IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 15, 2020, accepted August 26, 2020, date of publication September 1, 2020, date of current version September 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021055

An Efficient Generation and Security Analysis of Substitution Box Using Fingerprint Patterns

ÖZNUR ŞENGEL[1], MUHAMMED ALİ AYDıN[2], AND AHMET SERTBAŞ[2]
[1]Computer Engineering Department, İstanbul Kültür University, 34158 İstanbul, Turkey
[2]Computer Engineering Department, İstanbul University-Cerrahpaşa, 34320 İstanbul, Turkey

Corresponding author: Öznur Şengel (o.sengel@iku.edu.tr)

ABSTRACT Information and its security have attracted the research community in recent years with increasing usage of mobile applications. Mobile devices have different security options in data transmission such as reading some biometric values. The keystone of the modern block and stream ciphers is the use of a substitution box (s-box) that obscures correlation between plaintext and ciphertext. In this study, we proposed a novel s-box generation algorithm by using the fingerprint pattern of the person who transfers information to the target. We generated several s-boxes by using bifurcation and ridge ending features of the fingerprint. Proposed s-boxes are compared with several known s-boxes over nonlinearity, bijectiveness, strict avalanche criterion, bit independence criterion, linear probability, and differential probability. Along with these properties, we analyzed confidence interval and randomness properties of new s-boxes as well. Also, the execution time of the proposed s-box generation algorithm is calculated and examined. The results of the cryptographic properties have shown that the proposed s-boxes by using ridge ending of the fingerprint performs better. The performances analysis show that the proposed s-box has satisfactory results according to the results of chaotic-based s-boxes. On the other hand, the fingerprint s-boxes are much better than the existing biometric s-boxes according to the s-box security metrics. The results have shown that the execution time of the proposed s-box generation algorithm is more minimum than the existing biometric s-box generation algorithms. Resulting from applying fingerprint biometric data to generate an s-box, such a successful algorithm is promising to be used in mobile devices.

INDEX TERMS Biometric data, cryptography, fingerprint, mobile devices, substitution box.

I. INTRODUCTION

Nowadays, information and its security are hot topics all over the world. Information security is based on good communication. It means that the processes between sender and receiver are designed to deliver information over uninterrupted, reliable, and accurate channels. If the channel is not reliable, the system must protect the information by using a cryptographic technique before the information is sent over that unreliable channel. Modern cryptography is generally divided into two categories: symmetric cryptography that has only one secret key and asymmetric cryptography that has a key pair and is based on the explicit key principle. In this study, Advanced Encryption Standard (AES), that was published in 2001 by the National Institute of Standards and Technology (NIST), has been adopted by the US government and other countries in the world to protect confidential data and information [1], is used.

Substitution box (s-box) is a vital part of block cipher since commercial computer cryptography [2] was introduced by Horst Feistel. An s-box includes 2^n elements, n is the number of bits in input. If all elements in the s-box are unique and different, it is called a crypto s-box. There must not be a correlation between the sequence of the values to generate a good s-box [3]. Rijndael algorithm [4], [5] uses Nyberg's s-box [6] that is constructed with inverse mapping based on finite fields.

The strength of the AES algorithm is related to the s-box and mix column operation. These two elements ensure security against the linear and differential attacks. There is a lot of scientific research focus on the key generation [7], [8], the round operations of algorithm [9], [10], an s-box design [11], [12], and the mathematical operations of the AES algorithm to get a more robust system. There are

The associate editor coordinating the review of this manuscript and approving it for publication was Zahid Akhtar.

lots of studies and papers that improve the algorithm's key and s-box with either biometric traits [8], [13]–[19] to get unique data or machine learning techniques [20]–[22] to get a function with a sequence of mathematical operations.

In this article, to address generating with minimum time consumption, a more secure, and person specific s-box problem, a new biometric s-box is proposed. Our main contribution is developing a personal s-box with personal biometric traits. The most person specific biometric data is fingerprint that has good entropy with its unique pattern. In this study, fingerprint biometric data have been used to get a good source of randomness. The user can add their fingerprint pattern while using their mobile devices to transfer data to the target. The logical and mathematical processes are applied to generate values from the fingerprint patterns. Feature extraction from a fingerprint takes time and time complexity is vital for mobile applications. When any modification is applied to the system, it must be completed with minimum time consumption, otherwise it could not be acceptable [23]. Another signifint contribution of this study is that the proposed s-box generation algorithm is completed within a few milliseconds with high privacy. The proposed algorithm has minimum time consumption, great success in randomization and personal sequence. To improve the shortcomings of existing s-box construction methods, this article presents a novel and person specific s-box construction method by using strong characteristics of biometric data.

The rest of the paper is organized as follows: the different kinds of existing s-boxes are introduced in Section 2. The proposed s-box and inverse s-box generation algorithms are presented in Section 3. Section 4 introduces biometric parameters and explains why fingerprint biometric is preferred. The results of the good s-box criteria and performance assessments are presented in the same section. Moreover, the proposed fingerprint s-box generation algorithm is compared with existing s-boxes in literature according to the good s-box criteria in Section 5. Finally, the conclusion summarizes the proposed fingerprint s-box generation algorithm and presents comments and discussions.

II. RELATED WORKS

Information security and privacy are hot topics in the information technology industry, finance and banking, and scientific research. All applications for these fields need authentication to verify the users. There are different kinds of verifications such as password, biometric traits that are stored in servers of the applications. The transfer of information is protected by using cryptographic algorithms. The more the information is important, the more the application is strong. Researchers develop different parts of the cryptographic algorithms such as key operation, rounds operation, s-box operation because using the only cryptographic algorithm does not ensure security. One of the most important operations for developing a cryptographic algorithm is generating the s-box. There are different methods to produce an s-box such as heuristic techniques, finite field inverse, finite field exponent, pseudo-random [3].

The chaotic map has a spread spectrum to increase the performance of random number generators in heuristic techniques. There are a significant number of published chaotic s-boxes. Chen [24] presented an efficient algorithm that is based on chaotic maps and simulated annealing to obtain an 8 × 8 s-box. Çavuşoğlu *et al.* [25] designed the random number generation algorithm by using the new scaled Zhongtang chaotic system to generate a complicated and dynamic s-box. Lambic [26] proposed a discrete chaotic map based on the composition of permutations. The 3-D four-wing autonomous chaotic system is used by [27] to generate an s-box. The Gingerbreadman chaotic map and S_8 permutations are synthesized by [28] to present resilient nonlinear mechanisms. Lambic [29] presented an s-box with low complexity and large key space and applied composition of operations on existing s-boxes.

Researchers focused on the genetic algorithm that is the best in complex multi-dimensional search space to generate chaotic s-boxes. The chaotic logistic and chaotic tent map are iterated to generate the initial population of the genetic algorithm [30]. The logistic map is used to generate the initial s-box and the three-dimensional chaotic Lorenz system is applied to generate the control parameters of the genetic algorithm [31]. The s-box generation problem transformed the travelling salesman problem and an s-box designed based on the chaotic map and the genetic algorithm [32].

There are various optimization approaches to design an efficient s-box. Ahmad *et al.* [33] proposed an ant colony optimization-based scheme to design an s-box by iterating the chaotic logistic map and chaotic tent map for initialization. The differential uniform and nonlinearity are considered as the fitness function in the optimization step of the algorithm [34], [35]. A chaotic s-box is generated by using the six-dimensional hyper-chaotic map and artificial bee colony optimization algorithm [34]. Many s-boxes are generated by using the intertwining logistic map and then the bacterial foraging optimization algorithm is applied to find the optimal s-box [35]. The travelling salesman problem and piece-wise linear chaotic map are synthesized by [36]. The chaotic map and new Teaching–Learning-Based Optimization (TLBO) are presented to optimize keys generated as a result of round [37]. An s-box is generated by using the discrete space chaotic map and the firefly algorithm optimized the initial s-box [38]. According to ancient Chinese I-Ching philosophy, there are three innovative I-Ching operators (ICOs): intrication, turnover, and mutual operators are used to generate an s-box [39].

The cryptographic primitives and the properties of the chaotic system share unique characteristics that are unpredictable operations and random data. Ullah *et al.* [40] constructed an s-box with the chaotic system and linear fractional transformation and Ullah *et al.* [41] constructed an s-box with arithmetic background based on group action of projective general linear group on units of finite local ring.

Özkaynak [42] designed two s-boxes by applied the chaotic Chen system and chaotic Henon map, and Özkaynak [43] used the chaotic logistic map and chaotic sine map with the properties of existing iris dataset as initial condition and control parameters to improve randomness. A six-dimensional fractional Lorenz-Duffing chaotic system and O-shaped path scrambling algorithm are offered to construct an s-box [44]. The five-dimensional hyperchaotic system [45] and four-dimensional hyperchaotic Lorenz system [46] is discussed to construct an s-box. The one-dimensional discrete chaotic map and β-hill climbing search technique are introduced to construct an s-box [47].

A mathematical operation named as Determinant Rotation is introduced by [48] to produce different s-boxes for each round of AES. Ahmad and Malik [49] proposed the chaos-based neural network with four layers; input layer with eight neurons, two hidden layers with four neurons, two neurons respectively and output layer with one neuron. The sequence of the corresponding multiplicative inverse in GF(2^8) and artificial neural network with seven perceptron neurons in three layers operations are synthesized to generate an s-box [50].

The standard s-box is constructed by using $11B_{16}$ (x^8+ x^4+ x^3+ x + 1) irreducible polynomial in finite field of GF(2^8) and additive constant 63_{16}. An irreducible polynomial is not a constant polynomial that cannot be a product of two non-constant polynomials and ends with a constant one. The affine transformation is used to construct the AES s-box with an affine matrix that should be nonsingular. Approximately 2^{63} affine matrices can be generated with each irreducible polynomial according to [51] and they generated fifty s-boxes over $17B_{16}$, $1BD_{16}$, $14D_{16}$, 165_{16} with maximum avalanche criteria. Some irreducible polynomials with different affine matrices are compared by [52]. A system can use different irreducible polynomials every time to obtain a ciphertext and this irreducible polynomial is sent with a secret key to the receiver to obtain a plaintext [53]. Several irreducible polynomials are paired with the valid additive constants to generate a secure s-box [54]. The level of security with cryptographic properties of s-boxes with all irreducible polynomials is evaluated and the polynomial 163_{16} has an extremely good result according to the s-box security metrics [55].

The Deoxyribose Nucleic Acid (DNA) cryptography [56] is also a hot topic in generating an s-box. The DNA based s-box inspired by the strands of DNA that is a sequence of nucleotides. There are four nucleic acid bases: adenine (A), cytosine (C), guanine (G), thymine (T) to form a DNA sequence. Each nucleic acid base is represented with two-bits that are 00, 01, 10, 11 [57]. The DNA based s-box is generated in four steps: generating DNA strands, reverse complement, XOR operation, central dogma operation [58]. Two DNA strands are presented: first one is for the value of s-box, second one is for the location of the value in s-box [59]. Either logical operations or arithmetic operations are used to generate a DNA s-box. Data processing, DNA addition, DNA subtraction, the logical and arithmetic operations, and searching operation are applied to generate both s-box and inverse s-box by [60]. The RNA based multi s-boxes [61] with secret key initialization, inspired by the DNA based s-boxes [59], [60]. Data translation, addition, subtraction, XOR operation, and transcription processes are applied to generate an RNA s-box from a DNA strand [61].

III. PROPOSED ALGORITHM

In this study, a more robust s-box designed against the linear and differential attacks while minimizing input-output transformation correlation and difference propagation probability.

A. SUBSTITUTION BOX

Substitution Box (s-box) is the most important variable of the symmetric key encryption algorithm in block cipher algorithms. S-box is the only nonlinear part of the algorithm so constructing the best s-box affects the complexity of the ciphertext. The aim of using an s-box is to find a byte change in an algorithm and obscure the relationship between the ciphertext and the key.

The y = S(x) function with an 8-bit input (x) and an 8-bit output (y) as polynomials over GF(2^8) are used by the sub byte operation of the AES algorithm. The least significant nibble of the input represents the column of the s-box, the most significant nibble of the input represents the row of the s-box. The output value is the intersection of row and column on an s-box. If the input is 01011110, the least significant nibble of input is 1110 that has a value of 14 (0xE), and the most significant nibble of input is 0101 that has a value of 5 (0 × 5). The result of the function for the AES s-box is 01011000 (0 × 58). If the output value of the s-box is used as the input for the inverse s-box, the result will be the input value of the s-box.

