Design and Implementation of FPGA based Dual Key Encryption

B.Lakshmi,
Nehru Memorial College,
Puthanampatti

E. Kirubakaran
Bharath Heavy Electricals Ltd, Tiruchirappalli

T.N.Prabakar
Saranathan College of Engg, Tiruchirappall

ABSTRACT

In this paper, design and FPGA (Field Programmable Gate Array) implementation of embedded system for time based IDEA encryption is presented. Presently available encryption systems, suffer from Brute Force attack in which all key combinations are tried out to find the correct key. In such a case, the time taken for breaking the code depends on the system used for cryptanalysis. In the proposed system, time is used as a second dimension of the key. That is, the correct key entered at the correct time is needed for proper decryption. The proposed scheme uses a dynamically varying number of shifts for both encryption and decryption thereby the system needs to wait till that time and this forms the time based key input. Hence, the possibility of brute force attack is minimized and is free from the system capability. IDEA encryption algorithm is taken as the base and time factor is implemented as a second dimension of the key. The proposed system adds complexity to the IDEA encryption algorithm by including the time as a second dimension besides increasing the time required for cryptanalysis. As the proposed system needs concurrent execution and real time processing, the system is implemented using Altera Stratix III FPGA and the results are presented.

General Terms

Security

Keywords

Encryption, Decryption, Real Time Systems, Time Based Key, Brute Force attack, Cryptanalysis, FPGA

1. INTRODUCTION

Symmetric cryptosystems are based on algorithms in which identical keys are used for encryption and decryption. The secret key used for encryption/decryption should be known only to the legitimate senders and receivers in order to protect data. The key algorithms can be further divided into block ciphers for fixed transformations on plain-text data, and stream ciphers for time varying transformations. Block ciphers are the most basic type of ciphers and operate on the principle of encrypting/decrypting fixed size blocks. The size of the block is algorithm specific. For example the conventional cryptographic algorithms such as International Data Encryption Algorithm (IDEA) or Data Encryption Standard (DES) or BLOWFISH are symmetric, block-oriented cryptographic algorithms. Cryptanalysis is used to refer to any attempt to circumvent the security of cryptographic algorithms and protocols. Brute force attack means an exhaustive search of the key space. That is, trying all possible keys in order to recover the plain text from cipher text. In Brute Force attack, the expected number of trials before the correct key is found is equal to half of the key space. For example, if there are 2^{64} possible keys, a brute force attack would, on average, be expected to find a key after 2^{63} trials. Symmetric ciphers with keys of length of 64 bits are broken by brute force attacks. DES (Data Encryption Standard) a widely used block cipher [7] was broken by custom hardware in 1998. For applications requiring long term security, 128 bits is, as of 2004, a sufficient key length to defend against brute force attack. But embedded hardware, such as, COPACOBANA machine, which uses 120 FPGAs are used for exhaustive key search. In that case, the time taken to try one key is minimized and hence, it requires the key size to increase further. This paper presents a solution for this problem. In this paper, a method is proposed, in which time is taken as a second dimension of the key. Hence, even though the system is capable of performing so many million instructions per second, because of the time factor, brute force attack becomes difficult.

In the proposed system, IDEA is taken as the base algorithm. IDEA is a block cipher which uses 128 bit length key to encrypt successive 64 bit blocks of plain text. The procedure is quite complicated using sub keys (fifty two 16 bit sub keys) generated from the key to carry out a series of modular arithmetic and XOR operations on segments of 64 bit plain text block. In the proposed system, the sub key generation is made as a separate process and the keys are continuously rotated. Based on the time factor, the correct key is picked out and is given for encryption/decryption. The pick up time is the second dimension of the key. The process is explained in the following sections in detail. As this process requires the time interval to be measured accurately, conventional computers cannot be used because they lack real time operations. Real time operations can also be incorporated through additional software such as RTX (Windows real time extension) but implementing such systems in FPGA will make a hardware cryptographic embedded machine which can be specifically used for security aspects. The remainder of the paper is organized as follows: Section II says previous works and Section III gives the conventional IDEA algorithm. Section IV describes the proposed algorithm. Section V is concerned with FPGA implementation of both IDEA and proposed method. Section VI explains the comparison of IDEA and proposed method. Section VII describes the implementation results and Section VIII tells the suitability of the proposed scheme for computer based implementation. Section IX speaks the

advantages of the proposed system and Section X concludes the paper.

