Generation of Addition Chain using Deterministic Division Based Method

Mr. K. Mani
Associate Professor in Computer Science
Nehru Memorial College, Puthnampatti
Trichy, INDIA – 621 007
nitishmanik@yahoo.com

Abstract— Efficiency of a cryptosystem depends not only on the security it provides, but also it increases the operational speed thereby it reduces the time taken for encryption and decryption. In most number-theoretic cryptographic algorithms like RSA, ElGamal, Massey-Omura etc., the encryption and decryption functions often involve raising large elements (x^e mod n) of group fields $GF(2^n)$ or large powers (exponents). If they are not properly implemented, they increase the operational time which ultimately lead to customer dissatisfaction. Thus, group exponentiation has received much attention by the researchers in recent times owing to their central role in modern cryptography and it is effectively computed using the concept of addition chain. Several deterministic and stochastic algorithms have been proposed in literature to generate the shortest addition chains. Normally, stochastic algorithms produce the optimal addition chains but it is not obtained from the single run which is a time consuming process. Thus, a deterministic algorithm has been proposed which is simply based on division in this paper and it is compared with other deterministic and stochastic algorithms.

Keywords – public-key cryptography, modular exponentiation, and addition chain

I. INTRODUCTION

The concept of public-key cryptography was presented by Whitefield Diffie and Martin Hellman at National Computer Conference[1]. A few months later, their seminal paper "New Directions in Cryptography" was published[2]. Since 1976, numerous public-key cryptography algorithms have been proposed. A public-key cryptographic system consists of a plaintext message space M, a ciphertext message space C, an encryption key space K, the decryption key space K´, an efficient key generation algorithm G: $N \rightarrow K \times K'$, an efficient encryption algorithm E: $M \times K \rightarrow C$, and an efficient decryption algorithm D: $C \times K^{-1} \rightarrow M$. For $k \in K$, $x \in M$, and $y \in C$, we denote by $y=k_e(E(x))$ and $x=k_d(D(y))$. In public-key cryptosystem, encryption and decryption processes use different keys; i.e., for every $k_e \in K$, there exists $k_d \in K'$ and $k_e \neq k_d$, where k_e and k_d are encryption and decryption key respectively. The security of the public-key cryptosystem lies in the hardness of some underlying mathematical problem which is believed to be computationally difficult.

Most of the public-key cryptosystems like RSA, ElGamal, etc., modular exponentiation is the cornerstone operation which plays a vital role in performing encryption/decryption operations. They often involve raising large elements of some group fields to large powers. Successive multiplication is normally used to perform modular multiplication but it is a time consuming process. For example, to compute x^e based on paper-and-pencil method, it requires (e-1) multiplication of x. i.e $x^1 \rightarrow x^2 \rightarrow x^3 \rightarrow \cdots \rightarrow x^{e-1} \rightarrow x^e$.

Fast exponentiation is becoming increasingly important with the widening use of encryption. In order to improve the time requirements of the encryption process, minimizing the number of multiplication is essential. Several authors have proposed the fast modular exponentiation like left-to-right, right-to-left, multi-exponentiation by interleaved exponentiation, sliding window and window NAF exponential methods etc. A large number of field exponentiation algorithms have been reported in the literature. Some of the known algorithms are: binary, m-ary, adaptive m-ary, power tree, factor method, etc. But these algorithms have a common problem that they strive to keep the number of required multiplications to compute the exponent as low as possible through the use of a particular heuristic. Also, the said algorithms are not considered to yield an optimal addition chain for every possible field size.

It is a known fact that larger the size of the field utilized, harder the problem of optimizing the computation of the field exponentiation. This is because a heuristic strategy is normally used to find the optimal addition chain for hard optimization problems. Since these problems have huge search spaces, they do not provide the guarantee on the quality of the solutions. Normally, a heuristic method starts from a non-optimal solution (partial solution) and iteration. After performing some iteration, it improves the solution until a reasonable valid solution could be achieved. Thus, to improve the partial solution which is considered at the initial stage, either deterministic or probabilistic search criteria is used.

Several methods are available to generate the addition chain of minimal length. They are classified into two types namely: deterministic and evolutionary algorithms. In deterministic type of algorithms, steps used are predetermined. Examples of these types include binary, the factor, the window method, and sliding window

methods. Evolutionary algorithms are stochastic optimization methods which are inspired by the idea of either natural evolution or social behaviour. Examples of these types include: Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Artificial Immune System (AIM) [3].

In deterministic type, given a fixed set of initial conditions, the optimized solution obtained by a deterministic heuristic will remain unchanged from run to run. On the other hand, repeated executions of stochastic heuristic may produce different final solutions. When compared with deterministic heuristic, stochastic heuristic like GA (Genetic Algorithm) and EP (Evolutionary Programming) always produce the optimal addition chain for the given exponent. Thus, there is a need to develop a deterministic algorithm based on merely division called as Division Based Method (DBM) which would generate the shortest addition chain as generated by stochastic heuristic.