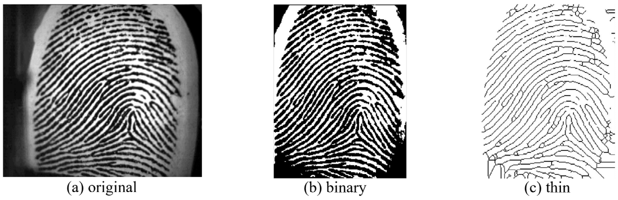


FIGURE 1. Fingerprint processing.

B. PROPOSED S-BOX AND INVERSE S-BOX GENERATION ALGORITHM

In this study, an s-box and inverse s-box are generated using fingerprint patterns of people. The general block diagram of the proposed substitution box generation algorithm consists of the four main stages as illustrated in Fig. 2. The first stage is for extracting features of a fingerprint. The fingerprint of persons can be collected from a fingerprint reader of a mobile device (Fig. 1(a)). If the user uploads the mobile application, the fingerprint is uploaded by the fingerprint reader of the device. The fingerprint in binary format (Fig. 1(b)) is obtained from Fig. 1(a) after binary conversation and size reduction to remove redundant space. The fingerprint processing step includes thinning the lines of fingerprint as shown in Fig. 1(c).

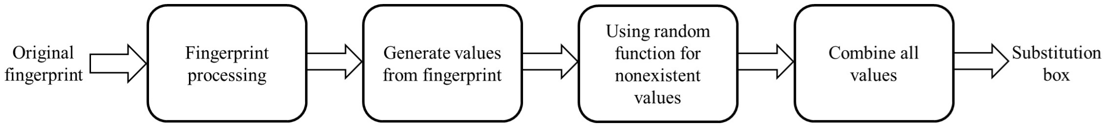

Original fingerprint
Fingerprint processing
Generate values from fingerprint
Using random function for nonexistent values
Combine all values
Substitution box

FIGURE 2. General block diagram of the proposed substitution box generation algorithm.

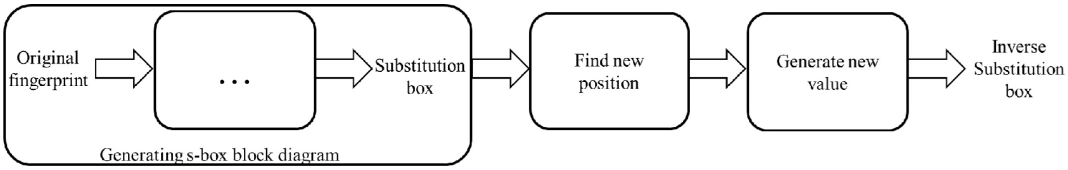


FIGURE 3. General block diagram of the proposed inverse substitution box generation algorithm.

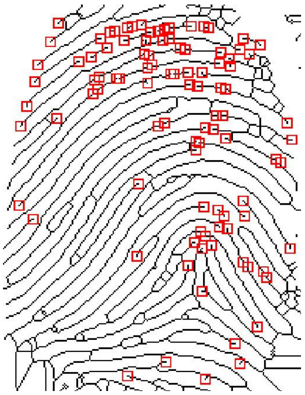


FIGURE 4. Ridge ending points.

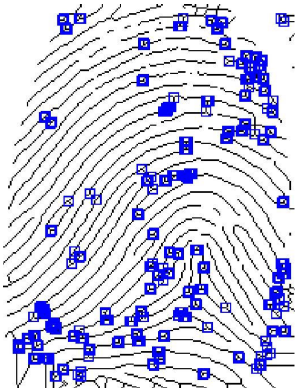


FIGURE 5. Bifurcation points.

Each pixel of the fingerprint is scanned. There is a line if both the selected pixel and the adjacency pixel of the selected pixel is not empty. The system scans all the adjacency pixels of the selected pixel and counts the lines. If the number of lines is one, it is labeled as ridge ending. Ridge ending points of the fingerprint are shown in Fig. 4. If the number of lines is three, it is labeled as bifurcation. Bifurcation points of the fingerprint is shown in Fig. 5.

The second stage generates the values of s-box after some operations from the fingerprint features. The system uses either one of these features or a combination of these features to generate an s-box. The position values of the feature points are stored as x-coordinate and y-coordinate. The system performs the exclusive disjunction (XOR) operation on the x-coordinate and y-coordinate of the selected feature. The result of the calculation is checked for duplicate values. If the result value exists more than one time in the result array, the system stores duplicate values once and removes the others from the result array.

The third stage is for using a random function for nonexistent values of s-box. An s-box has values between 0 and 255 (0xFF). The system checks that each value of the result array is between 0 and 255. If the values between 0 and 255 do not exist in the result array, the system stores this value in a nonexistence array. The values in the nonexistence array are assigned the rest of the result array with random permutation.

In the fourth and final stage, the result array values fill the s-box.

Each s-box has a unique inverse s-box that is used to decrypt the ciphertext. The general block diagram of the proposed inverse substitution box generation algorithm consists of four main stages from the proposed s-box generation algorithm and extra two stages as illustrated in Fig. 3. The first addition stage reads a value from the s-box to find the position of the inverse s-box. The value of the s-box has two digits in hexadecimal format. The left-hand digit represents the row of the inverse s-box. The right-hand digit represents the column of the inverse s-box. The second addition stage reads the position of the selected value from the s-box to generate the value of the inverse s-box. The row position of the s-box value is the left-hand digit of the value of the inverse s-box. The column position of the s-box value is the right-hand digit of the value of the inverse s-box. All s-box values are read to generate an inverse s-box.

An overview of the proposed algorithm is presented in Algorithm 1 given in the Appendix, which consists of these eleven steps:

Step 1: The original input data is a fingerprint that is read by a biometric fingerprint reader, stored as an image file F.

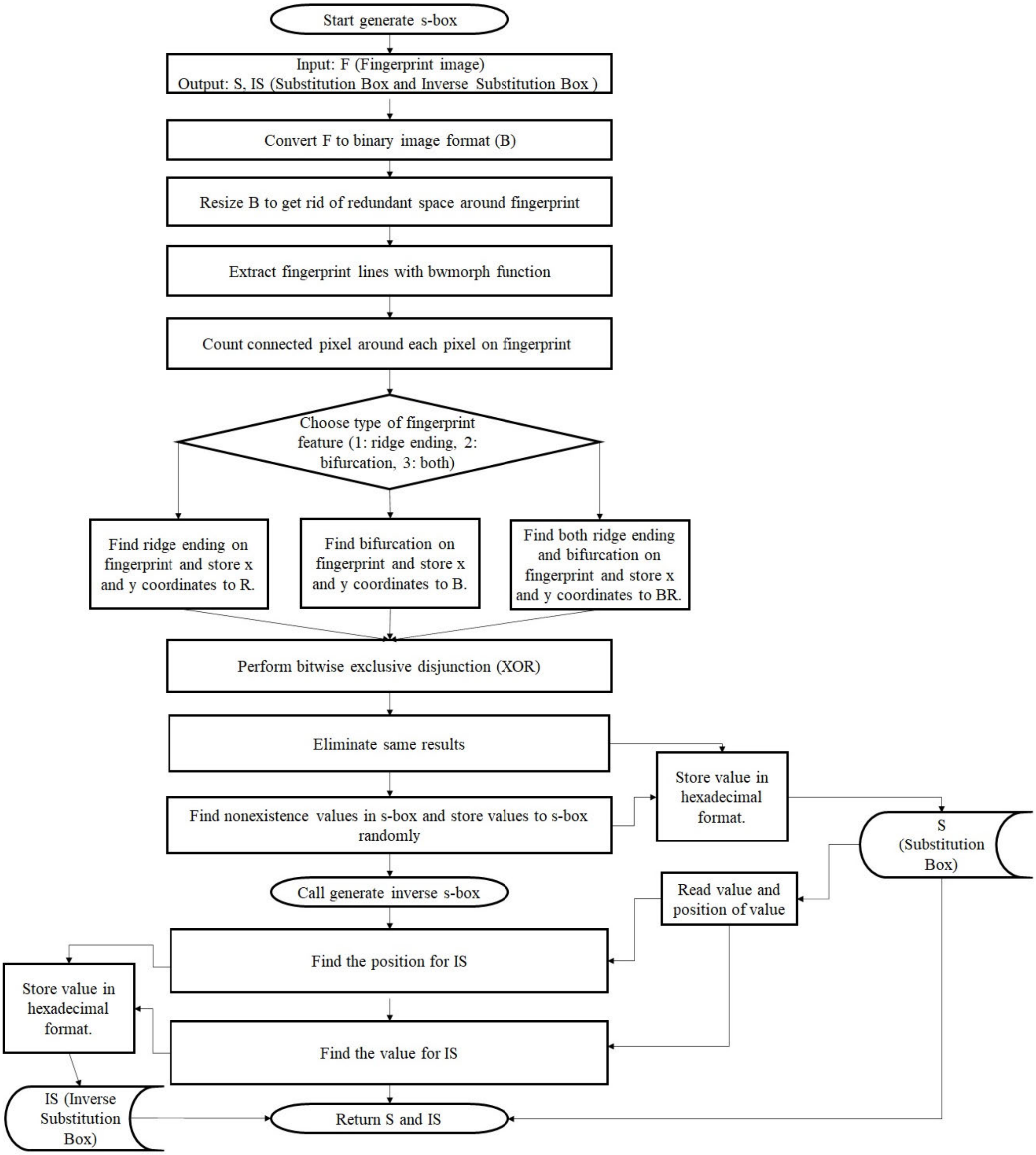

Start generate s-box
Input: F (Fingerprint image)
Output: S, IS (Substitution Box and Inverse Substitution Box)
Convert F to binary image format (B)
Resize B to get rid of redundant space around fingerprint
Extract fingerprint lines with bwmorph function
Count connected pixel around each pixel on fingerprint
Choose type of fingerprint feature (1: ridge ending, 2: bifurcation, 3: both)
Find ridge ending on fingerprint and store x and y coordinates to R.
Find bifurcation on fingerprint and store x and y coordinates to B.
Find both ridge ending and bifurcation on fingerprint and store x and y coordinates to BR.
Perform bitwise exclusive disjunction (XOR)
Eliminate same results
Store value in hexadecimal format.
Find nonexistence values in s-box and store values to s-box randomly
S (Substitution Box)
Call generate inverse s-box
Read value and position of value
Find the position for IS
Store value in hexadecimal format.
Find the value for IS
IS (Inverse Substitution Box)
Return S and IS

FIGURE 6. Flowchart of proposed algorithm.

Step 2: Fingerprint image is converted to binary image format B.

Step 3: Binary image is resized to get rid of redundant space around the fingerprint and is stored as resized image file R.

Step 4: The fingerprint lines (T) are extracted by using bwmorph function from R.

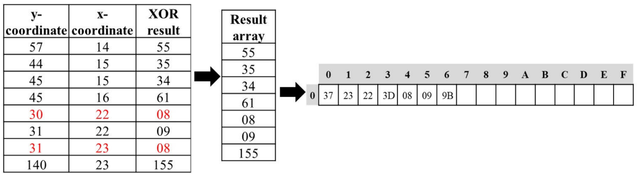
Step 5: The number of connections is counted around each pixel of T and stores in count matrix C.

Step 6: The type of the fingerprint feature is chosen (1: ridge ending, 2: bifurcation, 3: both). If the type is 1, the location of the pixel where C(x,y) value is equal to 1 is stored in R. If the type is 2, the location of the pixel where C(x,y) value is equal to 3 is stored in B. If the type is 3, the location of the pixel where C(x,y) value is either equal to 3 (for bifurcation) or equal to 2 (for ridge ending), is stored in BR.

Step 7: The algorithm performs the bitwise XOR operation on the pixel components (x-coordinate and y-coordinate) of the features. The x-coordinate and the y-coordinate are in decimal format. The result of the XOR operation is in bitwise binary format as shown in Table 1.

TABLE 1. XOR calculation on one pixel.

Coordinate	Decimal	Hexadecimal	Bitwise operation							
x	14	0E	0	0	0	0	1	1	1	0
y	57	39	0	0	1	1	1	0	0	1
x⊕y	**55**	**37**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**1**



y-coordinate	x-coordinate	XOR result
57	14	55
44	15	35
45	15	34
45	16	61
30	22	08
31	22	09
31	23	08
140	23	155

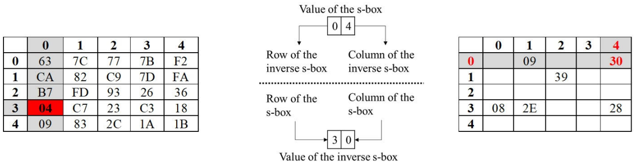
Result array
55
35
34
61
08
09
155

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	37	23	22	3D	08	09	9B									

FIGURE 7. An example for process of step 8.

