iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** IX **Month of publication:** September 2017

DOI: http://doi.org/10.22214/ijraset.2017.9190

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

HDL Design and Verification of Hash Based Authentication Algorithms

Amrit Singh[1], Gurmohan Singh[2], Manjit Kaur[3]
[1, 2, 3]*Centre for Development of Advance Computing, Mohali*

Abstract: ***In today's world everything is computer based, like internet application, banking application, electronic transaction, electronic commerce and electronic learning. All these electronic transactions needs high privacy and confidentiality i.e. the access to only authorized users. So, we want to enhance the security level of these entire computer or internet based applications. For high privacy and confidentiality the authentication system is proposed. Authentication system provides essential security to password and username for any of the above applications. Authentication systems and digital signature application uses hashing algorithms to make unreadable code from password and kept the code in database. So that no one can understand it by reading from database. To access the application or web, the user must be going through the authentication system by entering correct username and password and hash code generated from password will be compared with hash values in database and if equals then only user have authority to access that application. In this thesis work, we designed MD5 (message digest) and SHA512 (standard hash algorithm) hashing algorithms in a single hash unit. To optimize the speed of algorithms, pipelining technique is used. The tool used for coding is Xilinx ISE design suite 12.4. The HDL design entry has been done using ISE text editor and synthesis tool used is Xilinx Synthesis Tool (XST) with Virtex-5 FPGA device XC5VLX30. Function verification has been done using ISim simulator. The obtained throughput is 185.05 Mbps for SHA512 and 308.9 Mbps for MD5 resulting into a combined throughput for designed hash unit is 297.3 Mbps. The maximum frequency of designed SHA512 is 277MHz and of MD5 is 162 MHz combining into a single hash unit with frequency of 162MHz.***

Keywords: Hash, MD5, Authentication, SHA, Cryptography.

I. INTRODUCTION

In modern era almost everything is computer based, like internet application, banking application, Electronic transaction, E-commerce, E-government and E-learning. These all are electronic applications, used with the help of computer network and internet media. This continues development in the current technology, needs high privacy mean only authorized user can access. So, we want to increase the security level of these entire computer or internet based application. For high privacy and confidentiality the authentication system is proposed. Authentication system provides guard in contradiction of password and username. Authentication system used Hash algorithms to make unreadable code from password and kept the code in database. So if someone wants to read it from database will not able to understand it. To access the application or web, the user must be going through the authentication system by entering correct username and password and hash code generated from password will compared with hash values in database if it equals then user have authority to access that application [1]-[5].

But user name and password authorization faces number of security problems, some problems are mention below:

Monitor during data transmission. Authenticated data is mainly transmitted over network, hackers might be monitoring the transmitted data with the help of superior tools and track user's password and other information.

There are many unreliable programs(or virus) coming from various media, which could be triggered by operator's sloppiness and give opportunity to unauthorized user to record authorized user's actions, including his username and password.

Authentication is confirmation of identity of source of information. Authentication is the heart of cryptography. Authentication gives guarantee for secure communication. Authentication covers a number of security goals. It is classified into two types:

A. Hash Function

Cryptography have many essential parts corresponding hash function, encryption and message authentication code (MAC). Hash function does not need any individual key to produce code. In case of encryption and message authentication code key is necessary. Hash function is used to convert arbitrary length message into a fixed length hash code, this code will use as authenticators. The necessary condition for safe hash function is [2]:

Two different plain text generate the same hash code that should not be calculated is called as collision. Certain data abstract can be uncalculated over the additional plain data producing the same hash code, which means first state cannot be concluded from the result. The hash code does not depend on input data length, it generates same length hash code.
If there is very minor change in plain text, the hash code that was generated entirely different. Mean single bit changes in main message will create a unique hash value.
Mainly there are two different approaches which may be used to attack the security of a hash function. Cryptanalysis and brute-force attack [2]. Cryptanalysis will attack on the logically weak hash algorithm.
The strength of a hash code can be verified by brute force attack, which depends on the length of hash code and complexity of process used in hash algorithm.

II. PROBLEM FORMULATION

As a number of hash based authentication algorithms are proposed for different applications. But still the problem is that only one hash function is implemented or hash function combined with operation cryptographic algorithms (i.e. encryption or decryption). So in the application of a unified hash based authentication algorithm (i.e. SHA-512 and MD5) where we need to use the combined operation of these two algorithms, there comes a problem.
So the need arises to develop an application in which we can use the combined operation of these two algorithms efficiently so that we can develop the unified module of hash based authentication algorithms. So the main need of the research work to be carried out is to develop the application of unified module.

