Wireless Personal Communications (2019) 105:993–1007
https://doi.org/10.1007/s11277-019-06132-4

Fuzzy Genetic Elliptic Curve Diffie Hellman Algorithm for Secured Communication in Networks

Priya Sethuraman[1] · P. S. Tamizharasan[2] · Kannan Arputharaj[3]

Published online: 6 February 2019

Abstract
More computations have to be done through less powerful mobile devices which includes ultra modern wearables. The huge overhead lies in the processing of the humongous key space each and computation of the intelligible message. The uniqueness of the elliptic curve cryptography (ECC) lies in the processing of data using shorter keys which are capable to achieve the performance of long key requirement of RSA. In order to reduce the overhead involved in the computation of less powerful mobile devices the fuzzy genetic elliptic curve Diffie Hellman is proposed in this paper. The intelligent rules are used for ranking during key selection process, multi attribute decision making model with fuzzy reasoning for obtaining keys and genetic algorithms for effective optimization of computation in ECC contributes to obtain the proposed FGECDH algorithm.

Keywords FGECC · RSA · ECC · MADM

1 Introduction

Elliptic key cryptography (ECC) based algorithms are renowned for functions like Key Exchange [1] and Digital Signature [2] and hence the National Security Agency (NSA) has strongly recommended ECC. Due to the advantages possessed by the ECC, it is further refined with fuzzy rules and genetic algorithms and hence introduced the evolution of fuzzy genetic version of ECC. Fuzzy logic is the computation approach based on degrees of truth rather than the usual Boolean logic on which the modern computer is based [3]. The genetic algorithm [4] is a process of populating new items by the creation of crossover among two or more existing set of basics and thus creates a new offspring by the process of mutation(changing it minutely) at every step. Now, this new offspring is evaluated by a certain mechanism to check the feasibility of this offspring to satisfy the required surviving criteria. ECC [5] irrespective of it's advantages, has a distinctive disadvantage, like

✉ Priya Sethuraman
priyaprabhaharan@gmail.com

[1] Anna University, Chennai, Tamilnadu, India

[2] Department of Information Technology, Jerusalem College of Engineering, Chennai, Tamilnadu, India

[3] Department of IST, Anna University, Chennai, Tamilnadu, India

complexity and implementation difficulty compared to RSA [6]. It increases the implementation errors, which is a direct parameter to hinder the security of the algorithm. The complexity of the ECC lies in the generation and maintenance of the required key set.

Plenty of cryptographic algorithms [7, 8] are available. Number theory [9, 10] plays a vital role in each of the cryptographic algorithms. Each has a specific use case, and cater to a very specific kind of problem. This problem keeps changing with time and as the problem changes, modifying the existing system to support that change becomes necessary.

Mobile computing is the standard on which all kinds of cryptographic enhancements will be measured for the next decade. With the introduction of Apple Pay by Apple and Google Wallet by Google and numerous mobile based transaction portals are becoming the common place for most of the financial transaction. This creates a need for a cryptographic algorithm which is resource stringent but delivers yet delivering the security of a highly sophisticated system becomes a critical necessity. Genetic equations are largely known for its minimal resource utilization, fuzzy rules for decision making and elliptic curve cryptography is by far, the most sophisticated algorithm in cryptography. In this paper, a new cryptographic algorithm called fuzzy genetic elliptic curve Diffie Hellman algorithm is proposed to enhance the security of wireless networks. The proposed FGECDH algorithm is an amalgamation of fuzzy, genetic equations and elliptic curve cryptography with the best of the trio worlds and each of the technologies negates the disadvantages of the other. The contribution of this paper is to propose a new algorithm for secure transmission of messages which is mobile compliant suiting the day today requirements.

This paper demonstrates a mechanism, where we use a simplified version of fuzzy to rank the key selection involved in multi attribute decision making (MADM) algorithm, genetic algorithm to automate the key set generation hence increasing the security of ECCDH in the process. The rest of the paper is organized as follows: Sect. 2 presents the genetic programming approach for decision making, Sect. 3 presents the fuzzy decision making systems. Section 4 presents the fuzzy genetic decision making systems. Section 5 presents the proposed fuzzy genetic elliptic curve Diffie Hellman. Section 6 depicts the performance evaluation. Finally, Sect. 7 deals with the conclusion and future work.

2 The Genetic Programming Approach for Decision Making

Evolutionary computing is a division of artificial intelligence (AI) which gets the basic evolution principles and implements them in a program. Rather than the programmer giving a solution to the program, the program literally evolves a solution. A genetic algorithm (GA) [11–13] is a particular form of evolutionary computing which is used in this paper. A GA is similar to biological evolution. The fitness function is the key parameter of any problem handled by GA. In fact this is another heuristic. It is a feature of the problem that the programmer can use to guide the evolutionary process. In computing terms, this is simply a method which allows us to rank all potential solutions from hopeless to perfect. If such a ranking is possible then GA methods will at least in principle, be able to evolve a solution to the problem.