Step 8: The results of XOR operation are checked. If the result of XOR operation is calculated by the different pixel components before, the last result is ignored. Same values are eliminated during calculation and a distinct result is stored in the s-box (S). All elements of the S are filled after elimination shown in Fig. 7 in hexadecimal format.

Step 9: The nonexistence values of the s-box and empty index in s-box matrix are detected. The nonexistence values randomly stored in the s-box.



	0	1	2	3	4
0	63	7C	77	7B	F2
1	CA	82	C9	7D	FA
2	B7	FD	93	26	36
3	04	C7	23	C3	18
4	09	83	2C	1A	1B

	0	1	2	3	4
0		09			30
1			39		
2					
3	08	2E			28
4					

FIGURE 8. An example of an inverse s-box generation.

The process of an inverse s-box generation is demonstrated in Fig. 8 to clarify the algorithm better. The following steps to generate an inverse s-box:

Step 10: Read the first value of the s-box. The left-hand digit of this value represents the row of the inverse s-box, the right-hand digit of this value represents the column of the inverse s-box.

Step 11: Read the row of the s-box and store the left-hand digit of the value of the inverse s-box, read the column of the s-box and store the right-hand digit of the value of the inverse s-box (IS).

The flowchart of the proposed s-box and inverse s-box generation process is demonstrated in Fig. 6 to clarify the algorithm better.

IV. ANALYSIS AND RESULTS

In this section, the security metrics used for evaluation of the proposed s-box and inverse s-box generation algorithms are presented and the reason why we preferred fingerprint to generate an s-box and which fingerprint feature is more distinctive to generate an s-box are explained in detail. The security metrics are bijectiveness, nonlinearity, strict avalanche criterion, bit independence criterion, linear probability, differential probability, confidence interval, randomness, and the execution time of the proposed algorithm. The proposed algorithm is implemented in MATLAB R2019b running on Windows 10 64-bit operating system on a computer equipped with 16 GB RAM and an Intel Core i7-7500-U (2.70 GHz - 2.90 GHz) processor.

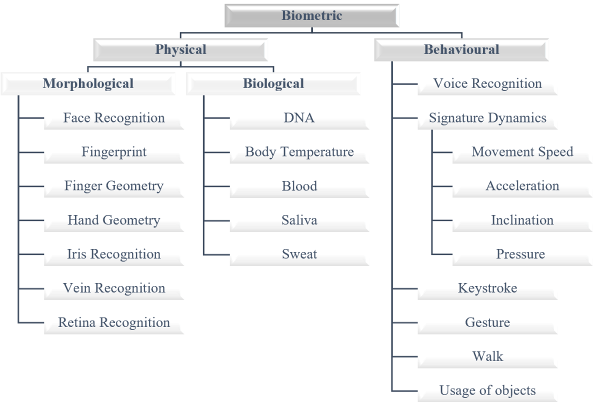


FIGURE 9. Biometric technologies.

A. BIOMETRIC PARAMETRICS (WHY FINGERPRINT?)

Biometric is the physical and behavioral characteristic feature of a human. These features are unique for each person, so the combination of these features is used to increase the accuracy of the system. Biometric technology divides into two categories: physical biometrics and behavioral biometrics as seen in Fig. 9. Physical biometrics occurs from birth and differs from person to person. The behavioral biometrics is characteristic and measurable features in human activities. The physical biometrics is more distinctive than the behavioral biometrics. The behavioral biometrics can be changed by momentary feelings such as stress.

Some of these biometric identifiers are compared according to seven factors as given in Table 2 [62]. The properties of a biometric identifier are universality, distinctiveness, permanence, and collectability. The attributes of a biometric system are performance, acceptability, and circumvention. The universality of a biometric identifier refers to the existence of the biometric feature for all people. The universality property ensures a high ratio for face recognition and iris recognition. The distinctiveness factor means that a biometric identifier can distinguish one from the other. The distinctiveness property ensures a high ratio for fingerprint and iris recognition. The permanence factor denotes the consistency of an identifier. Fingerprint and iris recognition satisfy a high ratio for the permanence property. The collectability factor represents how the identifier is captured and quantified. The collectability of face recognition, hand geometry, and signature recognition have a high ratio. Performance factor refers to speed and accuracy of the system. Fingerprint and iris recognition ensure the performance property with a high ratio. The acceptability of a system with an identifier represents how many users want to use that biometric identifier in the system. The users prefer to use face, voice, and signature identifiers with a high ratio in a system. The circumvention

TABLE 2. Comparison of biometric identifiers for mobile systems.

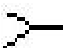
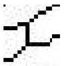
Identifiers	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	H
Fingerprint	**M**	**H**	**H**	**M**	**H**	**M**	**M**
Hand geometry	M	M	M	H	M	M	M
Iris	H	H	H	M	H	L	L
Keystroke	L	L	L	M	L	M	M
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

H: High, M: Medium, L: Low

factor refers to foolproof of the system. Face recognition, voice recognition, and signature recognition satisfy a high ratio for the circumvention property of a system.

Distinctiveness, collectability, performance, and universality are the most important factors for cryptography. The high value set is fingerprint, iris recognition, and face recognition for these factors. Face, fingerprint, and voice can be collected by mobile devices. Face recognition has two high and two low ratios, fingerprint has two high and two medium ratios, voice recognition has two medium and two low ratios for selected factors. As a result of the ratio, fingerprint, face, and voice identifiers would be preferable for a mobile system. The intersection of the factors with biometric identifiers is provided in Table 2. Distinctiveness and performance are the most important criteria to generate an s-box for a mobile payment system. Fingerprint satisfies a high ratio for both distinctiveness and performance factors. Therefore, we preferred the fingerprint metric for the proposed algorithm.

TABLE 3. Fingerprint features and definitions.

Pattern	Feature	Explanation
	Island or point	Island is the smallest pattern in fingerprint.
	Ridge ending	The line ends immediately.
	Bifurcation	The line branches two or more lines.
	Lake	The line branches and ends back to same line and occurs a space.
	Supur	A protrusion occurs from any line.
	Crossover	A new line occurs from connection of two lines.

Every person has a unique fingerprint. The characteristic points in a fingerprint is called the feature (minutiae). Generally, every line in a fingerprint looks the same but there are various structures in a fingerprint. A fingerprint has different features such as island, ridge ending, bifurcation, lake, supur, and crossover that are shown in Table 3. Ridge ending and bifurcation are the most common and distinctive features to detect a fingerprint, the other features are a combination of the ridge ending and the bifurcation.

B. SECURITY ANALYSIS OF PROPOSED S-BOXES

In this section, the cryptographic strength of the proposed s-boxes is tested with widely used analysis techniques such as bijectiveness, nonlinearity, strict avalanche criterion, bit independence criterion, linear probability, differential probability that are presented in [63], [64]. All security metrics are implemented in MATLAB R2019b. Besides these security properties of an s-box, we deal with randomness and confidence interval to specify a perfect n × n s-box. The execution time of the proposed s-box generation algorithm is calculated to test its suitability for mobile payment applications. 11 fingerprints are used to generate three different s-boxes from each fingerprint. We generated bifurcation, ridge ending, and bifurcation-ridge ending s-boxes from one fingerprint. The security analysis of each fingerprint s-boxes is demonstrated in Table 4. The results show that the fingerprint feature type is important to generate more robust s-boxes. The ridge ending s-box for each fingerprint is more robust according to the security parameters shown in Table 4. Each fingerprint feature is analyzed by an average of the security metrics of all s-boxes with the same fingerprint feature to decide which fingerprint feature is more secure to generate an s-box.

1) NONLINEARITY

Nonlinearity (NL) is the most important parameter of crypto s-boxes. It is the measurement of the difference among outputs. High nonlinearity means that there is no linear equation to generate the s-box. These linear equations make the system breakable so the most nonlinear s-box should be used. Nonlinearity is calculated by (1) which is a different form of the Walsh spectrum.

$$NL_f = 2^{n-1}(1{-}2^{-n}max_{\omega\epsilon GF(2^n)}|s_{<f>}(\omega)) \qquad (1)$$

The cyclic spectrum of f(x) is calculated by (2) where $\omega\epsilon$ GF(2^n), x. ω is the dot product.

$$s_{<f>}(\omega) = \Sigma_{x\epsilon GF(2^n)}(-1)^{f(x)\oplus x\omega} \qquad (2)$$

The maximum value of nonlinearity percentage is 0.80, the minimum value of nonlinearity percentage is 0.52, an average value of nonlinearity percentage is 0.73 for the fingerprint s-boxes, given in Table 4. The maximum nonlinearity is 180, the minimum nonlinearity is 62. According to nonlinearity results, the best nonlinearity value obtained with the bifurcation s-box and the bifurcation-ridge ending s-box, but the average of the ridge ending s-boxes got the best value with 0.75.

The maximum, average, and minimum nonlinearity values for each fingerprint feature are illustrated in Table 5. The best result of the nonlinearity percentage is 0.80 and both the bifurcation and the bifurcation-ridge ending s-boxes got. On the other hand, both the bifurcation and

TABLE 4. Security analysis of fingerprint s-boxes.

S-Box		SAC	Nonlinearity		Nonlinearity Percentage	DU	DP	LAT	LP	BIC-SAC	BIC-NL
Fingerprint	Fingerprint feature		max	min							
1	Bifurcation	0.4761	164	92	0.77	32	0.1250	36	0.1406	0.4949	101
	Ridge ending	0.5098	162	94	0.78	14	0.0547	34	0.1328	0.5019	103
	Bifurcation-Ridge ending	0.4753	166	88	0.73	32	0.1250	40	0.1563	0.4948	101
2	Bifurcation	0.5002	162	94	0.78	10	0.0391	34	0.1328	0.5047	103
	Ridge ending	0.5034	168	88	0.73	12	0.0469	40	0.1563	0.5013	103
	Bifurcation-Ridge ending	0.4941	166	88	0.73	10	0.0391	40	0.1563	0.4989	102
3	Bifurcation	0.4993	162	94	0.78	10	0.0391	34	0.1328	0.4983	102
	Ridge ending	0.4941	172	84	0.70	10	0.0391	44	0.1719	0.5004	103
	Bifurcation-Ridge ending	0.4888	172	84	0.70	10	0.0391	44	0.1719	0.4997	102
4	Bifurcation	0.5100	164	92	0.77	12	0.0469	36	0.1406	0.4976	102
	Ridge ending	0.5002	170	86	0.72	10	0.0391	42	0.1641	0.4996	102
	Bifurcation-Ridge ending	0.4924	180	76	0.63	18	0.0703	52	0.2031	0.4986	102
5	Bifurcation	0.4785	170	86	0.72	14	0.0547	42	0.1641	0.4974	102
	Ridge ending	0.5059	162	92	0.77	12	0.0469	36	0.1406	0.4987	102
	Bifurcation-Ridge ending	0.4839	164	90	0.75	12	0.0469	38	0.1484	0.4941	101
6	Bifurcation	0.4934	160	92	0.77	10	0.0391	36	0.1406	0.4960	102
	Ridge ending	0.5020	162	90	0.75	12	0.0469	38	0.1484	0.5043	103
	Bifurcation-Ridge ending	0.4980	164	74	0.62	14	0.0547	54	0.2109	0.4939	101
7	Bifurcation	0.4912	164	92	0.77	12	0.0469	36	0.1406	0.5021	103
	Ridge ending	0.5032	162	94	0.78	12	0.0469	34	0.1328	0.4970	102
	Bifurcation-Ridge ending	0.4971	160	96	0.80	14	0.0547	32	0.1250	0.4942	101
8	Bifurcation	0.4963	162	62	0.52	20	0.0781	66	0.2578	0.4911	101
	Ridge ending	0.4983	166	90	0.75	12	0.0469	38	0.1484	0.5033	103
	Bifurcation-Ridge ending	0.4875	164	66	0.55	24	0.0938	62	0.2422	0.4885	100
9	Bifurcation	0.5012	162	76	0.63	12	0.0469	52	0.2031	0.4987	102
	Ridge ending	0.5051	160	94	0.78	10	0.0391	34	0.1328	0.4973	102
	Bifurcation-Ridge ending	0.4951	164	84	0.70	14	0.0547	44	0.1719	0.4980	102
10	Bifurcation	0.4968	160	96	0.80	20	0.0781	32	0.1250	0.5003	103
	Ridge ending	0.5002	162	92	0.77	14	0.0547	36	0.1406	0.5015	103
	Bifurcation-Ridge ending	0.4954	164	84	0.70	26	0.1016	44	0.1719	0.5016	103
11	Bifurcation	0.5017	162	94	0.78	12	0.0469	34	0.1328	0.5075	104
	Ridge ending	0.4900	166	86	0.72	14	0.0547	42	0.1641	0.5005	103
	Bifurcation-Ridge ending	0.4846	162	94	0.78	10	0.0391	34	0.1328	0.5006	103

the bifurcation-ridge ending s-boxes have big fluctuations between minimum and maximum value of nonlinearity percentage. The nonlinearity percentage range for the bifurcation s-boxes is [0.52, 0.80], the nonlinearity percentage range for the bifurcation-ridge ending s-boxes is [0.55, 0.80]. On the other hand, the nonlinearity percentage range for the ridge ending s-boxes is [0.70, 0.78]. The nonlinearity percentage of the ridge ending s-boxes are greater than 0.70. According to the average result of each fingerprint feature, the nonlinearity percentage is higher than 0.70.