The rest of the paper is organized as follows. The concept of addition chain is discussed in section 2. Section 3 shows the proposed division based method to generate the addition chains. A comparative analysis of addition chains generated by the proposed method and the existing methods for some soft and hard exponents is discussed in section 4. Finally,Section 5 ends with conclusion.

II. ADDITION CHAIN

To understand the relevance of the work, the following mathematical concepts are used.

Definition (Modular Exponentiation): Let α be an integer in the range [1, n-1], and e an arbitrary positive integer. Then, modular exponentiation is defined as the problem of finding the unique integer that satisfies the equation:

$$\beta = \alpha^{e} \bmod n \quad (1)$$

It is noted that addition chains are normally used to find the correct sequence of multiplications in performing exponentiation [3].

Definition (Addition Chain): The problem of determining the correct sequence of multiplications required for performing a modular exponentiation can be elegantly formulated by using the concept of addition chains. Formally, it is defined as:

An addition chain [4] for an integer e is a sequence of integers

$$1= a_0, a_1, a_2, \ldots, a_r = e \quad (2)$$

with the property that

$$a_i = a_j + a_k, \text{ for some } k \leq j < i, \text{ for all } i = 1, 2, 3, \ldots, r \quad (3)$$

i.e., the first number is numbered one; every subsequent number is the sum of two early numbers and e occurs at the end of the chain.

The shortest length (l) for which there exists an addition chain for e is denoted by l(e). For the given exponent, it is possible to generate several addition chains, and the smallest length is better. If the shortest addition chain is found, then it will be useful to reduce the number of multiplications required in the exponentiation. Based on the shortest addition chain, modular exponentiation is performed very fast. Finding the best addition chain is very difficult, but it is enough to find near optimal addition chain. However, for the given integer, finding at least one of the shortest addition chains is an NP-hard problem. For example, the possible addition chains for the integer e (exponent) = 6271 is

(i) 1-2-4-8-10-20-30-60-90-180-360-720-1440-2880-5760-5970-6150-6240-6270-6271

(ii) 1-2-3-6-12-24-48-96-292-384-768-1536-3072-6144-6240-6264-6270-6271

(iii) 1-2-3-5-10-20-23-46-92-194-298-391-782-1564-1567-3134-6168-6271

The length of the addition chain for (i), (ii), and (iii) are 19, 17, and 17 respectively. The shortest length l(6271) is 17.

III. PROPOSED DIVISION BASED METHOD (DBM)

In this work, to generate an addition chain for the given exponent e, e is divided by 2 and only the integral part of the division is considered. Let it be d_1. This d_1 is again divided by 2 to get d_2. The process is repeated until e is reduced to 1. Let it be d_n. Reverse the quotients as $d_1, d_2, d_3 \ldots d_{n-1}, d_n$. Assign $e_1 = d_n$, $e_2 = d_{n-1}$, $e_3 = d_{n-2} \ldots e_{n-1} = d_1$, and $e_n = e$. Let $a_0, a_1, a_2 \ldots a_b \ldots a_{m-3}, a_{m-2}, a_{m-1}, a_m, \ldots a_{r-3}, a_{r-2}, a_{r-1}, a_r = e$ are the numbers which are to be computed in the addition chain. Without loss of generality, let $a_0=1$ and $a_1= 2$. To obtain a_3, it is either 3 or 4 depending on the value of e_3. If a_3 is 3, then a_4 is either 5 or 6. If $a_3 =4$, then a_4 is either 6 or 8 etc. In this way, a suitable addition chain of up to 10 is formed. The possible addition chains are:

$$1\rightarrow2\rightarrow3\rightarrow5\rightarrow7;\ 1\rightarrow2\rightarrow3\rightarrow5\rightarrow8;\ 1\rightarrow2\rightarrow3\rightarrow5\rightarrow10;\ 1\rightarrow2\rightarrow4\rightarrow8$$

$$1\rightarrow2\rightarrow3\rightarrow6\rightarrow9;\ 1\rightarrow2\rightarrow4\rightarrow5\rightarrow10;\ 1\rightarrow2\rightarrow4\rightarrow5\rightarrow9$$

These addition chains are called as base addition chains in the proposed model. For example, if e=20, then $d_1=10$; $d_2=5$; $d_3=2$; $d_4=1$. Then, the base addition chain is $1\rightarrow2\rightarrow3\rightarrow5\rightarrow10$. By selecting the suitable base addition chain, the rest of numbers in the addition chain are determined as follows.