In this work, genetic algorithm is used for the generation of every valid key in a shorter period of time where partial computation is done at the background. Using this process, the set of candidate solutions is generated as follows: most of the real life applications in key generation for enhancing the security of communication need an optimal and efficient method for selecting a key with less number of bits but providing high security equivalent

to large number of bits in the order of 4096 bits. In such a scenario, the time complexity of the algorithm for key generation is less when a genetic algorithm (GA) based optimization approach is used. Moreover, the security proof in such algorithms has guaranteed that the time complexity for decrypting the text without knowing is increasing exponentially in such a way that its complexity is in discrete logarithmic type. Therefore, a genetic algorithm based key generation approach is proposed in this work in which optimization is carried out by introducing an efficient fitness function. In the past, genetic algorithm was introduced as a heuristic technique [14] which is used to find the shortest path in a graph by applying the heuristic function. When it was compared with other search technique including hill climbing, the solution provided by genetic algorithms was more optimal than the other existing heuristic search technique [15]. Since, GA works on the set of candidate populations from possible solution that it performs better than most of the heuristics that works based on probabilistic methods and hence such techniques are not deterministic in nature. On the other hand, each individual to be used as a parent in the GA process will contribute more uniformly to provide the most optimal solution. In this work, the GA based key generation and optimization technique starts with a set of large prime numbers which are taken from the points on the elliptic curve represented by the equation

$$y^2 = x^3 + sx + t \tag{1}$$

are considered to be the initial chromosomes called as the initial set of candidate keys called the initial population where s and t are constants. Moreover, the large prime numbers P and Q taken from the above equation are used as the candidate keys for the optimization process in the key generation algorithm. Such candidate keys are consisting of large prime numbers which are encoded using binary encoding to provide 4096 numbers and two such candidates are selected at a time to form the initial set of chromosomes called parents in the GA re-production process. The solution obtained from this population is checked with the activation function called the fitness function shown in (2).

$$Fitness\ Value = WT_1 * Count(Zeros) + WT_2 * Count(Ones). \tag{2}$$

In (2), the values of WT1 and WT2 were fixed as 0.4 and 0.6 after performing repeated number of experiments using values between 0 and 1. The parents have been selected from the candidate chromosomes by using the random subset generation method since the set of candidates consists of a finite set of large prime numbers that have been obtained from the elliptic curve. The fitness function used in this work performs better global optimization when it is compared with other greedy techniques. The main advantage of applying the subset generation method is that it reduces the key space by controlling the direction of search. During the candidate key generation from the elliptic curve, the proposed algorithm starts taking a large set of keys and then it applies the subset generation technique to reduce the candidate sets from $K_1, K2, \ldots K_n$ to $KA_1, KA_2, \ldots KA_m$ where $m < n$. Finally, the tournament selection method is used in this work to select the suitable chromosomes as parents which are obtained from the subsets generated using the subset generation process.

In this work, genetic programming is introduced to set up a set of candidate solutions to the security problem. At this initial stage, they will be totally random. These candidate solutions are then ranked by means of a fitness function. Even though they were generated as completely random attempts, it is possible to rank them from first to one hundredth. Of course, the chances are that even the first in this ranking will not look like any sort of solution to the problem. The algorithm then follows natural selection by dispensing with, say, the bottom n members of the population. They are replaced by combinations of the top n members of the population in a process inspired by biological reproductions. Portions of

the pairs of the top n are combined in a process known as crossover [16, 17]. This is the computational equivalent of biological reproduction. With the population up to the original value again, the process of selection by means of fitness function [18] is repeated. Then the crossover is repeated, and so on for many generations. Eventually the program will converge towards an acceptable solution to the problem-always assuming that there is at least one solution and the fitness function can guide the selection process towards it. Multiple attribute decision making algorithm [19, 20] plays a vital role in key selection.

3 The Fuzzy Decision Making Systems

The fuzzy temporal rules [21] for developing an intelligent pattern classification system for analyzing periodic patterns in medical diagnosis based on symptoms which is used to identify the diseases more accurately by applying the fuzzy decision making approach. A fuzzy temporal approach [22] has a fuzzy temporal logic which predicts the energy level based on past and present data leading to suitable rotation of cluster heads based on energy to improve the network performance. An intelligent agent and fuzzy swarm optimization approach in optimal routing [23] improved the performance of the network by increasing the packet delivery ratio and by reducing the energy consumption using fuzzy swarm optimization.