TABLE 5. Minimum, average, and maximum security results of fingerprint feature of proposed s-boxes.

S-Boxes		SAC	Nonlinearity		Nonlinearity Percentage	DU	DP	LAT	LP	BIC-SAC	BIC-NL
Fingerprint feature	Evaluation		max	min							
Bifurcation	Maximum	0.5100	170	96	0.80	32	0.1250	66	0.2578	0.5075	104
	Average	0.4950	163	88	0.73	15	0.0582	40	0.1555	0.4990	102
	Minimum	0.4761	160	62	0.52	10	0.0391	32	0.1250	0.4911	101
Ridge ending	Maximum	0.5098	172	94	0.78	14	0.0547	44	0.1719	0.5043	103
	Average	0.5011	165	90	0.75	12	0.0469	38	0.1484	0.5005	103
	Minimum	0.4900	160	84	0.70	10	0.0391	34	0.1328	0.4970	102
Bifurcation-Ridge ending	Maximum	0.4980	180	96	0.80	32	0.1250	62	0.2422	0.5016	103
	Average	0.4902	166	84	0.70	17	0.0653	44	0.1719	0.4966	102
	Minimum	0.4753	160	66	0.55	10	0.0391	32	0.1250	0.4885	100

2) STRICT AVALANCHE CRITERION

The strict avalanche criterion (SAC) is the second important criterion that is an effect of any change in an input on an output. This criterion is the combination of both completeness and avalanche effect. The avalanche effect means that the one-bit change in the plaintext should affect half of the bits in the ciphertext as given (3). Completeness means every bit in the plaintext must contribute to each output bit. If some bits of the ciphertext change only a few bits of the plaintext, the cryptanalyst can detect this relationship between the input and the output and use this relation to search for the key with the chosen plaintext attack.

$$Avalanche\ Criterion = \#switched\ bits\ in\ output \\ /\#total\ bits\ in\ output = 0.5 \quad (3)$$

$$wt(a_j^{\Delta Xi})/2^n = 1/2 \quad where\ i, j\epsilon\{0, 1, 2, \ldots, n\} \quad (4)$$

The SAC is significant for a cryptographic s-box. If the s-box satisfies the completeness and avalanche criteria, this s-box can satisfy the strict avalanche criteria. The probability of the n-th bit of y value that output of the S function is approximately equal to 1/2 when the m-th bit of x (input value of S function) is changed as given (4).

The strict avalanche criterion of all fingerprint s-boxes is close to 0.5 as shown in Table 4. The maximum, minimum, and average SAC values are 0.5098, 0.4753 and, 0.4954, respectively. The results show that using a fingerprint for a biometric s-box reveals good SAC value. The maximum, average, and minimum SAC values for each fingerprint feature are illustrated in Table 5. The SAC range of the bifurcation s-boxes is [0.4761, 0.5100], the SAC range of the ridge ending s-boxes is [0.4900, 0.5098], the SAC range of the bifurcation-ridge ending s-boxes is [0.4753, 0.4980]. The average SAC value of the bifurcation s-boxes is 0.4950, the average SAC value of the ridge ending s-boxes of fingerprint is 0.5011, the average SAC value of the bifurcation-ridge ending s-boxes is 0.4902. The SAC results show that generating an s-box with the ridge ending feature of fingerprint has the best SAC value.

3) LINEAR PROBABILITY

The Linear Approximation Table (LAT) is a good method for testing an s-box against the linear cryptanalyze, because linear probability is a significant property. The achievement of linear attacks will be difficult when the maximum value of the LAT is minimum. The LAT is 2^n x 2^n table for n-bit input and n-bit output s-box (S: GF(2^n) →GF(2^n)). LAT table is filled with the result of (5) for each a, b, Γ_a, $\Gamma_b\epsilon$ GF(2^n).

$$LAT(\Gamma_a, \Gamma_b) = \#\{x\epsilon GF(2^n) : \Gamma_a.x = \Gamma_b.S(x)\} - 2^{n-1} \quad (5)$$

The maximum LAT is 66, the minimum LAT is 32, the average LAT is 41 is given in Table 4. The maximum, average, and minimum LAT values for each fingerprint feature are illustrated in Table 5. The LAT range of the bifurcation s-boxes is [31], [65], the LAT range of the ridge ending s-boxes is [33], [43], the LAT range of the bifurcation-ridge ending s-boxes is [31], [61]. The average LAT value of the bifurcation, ridge ending, and bifurcation-ridge ending s-boxes are 40, 38, and 44, respectively. The LAT results show that the ridge ending s-boxes have the best LAT value.

The Linear Probability (LP) is expressed mathematically as:

$$LP = max_{\Gamma a, \Gamma b \neq 0} |(\#\{x\epsilon GF(2^n) : \Gamma_a.x = \Gamma_b.S(x)\} - 2^n)/2| \quad (6)$$

where Γ_a is corresponding input mask and Γ_b is corresponding output mask, "." denotes the dot product operation, "#" denotes the number of x satisfying the condition.

If an s-box has a low linear probability, the system resists against the linear cryptanalysis. The maximum, minimum, and average LP values are 0.2578, 0,1250, and 0.1586, respectively. The maximum, average, and minimum LP values for each fingerprint feature are illustrated in Table 5. The LP range of the bifurcation s-boxes is [0.1250, 0.2578], the LP range of the ridge ending s-boxes is [0.1328, 0.1719], the LP range of the bifurcation-ridge ending s-boxes is [0.1250, 0.2422]. The average LP value of the bifurcation, ridge ending, and bifurcation-ridge ending s-boxes are

0.1555, 0.1484, and 0.1719, respectively. The LP results show that the ridge ending s-boxes has the best LP value.

4) DIFFERENTIAL PROBABILITY

Differential Uniformity (DU) is the maximum similarity of generating a ciphertext differential when plaintext changes. The XOR distribution is $2^n \times 2^n$ table for n-bit input and n-bit output s-box (S: GF(2^n) →GF(2^n)). The XOR table is filled with the result of (7) for each Δx, Δy ∈ GF(2^n).

$$DU = max_{\Delta x \neq 0, \Delta y}(\#\{x \epsilon GF(2^n) : S(x) \oplus S(x \oplus \Delta x) = \Delta y\}) \quad (7)$$

The maximum DU is 32, the minimum DU is 10, the average DU is 15 is given in Table 4. The maximum, average, and minimum DU for each fingerprint feature are illustrated in Table 5. The DU range for the bifurcation s-boxes is [10], [31], the DU range for the ridge ending s-boxes is [10], [14], the DU range for the bifurcation-ridge ending s-boxes is [10], [31]. The average DU value of the bifurcation, ridge ending, and bifurcation-ridge ending s-boxes are 15, 12, and 17, respectively. The DU results show that generating an s-box with the ridge ending feature of fingerprint has the best DU value.

Differential Probability (DP) finds the same differential pairs between the plaintext and corresponding ciphertext.

Differential probability is calculated by (8) where Δx, Δy are the differential pairs for input and output.

$$DP = max_{\Delta x \neq 0, \Delta y}(\#\{x \epsilon GF(2^n) : S(x) \oplus S(x \oplus \Delta x) = \Delta y\}/2^n) \quad (8)$$

If an s-box has low differential probability, the system is more robust against the differential cryptanalysis. The maximum DP is 0.1250, the minimum DP is 0.0391, the average DP is 0.0568 is given in Table 4. The maximum, average, and minimum DP for each fingerprint feature are illustrated in Table 5. The DP range for the bifurcation s-boxes is [0.0391, 0.1250], the DP range for the ridge ending s-boxes is [0.0391, 0.0547], the DP range for the bifurcation-ridge ending s-boxes is [0.0391, 0.1250]. The average DP value of the bifurcation, ridge ending, and bifurcation-ridge ending s-boxes are 0.0582, 0.0469, and 0.0653, respectively. The DP results show that generating an s-box with the ridge ending feature of fingerprint has the best DP value.

5) BIT INDEPENDENCE CRITERION

Bit Independence Criterion (BIC) means that a change in one bit of an input does not affect any change in the output bit of the s-box. If the a-th bit of the input flips, the b-th bit and c-th bit of the output changes independently. There are two indicators of bit independence criterions to measure this feature of an s-box. One of them is the bit independence criterion for strict avalanche criterion (BIC–SAC) which calculates the average of (9) where x and w are the n-bit input and have only one-bit difference for each calculation.

$$BIC - SAC = \Sigma_{x=0 \rightarrow 2^n - 1}(S_i(x) \oplus S_j(w) - S_i(x) \oplus S_j(x)) \quad (9)$$

An s-box has an ideal BIC-SAC value when the average BIC-SAC is close to 0.5. The maximum, minimum, and average BIC-SAC values are 0.5075, 0.4885, and 0.4987, respectively. The maximum, average, and minimum BIC-SAC values for each fingerprint feature are illustrated in Table 5. The BIC-SAC range of the bifurcation s-boxes is [0.4911, 0.5075], the BIC-SAC range of the ridge ending s-boxes is [0.4970, 0.5043], and the BIC-SAC range of the bifurcation-ridge ending s-boxes is [0.4885, 0.5016]. The average BIC-SAC value of the bifurcation, ridge ending, and bifurcation-ridge ending s-boxes are 0.4990, 0.5005, and 0.4966, respectively. The BIC-SAC results show that generating an s-box with the ridge ending feature of fingerprint has the best BIC-SAC value.

The other indicator is the bit independence criterion for nonlinearity (BIC–NL) which calculates the average of equation (10) where b and c are the n-bit output.

$$y_b \oplus y_c \quad (10)$$

An s-box has an ideal BIC-NL value when the average BIC-NL is close to 103. The maximum BIC-NL is 104, the minimum BIC-NL is 100, the average BIC-NL is 102 is given in Table 4. The maximum, average, and minimum BIC-NL values for each fingerprint feature are illustrated in Table 5. The BIC-NL range for the bifurcation s-boxes is [101,104], the BIC-NL range for the ridge ending s-boxes is [102, 103], and the BIC-NL range for the bifurcation-ridge ending s-boxes is [100, 103]. The average BIC-NL of the bifurcation s-boxes is 102, the average BIC-NL of the ridge ending s-boxes is 103, and the average BIC-NL of the bifurcation-ridge ending s-boxes is 102. The BIC-NL results show that generating an s-box with the ridge ending feature of fingerprint has the best BIC-NL value, but the other features are also very good.

6) BIJECTIVENESS

Bijective means that a function must be both one to one and surjective (onto). Bijective function is that each element of the domain has only one match in the co-domain and there is no element in the co-domain without matching. Therefore, the number of elements in domain must be equal to the number of elements in co-domain.

S-box uses the y = S(x) function where S: x ∈ N → y ∈ . N is bounded in [0, 255] and output values of s-box are distinct. The hexadecimal digits of x represent the row and column of s-box. S(r, c) = S(hex(x)) where r = 1, 2, …, F; c = 1, 2, …, F. The s-box function always generates a different output for each different input. On the other hand, there is a different value in each row and column pair in an s-box. The position of the y gives a distinct (r, c) pair. The inverse transformation of S(x) is $S^{-1}(y)$. The inverse s-box function always generates a different output for each different input. The position of the x gives a distinct (r, c) pair. On the other hand, there is a different value in each row and column pair in an inverse s-box. The proposed s-box and inverse s-box have the property of bijectiveness.

7) RANDOMNESS

Randomness means that an s-box generated with independent random numbers, each value of the s-box is obtained completely randomly and there is no correlation among the sequence values. When the s-box is produced with a random function, there is no mathematical correlation between the columns of the s-box. Proposed fingerprint s-box can generate different s-boxes from one fingerprint due to the fingerprint pattern. The fingerprint is unique to each person, so the pattern list is different. The proposed s-box generation algorithm uses ridge ending and bifurcation features, each of the features shows different points on the fingerprint. Therefore, unique s-boxes are generated from one fingerprint. The proposed s-box generation algorithm satisfies randomness fully, so this attribute makes the s-boxes more robust, unpredictable, and untraceable.