2. PREVIOUS WORKS

Symmetric Cryptosystems are based on algorithms in which identical keys are used for both encryption and decryption [18].Symmetric ciphers with keys of length up to 64 bits have been broken by brute force attacks. DES, a widely used block cipher which uses 56 bit keys, was broken by custom hardware in 1998 [10] and 12-round RC5 (64 bit blocks) is susceptible to a differential attack using 2^{44} chosen plaintexts by Distributed.net [11]. The 8-round DES was broken with 2^{21} known plaintexts and 16 rounds DES was broken with 2^{47} known plaintexts [12].

Various cryptographic algorithms have been proposed and implemented continuously to encrypt data effectively. The proposed time based scheme is a novel algorithm in [1]. An approach in which is a combination of Elliptic Curve Cryptography (ECC) and Data Encryption Standard (DES) is used [2]. The algorithm justified that DES being an efficient algorithm, the key can easily be revealed. An algorithm [3] based on the difficulty in factoring composite integer into its component primes is named as matrix based asymmetric bulk encryption algorithm. An encryption algorithm based on the application of Optimal Alphabetic Trees (OATs) is used [4]. A new word oriented stream cipher called *RAINBOW* uses two keys namely 'temporal key' and 'real key' for encryption, in which the temporal key is a sub key that is derived from the real key[5]. According to LIU Shu Tang [17] permutation and substitution methods are incorporated to present a strong image encryption algorithm.

[6] discusses an encryption algorithm that is suitable for VLSI (Very Large Scale Integrated Circuits) implementations. Diffie and Hellman [15] suggested an exhaustive search of the entire key space on a parallel machine. They estimate that a VLSI chip may be built which can search one key every microsecond. By building a search machine with a million such chips, all searching in parallel, 10^{12} keys can be searched per second. The key size of 128 bits can offer highest security with today's system but in near future as the speed of processing is continuously increasing, the key size has to be increased to protect the data.

It is said that large classes of weak keys have been found for block cipher algorithm IDEA. A chosen plain text attack recovers 32 bits of the key using 6 chosen plaintexts two under first key 4 under second key [13]. A successful differential attack was presented for 2.5 rounds of IDEA requiring 2^{10} chosen plain texts and 2^{32} time (one day in a standard PC). The algorithm has a few classes of weak keys [20]. In IDEA a class of 2^{23} keys exhibits a linear factor. For a certain class of 2^{35} keys the cipher has a global characteristic with probability 1[14]. Symmetric ciphers with keys of length up to 64 bits have been broken by brute force attack. DES, a widely used block cipher which uses 56 bit keys, was broken by custom hardware in 1998. For applications requiring long term security, 128 bits is as of 2004 is sufficient key length using symmetric key algorithms. NIST has recommended that 80 bit designs will be phased out by 2015. Image compression Encryption schemes (ICES) uses FPGA [16] for implementation of Discrete Wavelet Transform. The design and analysis of various hardware reconfigurable models of RC5 encryption algorithm is implemented using FPGA [18].

The above encryption algorithms are all based on various techniques that are susceptible to brute force attack. The security of an encryption system is primarily based on the size of the key used to encrypt the messages. The key size of 128 bits can offer highest security with today's system but in near future as the speed of processing is continuously increasing the key size also needs to be increased to protect the data. Also, when the number of systems used for brute force attack increases the key space can be shared between the systems that again weakens the security level. To summarize, the level of security is determined by the time taken for searching the key space by the system. The proposed algorithm tries to overcome this defect by introducing the time as a second dimension of the key.

3. CONVENTIONAL IDEA ALGORITHM

The existing system comprises of a number of conventional cryptographic algorithms such as International Data Encryption Algorithm (IDEA) or Data Encryption Standard (DES) or BLOWFISH which are symmetric, block-oriented cryptographic algorithms. In this paper IDEA is taken as the base algorithm. The IDEA operates on 64-bit plain text blocks and uses 128-bit keys, which makes it practically immune to brute-force attack. Figure 1 shows the block diagram of IDEA algorithm. IDEA is based upon a basic function, which is iterated eight times. The first iteration operates on the input 64-bit plain text block and the successive iterations operate on the 64-bit block from the previous iteration. After the last iteration, a final transformation step produces a 64-bit cipher block. IDEA uses both confusion and diffusion to encrypt the data. The 64-bit input data is divided into four 16-bit sub-blocks X1, X2, X3, and X4.