III. UNIFIED HASH BASED AUTHENTICATION ALGORITHMS

Hash based authentication algorithm for authorization purpose has been designed and discussed in purposed research work. In research work, two different algorithms i.e. SHA-512 and MD5 are unified in a single hash unit and implemented on FPGA. Integration of two authentication algorithms i.e. SHA-512 and MD5 will be discussed.

A. Authentication Algorithms

Hash based authentication algorithm is generally used to generate a hash code of fixed length from arbitrary length. Figure 1 shows the general flow of diagram of hash algorithm common for every algorithm.

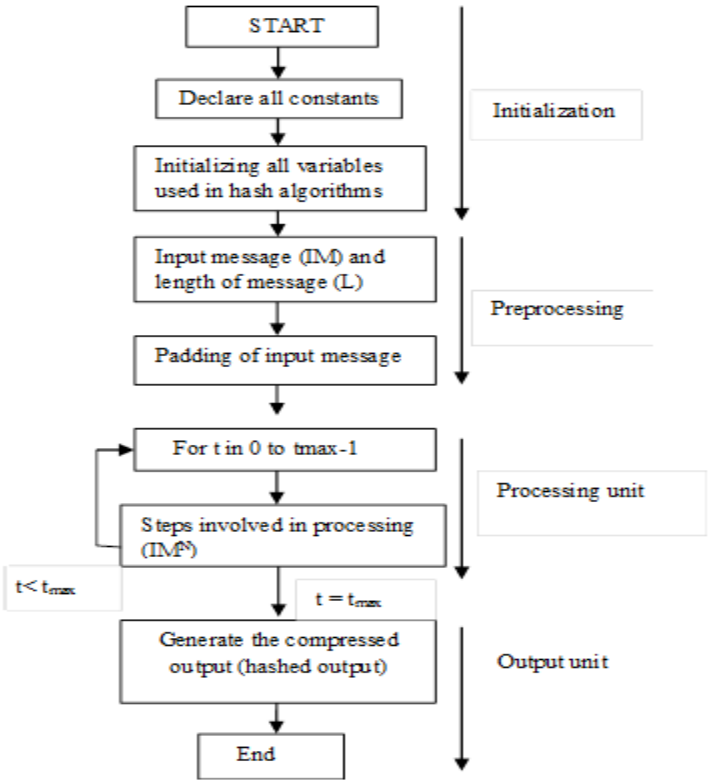


Fig. 1 Flow diagram for Hash based Authentication algorithms

1) *Initialization:* After the start, initialization of all the constants and working variable is common in every hash algorithm.
2) *Pre-processing of Input Message:* In this step, we take the input message of any length and padding is done to make the message length of exactly divisible of block size.
3) *Processing Unit:* In this step, there are number of operations in each iteration performed over padded message. When the loop is ended mean all the rounds are complete then working variable store final values.
4) *Output Unit:* When all the rounds are completed the result stored in working variable are operated in final step to obtain the result of authentication hashing algorithm.

Figure 1 shows the steps in flow diagram except processing unit is common for every hash based authentication algorithm. Different types of architecture are developed for each algorithm in processing unit to obtained hash code. If the number of bits increases at output then strength of the hash code is increased. More the strength means highly secure network.

B. SHA-512 Algorithm

SHA-512 is basically an iterative algorithm because it performs number of transformation (or iterations) round to obtain hash result. There are some steps to execute the input message to generate hash output as discussed below:-

1) *Step 1:* The message of any length is taken as an input of hash function. The input message is padded to make the input block size of 1024 bit each. The padding is done by taking first bit '1' and then all other bit are '0' (i.e 1000…….000). Divide the padded input message in 'N' number of blocks each of size 1024-bit and denoted as IM^1, IM^2,….., IM^N.
2) *Step 2:* In this each block of 1024 bits is divided into sixteen sub blocks of 64 bit each. First sub block is denoted as $IM_0^{(j)}$ and next sub blocks are denoted as $IM_1^{(j)}$……., $IM_{15}^{(j)}$. Value of 'j' ranges from 1 to N. 'N' denotes how many number of 1024 bit blocks are there.
3) *Step 3:* Scheduling of 64 bit message block is done in this step. In first sixteen iterations, the value of 16 bit message block is directly loaded in working register 'w'. For next iterations the value of register 'w' is depends on the rotation function of 'x' and previous values of register 'w'.