Let the last number in the base addition chain is a_b and select the suitable e_l which is exactly or nearer to a_b. To obtain the next number $a_{b+1,}$ a_b is multiplied by 2. i.e. $a_{b+1=} a_b \times 2$. Compare a_{b+1} with e_{l+1} and find the difference (d). If d is zero, then $a_{b+2 =} a_{b+1 *} 2$. Otherwise, check whether the difference is found in the base addition chain or not. If it is found, then $a_{b+2 =} (a_{b+2} + d)$. Otherwise, choose the number from the base addition chain which is nearer to d. It is noted that in some situation the number 1 or 2 in the base addition chain may be nearer to d. If it is the case, it could be avoided because it may increase the length of the addition chain. Thus, after two or three numbers in the addition chain are generated, the difference d=1 or d=2 may disappear. The process is repeated until a_n= e or a_{n-1}+d= e.

If the exponent e is an even number, then it is enough to generate the addition chain for e/2. In that case, to obtain the addition chain for e, e/2 is multiplied by 2. Thus, if l(e/2)=r, then l(e)= r+1. The generated addition chains may be the shortest addition chain or very close to the shortest one which can greatly improve the efficiency of modular exponentiation in RSA and other public-key algorithms. The generation of the shortest addition chain based on the proposed division method is illustrated in algorithm 1.

Algorithm 1: Generation of Addition Chain for an Exponent e Based on DBM

/ / The proposed division based algorithm reads the exponent e and it generates the shortest addition chain for e

INPUT**:** exponent e

OUTPUT: addition chain a_i , i →1,…,r with a_1 =1 and a_r = e

1. i←1; j ←1; e_1= e
2. if (e % 2 = 0) then e ← int (e/2)
   / / Computation of quotients when e is divided by 2 recursively
3. do
   d_i ← int(e_1 / 2); i ← i+1; e_1← e_1 / 2
   while ($d_i \geq 1$)
4. k ← i; e_{k+1} = e
   / / assigning d_i to e_i
5. while (k ≥ 1) do
   e_j ← $d_{k;}$ j ← j +1; k ← k -1
   end while
   / / determination of base addition chain
6. if (e%5 and e%3=0) then l ← 6; m ← 6
   base_add_chain ← (1→2→3→5→10 → 15) go to main_computation
   end if
7. for k from 3 to i
   begin
   if (e_k =9) then
   l← k; m ← 5; base_add_chain ← (1→2→3→6→9) || (1→2→4→5→9)
   else if (e_k=8) then
   l ← k ; m ← 4; base_add_chain ← (1→2→4→8)
   else if (e_k=7) then
   l ← k; m ← 5; base_add_chain ← (1→2→3→5→7)
   else if (e_k=6) then
   l ← k; m ← 4; base_add_chain ← (1→2→3→6)
   else if (e_k=5) then
   l ← k; m ← 4; base_add_chain ← (1→2→3→5)
   else (e_k =4) then
   l ← k; m ← 3; base_add_chain ← (1→2→4)
   end if
   end for

main_computation:

8. add_chain ← base_add_chain

9. while ($a_m \neq e$) do
begin
$d \leftarrow e_l - a_m$
if (d=0) then
$m \leftarrow m+1$; $l = l+1$; $a_m \leftarrow a_{m-1} \times 2$; add_chain ← add_chain || a_m; $s_1 \leftarrow e-2 \times a_m$
end if
if((d =1) and (a_m = e-1)) then
$m \leftarrow m+1$; $a_m \leftarrow a_{m-1} + d$; add_chain ← add_chain || a_m; $a_{m+1} \leftarrow e$
exit from while loop
end if
if((d >1) and (s_1 in base_add_chain) then
$m \leftarrow m+1$; $a_m \leftarrow a_{m-1} + s_1$; add_chain ← add_chain || a_m
exit from the do loop
end if
if((d >2) and (d in base_add_chain) then
$m \leftarrow m+1$; $a_m \leftarrow a_{m-1} + d$; add_chain ← add_chain || a_m
end if
end while
10. l(e)=length(add_chain)
11. return (add_chain, l(e))

A. Generation of Addition Chains Based on DBM – Examples

i. To determine the addition chain for 31

Here, e=31, then d_1= 31/2=15, d_2=15/2=7, d_3=7/2=3, and d_4=3/2=1. Correspondingly, e_1=1, e_2=3, e_3=7, e_4=15, and e_5=e=31. The suitable base addition chain which is ≤10 is 1→2→3→5→7. Thus, a_1=1, a_2=2, a_3=3, a_4=5, and a_5=7. The last number in the base addition chain is 7 and the corresponding index value (m) is 5. The number (a_5=7) found in e_i is e_3=7 (exactly) and the corresponding index value (l) is 3. The difference $d=e_3-a_5=0$. Since d=0, compute $a_6=7\times2=14$. Now e_4=15 and $d=e_4-a_6=1$. Since d=1 and a_6 or $a_6+1 \neq e$, compute $a_7=a_6\times2=28$. Now, e_5=e=31(the given exponent), $d=e_5-a_7$ is computed as 3. Since 3 is found in the base addition chain and also $a_8=28+3=e(=31)$, the process is stopped. The shortest addition chain for 31 is: 1→2→3→5→7→14→28→31 and l(31) is 7.