These four sub-blocks become the input to the first round of the algorithm. There are eight such rounds. In each round, the four sub keys are XORed, added, and multiplied with one another and with six 16-bit sub-keys. In order to retain the same data size modulo operations are performed. The encryption involves modular multiplication with a modulus of ((2^16) +1) and addition with a modulus of (2^16). Between the rounds, the results of second and the third sub-blocks are swapped. Finally, after the eighth round, the four sub-blocks are collected and combined with four sub-keys in an output transformation.

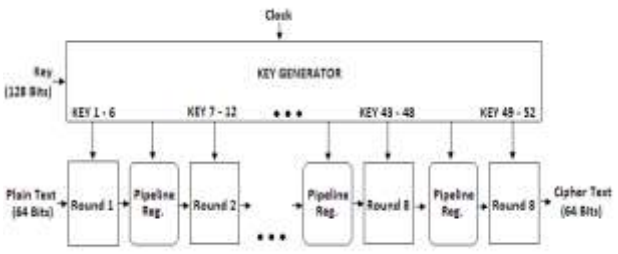

Figure 1.Block Diagram of Conventional IDEA Algorithm

The conventional algorithm suffers from the following drawbacks:

The main limitation of the existing system is that it purely depends on the key alone for encryption. The encryption is to be performed for a predetermined number of rounds.

The existing system is vulnerable to brute force attack and a proper cryptanalysis can easily bring out the message content easily. This is true because, in most cases, in a random code space of 'n' sets, the key can be found out with in 'n/2' sets of data.

4. PROPOSED ALGORITHM

In any security based algorithms the key must be known to the legitimate sender and receiver for encryption /decryption. The proposed algorithm involves the time by randomly selecting any four bit positions that comprise the time which must be known to the sender and receiver. The proposed scheme is a temporal IDEA algorithm. Figure 2 shows the block diagram of temporal algorithm. The plain text of size 64 bits is given to the first round of the IDEA encryption algorithm. The 128 bit key is given to the Sub key generator and Time Controller module. Once the process is started the first set of sub keys are generated and given to round 1 for processing. When the clock is applied to these modules, for every clock, the 128 bit key is rotated towards left by a bit. In addition four bits out of 128 Key bits are selected as a time factor. For example, the key bits 120, 73, 57 and 35 are selected and are fixed for both encryption and decryption. These four bits are concatenated together to form a 4 bit number whose maximum and minimum values will be '1111' and '0000' respectively. Suppose, if the key bits in these positions are '1100', the time controller waits for the counter to increment up to '1100' and this takes 12 clock cycles. By this time, the key is also rotated by 12 bits. From this key value, the next set of sub keys are picked out at this time and is used as sub keys for round 2. To control the output of round 1 to reach round 2, a register is placed in between and when the counter reaches '1100', the controller opens this register and allows the data to pass to the round 2. This setup ensures the synchronization of the processes such as the sub key of round 2 and output of round 1 to reach round2 to happen simultaneously.

In the proposed algorithm when the data enters in to round 2 again the four bits 120, 73, 57 and 35 are picked from the current key. The counter is reset to zero. Pipeline register after round 2 will be opened when this value equals the counter value. This procedure continues till the cipher text comes out the encryption system.

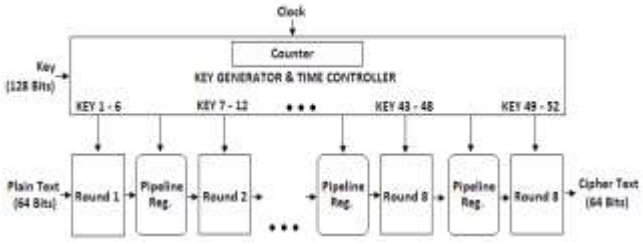

Figure 2.Block Diagram of Temporal IDEA Algorithm

5. FPGA IMPLEMENTATION

5.1 FPGA Architecture

It is a fact that, the invention of devices like embedded FPGAs can be used for some dedicated applications has decreased the level of security of the encryption algorithms. These devices serve as special purpose hardware and are used to break ciphers. This is because these devices have a hardware implementation of the software. The high level of fabrication techniques and much importantly the concurrency in processing data can be used to exploit the security holes through Brute force attacks. In addition Image compression Encryption schemes (ICES) also uses FPGA [Shih 06] for implementation of Discrete Wavelet Transform. The design and analysis of various hardware reconfigurable models of RC5 encryption algorithm is implemented using FPGA [Omar 08].