4 The Fuzzy Genetic Decision Making Systems

Genetic fuzzy systems are fuzzy systems constructed based on genetic programming, which mimics the process of natural evolution, to identify its structure and parameter. To automatically identify and build a fuzzy system traditional linear optimization techniques have several limitations, given the high degree of non linearity of the output. Hence the genetic programming has been imbibed on fuzzy systems in this work for proper identification of structure and parameters of fuzzy systems in the decision making process involved in the ranking process of key selection. The fuzzy rules applied on the genetic algorithm yields better results and it is evident in the novel weighted fuzzy C-means clustering based on immune genetic algorithm for intrusion detection [24] to carry out the communication through the cluster heads in order to develop an efficient intrusion detection system for wireless networks.

5 The Proposed Fuzzy Genetic Elliptic Curve Diffie Hellman(FGECDH)

The fuzzy genetic elliptic curve Diffie Hellman is the fuzzy genetic version of ECC along with the inclusion of multi attribute decision making model (MADM) in key selection. The baseline in imposing fuzzy genetic in ECC is that it has the capability to analyze the design process in the space of rule sets by coding the model in chromosome. The primary step in the design of the fuzzy genetic algorithm is to analyze and decide which parts of the fuzzy system are subjected to optimization by genetic algorithm coding in chromosomes. The ranking for key selection in MADM is based on fuzzy rules.

ECC has shorter key which is as strong as larger keys of RSA. Hence the genetic version has the refined key selection methodology having shorter keys produces the best security.

The key size is the prime factor which reflects the encryption time and decryption time. The proposed FGECDH being mobile compliant is designed to have lesser encryption and decryption time and hence the key size and key selection is given at most importance. The security of the algorithm is strengthened by identifying the prime parameters and optimizing it. The parameters identified for optimization are selecting the keys, generating the plot, obtaining the co-ordinates, encrypting the message, decrypting the message. Most of the algorithms concentrate more on generating the plot, obtaining the co-ordinates whereas the primary focus of this paper is to optimize the key selection. In this work, the elliptic curve $x^3 + sx + t = y^2$ has been considered in which P and Q are two points on the elliptic curve. $Q = A + B$ where, A and B are any other two points that have been selected from the elliptic curve. Here, the members $a, b, c, d \ldots$ are the members of the key set selected from the original key set by applying the subset generation process. The original key set is consisting of the set of points representing the keys obtained from the elliptic curve.

Algorithm 1 The FGECDH algorithm

1: Initially a 128 bit key-phrase set is chosen for ECC.
2: A set of keys are selected to plot the P and $A + B$ quotient of the elliptic curve.
3: This selection is done by the genetic algorithm to obtain the initial set of keys using Multi Attribute Decision Making Model (MADM) which incorporates fuzzy rules for key ranking
4: The elliptic curve is plotted and the message is encoded with the obtained keys from the above step.
5: Only the mutation limit is sent as the public key
6: The receiver performs reverse mutation to process the cryptic message obtained, by using the same crossover method and limiting the mutation to the publicly available limit.
7: The key hence obtained by the above mentioned method is used to decode the message using normal ECC decoding method

5.1 Key Selection Using MADM

The key selection involves a modified version of the multiple attribute decision making algorithm. MADM consists of two levels. In level 1 the judgments are aggregated on the basis of goals and alternative decisions. The derived judgments are sorted in level 2. According to the algorithm the final decision is obtained by gradually altering the attribute value available at each decision node.

The key set comprises of be any set of preselected keys that can be used for $ECCQ = \{a, b, c, d \ldots\}$ where $a, b, c, d \ldots$ are the preselected keys. At each instance the genetic algorithm selects any key, $(b, c, d \ldots)$ for a specific a. This is achieved by a careful crossover and eliminating the most illogical candidate at each stage.

The attribute selected at each node is programmatically maintained to help in the subsequent stages of decryption to arrive at the exact key as generated for the encryption. The MADM based model used for the generation of keys in FGECDH are illustrated as follows. Let the alternatives set be defined as $A = \{a_i | i = 1, \ldots, n\}$ and the goal set be defined as $G = \{g_i | j = 1, \ldots, m\}$. r_{ij} represents alternative i corresponding to goal j and $w_j \in \mathbb{R}$ denotes weight of goal j. The fuzzy membership function of i mapped to j is denoted as $\mu_{R_{ij}(r_{ij})}$ on $\mathbb{R}$. Likewise, The fuzzy membership function of w_j is denoted as $\mu_{wj(w_j)}$. Assumption: Every fuzzy set is normalized.

Step 1 The x_i is evaluated based on r_{kj} and w_j Considering a function $g\colon \mathbb{R}^{2n} \to \mathbb{R}$ is defined by,

$$g(z) = \frac{\sum_{j=1}^{m} w_j r_j}{\sum_{j=1}^{m} w_j} \tag{3}$$

with $z = (w_1, \ldots, w_m, r_1, \ldots, r_m)$. The member function of the product space $\mathbb{R}^{2n}$ is as follows

$$\mu_{zi}(z) = min\{min_j = 1 \ldots m(\mu_{wj}(w_j), min_{k=1,\ldots,m}(\mu_{R_{ik}}(r_k))\} \tag{4}$$

the fuzzy set $Z = (\mathbb{R}^{2m}, \mu_{zi})$ induces a fuzzy set $R_i = (\mathbb{R}, \mu_{Ri})$ using function g with the membership function

$$\mu_{Ri}(r) = sup_{z:\, g(z)=r} \mu_{zi}(z) r \in R. \tag{5}$$

$\mu_{Ri}(r)$ is the alternative x_i final rating based on rank is evaluated in step 2.