ii. To determine the addition chain for 349

Here, e=349, then d_1=349/2=174, d_2=174/2=87, d_3=87/2=43, d_4=43/2=21, d_5=21/2=10, d_6=10/2=5, d_7=5/2=2, and d_8=2/2=1. Correspondingly, e_1=1, e_2=2, e_3=5, e_4=10, e_5=21, e_6=43, e_7=87, e_8=174, and e_9=e=349. The suitable base addition chain which is ≤10 is 1→2→3→5→10. Thus, a_1=1, a_2=2, a_3=3, a_4=5, and a_5=10. The last number in the base addition chain is 10 and the corresponding index value (m) is 5. The number a_5=10 is found in e_i, is e_4=10 and the corresponding index l is 4. Find $d=e_4-a_5=0$. Since d=0, compute $a_6=10\times2=20$. Now e_5=21. Find $d=e_5-a_6=1$. Since, d=1 and a_6 or $a_6+1 \neq e$, d=1 is not considered at this stage. Compute $a_7=a_6\times2=40$. Now e_6=43. Find $d=e_6-a_7=3$. Since 3 is found in the base addition chain $a_8=a_7+3=43$. Compute $a_9=a_8\times2=86$. Now e_7=87 and $d=e_7-a_9=1$. Since, d=1 and a_9+1 or $a_9 \neq e$. Compute $a_{10}=86\times2=172$. Now, e_8=174 and $d=e_8-a_{10}=2$. Since d=2 and $a_{10}+2 \neq e$. Compute $a_{11}=172\times2=344$. Now e_9=379. The difference $d=e_9-a_{11}=5$ which is found in the base addition chain. Thus, a_{12} is computed as: $a_{12}=344+5=349$. Since $a_{12}=e_9=e=349$(the given exponent) the process is stopped. The shortest addition chain for 349 is: 1→2→3→5→10→20→40→43→86→172→344→349 and l(349)=11. For other exponents the addition chains are generated in this manner.

IV. RESULTS AND DISCUSSION

Table 1 shows the comparison of addition chains generated by the DBM and the addition chains illustrated in the literature [5-9]. Tables 2 and 3 show that the length of addition chains generated by the proposed method for field exponentiation computations are almost same with additions chains generated by GA[10] and EP [11] respectively for small exponent. But, for large exponents, the proposed method generates addition chain with thin increase (at most three) in length when compared with GA and EP. Though there is a thin increase in length for some large exponent, these exponents are hard to optimize.

Table 4 shows the length of the optimal addition chains (beyond that reducing the length of the addition chain for the exponents is not possible) given in the literature [3]. Table 5 gives the addition chains generated

by the proposed DBM. The proposed method also yields the same length of the addition chains as illustrated in Table 4.

Table 1: Shortest Addition Chains for some Exponents

Proposed by Several Authors			Proposed Division Based Method		
exponent e	Addition Chain	Length l	Proposed by-Algorithm Name	Addition Chain	Length l
39	1-2-3-4-7-11-18-25-32-39	9	Febien et al Euclidean Addition Chains	1-2-3-6-9-18-36-39	7
95	1-2-4-5-7-14-21-42-84-91-95	10	Brun's -Genetic Algorithm	1-2-3-5-10-20-23-46-92-95	9
197	1-2-3-4-5-6-12-24-48-49-98-196-197	12	QinJiushao-Genetic Algorithm-	1-2-3-6-12-24-48-49-98-196-199	10
221	1-2-3-4-5-6-7-12-24-48-55-110-220-221	13	QinJiushao-Blocking algorithm	1-2-3-6-12-24-27-54-108-110-220-221	11
255	1-2-4-8-16-17-34-68-85-170-255	10	DongFiguo et al. Fast Fourier Transform	1-2-3-5-10-15-30-60-120-240-255	10
343	1-2-4-6-8-15-17-30-47-94-109-117-234-347	13	Brun's -Genetic Algorithm	1-2-3-5-10-20-40-80-85-170-340-343	11
349	1-2-4-8-9-17-34-68-136-272-340-349	11	Colin D-Walter-Exponentiation using division chain	1-2-3-5-10-20-40-63-86-172-344-349	11
445	1-2-3-6-7-12-24-42-55-110-111-222-444-445	13	Noboru KUNIHRO et al-run length method	1-2-3-4-6-12-24-27-54-108-111-222-444-445	13
22531	1-2-4-8-16-32-64-128-256-512-1024-1025-2048-2050-4096-4097-8192-16384-20481-22531	19	DongFiguo et al. Fast Fourier Transform	1-2-3-5-10-20-22-44-88-176-352-704-1408-2816-5632-11264-22528-22531	17
30578	1-2-4-8-16-32-64-128-256-257-514-1028-2056-4112-4368-4369-8736-8738-17472-21840-30578	20	DongFiguo et al. Fast Fourier Transform	1-2-3-5-7-14-28-56-59-118-236-472-477-954-1908-1911-3822-7644-15288-30576-30578	20
75064310	1-2-3-4-5-7-8-14-15-16-30-31-32-64-128-143-286-572-1144-2288-4576-4581-9162-9163-18326-36652-73304-146608-293216-586432-1172864-2345728-2345759-4691518-9383036-18766072-18766077-37532154-37532155-75064310	39	Noboru KUNIHRO et al-hybrid method	1-2-3-5-10-15-30-60-70-140-143-286-572-1144-2288-2290-4580-9160-9163-18326-36652-73304-146608-293216-293219-586438-1172876-1172879-2345758-4691516-4691519-9383038-18766076-37532152-37532155-75064310	35