FPGAs are an array of programmable logic cells interconnected by a matrix of wires and programmable switches. Each cell performs a simple logic function defined as per the user instructions. An FPGA has a large number (64 to over 20,000) of these cells available to use as building blocks in complex digital circuits. Custom hardware has never been so easy to develop. The ability to manipulate the logic at the gate level enables the construction of a custom processor to efficiently implement the desired function.

FPGAs have become key components for implementing high-performance digital signal processing (DSP) systems, especially in digital communications, image and video processing applications. FPGAs have memory bandwidth that far exceeds that of microprocessors and DSP processors running at clock rates two to ten times faster, and unlike processors, FPGAs possess the ability to implement highly parallel, real time custom signal processing architectures.

Altera's FPGAs have 8 input fractural look up table and adaptive logic module (ALM) at its core. This can implement a full 6 input LUT or select 7 input functions. ALM consists of combinational logic, two registers and two adders. Unlike the conventional computers or microcontrollers the FPGAs work concurrently. This makes FPGAs more popular. In the proposed system, the key shifting (for sub key generation) is done concurrently with the decryption process so that right key is picked up at the right time for decryption. This requires two concurrent processes to run in tandem and hence, the FPGA implementation is opted for this purpose.

5.2 Conventional IDEA Algorithm

IDEA uses 64-bit input data that is divided into four 16-bit sub-blocks X1, X2, X3, and X4. Figure 3 explains the FPGA implementation of conventional IDEA algorithm.

In each round, the sequence of events is as follows:

1. Multiply X1 by the first sub key.
2. Add X2 and the second sub key.
3. Add X3 and the third sub key.
4. Multiply X4 by the fourth sub key.
5. XOR the results of Steps 1 and 3.
6. XOR the results of Steps 2 and 4.
7. Multiply the results of Step 5 by the Fifth sub key.
8. Add the results of Steps 6 and 7.
9. Multiply the results of Step 8 by the Sixth sub key.
10. Add the results of Step 7 and 9.

11. XOR the results of Steps 1 and 9.
12. XOR the results of Steps 3 and 9.
13. XOR the results of Steps 2 and 10.
14. XOR the results of Steps 4 and 10.

Between the rounds, the results of second and the third sub-blocks are swapped. Finally, after the eighth round, the four sub-blocks are collected and combined with four sub-keys in an output transformation. The decryption is exactly the reverse process.

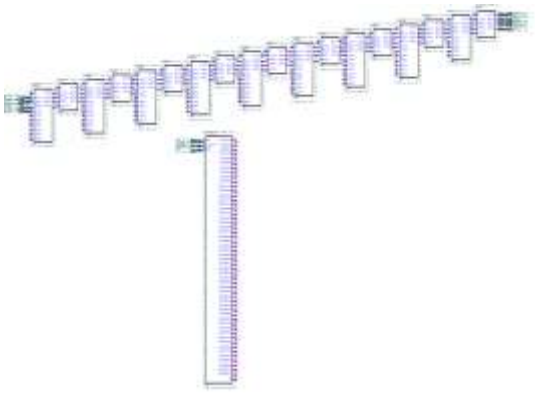

Figure 3.FPGA Implementation of Conventional IDEA Algorithm

5.3 Proposed Scheme

In the FPGA implementation, the eight rounds are implemented and pipelined using registers so that concurrent running of all rounds and hence, high throughput can be achieved. Figure 4 illustrates the same. Key generator and time controller (KGTC) modules are designed as a separate module. As it is a hardware implementation of the algorithm, all the modules can run concurrently. It is worthy to note that if the same is to be implemented in a computer, two computers are needed, one to do the encryption/decryption and the other for key generation cum delay timer.