Step 2 Baas and Kwakernaak proposed that if x_i has got higher rating a valid algorithm naturally selects x_i and makes this as its preferred alternatives as

$$\{i \in I | r_i \geq r_j, \forall j \in I\}, \quad I = \{1, \ldots, n\}.$$

Baas and Kwakernaak [25] proposed in their model two fuzzy sets to choose preferability of the alternative.

The conditional set $(I|R)$ is determined along with the characteristic function.

$$\mu_{(I|R)}(i|r_1, \ldots, r_n) = \{{}^{1}_{0}\, if\ r_j \forall j \in I, else\} \tag{6}$$

The above stated function denotes that xi is a member of preferred set if and only if the following equation is satisfied.

$$r_i \geq r_j \forall_{i,j} \in I \tag{7}$$

The R defined on $\mathbb{R}^{2n}$, the fuzzy set $\mathrm{R} = (\mathbb{R}^n, \mu_R)$ is depicted with the membership function

$$\mu_R(r_1, \ldots, r_n) = \min(\mu_{Ri(ri)}) \tag{8}$$

The fuzzy set and conditional fuzzy set together includes $I = (I, \mu_i)$ and its associated function is

$$\mu_{I^{(i)}} = sup(min)\{(\mu_{I|R)}(i|r_1, \ldots, r_n), \mu_R(r_1, \ldots, r_n)\} \tag{9}$$

If x_i is not the best alternative then the scenario may be represented by r_ij which is another fuzzy set.

The final ratings for $r_1, \ldots, r_n$ is

$$p_i = r_i - \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} r_j \quad i\ takes\ the\ value\ 1\ to\ n. \tag{10}$$

Inorder to increase the accuracy of key ranking fuzzy rules are used. This work uses Fuzzy IF...THEN rules for ranking effectively. Based on the definition of fuzzy logic [26, 27], overlapping regions are used to determine the degree of fuzziness. In this work, fuzzy rules have been formed by applying the triangular membership function. The fuzzy linguistic variables are shown in Table 1. Fuzzy rules for normalized key and ranking is shown in Table 2.

The associated mapping for the rating of r_i is $\mathbb{R}^{2n} \rightarrow \mathbb{R}$

$$\mu_{pi}(p) = \sup \mu_i(r_1, \ldots, r_n) \mu_R(r_1, \ldots, r_n). \tag{11}$$

Table 1 Fuzzy rules for initial decision

Score	Class
Low (L)	$0.0 < P_i < 0.5$
Medium (M)	$0.4 < P_i < 0.8$
High (H)	$0.7 < P_i < 0.85$
Medium high (MH)	$0.8 < P_i < 0.95$
Very high (VH)	$0.9 < P_i < 1.0$
Threshold (th)	0.65

Table 2 Fuzzy rules for normalized key and ranking

Final rating (P_i)	Normalized key size (s)	Rank
$0 < P_i < L$	$s \leq th$	10
$0 < P_i < L$	$s > th$	9
$L < P_i < M$	$s \leq th$	8
$L < P_i < M$	$s > th$	7
$M < P_i < H$	$s \leq th$	6
$M < P_i < H$	$s > th$	5
$H < P_i < MH$	$s \leq th$	4
$H < P_i < MH$	$s > th$	3
$MH < P_i < VH$	$s \leq th$	2
$MH < P_i < VH$	$s > th$	1

The stage at which the most suitable key-pair is obtained is maintained as a public key X.

This process of crossover is a hidden function $f(x)$.

$$f(x) = 011010100010101010. \tag{12}$$

This binary code has to be reverse mutated with the same genetic algorithm used before

$$f - 1(011010100010101010) = x \tag{13}$$

Such x, y and other details are then mapped to their original keys using original ECC. Hence the whole process of selecting the key set for the required ECC is automated and managed by the designated computer itself hence reducing the complexities and providing an artificial intelligence based support to ECC hence making it more robust

5.2 The Security Feature Enhancement

Let the key function be

$$f(k) = f(ka + kb) \tag{14}$$

where ka and kb are the two selected keys respectively. If the encryption technique's execution time includes the time involved in deciding the initial keys, Then the total time involved is

$$t(k) = t(ka) + t(kb) \tag{15}$$

Usually an ECC encryption algorithm takes time for the following:

selecting the keys = $t(k)$.
Generating the plot = $t(p)$.
Obtaining the co-ordinates = $t(o)$.
Encrypting the message = $t(e)$.