Table 2 : Comparison of Addition Chain Generated by GA in[10] with Proposed DM for some Hard Exponents

e	Addition Chain Proposed in[10]	GA	Proposed Division based Addition Chain	DM
6271	1 - 2 - 3 - 6 - 12 - 24 - 48 - 96 - 192 - 384 - 768 - 1536 - 3072 - 6144-6240 - 6264 - 6270 - 6271	17	1-2-3-6-12-24-48-96-192-195-390-780-783-1566-3132-3135-6270-6271	17
11231	1 - 2 - 3 - 6 - 12 - 24 - 25 - 50 - 100 - 200 - 400 - 800 - 1600 - 3200 - 6400 - 9600 - 11200 - 11225 - 11231	18	1-2-3-5-10-20-40-43-86-172-175-350-700-1400-1403-2806-5612-5615-11230-11231	19
18287	1 - 2 - 3 - 6 - 9 - 15 - 30 - 45 - 47 - 94-188 - 190 - 380 - 760 - 1520 -3040 - 6080 - 12160 - 18240 - 18287	19	1-2-3-5-8-16-32-64-69-138-276-284-568-1136-1141-2282-2285-4570-9140-9143-18286-18287	21
34303	1 - 2 - 3 - 6 - 12 - 14 - 28 - 56 - 112 - 224 - 448 - 504 - 1008 - 2016 - 4032 - 8064 - 16128 - 32256 - 34272 - 34300 - 34303	20	1-2-3-5-8-16-32-64-128-133-266-532-535-1070-2140-4280-8570-8575-17150-34300-34303	20
65131	1 - 2 - 3 - 6 - 12 - 24 - 48 - 72 - 144 - 288 - 576 - 1152 - 2304 - 4608 - 4611 - 9222 - 18444 - 27666 - 55332 - 55908 - 65130 - 65131	21	1-2-3-5-10-15-30-60-63-126-252-504-507-1014-1017-2034-4068-8136-8141-16282-32564-65128-65131	22
110591	1 - 1 - 2 - 4 - 5 - 10 - 20 - 40 - 80 - 160 - 320 - 640 - 1280 - 2560 - 2570 - 5140 - 7710 - 12850 - 25700 - 51400 - 102800 - 110510 - 110590- 110591	22	1-2-3-5-10-13-26-52-104-208-213-426-431-862-1724-1727-3454-6908-6911-13822-27644-27647-55294-110588-110591	24
196591	1 - 2 - 3 - 6 - 12 - 15 - 30 - 60 - 120 - 240 - 480 - 720 - 1440 - 2880 - 5760 - 11520 - 23040 - 46080 - 92160 - 184320 -19584 - 196560 - 196590- 196591	23	1-2-3-5-10-20-23-46-92-95-190-380-383-766-1532-1535-3070-6140-6143-12286-24572-49144-49147-98294-196588-196591	25
357887	1 - 2 - 3 - 4 - 8 - 16 - 32 - 64 - 128 - 256 - 257 - 514 - 771 - 11542 - 3084 - 6168 - 12336 - 24672 - 49344 - 49347 - 98691 - 148038 - 296076- 345423 - 357759 - 357887	24	1-2-3-5-10-20-40-43-86-172-344-349-698-1396-2792-2795-5590-11180-22360-22365-44730-44735-89470-178940-178943-357886-357887	26
685951	1 - 2 - 4 - 6 - 7 - 14 - 21 - 42 - 84 - 168 - 336 - 504 - 840 - 1680 - 3360 - 6720 - 13440 - 26880 - 53760 - 57120 - 114240 - 228480- 342720 - 685440 - 685944 - 685951	25	1-2-3-5-7-10-20-40-80-160-167-334-668-1336-2672-2679-5358-10716-21432-42864-42871-85742-141484-342968-342975-685950-68591	26
1176431	1 - 2 - 4 - 5 - 10 - 15 - 19 - 38 - 76 - 152 - 304 - 608 - 612 - 1224 - 2448 - 4896 - 9792 - 19584 - 29376 - 58752 - 117504 - 235008 - 352512 - 587520 - 1175040 - 1176264 - 1176416 - 1176431	27	1-2-3-5-8-16-32-35-70-140-280-285-570-1140-1148-2296-4592-4595-9190-18380-36760-36763-73526-147052-294104-588208-588210-1176420-1176428-1176431	29
2211837	1 - 2 - 3 - 6 - 9 - 15 - 30 - 60 - 120 - 126 - 252 - 504 - 1008 - 2016 - 4032 - 8062 - 16128 - 16143 - 32286 - 64572 - 129144 - 258288- 516576 - 1033152 - 2066304 - 2195448 - 2211591 - 2211717 - 2211837	28	1-2-3-8-16-32-64-67-134-268-526-539-1078-2156-2159-4318-8636-8639-17278-34556-34559-69118-138236-138239-276478-552956-552959-1105918-2211836-2211837	30
4169527	1 - 2 - 3 - 6 - 12 - 24 - 48 - 96 - 192 - 384 - 768 - 1536 - 2304- 4608 - 9216 - 18432 - 36864 - 73728 - 147456 - 294912 - 589824 - 589825- 1179650 - 1769475 - 3538950- 4128775 - 4165639- 414167943 - 4169479 - 4169527	29	1-2-3-5-7-14-28-31-62-124-127-254-508-1016-2032-2035-4070-8140-8143-16286-32572-32574-65148-130296-260592-260595-521190-1042380-2084760-4169920-4169527	31
7624319	1 - 2 - 3 - 6 - 12 - 18 - 36 - 72 - 144 - 288 - 576 - 1152 - 1224 - 2448 - 4896 - 9792 - 19584 - 39168 - 78336 - 156672 - 313344- 626688 - 1253376 - 1254600 - 1274184 - 1274185 - 2548370- 5096740 - 6370925 - 7624301 - 7624319	30	1-2-3-5-7-14-28-56-112-224-231-462-465-930-1860-3720-7440-7445-14890-29780-59560-119120-119127-238254-238259-476518-953036-1906072-1906079-3812158-7624316-7624319	31

