography System with DNA Encoding," *Int. Res. J. Comput. Sci. (IRJCS)*, vol. 2, no. 5, 2015, pp. 33–39.

[16] P. Vijayakumar, V. Vijayalakshmi, and G. Zayaraz, "DNA Computing based Elliptic Curve Cryptography," *Int. J. Comput. Applicat.*, vol. 36, no. 4, Dec. 2011, pp. 18–21.

[17] P. Vijayakumar, V. Vijayalakshmi, and G. Zayaraz, "Enhanced Level of Security Using DNA Computing Technique with Hyperelliptic Curve Cryptography," *ACEEE Int. J. Netw. Sec.*, vol. 4, no. 1, 2013, pp. 1–5.

[18] R. Bama, S. Deivanai, and K. Priyadharshini, "Secure Data Transmission Using DNA Sequencing," *IOSR J. Comput. Eng. (IOSR-JCE)*, vol. 16, no. 2, Mar.–Apr. 2014, pp. 19–22.

[19] European Commission. http://gmo-crl.jrc.ec.europa.eu/jrcgmoamplicons/

[20] N. Koblitz, "Elliptic Curve Cryptosystems," *Math. Comput.*, vol. 48, 1987, pp. 203–209.

[21] Certicom Corp., "Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters," Version 1.0, Certicom, Sept. 2000.

[22] D. Xu and W. Chen, "3G Communication Encryption Algorithm Based on ECC-ElGamal," *Int. Conf. Signal Proc. Syst.*, Dalian, Chian, July 5–7, 2010, pp. V3-291–V3-293.

[23] S. Sutikno, A. Surya, and R. Effendi, "An Implementation of El Gamal Elliptic Curves Cryptosystems," *Proc. IEEE Asia-Pacific Conf. Circuits Syst.*, Chiangmai, Thailand, vol. 24–27, 1998, pp. 483–486.

[24] W. Khudri and M. Sutanto, "Implementation of ELGamal Elliptic Curve Cryptography Using Matlab," *Int. Conf. Instrum., Commun. Inform. Technol. (ICICI) Proc.*, Bandung, Indonesia, Aug. 2005, pp. 1–6.

[25] H.D. Tiwari and Y.B. Cho, "Reduced Modulo Function Implementation for Elliptical Curve. Cryptography for Mobile Devices," *Int. Conf. Telecommun. Technol. Applicat.*, Jeju, Rep. of Korea, Apr. 2014, pp. 1–6.

[26] K. Rajendiran, R. Sankararajan, and R. Palaniappan, "A Secure Key Predistribution Scheme for WSN Using Elliptic Curve Cryptography," *ETRI J.*, vol. 33, no. 5, Oct. 2011, pp. 791–801.

Harsh Durga Tiwari received the B.Tech. degree in electronics engineering from Nagpur University, India, in 2008. He received the Master's degree in electronics, information, and communication engineering, from Konkuk University, Seoul, Rep. of Korea. He is pursuing his Ph. D. degree in Control and Instrumentation Engineering from the College of Mechatronics, Changwon National University, Rep. of Korea.

Jae Hyung Kim was born in Seoul, Rep. of Korea. He received the B.S. and M.S. degrees in electronics engineering and the Ph.D. degree in communication engineering from Korea University, Seoul, Rep. of Korea, in 1983, 1985, and 1989, respectively. Since 1991, he has been with Changwon National University, Rep. of Korea, where he is currently a Professor in the School of Electrical Electronics and Control Engineering. His current research interests include wireless modem design and implementation, focusing especially on compact MIMO system design and telemetry system design.