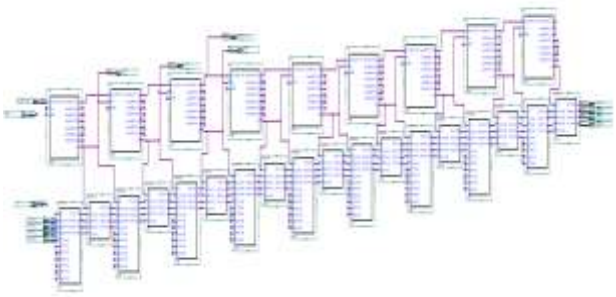

Figure 4.FPGA Implementation of Proposed IDEA Algorithm

The data manipulations inside the key generator and time controller module is explained below:

a. The 128 bit key is given to the KGTC module.
b. The KGTC module has a clock input and has an in-built counter.
c. For every clock input, the 128 bit key is rotated by one bit.
d. 4 bits from predefined locations from the key are picked out and concatenated. For example, key bits 126, 79, 39, 12 are selected. Assume the bits are 1010 after concatenation. (Decimal equivalent is 10)
e. The counter value is compared with this time value. When these two values are same, that indicates, the actual key has been rotated by 10 bits and time is consumed for 10 counts. If the clock is of 4 ns, then for 10 counts, 40 ns are consumed.
f. At this point of time, the first 96 bits of the key is taken out for encryption / decryption as per IDEA algorithm. (6 keys of size 16 bits for every round)
g. Round 1 of IDEA algorithm is processed, in two clock cycles and the first pipelined register is opened when the counter equals 12. This completes the first round of encryption / decryption.
h. The counter is reset to zero.
i. From the current key position, the key bits 126, 79, 39 and 12 are selected again and concatenated. This value may be the same or different. In both cases, again the counter runs and the equal value is checked with the counter value. Suppose, if the new value is 15, then the key is rotated by 15 bits and 60 ns are consumed.
j. Once the counter reaches 15, the first 96 bits are picked out from the key and given to the next round of the encryption / decryption.
k. When the counter reaches 17, the second flipflop is opened to all the data to pass to the third round.
l. The same process continues for all rounds.

In this algorithm, the following issues are analyzed for proper functioning.

1. In the implemented system, 4 bits are taken from the said positions. Hence, the maximum number realized is 16 and hence maximum delay achieved is (16 * 4 ns =) 64 ns. However, if the delay needs to be increased, more number of bits can be used. If 10 bits are picked out 2^{10} combinations are possible, hence (1024 * 4 ns =) 4096 ns delay is realized.
2. The counter size (number of bits) is a parameter to design. It is selected based on the number of bits selected in previous step. As the counter bits increases, number of transitions increase and dynamic power consumption increases. Hence, if four bits are selected in the previous step, maximum value is 15 and hence the counter should count up to 17. (Two counts are added to provide delay to perform the encryption / decryption as stated in step 'g' of the data manipulations, explained earlier.)
3. The clock's time period can be altered. Instead of 4 ns clock, higher time period clocks can be selected which increases the delay.

5.4 Temporal IDEA Decryption Algorithm

The conventional IDEA decryption algorithm uses exactly the same sequence of operations of successive 64-bit blocks of the cipher text, but with a different set of sub keys. The decryption sub keys are worked out from the encryption sub keys being either multiplicative or additive inverses of them. The decryption sub keys (relative to the encryption sub keys s1 to s52) are shown below:

1st round	s49*	s50#	s51#	s52*	s47	s48
2nd round	s43*	s45#	s44#	s46*	s41	s42
3rd round	s37*	s39#	s38#	s39*	s35	s36
4th round	s31*	s33#	s32#	s34*	s29	s30
5th round	s25*	s27#	s26#	s28*	s23	s24
6th round	s19*	s21#	s20#	s22*	s17	s18
7th round	s13*	s15#	s14#	s16*	s11	s12
8th round	s7*	s9#	s8#	s10*	s5	s6
Final transformation			s1*	s2#	s3#	s4*

Where,

sXX* = multiplicative inverse of sXX modulus ((2^16)+1)

sXX# = additive inverse of sXX modulus (2^16)

Hence, the same implementation is used for decryption also with the inverse of keys used. Modules for key inversion alone need to be added to do the decryption.

6. COMPARISON BETWEEN IDEA AND PROPOSED ALGORITHM

In the conventional IDEA algorithm, once the sub keys are generated from the given 128 bits keys, the key is rotated left by 25 bits and then next set of sub keys are generated. Hence, the shifting remains fixed. But, in the proposed algorithm, the shifting is done continuously without restricting to fixed number of bits. The shifting is dynamically performed based on the random selection of 4 bit positions. These bit positions must be known to the sender and receiver in addition to the key. The randomly selected bits 120, 73, 57 and 35 decide how many shift operations are to be performed before the next set of sub keys are generated. As explained in the previous example when the value is '1100' it takes 12 counts. When the clock runs at 4 ns rate, 12 counts takes 48 ns delay. Hence, this delay varies depending upon the key. Hence, the true complexity is introduced in the encryption process. When the correct key is given, both encryption and decryption will be done after a small added up delay. But when an intruder tries to crack the code with random generated key or with brute force search, the cryptanalysis consumes much time because, for various wrong key inputs, wrong delays are used and the system consumes more time. This increases the complexity of Brute force attack. Hence, the process of encryption and the decryption consumes time – a *minimum prescribed* for legitimate users and *infinite* for illegitimate users.