Table 3 : Comparison of Addition Chains Generated by GA and EP in[11] with Proposed DM for some Hard Exponents

e	Addition Chain Proposed in[11]	GA	PE		Proposed Division based Addition Chain	DM
3243679	1-2-3-5-10-20-40-80-83-166-206-412-824-1648-3296-6592-9888-19776-39552-79104-158208-316416-632832-632915-1265830-2531660-3164575-3243679	27	27		1-2-3-6-12-24-48-49-98-196-392-395-790-1580-3160-3166-6332-12664-12670-25340-50680-101360-202720-405440-810880-1621760-3243520-3243618-3243667-3243679	29
3493799	1-2-3-4-8-16-32-64-128-256-512-515-1030-2060-4120-8240-16480-32960-65920-131840-263680-263683--527366-1054732-2109464-3164196-3427879-3493799	27	27		1-2-3-6-12-24-48-51-53-106-212-424-848-1696-1704-3408-6816-6821-13642-13647-27294-54588-109176-109181-218362-436724-873448-1746896-1746899-3493798-3493799	30
3459835	1-2-4-5-10-20-25-35-70-140-280-560-1120-2240-4480-8960-17920-35840-71680-71705-143410-286820-573640114728 0-1147305-2294610-3441915-3459835	27	27		1-2-3-5-10-20-50-100-200-210-420-422-844-1688-3376-6752-6757-13514-27028-54056-108112-108117-216234-432468-432478-864956-1729912-1729917-3459834-3459835	29
3235007	1-2-3-6-12-24-27-54-108-162-324-648-1296-1944-3888-7776-15552-31104-62208-124416-248832-248859-497691-995382-1990764-2986146-3235005-3235007	27	27		1-2-3-6-12-24-48-96-98-196-197-394-788-1576-1579-3158-6316-12632-25264-25270-25273-50546-101092-202184-404368-808736-808748-1617496-3234992-3235004-3235007	30
3230591	1-2-3-4-8-16-32-64-128-131-262-524-1048-2096-4192-8384-16768-33536-67072-134144-268288-536576-538672-538803-1077606-2155212-2694015-3230591	27	27		1-2-3-5-6-12-24-48-96-192-197-394-788-1576-3152-6304-6309-12618-25236-50472-100944-201888-403716-807432-1614864-1615258-1615282-1615294-3230588-3230591	29
3182555	1-2-3-6-9-18-36-72-144-288-576-1152-1728-2880-5760-11520-2304046080-92160-184320-187200-187209-374418-748836-1497672-2995344-3182553-3182555	27	27		1-2-3-6-12-24-48-96-192-194-388-776-1552-3104-3107-6214-12428-24856-49712-99424-99427-198454-198908-397816-765632-1591638-1591276-3182552-3182555	29
3440623	1-2-4-8-16-32-48-96-192-193-386772-773-1546-3092-6184-12368-24736-49472-50245-99717-199434-398868-797736-1595472-3190944-3390378-3440623	27	27		1-2-3-5-6-12-24-48-96-192-204-209-418-836-839-1678-3356-6712-13424-26848-53696-107392-214784-429568-859136-1718272-3436544-3439900-3440318-3440527-3440623	30
3926651	1-2-4-5-9-18-36-72-144-288-576-1152-2304-4608-9216-18432-18437-36869-73738-92175-184350-368700-737400-811138-1548538-3097076-3908214-3926651	27	27		1-2-3-5-6-12-24-29-58-116-119-238-476-479-958-1916-3832-7664-7669-15338-30676-61352-122704-245408-245414-490828-490831-981662-1963324-3926648-3926651	30
3234263	1-2-3-6-12-24-48-96-192-194-388-776-1552-3104-3107-3155-6310-12620-25240-50480-100960-201920-403840-807680-1615360323072 0-3233875-3234263	27	27		1-2-3-5-10-12-24-48-96-192-197-394-788-1576-1579-3158-6316-12632-25264-25267-50534-101068-202136-202141-404282-808564-1617128-1617131-3234263	28
3352927	1-2-4-8-16-17-34-68-136-272-544-1088-2176-4352-8704-1740834816-69632-139264-278528-557056-1114112-1114656-1114724-1114741-2229482-3344223-3352927	27	27		1-2-3-6-12-24-48-96-192-384-768-816-1632-3264-6528-13056-26112-52224-10448-208896-417792-835584-1671168-3342336-3348864-3352812-3352896-3352920-3352926-3352927	29