The optimization of ECC algorithm involves a consideration of all these factors. The same factors has to considered for improving the security also. Each of these steps has to individually secure in order to affect the overall security performance of the ECC algorithm. Since most of the algorithms mainly concentrate on generating the plots ($t(p)$) and obtaining the co-ordinates ($t(o)$), the maximum time and security performance of any algorithm is staggered. True potentials involved in the key selection and encryption process and usually ignored which is the primary focus of this paper. Initially the whole without the help of a genetic algorithm the time function of ECC is

$$f(encryption) = (t(k) + (t(p) + t(o)) + t(e))^{keySize} \tag{16}$$

where $t(e)$ is the time taken to actually encrypt the pattern. Similarly the decryption is done using

$$f(decryption) = (t(lk) + (t(p) + t(o)) + t(d))^{keySize} \tag{17}$$

where $t(lk)$ is the time considered for the local key which plays a vital role in decryption When FGECDH is used, $t(k)$ is omitted as the key generation and identification is done prematurely and due to this, even the plotting time $t(p)$ and the co-ordinates obtaining time is highly minimised as it is partially computed In the background for every valid key generated by the Genetic algorithm. bp is a variable partially computed in the background that represents the plot value. bo is a variable that partially computed in the background that for obtaining the coordinates. Hence the updated time functions for FGECDH are

$$f(fgecdhEncryption) = (t(bp + bo) + t(e))^{keySize} \tag{18}$$

$$f(fgecdhDecryption) = (t(bp + bo) + t(d))^{keySize} \tag{19}$$

The security feature of ECC is also enhanced in a similar fashion.

$$fs(encryption) = (s(k) + (s(p) + s(o)) + s(e))^{keySize} \tag{20}$$

$$fs(decryption) = (s(lk) + (s(p) + t(o)) + t(d))^{keySize} \tag{21}$$

$$fs(gecdhEncryption) = (S(bp + bo) + S(e))^{keySize} \tag{22}$$

$$fs(gecdhDecryption) = (S(bp + bo) + S(d))^{keySize}. \tag{23}$$

6 Performance Evaluation

The performance evaluation environment used is Visual studio 2013 Integrated Development Environment. The performance evaluation environment further uses Visual C++ programming language with Block Cipher Cryptography 2015 as Software Development Kit. The security analyser used is Hackman Tools 2015

Table 3 Encryption time analysis (ms)

File size (MB)	AES_DFA	DES_DFA	ECC_DH	FGECDH
1	3085	2785	3708	3368
2	5392	4961	6417	5814
3	8417	7731	10,016	9058
4	11,289	10,611	13,594	12,513
5	14,192	13,163	16,999	15,691

Table 4 Decryption time analysis (ms)

File size (MB)	AES_DFA	DES_DFA	ECC_DH	FGECDH
1	2953	2886	3746	3372
2	5354	5089	6599	5823
3	8286	7721	9931	9082
4	11,239	10,473	13,538	12,420
5	14,208	13,144	17,055	15,736

6.1 Simulation Scenario 1: Varying File Size

The scenario 1 of the simulation is to study the performance of the proposed FGECDH over the other advanced encryption standard differential fault analysis (AES_DFA), data encryption standard differential fault analysis (DES_DFA), elliptic curve cryptography Diffie Hellman (ECC_DH) by varying the file size. The four key parameters such as encryption time, decryption time, memory usage and security are considered.

In Table 3, encryption time is directly proportional to the file size. The encryption time of the FGECDH is lesser compared to the ECC_DH. This is achieved because genetic versions of the algorithms are capable of processing quickly by matching the data from the learned set. The values enlisted are from the operations involving learning set for the first time. The secondary operations in the same category take very little time to encrypt making the algorithm more mobile friendly.

Similarly, Table 4 reveals that the decryption time of the FGECDH is lesser compared to ECC_DH but higher than the other cryptographic algorithms like AES_DFA, DES_DFA. Even though AES_DFA, DES_DFA have lesser encryption and decryption time the security provided by the two algorithms are lesser than the ECC_DH and FGECDH. To avoid system dependent complexity analysis, the proposed model provides the complexity of the existing algorithms namely AES_DFA, DES_DFA and the proposed algorithm using the Big ∅ notation and hence it is not depending on the system configuration. The complexities of the algorithms using Big O notations are as follows:

$$AES\_DFA = \varnothing(2K)$$
$$DES\_DFA = \varnothing(2K/2)$$
$$ProposedAlgorithm = \varnothing\left(2^{\sqrt{K}}\right)$$

where K is the key size.