1. In case of Conventional IDEA algorithm, key generation is done for one time. Thereafter for any amount of plain data input, the same set of keys is used repeatedly. But in the case of proposed algorithm, rather than a constant shifting of 25 bits, a dynamic key generation is done. Based on the new data, the key is generated for individual data input.
2. In case of Conventional IDEA algorithm, key is rotated by 25 bits and this is done in a single clock cycle. But in the proposed algorithm, depending upon the values available in those 4 bits, keys are rotated. A constant shift is not possible. Hence, till that time, the system needs to wait and this forms the time based key input. For an intruder, as the exact key is unknown, a try of all possible rotations need to be performed and this takes a long time for cryptanalysis.
3. It is also possible to select any number of bits from the key other than the specified 4 bit positions to represent the delay. That is the proposed scheme allows the sender and receiver to have a common agreement in selecting the number of bits and their positions. In case of conventional IDEA algorithm this is not possible.
4. In case of conventional IDEA algorithm, every process can be represented as a linear equation. That is every output is related to its input by a fixed relation. But in the proposed scheme, key rotation is dynamically varied, it provides a virtual 'S' box type architecture where no equation can be used to relate the input and output of the 'S' box.

7. IMPLEMENTATION RESULTS

The proposed scheme is implemented on an Altera Cyclone II (EP2C20F484C7) device available on a DE1 FPGA Development kit. The system consumes 1176 combinational logic registers and 239 dedicated logic registers. The following Figure 5 compares the conventional IDEA and proposed scheme in terms of silicon area, pins and speed of operation. Table I compares the IDEA and proposed scheme.

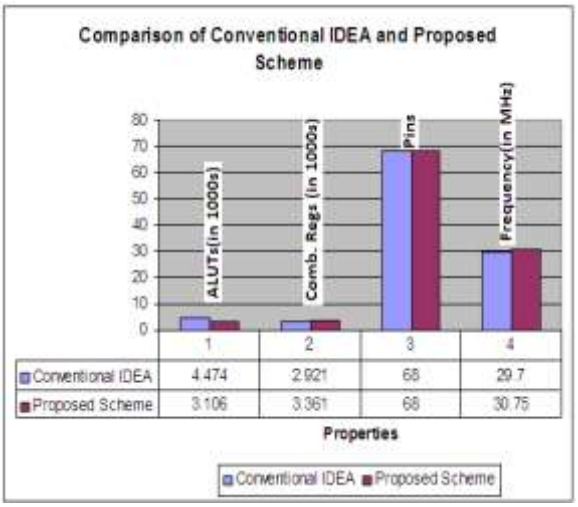

Figure 5. Implementation results of Conventional IDEA and proposed scheme

In terms of area, there is a marginal decrease in the computational ALUTs (Arithmetic LUTs) while increase in registers used. The operating frequency decreases by a factor around 1%. But when the time is used as a second dimension of key, the process of encryption and the decryption consumes time – a *minimum prescribed* for legitimate users and *infinite* for illegitimate users

Table 1. Comparison between IDEA & Proposed scheme

S. No	Title	Conventional IDEA	Proposed Scheme
1	Device	Altera EP3SL 150F1152C4	Altera EP3SL 150F1152C4
2	Combinational ALUT	4474	3106
3	Dedicated Registers	2921	3361
4	Pins	258	258
5	DSP Multipliers Used	68	68
6	Frequency	29.7 MHz	30.75 MHz

8. SUITABILITY OF THE PROPOSED SCHEME FOR COMPUTER BASED IMPLEMENTATION

The proposed scheme requires two processes to run concurrently. First process is key manipulation with selection of time key and the second process is encryption/decryption. Hence, the proposed algorithm can also be implemented in computers with one more microcontroller or microprocessor supplying keys at required intervals. This makes a hardware key to encrypt/decrypt software data.