The proposed method may gain some attention because it is deterministic whereas the other two approaches GA and EP are evolutionary algorithms. Further, to determine the optimum addition chain for the given exponent using evolutionary algorithms, the code is run for several times, preferably 30 independent times and the optimal addition chain for an exponent is obtained out of 30 independent runs.

The reason behind the proposed approach generates the optimal addition chain for the given exponent is that each time the exponent is reduced to half and the process is repeated until the exponent is reduced to one. To form the addition chain, the process starts with base addition chain to get the next number of the addition chain, and the last number in the base addition chain is multiplied by 2. The number obtained in this way is compared with numbers involved in the addition with the quotients obtained from the exponents when it is divided by two. If the difference between the two numbers is less than 10 and if it is found in the base addition chain, the difference is added to the number last generated in the base addition chain so that the vast difference between the two numbers could be avoided. Since, this checking process is performed at each stage; ultimately it produces the optimal addition chain.

Table 4 shows the length of optimal addition chain for some exponents up to 512 shown in [3]

length	Solutions
1	{2}
2	{3,4}
3	{5,6,8}
4	{7,9,10, 12,16,}
5	{11, 13, 14, 15, 17, 18, 20, 24, 32}
6	{19, 21, 22, 23, 25, 26, 27,28,30,33,34,36,40,48,64}
7	{29, 31, 35, 37, 38, 39, 41, 42, 43, 44,45,46,49,50,51,52, 54,56,60,65,66,68,72,80,96,128}
8	{47,53,55,57,58,59,61,62,63,67, 69,70,73,74,75,76,77,78,81,82, 83,84,85,86,88,90,92,97,98, 99,100,102,104,108,112,120,129, 130,132,136,144,160,192,256}
9	{71,79,87,89,91,93,94,95, 101,103,105,106,107,109,110,111,113,114, 115,116,117,118,119,121,122,123,124,125, 126,131,133,134,135,137, 138,140,145,146,147,148,149,150,152,153,154,156,161,162,163,164, 165,166,168,170,172,176,180,184,193,194,195,196,198,200,204,208, 216,224,240,257,258,260,264,272,288,320,384,512}