Table 5 Memory consumption analysis (bytes)

File size (MB)	AES_DFA	DES_DFA	ECC_DH	FGECDH
1	6046	5746	8061	8031
2	6714	6372	8889	8890
3	10,245	9444	13,219	13,135
4	13,528	12,637	17,700	16,999
5	16,804	15,690	22,127	21,336

Table 6 Security analysis %

File size (MB)	AES_DFA	DES_DFA	ECC_DH	FGECDH
1	85	79	95	96
2	84	75	94	95
3	84	76	94	96
4	86	74	94	96
5	85	75	93	95

The memory consumption enlisted in Table 5 reveals that the ECC_DH and FGECDH have almost similar memory consumption and it is found that the FGECDH has slightly less memory consumption than ECC_DH. The reason behind this is the selection of smaller key size using MADM which are as strong as larger key size. Moreover the fuzzy rules contributed to the ranking

From Table 6, it is clear that the proposed FGECDH out performs all the other cryptographic algorithms by providing good security. Even though security is ensured the memory consumption is not compromised and encryption and decryption is done in lesser time than ECCDH. This is achieved due to the accuracy in key selection by incorporating fuzzy rules. The security analysis percentage is calculated based on the time complexity indicating the time taken for decryption without knowing the key with each algorithm with a relative scale of DES algorithm. Here, the attacks considered are Denial of Service attacks, Probe, User to Root (U2R) and Remote to User (R2L).

6.2 Simulation Scenario 2: Varying Key Size

The scenario 2 of the simulation is to study the performance of the proposed FGECDH over the other advanced encryption standard differential fault analysis (AES_DFA), data encryption standard differential fault analysis (DES_DFA), elliptic curve cryptography Diffie Hellman (ECC_DH) by varying the key size. The four key parameters such as encryption time, decryption time, memory usage and security are considered. The key size is taken small but the security is high irrespective of the key size being small. It can give same security as the larger key size. The MADM is used during key generation and the genetic version of the algorithm have refined it and give better encryption time, decryption time, almost similar consumption of memory and better Security as enlisted in Tables 7, 8, 9 and 10.

Table 7 reveals that FGECDH increase with increase in the key size. AES_DFA, DES_DFA can encrypt better than ECC_DH but the security quotient of the latter algorithms holds good. The prime objective of the message transfer is the security and

Table 7 Encryption time analysis (ms)

Key size	AES_DFA	DES_DFA	ECC_DH	FGECDH
32b	1787	1674	3438	3068
64b	3666	3513	6510	5530
128b	5464	5295	10,013	8584
256b	7482	7086	13,515	11,431
512b	9383	8858	17,047	14,597

Table 8 Decryption time analysis (ms)

Key size	AES_DFA	DES_DFA	ECC_DH	FGECDH
32b	1923	1743	3562	3187
64b	3548	3500	6587	5666
128b	5574	5194	10,039	8631
256b	7533	7076	13,632	11,538
512b	9306	9001	17,146	14,603

Table 9 Memory consumption analysis (bytes)

Key size	AES_DFA	DES_DFA	ECC_DH	FGECDH
32b	4806	4503	6629	6689
64b	6143	5893	7539	7479
128b	9142	8779	11,121	11,137
256b	12,218	11,671	14,817	13,645
512b	15,191	14,625	18,513	17,998

Table 10 Security analysis %

Key size	AES_DFA	DES_DFA	ECC_DH	FGECDH
32b	76	67	84	86
64b	76	70	87	89
128b	81	74	91	93
256b	82	77	92	95
512b	85	80	94	97

then comes the encryption and decryption time. Even though FGECDH have greater encryption time than AES_DFA and DES_DFA, it has proved to perform better than the ECCDH by having refined it with genetic technology and MADM algorithm with fuzzy decision making systems.

Table 8 reveals the decryption time analysis of the four algorithms AES_DFA, DES_DFA, ECC_DH, FGECDH on the basis of key size which is the prime parameter of performance analysis. FGECDH performs better than the ECCDH. It can perform much better in the consequent runs as genetic versions can perform better by retrieving result from the learned set fuzzy decision making in generation of keys yields better performance which is revealed in the table decryption time analysis.

The memory consumption analysis of the four algorithms are enlisted in Table 9. The memory consumption of the proposed algorithm is due to the incompatibility of the

proposed algorithm in the crypto processors of the mobile devices. Once the algorithm is declared best in terms of security proposals may be sent to the concerned agencies for incorporating the compatibility of the proposed algorithm with its crypto processors. Basically the algorithm is designed to utilize lesser resource.

Table 10 reveals the ultimate objective of the proposed algorithm FGECDH by having accomplished higher security than the other three algorithms AES_DFA, DES_DFA, ECC_DH. The security quotient is directly proportional to the key size.

Figure 1 shows the security comparisons between the proposed algorithm and the existing algorithms namely ECC_DH, modified DES [28], enhanced AES [29], DES + network coding [30].