8) CONFIDENCE INTERVAL

Confidence interval is a kind of statistical range calculation of the observed data. It obtains the upper and lower bound of the interval to show confidence level. The 95% confidence level is used commonly to examine the observed data. The interval calculates by (11) where X is the mean of observed data, Z chosen z-value according to confidence level, s is the standard deviation, n is the number of observations.

$$X \pm Z(s/n^{1/2}) \tag{11}$$

The mean of proposed s-boxes is 12.5, Z is 9.11 according to confidence level, and the standard deviation is 74. As a result of (11) the upper bound of s-boxes is 170 and the lower bound of s-boxes is 85. The confidence interval of all fingerprint s-boxes is [85, 170]. The confidence interval of the fingerprint s-boxes has the same value as the AES s-box.

9) EXECUTION TIME

The execution time of a given algorithm is described as the time spent by the system while executing the algorithm. Time consumption is very important for mobile applications. If any additional part of the cryptographic algorithm takes a long time, this additional part cannot be appropriate for a mobile application. The execution time of the proposed s-box and inverse s-box generation algorithms for bifurcation, ridge ending and bifurcation-ridge ending feature are demonstrated in Fig. 10, Fig. 11, and Fig. 12, respectively. The execution time of the proposed inverse s-box generation algorithm for all fingerprint patterns is approximately the same. The execution time of the proposed s-box generation algorithm by using the bifurcation feature of the fingerprint is between 0.1346 milliseconds and 0.3794 milliseconds. The execution time of the proposed s-box generation algorithm by using ridge ending feature of the fingerprint is between 0.1523 milliseconds and 0.4218 milliseconds. This fluctuation occurred according to the number of the bifurcation and ridge ending points in a fingerprint. The execution time of the proposed s-box generation algorithm will increase when both bifurcation and ridge ending features are used. The execution time changed between 0.1552 milliseconds and 0.4922 milliseconds to generate the bifurcation-ridge ending s-boxes.

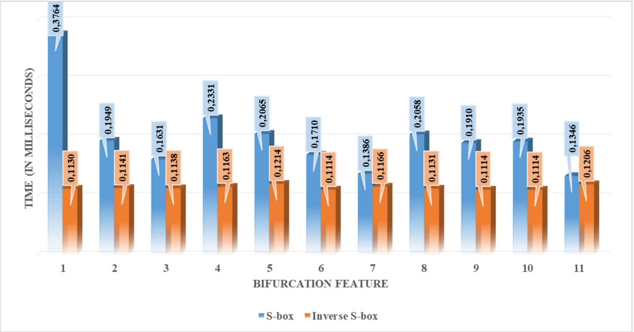


FIGURE 10. The execution time of the proposed s-box and inverse s-box generation algorithms for bifurcation feature.

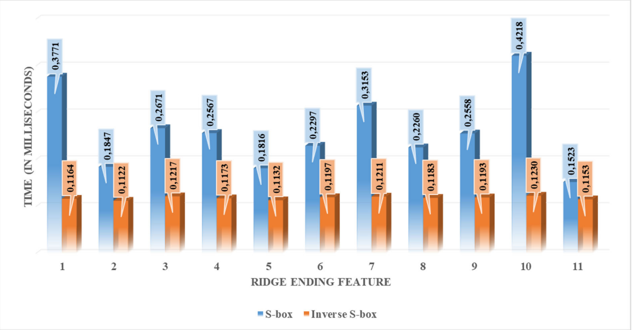


FIGURE 11. The execution time of the proposed s-box and inverse s-box generation algorithms for ridge ending feature.

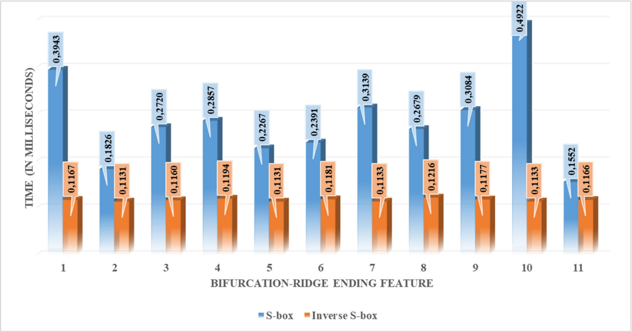


FIGURE 12. The execution time of the proposed s-box and inverse s-box generation algorithms for bifurcation-ridge ending feature.

The average execution time of the proposed s-box and inverse s-box generation algorithms for each feature and all fingerprint s-boxes are given in Table 6. The average execution time of the proposed s-box generation algorithm is 0.248923 milliseconds which is much better than [60], [61],

TABLE 6. The average execution time of the proposed s-box and inverse s-box generation algorithms.

Fingerprint Feature	S-box	Inverse S-box
All fingerprint s-boxes	0.248923	0.116345
Bifurcation	0.200769	0.114827
Ridge ending	0.260732	0.117938
Bifurcation-Ridge ending	0.285266	0.116271

the average execution time of the proposed inverse s-box generation algorithm is 0.116345 milliseconds which is much better than [60], [61]. The execution time changes according to the fingerprint and its feature. The average execution time of the s-box generation algorithm for the bifurcation fingerprint feature is 0.200769 milliseconds, the average execution time of the inverse s-box generation algorithm for the bifurcation fingerprint feature is 0.114827 milliseconds. The average execution time of the s-box generation algorithm for the ridge ending fingerprint feature is 0.260732 milliseconds, the average execution time of the inverse s-box generation algorithm for the ridge ending fingerprint feature is 0.117938 milliseconds. The average execution time of the s-box generation algorithm for the bifurcation-ridge ending fingerprint feature is 0.285266 milliseconds, the average execution time of the inverse s-box generation algorithm for the bifurcation-ridge ending fingerprint feature is 0.116271 milliseconds.

The execution time of the proposed s-box generation algorithm by using the bifurcation fingerprint feature has the smallest time. The number of bifurcation features for any fingerprint is fewer than the number of the ridge ending features for any fingerprint. Therefore, there are a few milliseconds difference between the execution time of algorithm with bifurcation and the execution time of algorithm with ridge ending. The execution time of the proposed s-box generation algorithm is less than 0.29 milliseconds and this value is acceptable.

V. PERFORMANCE COMPARISON

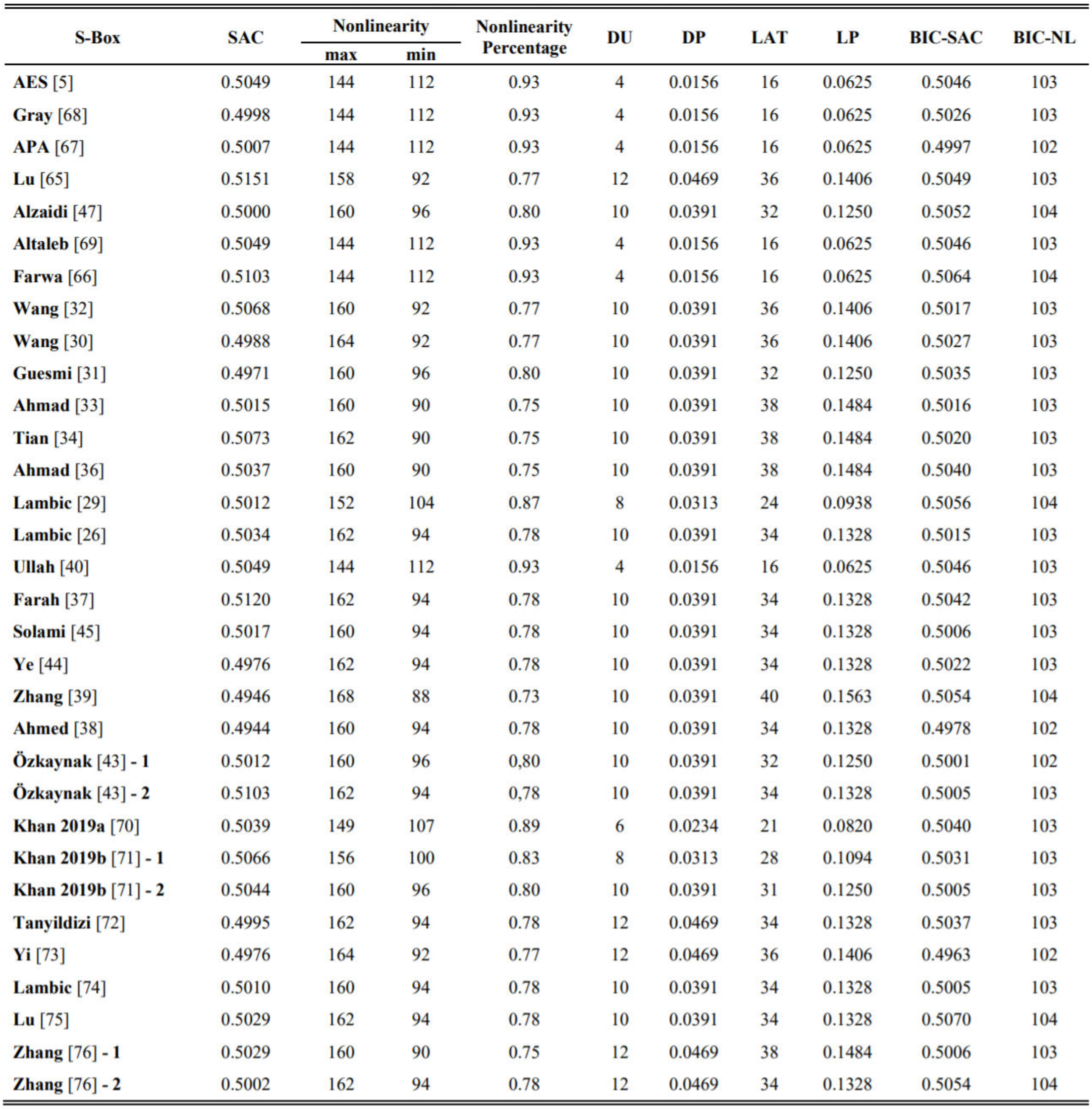
The performance comparison of the existing s-boxes is demonstrated in Table 7. The following points are important when we compare the performance of fingerprint s-boxes given in Table 4 and the average value of the fingerprint feature given in Table 8 with existing s-boxes:

- The ideal nonlinearity percentage is 93%, the average nonlinearity percentage is 81%, the maximum nonlinearity percentage is 93%, and the minimum nonlinearity percentage is 73% for the existing s-boxes. The average nonlinearity percentage of all fingerprint s-boxes, bifurcation s-boxes, and ridge ending s-boxes is in the range of the nonlinearity percentage of the existing s-boxes. However, the average nonlinearity percentage of the bifurcation-ridge ending s-boxes is not in the range of the nonlinearity percentage of the existing s-boxes.
- The DP of the existing s-boxes is in the [0.0156,0.0469] range. Most of the s-boxes have a probability less than 0.0547. The DP of bifurcation and ridge ending s-boxes is in the range of the existing s-boxes. The ridge ending s-boxes are more robust and the average DP for ridge ending s-boxes is 0.0469 such as [65], [72], [73], [76]-1, [76]-2.
- The existing s-boxes have differential uniformity between 4 and 12. The average DU of the fingerprint s-boxes is 15. If we prefer the ridge ending feature to generate a fingerprint s-box, the average DU is 12 like [65], [72], [73], [76]-1, [76]-2. The ridge ending s-boxes are more robust against the differential attack than the bifurcation and bifurcation-ridge ending s-boxes.
- The average LAT value is 31 and the range of the LAT is [16, 40] for the existing s-boxes. The average LAT value for all bifurcation s-boxes is 40 and the average LAT value for all ridge ending s-boxes is 38. The fingerprint s-boxes with not only bifurcation but also ridge ending features have the same LAT value with [33], [34], [36], [39], [76]-1.
- The LP of the existing s-box is between 0.0625 and 0.1563. The average LP value for all fingerprint s-boxes is 0.1586 and all fingerprint s-boxes are in the [0.0625, 0.1563] range approximately. The average LP value for bifurcation s-boxes is 0.1555 that is better than [39]. The average LP value for ridge ending s-boxes is 0.1484 like [33], [34], [36], [76]-1 and better than [39].
- The ideal BIC-SAC is 0.5 and the BIC-SAC value of all fingerprint s-boxes has an ideal value nearly. The average BIC-SAC value for all fingerprint s-boxes is 0.4987. The average BIC-SAC value for ridge ending s-boxes is 0.5005.
- The average BIC-NL value for ridge ending s-boxes is 103 which is better than [29], [38], [39], [43]-1, [47], [66], [67], [73], [75], [76]-2.