Table 5: Generation of addition chain based on proposed DBM for small exponents

Exponent (e)	Addition Chain	Length l(e)	Exponent (e)	Addition Chain	Length l(e)
2	1-2	1	37	1-2-3-6-9-18-36-37	7
3	1-2-3	2	38	1-2-3-6-9-18-19-38	7
4	1-2-4	2	39	1-2-3-6-9-18-36-39	7
5	1-2-3-5	3	41	1-2-3-5-10-20-40-41	7
6	1-2-3-6	3	42	1-2-3-5-10-20-21-42	7
8	1-2-4-8	3	43	1-2-3-5-10-20-40-43	7
7	1-2-3-5-7	4	44	1-2-3-5-10-11-22-44	7
9	1-2-3-6-9	4	45	1-2-3-5-10-20-40-45	7
10	1-2-3-5-10	4	46	1-2-3-5-10-20-23-46	7
12	1-2-3-6-12	4	49	1-2-3-6-12-24-48-49	7
16	1-2-4-8-16	4	50	1-2-3-6-12-24-25-50	7
11	1-2-3-5-10-11	5	51	1-2-3-6-12-24-48-51	7
13	1-2-3-6-12-13	5	52	1-2-3-6-12-13-26-52	7
14	1-2-3-5-7-14	5	54	1-2-3-6-12-24-48-54	7
15	1-2-3-5-10-15	5	56	1-2-3-5-7-14-28-56	7
17	1-2-4-8-16-17	5	60	1-2-3-5-10-15-30-60	7
18	1-2-3-6-9-18	5	65	1-2-4-8-16-32-64-65	7
20	1-2-3-5-10-20	5	66	1-2-4-8-16-32-33-66	7
24	1-2-3-6-12-24	5	68	1-2-4-8-16-17-34-68	7
32	1-2-4-8-16-32	5	72	1-2-3-6-9-18-36-72	7
19	1-2-3-6-9-18-19	6	80	1-2-3-5-10-20-40-80	7
21	1-2-3-5-10-20-21	6	96	1-2-3-6-12-24-48-96	7
22	1-2-3-5-10-11-22	6	128	1-2-4-8-16-32-64-128	7
23	1-2-3-5-10-20-23	6	47	1-2-3-5-10-20-40-45-47	8
25	1-2-3-6-12-24-25	6	53	1-2-3-5-12-13-26-52-53	8
26	1-2-3-6-12-13-26	6	55	1-2-3-6-12-24-48-54-55	8
27	1-2-3-6-12-24-27	6	57	1-2-3-5-7-14-28-56-57	8
28	1-2-3-5-7-14-28	6	58	1-2-3-5-7-14-28-29-58	8
30	1-2-3-5-10-15-30	6	59	1-2-3-5-7-14-28-56-59	8
33	1-2-4-8-16-32-33	6	61	1-2-3-5-10-15-30-60-61	8
34	1-2-4-8-16-17-34	6	62	1-2-3-5-10-15-30-31-62	8
36	1-2-3-6-9-18-36	6	63	1-2-3-5-10-15-30-60-63	8
40	1-2-3-5-10-20-40	6	67	1-2-4-8-16-32-33-66-67	8
48	1-2-3-6-12-24-48	6	69	1-2-4-8-16-17-34-68-69	8
64	1-2-4-8-16-32-64	6	70	1-2-4-8-16-17-34-35-70	8
29	1-2-3-5-7-14-28-29	7	73	1-2-3-6-9-18-36-72-73	8
31	1-2-3-5-10-15-30-31	7	74	1-2-3-6-9-18-36-37-74	8
35	1-2-4-8-16-17-34-35	7	75	1-2-3-6-9-18-36-72-75	8

V. CONCLUSION

The proposed division based addition chain generates the optimal addition chains for the small exponents which are exactly matched with addition chains generated by the latest methods. But, for some large exponents, there is very small increase in chains length (at most three). This result in turn reduces the encryption/decryption time due to the fact that most of the public-key algorithms exponentiation operation plays a key role in encryption/decryption process where the number of multiplications involved in the given exponent is equal to the length of the addition chain. Also, the proposed method is simple, deterministic and is based on simply the division.

REFERENCES

[1] Whitfield Diffie and Martin E. Hellman, "Multiuser Cryptographic Techniques", Proceedings of AFIPS National Computer Conference, 1976, pp. 109-112.
[2] Whitefield Diffie and Martin E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, 22(6), November 1976, pp. 644-654.
[3] N.Cruz-Cortés, F. Rodriquez-Hanriquez, and C. A. Coello-Coello, "An artificial immune system heuristic for generating short-addition chains", IEEE transactions on Evolutionary Computation, 12(1):1-24, February 2008.
[4] D. Knuth, The Art of Computer Programming–Semi Numerical Algorithms, Vol. 2, Addison-Wesley, Third Edition, 1998.
[5] Colin D. Walter, "Exponentiation Using Division Chains", IEEE transactions on Computers, Vol. 47, No. 7, 1998.
[6] Noboru Kunihiro and Hirosuke Yamamoto, "New Methods for Generation of Short Addition Chains", IEICE Trans. Fundamental, Vol. E83, No. 1, 2000.
[7] Noboru Kunihiro and Hirosuke Yamamoto, "Optimal addition chains classified by Hamming weight", IEICE, Technical Report, ISEC96-74, 1997.
[8] Nadia Nédjah and Luiza Macedo Mourelle, "Fast Pre-Processing for the Sliding Window Method Using Genetic Algorithms", International Journal of Computers, Systems and Signals, Vol. 4, No. 2, 2003.
[9] R. Begeron, J. Berstel, S. Brlek, and C. Duboc, "Addition chains using continued fractions", Journal of Algorithms, No. 10, 1989, pp. 403-412.
[10] Nareli Cruz-Cortés, Francisco Rodriguez-Henriquez, Raúl Juárez-Morales, and Carlos A. Coello Coello, "Finding Optimal Addition Chains Using a Genetic Algorithm Approach", Springer-Verlag, 2005, pp. 208-215.
[11] S. Domínguez-Isidro and E. Mezura-Montes, "An Evolutionary Programming Algorithm to Find Minimal Addition Chains", I Congreso Internacional de Ingeniería Electrónica, Instrumentación y Computación, de Junio del, Minatitlán Veracruz, México, 2011.