From Fig. 1, it is observed that the performance of the proposed security algorithm called FGECDH is better when it is compared to the other security algorithms such as ECC_DH, modified DES, enhanced AES and DES with Network Coding. This is due to the fact that the proposed algorithm uses a genetic based key generation and optimization approach in which a new activation function is introduced. Moreover, the elliptic key cryptography algorithm is modified with optimization technique leading to overall improvement in performance and hence the time consumption is reduced and security level is enhanced. This improvement in performance of FGECDH shown in Tables 3, 4, 5, 6, 7, 8 and 9 is due to the effective handling of uncertainty in the growth of values of Pi using fuzzy rules. Moreover, the rules used in this work can be used to modify the exiting decisions by adding additional constraints. This leads to make decisions with non-monotonic reasoning starting with a default reasoning to make an initial decision. The final decision is made non-monotonically by the final decisions rules which considers qualitative reasoning and hence is able to handle incomplete information by making prediction using past and present data.

7 Conclusion and Future Work

Encryption is a highly volatile phenomenon and updating before the existing methods fail is highly necessary. This paper emphasises this by providing a solid update to an already robust security system. This not only enhances the security but also tackles

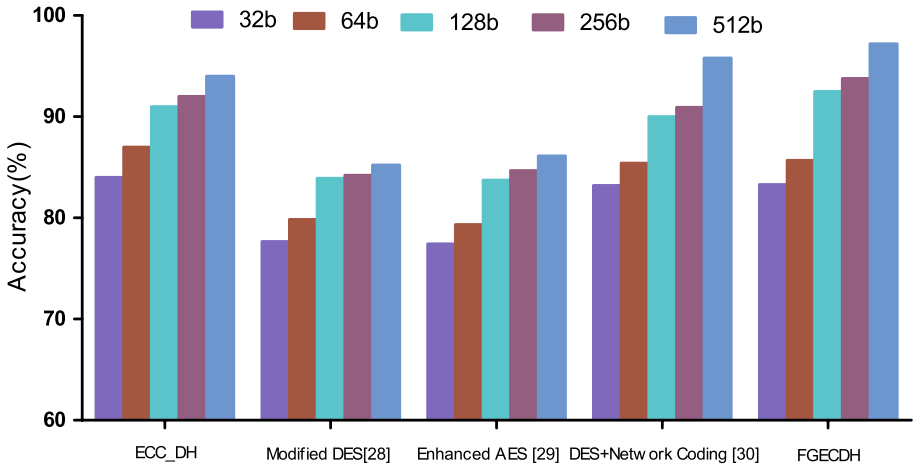

Fig. 1 Comparative analysis of accuracy in security algorithms

few vital flaws existing in the previous methods. The proposed FGECDH is the secured algorithm which can decrypt and encrypt at lesser time compared to ECC. FGECDH outperforms the other algorithms by maintaining the same processing requirements and just varying the amount of data based on the level of encryption required which has been achieved by using fuzzy rules for decision making and genetic algorithm for optimization. Hence, this makes FGECDH a mobile friendly algorithm. The future work suggests to reduce the memory consumption by making the FGECDH compliance with the crypto processors of all models in mobile devices.