We find following points when we compare the average performance of each fingerprint features is given in Table 8 with the lower and upper bound of the existing s-boxes is given in Table 7:

- The average nonlinearity percentage for all types of fingerprint s-boxes is 73% like [39] and the nonlinearity range is [87, 165]. The ridge ending s-boxes have the best average nonlinearity with 75% in the [90, 165] range, same with [33], [34], [36], [76]-1 and better than [39]. The bifurcation-ridge ending s-boxes have the worst average nonlinearity with 70% in the [84, 166] range.
- The fingerprint s-box satisfies the strict avalanche criterion excellently. The average SAC for all types of fingerprint s-boxes is 0.4954. The ridge ending s-boxes have the best average SAC value with 0.5011. The bifurcation-ridge ending s-boxes have the worst average SAC value with 0.4902.
- The fingerprint s-box satisfies the BIC-SAC excellently for all fingerprint features. The average BIC-SAC value

TABLE 7. Security analysis of existing s-boxes.



S-Box	SAC	Nonlinearity		Nonlinearity Percentage	DU	DP	LAT	LP	BIC-SAC	BIC-NL
		max	min							
AES [5]	0.5049	144	112	0.93	4	0.0156	16	0.0625	0.5046	103
Gray [68]	0.4998	144	112	0.93	4	0.0156	16	0.0625	0.5026	103
APA [67]	0.5007	144	112	0.93	4	0.0156	16	0.0625	0.4997	102
Lu [65]	0.5151	158	92	0.77	12	0.0469	36	0.1406	0.5049	103
Alzaidi [47]	0.5000	160	96	0.80	10	0.0391	32	0.1250	0.5052	104
Altaleb [69]	0.5049	144	112	0.93	4	0.0156	16	0.0625	0.5046	103
Farwa [66]	0.5103	144	112	0.93	4	0.0156	16	0.0625	0.5064	104
Wang [32]	0.5068	160	92	0.77	10	0.0391	36	0.1406	0.5017	103
Wang [30]	0.4988	164	92	0.77	10	0.0391	36	0.1406	0.5027	103
Guesmi [31]	0.4971	160	96	0.80	10	0.0391	32	0.1250	0.5035	103
Ahmad [33]	0.5015	160	90	0.75	10	0.0391	38	0.1484	0.5016	103
Tian [34]	0.5073	162	90	0.75	10	0.0391	38	0.1484	0.5020	103
Ahmad [36]	0.5037	160	90	0.75	10	0.0391	38	0.1484	0.5040	103
Lambic [29]	0.5012	152	104	0.87	8	0.0313	24	0.0938	0.5056	104
Lambic [26]	0.5034	162	94	0.78	10	0.0391	34	0.1328	0.5015	103
Ullah [40]	0.5049	144	112	0.93	4	0.0156	16	0.0625	0.5046	103
Farah [37]	0.5120	162	94	0.78	10	0.0391	34	0.1328	0.5042	103
Solami [45]	0.5017	160	94	0.78	10	0.0391	34	0.1328	0.5006	103
Ye [44]	0.4976	162	94	0.78	10	0.0391	34	0.1328	0.5022	103
Zhang [39]	0.4946	168	88	0.73	10	0.0391	40	0.1563	0.5054	104
Ahmed [38]	0.4944	160	94	0.78	10	0.0391	34	0.1328	0.4978	102
Özkaynak [43] **- 1**	0.5012	160	96	0,80	10	0.0391	32	0.1250	0.5001	102
Özkaynak [43] **- 2**	0.5103	162	94	0,78	10	0.0391	34	0.1328	0.5005	103
Khan 2019a [70]	0.5039	149	107	0.89	6	0.0234	21	0.0820	0.5040	103
Khan 2019b [71] **- 1**	0.5066	156	100	0.83	8	0.0313	28	0.1094	0.5031	103
Khan 2019b [71] **- 2**	0.5044	160	96	0.80	10	0.0391	31	0.1250	0.5005	103
Tanyildizi [72]	0.4995	162	94	0.78	12	0.0469	34	0.1328	0.5037	103
Yi [73]	0.4976	164	92	0.77	12	0.0469	36	0.1406	0.4963	102
Lambic [74]	0.5010	160	94	0.78	10	0.0391	34	0.1328	0.5005	103
Lu [75]	0.5029	162	94	0.78	10	0.0391	34	0.1328	0.5070	104
Zhang [76] **- 1**	0.5029	160	90	0.75	12	0.0469	38	0.1484	0.5006	103
Zhang [76] **- 2**	0.5002	162	94	0.78	12	0.0469	34	0.1328	0.5054	104

for all types of fingerprint s-boxes is 0.4987. The ridge ending s-boxes have the best average BIC-SAC value with 0.5005.

- The fingerprint s-box satisfies the BIC-NL excellently for all fingerprint features. The average BIC-NL value for all types of fingerprint s-boxes is 102. The ridge ending s-boxes have the best average BIC-NL value with 103.
- The fingerprint s-box satisfies the LP for both ridge ending and bifurcation features. The average LP value for all types of fingerprint s-boxes is 0.1586. The ridge ending s-boxes have the best average LP value with 0.1484. The bifurcation-ridge ending s-boxes have the worst average LP value with 0.1719.
- The fingerprint s-box satisfies the DP excellently for ridge ending feature. The average DP value for all types of fingerprint s-boxes is 0.0568. The ridge ending s-boxes have the best average DP value with 0.0469. Both bifurcation and bifurcation-ridge ending s-boxes have the worst average DP values with 0.0582, 0.0653, respectively.

The security analysis of the DNA-1 (SHARED KEY = COMPUTER AND SERVER NAME = SCIENCE) and the DNA-2 (SHARED KEY = BOMPUTER AND SERVER

TABLE 8. Performance metrics of 8 × 8 fingerprint s-boxes on average.

S-Box Type	SAC	Nonlinearity		Nonlinearity Percentage	DU	DP	LAT	LP	BIC-SAC	BIC-NL	Time	Inv Time
		max	min									
Average of Bifurcation S-boxes	**0.4950**	**163**	**88**	**0.73**	15	0.0582	**40**	**0.1555**	**0.4990**	**102**	0.201	0.115
Average of Ridge ending S-boxes	**0.5011**	**165**	**90**	**0.75**	**12**	**0.0469**	**38**	**0.1484**	**0.5005**	**103**	0.261	0.118
Average of Bifurcation-Ridge ending S-boxes	0.4902	**166**	84	0.70	17	0.0653	44	0.1719	0.4966	**102**	0.285	0.116
Average of All Fingerprint S-boxes	**0.4954**	**165**	87	**0.73**	15	0.0568	41	0.1586	**0.4987**	**102**	0.249	0.116

TABLE 9. Security analysis of the dna and the rna s-boxes.

S-Box	SAC	Nonlinearity		Nonlinearity Percentage	DU	DP	LAT	LP	BIC-SAC	BIC-NL	Time	Inv Time
		max	min									
Kadhim [60] **DNA-1**	0.4785	170	80	0.67	26	0.1016	48	0.1875	0.5020	103	0.216	0.240
Kadhim [60] **DNA-2**	0.4524	182	54	0.45	58	0.2266	74	0.2891	0.4761	98	0.388	0.400
Average of DNA	**0.4655**	**176**	**67**	**0.56**	**42**	**0.1641**	**61**	**0.2383**	**0.4890**	**100**	**0.302**	**0.320**
Farhan [61] **RNA-1**	0.4890	168	74	0.62	16	0.0625	54	0.2109	0.4934	101	130	130
Farhan [61] **RNA-2**	0.4824	172	84	0.70	12	0.0469	44	0.1719	0.4973	102	130	130
Farhan [61] **RNA-3**	0.4971	164	92	0.77	12	0.0469	36	0.1406	0.5016	103	130	130
Farhan [61] **RNA-4**	0.4741	184	72	0.60	14	0.0547	56	0.2188	0.4971	102	143	143
Farhan [61] **RNA-5**	0.4932	166	84	0.70	12	0.0469	44	0.1719	0.5020	103	143	143
Average of RNA	**0.4872**	**171**	**81**	**0.68**	**13**	**0.0516**	**47**	**0.1828**	**0.4983**	**102**	**135**	**135**
Average of All	**0.4810**	**172**	**77**	**0.64**	**21**	**0.0837**	**51**	**0.1987**	**0.4957**	**102**	**97**	**97**

NAME = SCIENCE) s-boxes from [60] and RNA-1 (Secret key = “ABC”), RNA-2 (Secret key = “!kh”), RNA-3 ((Secret key = “ U@m”), RNA-4 (Secret key = “%: %”), RNA-5 (Secret key = “%: %”) s-boxes from [61] are given in Table 9. The comparison comments are as follows:

- The DNA-based s-boxes and the RNA-based s-boxes do not reach the ideal SAC value. The average SAC value of the fingerprint s-boxes is much better than the average SAC of the DNA and RNA s-boxes. The SAC value of the fingerprint s-boxes is between 0.4753 and 0.5100 but the SAC value of the DNA s-boxes is between 0.4524 and 0.4785 and the SAC value of RNA s-boxes is between 0.4741 and 0.4971. The average SAC value of the fingerprint s-boxes is 0.4954 and the average SAC value of the existing biometric s-boxes is 0.4810. As a result of SAC value, the fingerprint s-boxes have the best SAC value.
- The ideal nonlinearity percentage is 93% and the ideal nonlinearity is 112 according to the AES s- box. The nonlinearity percentage of the existing biometric s-boxes is 82% on average and the average nonlinearity of all the existing biometric s- boxes is 98 on average. The nonlinearity percentage of the DNA s-boxes is 56% on average, the nonlinearity percentage of the RNA s-boxes is 68% on average, the nonlinearity percentage of all the existing biometric s-boxes is 64% on average, the nonlinearity percentage for the fingerprint s-boxes is 73% on average. The average nonlinearity of the DNA s-boxes is 67, the average nonlinearity for the RNA s-boxes is 81, the average nonlinearity of all the existing biometric s-boxes is 77, and the average nonlinearity for the fingerprint s-boxes is 87. According to nonlinearity, the fingerprint s-boxes are more nonlinear than the existing biometric s-boxes.
- The average DP value of the existing s-boxes is 0.0340 and the average DU value of the existing s-boxes is 9. The average DP value of the DNA s-boxes is 0.1641, and the average DU value of the DNA s-boxes is 42. The average DP value of the RNA s-boxes is 0.0516, and the average DU value of the RNA s-boxes is 13. The average DP value of all the existing biometric s-boxes is 0.0837, and the average DU value of all the existing biometric s-boxes is 21. The average DP value of the fingerprint s-boxes is 0.0568 and the average DU value of the fingerprint s-boxes is 15. According to DP and DU, the fingerprint s-boxes are more robust against the differential attacks than the DNA s-boxes.
- The average LP value of the existing s-boxes is 0.1163 and the average LAT value of the existing s-boxes is 30. The average LP value of the DNA s-boxes is 0.2383, and the average LAT value of the DNA s-boxes is 61. The average LP value of the RNA s-boxes is

TABLE 10. An example of the fingerprint s-box and its analysis results.

SAC	NL max	NL min	NL Percentage	DU	DP	LAT	LP	BIC-SAC	BIC-NL
0.5002	162	94	0.78	10	0.0391	34	0.1328	0.5047	103

row\column	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	64	79	28	31	30	89	34	35	10	50	45	49	101	104	77	133
1	38	240	125	68	140	83	91	194	192	170	234	213	63	209	254	88
2	150	27	65	222	164	156	229	168	82	115	111	71	143	251	244	26
3	4	197	199	43	183	74	137	131	239	93	211	80	6	9	20	193
4	134	232	123	190	90	205	129	51	33	72	117	174	108	212	21	202
5	48	121	112	14	173	187	151	169	243	119	175	223	204	160	221	29
6	86	216	210	245	128	78	179	13	185	12	220	53	25	198	149	95
7	59	238	62	17	176	225	230	92	116	159	161	54	253	135	141	132
8	58	41	147	237	171	5	178	227	52	189	81	99	224	69	152	127
9	148	200	235	98	231	166	146	158	56	233	184	219	208	182	153	145
A	22	110	126	87	226	252	37	67	157	248	206	162	191	139	100	188
B	18	70	97	138	177	16	144	61	3	218	241	124	44	113	106	172
C	60	102	39	36	130	215	242	165	66	114	23	73	228	105	246	32
D	7	217	47	42	136	196	247	236	1	24	120	15	122	2	75	11
E	195	255	214	180	0	167	109	40	118	76	46	103	8	154	84	250
F	19	57	201	96	163	249	181	142	85	207	155	203	186	94	55	107

0.1828, and the average LAT value of the RNA s-boxes is 47. The average LP value of all the existing biometric s-boxes is 0.1987, and the average LAT value of all the existing biometric s-boxes is 51. The average LP value of the fingerprint s-boxes is 0.1586 and the average LAT value of the fingerprint s-boxes is 41. According to LP, the fingerprint s-boxes are more robust against the linear attacks than all the existing biometric s-boxes.