9. ADVANTAGES OF PROPOSED SCHEME

1. Without FPGA the algorithm requires two computers, one to do the decryption and the other for key generation cum delay timer.
2. In contrast to conventional IDEA, the proposed scheme provides a virtual 'S' box type architecture where no equation can be framed that relates thee input and output by dynamically rotating the key.
3. In Conventional IDEA algorithm, key generation is done one time. For any plain data input, the same set of keys is used repeatedly. But in the proposed algorithm, rather than a constant shift of 25 bits, a dynamic key generation is done. Based on the new data, the key is generated for individual data input.

10. CONCLUSION

In this paper, a novel approach is proposed to enhance the security level of IDEA encryption algorithm against Brute force attack besides retaining its original strength. This approach impedes the fact that, cryptanalysis using brute force attack purely depends on the speed of the system used. Even though the system is capable of performing faster, the delays are included to make the cryptanalysis more complex. Similarly if more number of systems is used for cryptanalysis, then the key space will be shared between them which further reduce the level of security of the encryption algorithm. Though, 128 bit size of the key provides adequate level of security at the present time, due to the increase in the speed of the system with latest technologies in the IC fabrication houses, the key size needs to be increased proportionally. The proposed scheme provides an alternative method in which the strength of the IDEA algorithm is maintained without any modification but at the same the system can defend against brute force attack more vigorously.

11. REFERENCES

[1] B. Lakshmi, T.N.Prabakar, E. Kirubakaran, "Real time cryptography with dual key encryption", IEEE International Conference on Computing, Communication and Networking, December 2008. pp 1-4.

[2] Peng Gong au, Feng-jiao Qiu & Meng Liu, "A new algorithm based on DES and ECC for CSCW", The 8th International Conference on Computer Supported Cooperative Work in Design, 2004. Proceedings. May 2004 Volume: 1 pp: 481 – 486.

[3] Mukesh Kumar Singh, "Matrix based asymmetric bulk encryption algorithm", Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC, June 2004, pp: 161 – 167.

[4] Arafat S M, "An encryption algorithm based on alphabetic trees", The 3rd ACS/IEEE International Conference on Computer Systems and applications, 2005. pp - 92.

[5] Ya-PintZhang, Jizhou Sun au and Xu Zhang, "A stream cipher algorithm based on conventional encryption techniques", Canadian Conference on Electrical and Computer Engineering, May 2004, pp: 649 - 652 Vol. 2.

[6] Fournaris A. P, Sklavos N and Koufopavlou O, "VLSI architecture and FPGA implementation of ICE encryption algorithm", Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003, Dec.2003, pp.88 - 91 Vol. 1.

[7] Atul Kahate, "Cryptography & Network Security", Tata McGraw Hill, 2003.

[8] Bruce Schneider, "Applied Cryptography II Ed", Wiley Eastern, 1995.

[9] William Stallings, "Cryptography & Network Security", Prentice Hall, 1998.

[10] Mitsuru Matsui, "Linear Cryptanalysis Method for DES cipher", Computer & Information Systems Laboratory, Mitsubishi Electric Corporation.

[11] RC5-The Wikipedia free encyclopedia.

[12] E.Biham, A.Shamir, "Linear Cryptanalysis of DES like Cryptosystems" –Journal of cryptology, pp 3-72(1991)

[13] John Kelsey, Bruce Schneier, David Wagner, "Key – Schedule of IDEA, G-DES, GOST, SAFER and Triple DES".

[14] Joan Daemen, Rene Govaerts and Joos Vandewalle, "Weak Keys for IDEA"

[15] W.Diffie and M.E. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard, Computer", Vol. 10, No.6, pp 74-84, June 1977.

[16] Shih-Ching Ou ,Hung ,–Yuan Chung ,Wen-Tsai Sung , "Improving the Compression and Encryption of Images using FPGA-based Cryptosystems", Springer Science +Business Media, Inc 2006.

[17] LIU Shu Tang & SUN Fu Yan, "Spatial Chaos-based Image Encryption Design ", Springer -Verlag

[18] Omar Elkeelany, "Design and Implementation of Various Models Pof RC5-192 Embedded Information Security Algorithm", International Journal of Applied Mathematics and Informatics Vol 2, 2008.

[19] Schubert .A and Anheier .W, "Efficient VLSI Implementation of Modern Symmetric Ciphers", in The ICECS99. 1999.

[20] Limer Elbaz, & Hagai Bar-EI, "Strength Assessment of Encryption Algorithm", Whitepaper October 2000.