References

1. Bogdanov, A., Knudsen, L. R., Leander, G., Standaert, F. X., Steinberger, J., & Tischhauser, E. (2012). Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations. In D. Pointcheval & T. Johansson (Eds.), *Annual international conference on the theory and applications of cryptographic techniques* (Vol. 7237, pp. 45–62). Berlin, Heidelberg: Springer.
2. Libert, B., Peters, T., Joye, M., & Yung, M. (2015). Linearly homomorphic structure-preserving signatures and their applications. *Designs, Codes and Cryptography*, *77*(2–3), 441–477.
3. Zadeh, L. A. (2012). *Computing with words: Principal concepts and ideas* (Vol. 277). Berlin: Springer.
4. Lin, F.-T., & Kao, C.-Y. (1995). A genetic algorithm for ciphertext-only attack in cryptanalysis. In *Systems, man and cybernetics, 1995. Intelligent systems for the 21st century, IEEE international conference on* (Vol. 1, pp. 650–654). IEEE.
5. Faugère, J.-C., Perret, L., Petit, C., & Renault, G. (2012). Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In D. Pointcheval & T. Johansson (Eds.), *Annual international conference on the theory and applications of cryptographic techniques* (Vol. 7237, pp. 27–44). Berlin, Heidelberg: Springer.
6. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120–126.
7. Needham, R. M., & Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM*, *21*(12), 993–999.
8. Seredynski, F., Bouvry, P., & Zomaya, A. Y. (2004). Cellular automata computations and secret key cryptography. *Parallel Computing*, *30*(5), 753–766.
9. Bhasin, H. (2012). Corpuscular random number generator. *International Journal of Information and Electronics Engineering*, *2*(2), 197.
10. Burke, L. (1999). A review of optimization in operations research Ronald L. Rardin Prentice-Hall, 1998, 919pp, ISBN 0-02-398415-5. *Iie Transactions*, *31*(3), 279–280.
11. Goldberg, D. (1989). Genetic algorithms in search. *Optimization, and Machine Learning*.
12. Ruttor, A., Kinzel, W., Naeh, R., & Kanter, I. (2006). Genetic attack on neural cryptography. *Physical Review E*, *73*(3), 036121.
13. Khan, F. U., & Bhatia, S. (2012). A novel approach to genetic algorithm based cryptography. *International Journal of Research in Computer Science*, *2*(3), 7.
14. Holland, J. H., & Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.
15. Stein, G., Chen, B., Wu, A. S., & Hua, K. A. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. In *Proceedings of the 43rd annual Southeast regional conference* (Vol. 2, pp. 136–141). ACM.
16. Kaya, Y., Uyar, M., et al. (2011). *A novel crossover operator for genetic algorithms: Ring crossover*. arXiv preprint arXiv:1105.0355.
17. Picek, S., & Golub, M. (2010). Comparison of a crossover operator in binary-coded genetic algorithms. *WSEAS Transactions on Computers*, *9*, 1064–1073.
18. Jhingran, R., Thada, V., & Dhaka, S. (2015). A study on cryptography using genetic algorithm. *International Journal of Computer Applications, 118*(20), 10–14.
19. Ma, Z., & Zeng, S. (2014). Confidence intuitionistic fuzzy hybrid weighted operator and its application in multi-criteria decision making. *Journal of Discrete Mathematical Sciences and Cryptography*, *17*(5–6), 529–538.
20. Zimmermann, H.-J. (2011). *Fuzzy set theory and its applications*. Berlin: Springer.
21. Ganapathy, Sethukkarasi, Sethukkarasi, R., Yogesh, P., Vijayakumar, P., & Kannan, A. (2014). An intelligent temporal pattern classification system using fuzzy temporal rules and particle swarm optimization. *Sadhana*, *39*(2), 283–302.

22. Selvi, M., Logambigai, R., Ganapathy, S., Ramesh, L. S., Nehemiah, H. K., & Arputharaj, K. (2016). Fuzzy temporal approach for energy efficient routing in WSN. In *Proceedings of the international conference on informatics and analytics* (p. 117). ACM.
23. Selvi, M., Logambigai, R., Ganapathy, S., Nehemiah, H. K., & Arputharaj, K. (2017). An intelligent agent and FSO based efficient routing algorithm for wireless sensor network. In *Recent trends and challenges in computational models (ICRTCCM), 2017 second international conference on* (pp. 100–105). IEEE.
24. Ganapathy, S., Kulothungan, K., Yogesh, P., & Kannan, A. (2012). A novel weighted fuzzy C-means clustering based on immune genetic algorithm for intrusion detection. *Procedia Engineering*, *38*, 1750–1757.
25. Baas, S. M., & Kwakernaak, H. (1977). Rating and ranking of multiple-aspect alternatives using fuzzy sets. *Automatica*, *13*(1), 47–58.
26. Zadeh, L. A. (1998). Roles of soft computing and fuzzy logic in the conception, design and deployment of information/intelligent systems. In O. Kaynak, L. A. Zadeh, B. Türkşen & I. J. Rudas (Eds.), *Computational intelligence: Soft computing and fuzzy-neuro integration with applications* (pp. 1–9). Berlin, Heidelberg: Springer
27. Logambigai, R., Ganapathy, S., & Kannan, A. (2018). Energy-efficient grid-based routing algorithm using intelligent fuzzy rules for wireless sensor networks. *Computers & Electrical Engineering*, *68*, 62–75.
28. Mihret, Z., & Ahmad, M. W. (2016). The reverse engineering of reverse encryption algorithm and a systematic comparison to DES. *Procedia Computer Science*, *85*, 558–570.
29. Riyaldhi, R., Kurniawan, A., et al. (2017). Improvement of advanced encryption standard algorithm with shift row and S. box modification mapping in mix column. *Procedia Computer Science*, *116*, 401–407.
30. Tang, H., Sun, Q. T., Yang, X., & Long, K. (2018). A network coding and DES based dynamic encryption scheme for moving target defense. *IEEE Access*, *6*, 26059–26068.

Priya Sethuraman is the Deputy Registrar of Anna University having more than 10 years of teaching and research experience. Her areas of interest include network security, mobile computing, parallel computing and deep learning.

P. S. Tamizharasan received his master degree in Computer Science and Engineering from Anna University, Chennai in 2008. His research interests include digital system design, multi/manycore architecture, deep learning and GPU computing.

Kannan Arputharaj is a professor in the Department of IST, Anna University, Chennai having more than 30 years of teaching and research experience. His areas of interest include network security, data mining, artificial intelligence and software engineering.