- The average BIC-SAC value of the existing s-boxes is 0.5029 and the average BIC-NL value of the existing s-boxes is 103. The average BIC-SAC value of the DNA s-boxes is 0.4890 and the average BIC-NL value of the DNA s-boxes is 100. The average BIC-SAC value of the RNA s-boxes is 0.4983 and the average BIC-NL value of the RNA s-boxes is 102. The average BIC-SAC value of all the existing biometric s-boxes is 0.4957 and the average BIC-NL value of all the existing biometric s-boxes is 102. The average BIC-SAC value of the fingerprint s-boxes is 0.4987 and the average BIC-NL value of the fingerprint s-boxes is 102. As a result of BIC-SAC and BIC-NL, the fingerprint s-boxes are better than all the existing biometric s-boxes.
- The average execution time of the DNA s-box generation algorithm is 0.302 milliseconds and the average execution time of the DNA inverse s-box generation algorithm is 0.320 milliseconds. The average execution time of the RNA s-box generation algorithm is 135 milliseconds and the average execution time of the RNA inverse s-box generation algorithm is 135 milliseconds. The average execution time of the fingerprint s-box generation algorithm is 0.248 milliseconds and the average execution time of the fingerprint inverse s-box generation algorithm is 0.115 milliseconds. The execution time of the fingerprint s-box and inverse s-box generation algorithms are much lower than both the DNA and the RNA s-boxes and inverse s-boxes generation algorithms.

The best fingerprint s-box is given in Table 10 with its analysis results. The analysis results showed that the fingerprint s-box is much better than the existing biometric s-boxes.

VI. CONCLUSION

In this article, a new, simple, and effective fingerprint s-box generation algorithm is introduced by using different fingerprint patterns. The feasibility of the proposed algorithm is evaluated, and the fingerprint s-boxes are compared with not only some existing chaotic s-boxes but also biometric s-boxes according to security metrics of a good s-box. As evaluation security metrics, Nonlinearity, Nonlinearity Percentage, Strict Avalanche Criteria, Differential Uniformity, Differential Probability, LAT, Linear Probability, Bit Independence Criterion-SAC, Bit Independence Criterion-NL, Bijective, Randomness, Confidence Interval, Time Consumption are utilized. The obtained performance parameters show that;

(1) the fingerprint s-boxes get the different result according to fingerprint pattern,

(2) the fingerprint s-boxes with the ridge ending feature has the best security metrics results,

Algorithm 1 Fingerprint 8 × 8 Substitution Box and Inverse Substitution Box Algorithm

```
input: A minutiae type of fingerprint and fingerprint
output: A 8 × 8 substitution box and inverse substitution box
Read fingerprint F in binary format B
Resize B
Specify the pattern of fingerprint with bwmorph function P
Thinned the pattern of fingerprint T
for row 1 to size of T do
    for column 1 to size of T do
        if it has connection near square of pixels then
            add one to the pixel counter for each side
        end if
        set counter to matrix(row, column)
    end for
end for
if minutiae type is equal to 1 then
    set column and row of bifurcation where matrix(row, column) is greater than three
    for all bifurcation do
        set result of xor column and row to bifurcation
    end for
    for all xor result do
        if first element of bifurcation is same with next element of bifurcation then
            delete next element
        end if
    end for
    set all elements to substitution box
    set nonexistence values randomly to substitution box
end if
if minutiae type is equal to 2 then
    set column and row of ridge where matrix(row, column) is equal to two
    for all ridge do
        set result of xor column and row to ridge
    end for
    for all xor result do
        if first element of ridge is same with next element of ridge
            delete next element
        end if
    end for
    set all elements to substitution box
    set nonexistence values randomly to substitution box
end if
if minutiae type is equal to 3 then
    set column and row of bifurcation where matrix(row, column) is greater than three
    set column and row of ridge where matrix(row, column) is equal to two
    combine bifurcation and ridge
    for all values do
        set result of xor column and row to BR matrix
    end for
    for all xor result do
        if first element of BR matrix is same with next element of BR matrix then
            delete next element
        end if
    end for
```

Algorithm 1 *Continue:*

```
    set all elements to substitution box
    set nonexistence values randomly to substitution box
end if
for row 0 to 15 do
    for column 0 to 15 do
        assign row and column to value
        assign first digit of sbox(row,column) to new_row
        assign second digit of sbox(row,column) to new_column
        set value to inv_sbox(new_row,new_column)
    end for
end for
```

(3) the fingerprint s-boxes have satisfactory results according to the result of the existing chaotic s-boxes,

(4) the fingerprint s-boxes have a higher security performance than both the DNA and the RNA s-boxes,

(5) the execution time of the proposed fingerprint s-box generation algorithm is better than both the DNA and the RNA s-box generation algorithms,

(6) the execution time of the proposed fingerprint inverse s-box generation algorithm is better than both the DNA and the RNA inverse s-box generation algorithms

(7) the LP and DP results of the fingerprint s-boxes have a higher score than the DNA s-boxes showing that the proposed fingerprint s-boxes has apparent advantages to restrain the differential and linear cryptanalysis attacks.

Taking into consideration the success of the proposed fingerprint s-box generation algorithm based on utilizing biometric data, to ensure uniqueness and randomness the algorithm confirms its suitability for using as a good s-box in cipher algorithm.

We have more scenarios for the inverse s-box: (1) it can be generated by sender and send it over communication channel to the receiver, (2) the fingerprint image is sent over communication channel to the receiver and the receiver generates an inverse s-box, (3) the fingerprint image can be stored in application to generate an inverse s-box, (4) an s-box can be stored in the application to generate an inverse s-box. We will test the s-boxes on cipher algorithms to decide which scenario is suitable for inverse s-box as a future work.

APPENDIX

See Algorithm 1.

ACKNOWLEDGMENT

This work is a part of the Ph.D. thesis titled “Model Design and Performance Analysis for Secure Storage of Personal Data in Mobile Payment Systems” at the Institute of Graduate Studies, İstanbul University-Cerrahpaşa, İstanbul, Turkey.

REFERENCES

[1] M. J. Dworkin, E. B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, and F. James Dray, Jr., “Advanced encryption standard (AES) (FIPS PUB 197),” Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, Tech Rep. 197, 2001.

[2] H. Feistel, “Block cipher cryptographic,” U.S. Patent 158 360, Mar. 19, 1974.

[3] T. Sakalli, E. Buluş, A. Şahin, and F. Büyüksaraçoğlu, “S Kutulainda doğrusal eŞitlik,” *Proc. Sinyal İŞleme ve İletişim UygulamalarI Kurultayi-SİU*, Antalya, Turkey, 2006, pr. 2006.

[4] J. Daemen and V. Rijmen, “The block cipher Rijndael,” in *Proc. 3rd Int. Conf. Smart Card Res. Adv. Appl.* (Lecture Notes in Computer Science), vol. 1820. Berlin, Germany: Springer, 2000, pp. 227–284.

[5] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*. Berlin, Germany: Springer Publications, 2002.

[6] K. Nyberg, “Differentially uniform mappings for cryptography,” in *Advances in Cryptology-EUROCRYPT*, Berlin, Germany: Springer, 1994.

[7] S. R. Moosavi, E. Nigussie, S. Virtanen, and J. Isoaho, “Cryptographic key generation using ECG signal,” in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2017, pp. 1024–1031.

[8] Y.-J. Chang, W. Zhang, and T. Chen, “Biometrics-based cryptographic key generation,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jun. 2004, pp. 2203–2206.

[9] S. M. Wadi and N. Zainal, “Rapid encryption method based on AES algorithm for grey scale HD image encryption,” *Procedia Technol.*, vol. 11, pp. 51–56, 2013.

[10] P. Kumar and S. B. Rana, “Development of modified AES algorithm for data security,” *Optik*, vol. 127, no. 4, pp. 2341–2345, Feb. 2016.

[11] D. Canright, “A very compact s-box for AES,” in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Berlin, Germany, 2005.

[12] T. Tiessen, L. R. Knudsen, S. Kölbl, and M. M. Lauridsen, “Security of the AES with a secret s-box,” International Workshop on Fast Software Encryption, Berlin, Heidelberg, 2015.

[13] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, “Biometric encryption using image processing,” *Proc. SPIE*, vol. 3314, pp. 178–188, Apr. 1998.

[14] C. R. Costanzo, “Biometric cryptography: key generation using feature and parametric aggregation,” School Eng. Appl. Sci., Dept. Comput. Sci., George Washington Univ., Washington, DC, USA, 2004.

[15] M. Rashid and H. Zaki, “RSA cryptographic key generation using fingerprint minutiae,” *Iraqi J. Comput. Informat.*, vol. 41, no. 1, pp. 66–69, Dec. 2014.

[16] A. Jagadeesan and K. Duraiswamy, “Secured cryptographic key generation from multimodal biometrics: Feature level fusion of fingerprint and Iris,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 7, no. 2, pp. 028–037, 2010.

[17] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel, “Using voice to generate cryptographic keys,” in *Proc. Speaker Odyssey - Speaker Recognit. Workshop*, 2001, pp. 237–242.

[18] B. Chen and V. Chandran, “Biometric based cryptographic key generation from faces,” in *Proc. 9th Biennial Conf. Austral. Pattern Recognit. Soc. Digit. Image Comput. Techn. Appl. (DICTA)*, Dec. 2007, pp. 394–401.

[19] S. Abuguba, M. M. Milosavljevic, and N. Macek, “An efficient approach to generating cryptographic keys from face and iris biometrics fused at the feature level,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 15, no. 6, pp. 6–11, 2015.

[20] B. Santhi, K. Ravichandran, A. Arun, and L. Chakkarapani, “A novel cryptographic key generation method using image features,” *Res. J. Inf. Technol.*, vol. 4, no. 2, pp. 88–92, 2012.

[21] K. Kalaiselvi and A. Kumar, ‘‘Enhanced AES cryptosystem by using genetic algorithm and neural network in S-box,’’ in *Proc. IEEE Int. Conf. Current Trends Adv. Comput. (ICCTAC)*, Mar. 2016, pp. 1–6.

[22] M. N. A. Noughabi and B. Sadeghiyan, ‘‘Design of S-boxes based on neural networks,’’ in *Proc. Int. Conf. Electron. Inf. Eng. (ICEIE)*, Aug. 2010, p. 172.

[23] Ö. Şengel, M. A. Aydin, and A. Sertbaş, ‘‘A survey on white box cryptography model for mobile payment systems,’’ in *Int. Telecommun. Conf.*, Singapore, 2019, pp. 215–225.

[24] G. Chen, ‘‘A novel heuristic method for obtaining S-boxes,’’ *Chaos, Solitons Fractals*, vol. 36, no. 4, pp. 1028–1036, May 2008.

[25] Ü. Şavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, ‘‘A novel approach for strong S-Box generation algorithm design based on chaotic scaled zhongtang system,’’ *Nonlinear Dyn.*, vol. 87, no. 2, pp. 1081–1094, Jan. 2017.

[26] D. Lambić, ‘‘A novel method of S-box design based on discrete chaotic map,’’ *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2407–2413, Mar. 2017.

[27] G. Liu, W. Yang, W. Liu, and Y. Dai, ‘‘Designing S-boxes based on 3-D four-wing autonomous chaotic system,’’ *Nonlinear Dyn.*, vol. 82, no. 4, pp. 1867–1877, Dec. 2015.

[28] M. Khan and Z. Asghar, ‘‘A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and s8 permutation,’’ *Neural Comput. Appl.*, vol. 29, no. 4, pp. 993–999, Feb. 2018.

[29] D. Lambić, ‘‘A novel method of S-box design based on chaotic map and composition method,’’ *Chaos, Solitons Fractals*, vol. 58, pp. 16–21, Jan. 2014.

[30] W. Yong and L. Peng, ‘‘An improved method to obtaining S-Box based on chaos and genetic algorithm,’’ *HKIE Trans.*, vol. 19, no. 4, pp. 53–58, Jan. 2012.

[31] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, ‘‘A novel design of chaos based S-Boxes using genetic algorithm techniques,’’ in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2014, pp. 678–684.

[32] Y. Wang, K.-W. Wong, C. Li, and Y. Li, ‘‘A novel method to design S-box based on chaotic map and genetic algorithm,’’ *Phys. Lett. A*, vol. 376, nos. 6–7, pp. 827–833, Jan. 2012.

[33] M. Ahmad, D. Bhatia, and Y. Hassan, ‘‘A novel ant colony optimization based scheme for substitution box design,’’ *Procedia Comput. Sci.*, vol. 57, pp. 572–580, 2015.

[34] Y. Tian and Z. Lu, ‘‘S-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm,’’ *J. Syst. Eng. Electron.*, vol. 27, no. 1, pp. 232–241, Feb. 2016.

[35] Y. Tian and Z. Lu, ‘‘Chaotic S-box: Intertwining logistic map and bacterial foraging optimization,’’ *Math. Problems Eng.*, vol. 2017, pp. 1–11, 2017.

[36] M. Ahmad, N. Mittal, P. Garg, and M. Maftab Khan, ‘‘Efficient cryptographic substitution box design using travelling salesman problem and chaos,’’ *Perspect. Sci.*, vol. 8, pp. 465–468, Sep. 2016.

[37] T. Farah, R. Rhouma, and S. Belghith, ‘‘A novel method for designing S-box based on chaotic map and Teaching–Learning-Based Optimization,’’ *Nonlinear Dyn.*, vol. 88, no. 2, pp. 1059–1071, 2017.

[38] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, ‘‘A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map,’’ *Neural Comput. Appl.*, vol. 11, pp. 1–18, May 2018.

[39] T. Zhang, C. L. P. Chen, L. Chen, X. Xu, and B. Hu, ‘‘Design of highly nonlinear substitution boxes based on I-Ching operators,’’ *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3349–3358, Dec. 2018.

[40] A. Ullah, S. S. Jamal, and T. Shah, ‘‘A novel construction of substitution box using a combination of chaotic maps with improved chaotic range,’’ *Nonlinear Dyn.*, vol. 88, no. 4, pp. 2757–2769, Jun. 2017.

[41] Attaullah, S. S. Jamal, and T. Shah, ‘‘A novel algebraic technique for the construction of strong substitution box,’’ *Wireless Pers. Commun.*, vol. 99, no. 1, pp. 213–226, Mar. 2018.

[42] F. Özkaynak, ‘‘Construction of robust substitution boxes based on chaotic systems,’’ *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3317–3326, Aug. 2019.

[43] F. Özkaynak, ‘‘From biometric data to cryptographic primitives: A new method for generation of substitution boxes,’’ in *Proceedings Int. Conf. Biomed. Eng. Bioinf.*, Bangkok, Thailand, 2017.

[44] T. Ye and L. Zhimao, ‘‘Chaotic S-box: Six-dimensional fractional Lorenz–Duffing chaotic system and O-shaped path scrambling,’’ *Nonlinear Dyn.*, vol. 94, no. 3, pp. 2115–2126, 2018.

[45] E. Al Solami, M. Ahmad, C. Volos, M. Doja, and M. Beg, ‘‘A new hyperchaotic system-based design for efficient bijective substitution-boxes,’’ *Entropy*, vol. 20, no. 7, p. 525, Jul. 2018.

[46] J. Peng, D. Zhang, and X. Liao, ‘‘A method for designing dynamical S-boxes based on hyperchaotic lorenz system,’’ in *Proc. IEEE 10th Int. Conf. Cognit. Informat. Cognit. Comput. (ICCI-CC)*, Aug. 2011, pp. 304–309.

[47] A. A. Alzaidi, M. Ahmad, M. N. Doja, E. A. Solami, and M. M. S. Beg, ‘‘A new 1D chaotic map and β-hill climbing for generating substitution-boxes,’’ *IEEE Access*, vol. 6, pp. 55405–55418, 2018.

[48] J. Juremi, R. Mahmod, Z. A. Zukarnain, and S. M. Yasin, ‘‘Modified AES s-box based on determinant matrix algorithm,’’ *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 1, pp. 110–116, 2017.

[49] M. Ahmad and M. Malik, ‘‘Design of chaotic neural network based method for cryptographic substitution box,’’ in *Proc. Int. Conf. Electr., Electron., Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 864–868.

[50] J. D. R. Arrañaga, J. A. S. Chavarin, J. J. R. Panduro, and E. C. B. Alvarez, ‘‘New s-box calculation for Rijndael-AES based on an artificial neural network,’’ *Electrónica*, vol. 6, no. 2, pp. 49–69, 2017.

[51] K. Shama, ‘‘S-boxes generated using affine transformation giving maximum avalanche effect,’’ *Int. J. Comput. Sci. Eng. (IJCSE)*, vol. 3, no. 9, pp. 3185–3193, 2011.

[52] S. Sinha and C. Arya, ‘‘Algebraic construction and cryptographic properties of rijndael substitution box,’’ *Defence Sci. J.*, vol. 62, no. 1, pp. 32–37, Jan. 2012.

[53] I. Das, S. Nath, S. Roy, and S. Mondal, ‘‘Random S-Box generation in AES by changing irreducible polynomial,’’ in *Proc. Int. Conf. Commun., Devices Intell. Syst. (CODIS)*, Dec. 2012, pp. 556–559.

[54] S. Das, J. K. M. S. U. Zaman, and R. Ghosh, ‘‘Generation of AES S-boxes with various modulus and additive constant polynomials and testing their randomization,’’ *Procedia Technol.*, vol. 10, pp. 957–962, 2013.

[55] B. R. Gangadari and S. R. Ahamed, ‘‘Analysis and algebraic construction of S-Box for AES algorithm using irreducible polynomials,’’ in *Proc. 8th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2015, pp. 526–530.

[56] A. Gehani, T. LaBean, and J. Reif, ‘‘DNA-based cryptography,’’ in *Aspects of Molecular Computing* (Lecture Notes in Computer Science), vol. 2950. Berlin, Germany: Springer, 2003, pp. 167–188.

[57] K. Ning, ‘‘A pseudo DNA cryptography method, *CoRR*, vol. abs/0903.2693, 2009. [Online]. Available: http://arxiv.org/abs/0903.2693

[58] A. H. S. Al-Wattar, R. Mahmod, Z. A. Zukarnain, and N. I. Udzir, ‘‘Generating a new S-Box inspired by biological DNA,’’ *Int. J. Comput. Sci. Appl.*, vol. 4, no. 1, pp. 32–42, 2015.

[59] A. H. Al-Wattar, R. Mahmod, Z. A. Zukarnain, and N. I. Udzir, ‘‘A new DNA-based S-box,’’ *Int. J. Eng. Technol. IJET-IJENS*, vol. 15, no. 4, pp. 1–9, 2015.

[60] F. A. Kadhim, G. H. A. Majeed, and R. S. Ali, ‘‘Proposal new s-box depending on DNA computing and mathematical operations,’’ in *Proc. Al-Sadeq Int. Conf. Multidisciplinary IT Commun. Sci. Appl. (AIC-MITCSA)*, May 2016, pp. 1–6.

[61] A. K. Farhan, R. S. Ali, H. R. Yassein, and N. M. G. Al-Saidi, ‘‘A new approach to generate multi s-boxes based on RNA computing,’’ *Int. J. Innov. Comput., Inf. Control*, vol. 16, pp. 331–348, 2020.

[62] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain, ‘‘Biometric cryptosystems: Issues and challenges,’’ *Proc. IEEE*, vol. 92, no. 6, pp. 948–960, Jun. 2004.

[63] A. Webster and S. Tavares, ‘‘On the design of S-boxes,’’ in *Advances in Cryptology-CRYPTO* (Lecture Notes in Computer Science), vol. 218. Berlin, Germany: Springer, 1986, pp. 523–534.

[64] C. Adams and S. Tavares, ‘‘Good s-boxes are easy to find,’’ in *Advances in Cryptology-CRYPTO* (Lecture Notes in Computer Science), vol. 435. New York, NY, USA: Springer, 1990, pp. 612–615.

[65] Lu, Zhu, and Wang, ‘‘A novel S-Box design algorithm based on a new compound chaotic system,’’ *Entropy*, vol. 21, no. 10, p. 1004, Oct. 2019.

[66] S. Farwa, T. Shah, and L. Idrees, ‘‘A highly nonlinear S-box based on a fractional linear transformation,’’ *SpringerPlus*, vol. 5, no. 1, p. 1658, Dec. 2016.

[67] L. Cui and Y. Cao, ‘‘A new S-box structure named affine-power-affine,’’ *Int. J. Innov. Comput., Inf. Control*, vol. 3, no. 3, pp. 751–759, Jun. 2007.

[68] M. T. Tran, D. K. Bui, and A. D. Duong, ‘‘Gray S-box for advanced encryption standard,’’ in *Proc. Int. Conf. Comput. Intell. Secur.*, Dec. 2008, pp. 253–258.

[69] A. Altaleb, M. S. Saeed, I. Hussain, and M. Aslam, ‘‘An algorithm for the construction of substitution box for block ciphers based on projective general linear group,’’ *AIP Adv.*, vol. 7, no. 3, Mar. 2017, Art. no. 035116.

[70] M. F. Khan, A. Ahmed, and K. Saleem, ‘‘A novel cryptographic substitution box design using Gaussian distribution,’’ *IEEE Access*, vol. 7, pp. 15999–16007, 2019, doi: 10.1109/ACCESS.2019.2893176.

[71] M. F. Khan, A. Ahmed, K. Saleem, and T. Shah, "A novel design of cryptographic SP-network based on gold sequences and chaotic logistic tent system," *IEEE Access*, vol. 7, pp. 84980–84991, 2019, doi: 10.1109/ACCESS.2019.2925081.

[72] E. Tanyildizi and F. Ozkaynak, "A new chaotic S-box generation method using parameter optimization of one dimensional chaotic maps," *IEEE Access*, vol. 7, pp. 117829–117838, 2019, doi: 10.1109/ACCESS.2019.2936447.

[73] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu, and J. Liu, "A novel block encryption algorithm based on chaotic S-box for wireless sensor network," *IEEE Access*, vol. 7, pp. 53079–53090, 2019, doi: 10.1109/ACCESS.2019.2911395.

[74] D. Lambić, "A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design," *Nonlinear Dyn.*, vol. 100, no. 1, pp. 699–711, Mar. 2020, doi: 10.1007/s11071-020-05503-y.

[75] Q. Lu, C. Zhu, and X. Deng, "An efficient image encryption scheme based on the LSS chaotic map and single S-box," *IEEE Access*, vol. 8, pp. 25664–25678, 2020, doi: 10.1109/ACCESS.2020.2970806.

[76] Y.-Q. Zhang, J.-L. Hao, and X.-Y. Wang, "An efficient image encryption scheme based on S-boxes and fractional-order differential logistic map," *IEEE Access*, vol. 8, pp. 54175–54188, 2020, doi: 10.1109/ACCESS.2020.2979827.

ÖZNUR ŞENGEL was born in İstanbul, Turkey. She received the B.S. and M.S. degrees in computer engineering from İstanbul Kültür University, İstanbul, in 2009 and 2013, respectively. She is currently pursuing the Ph.D. degree in computer engineering with İstanbul University-Cerrahpaşa, İstanbul.

Since 2010, she has been working as a Research Assistant at the Computer Engineering Department, İstanbul Kültür University. Her research interests include cryptography, information security, and bioinformatics. She has three journal articles and published and presented seven international conference papers and a national conference paper.

MUHAMMED ALİ AYDıN received the B.S. degree from İstanbul University, İstanbul, Turkey, in 2001, the M.Sc. degree from Istanbul Technical University, İstanbul, in 2005, and the Ph.D. degree from İstanbul University, in 2009, all in computer engineering.

He was a Postdoctoral Research Associate with the Department of RST, Telecom SudParis, Paris, France, from 2010 to 2011. He has been working as an Associate Professor at the Computer Engineering Department, İstanbul University-Cerrahpaşa, since 2009. He has also been the Vice Dean of the Engineering Faculty and the Head of the Cyber Security Department, since 2016. He received ten research projects consisting of over Turkey from local industries in Turkey and the İstanbul University-Cerrahpaşa Research Foundation. He has authored 20 journal articles and published and presented 70 papers at international conferences. His research interests include cryptography, network security, information security, and optical networks.

AHMET SERTBAŞ was born in İstanbul, Turkey. He received the B.S. and M.S. degrees in electronic engineering from Istanbul Technical University, İstanbul, in 1986 and 1990, respectively, and the Ph.D. degree in electric-electronic engineering from İstanbul University, İstanbul, in 1997.

Since 2000, he has been an Assistant Professor, an Associate Professor, and a Professor with the Computer Engineering Department, İstanbul University, and a Professor with the Computer Engineering Department, İstanbul University-Cerrahpaşa, since 2018. His research interests include image processing, artificial intelligence, computer arithmetic, and hardware security. He has 19 articles in indexed SCI-SCIE journals and many journal articles not indexed SCI-SCIE and international conference papers.