$$w_i = M_i^j \quad for \quad 0 \le i \le 15 \tag{1}$$

$$w_i = S1(W_{i-2}) \oplus S0(W_{i-15}) \oplus W_{i-7} \oplus W_{i-6} \quad for \quad 16 \le i \le i_{max} \tag{2}$$

w_i is working register for i^{th} iteration. M_i^j is 64 bit message block for i^{th} iteration of j^{th} block of input message. I_{max} is maximum number of iteration or transformation round. $S(x)$ is the XOR operation of rotation function of variable 'x'.

$$S0(x) = ROR(x,1) \oplus ROR(x,1) \oplus SHR(x,7) \tag{3}$$

$$S1(x) = ROR(x,19) \oplus ROR(x,61) \oplus SHR(x,6) \tag{4}$$

ROR(x,n) is circular right rotation of variable 'x' for 'n' times. *SHR(x,n)* is shift right operation of variable 'x' for 'n' times. Eight 64 bit variable registers are there. The value of each variable is generated from fractional part of square root of prime number. From fractional part first 64 bits are considered. H_0, H_1, H_2, H_3, H_4, H_5, H_6 and H_7 are variable registers used to store 64 bit standard values. Eighty 64 bit Constant C_0, C_1, …….., C_{79} are used in each iteration. The value of constants C_0, C_1, ….., C_{79} is generated by taking the fractional part of cube root of prime number. First 64 bit of fractional part is considered.

4) *Step 4:* Now all the working variable register *a, b, c, d, e, f, g* and *h* respectively initialized with standard values that are stored in variable registers H_0, H_1, H_2, H_3, H_4, H_5, H_6 and H_7. In each round the value of working variable registers are altered accordingly.

$$a = H_0;\ b = H_1;\ c = H_2;\ d = H_3;$$
$$e = H_4;\ f = H_5;\ g = H_6;\ h = H_7;$$

5) *Step 5:* After initialization of working variable register computation unit is there. In computation unit 80 rounds are there. Each round changes the value of working variable register. Each round use different value of constant 'C_i' from table.

$$(e,f,g) = ((e,f) \oplus (\bar{e},g)) \tag{5}$$

P is the function of *e, f, g* variable.

$$M(a,b,c) = ((a,b) \oplus (a,c) \oplus (b,c)) \tag{6}$$

M is the function of variable *a, b, c.*

$$Sm_0(x) = ROR(x,28) \oplus ROR(x,34) \oplus ROR(x,39) \tag{7}$$

$$Sm_1(x) = ROR(x,14) \oplus ROR(x,18) \oplus ROR(x,41) \tag{8}$$

$Sm_0(x)$ and $Sm_1(x)$ are the XOR operation of circular right rotation of variable '*x*'. $ROR(x, n)$ is circular right rotation of variable '*x*' for '*n*' times.

6) *Step 6:* When j[th] block of input message is computed then H^j of 512 bit is generated by executing the equation 9.

$$H^j = ((a + H_0^{j-1})\&(b + H_1^{j-1})\&(c + H_2^{j-1})\&(d + H_3^{j-1})\&(e + H_4^{j-1})\&(f + H_5^{j-1})\&(g + H_6^{j-1})\&(h + H_7^{j-1})) \quad (9)$$

Here **'&'** denoted the concatenation of signals.

When the value of '*j*' approached to '*N*' mean when the N[th] block of input message is computed then H^N is generated. The generated output is known as message digest of input message. H^N is of 512 bit.

C. MD5 Algorithm

MD5 is also an authentication algorithm. MD5 is an iterative algorithm because it performs number of transformation (or iterations) rounds to obtain hash result. There are some steps to execute the input message to generate hash output of 128 bits as discussed below

1) *Step 1:* The message of any length is taken as an input in this algorithm. The input message is padded to make the input block size of 512 bit each. The padding is done by taking first bit '1' and then all other bit are '0' (i.e 1000…….000). Divide the padded input message in 'N' number of blocks each block of 512 bit is denoted as IM^1, IM^2,….., IM^N.
2) *Step 2:* In this each block of 512-bit is divided into sixteen sub blocks of 32 bit each. First sub block is denoted as $IM_0^{(j)}$ and next sub blocks are denoted as $IM_1^{(j)}$…….., $IM_{15}^{(j)}$. Value of 'j' ranges from 1 to N. 'N' denoted that how many number of 512 bit blocks are there.
3) *Step 3:* In this step, buffers that are available in MD5 are initialized with some standard values. In MD5 there are basically four buffers *buf_A, buf_B, buf_C* and *buf_D.* Each buffer having memory size of 32 bits.
4) *Step 4:* This step is the main part of the algorithm in other words this step is basically heart of the algorithm. There are four rounds of operation to obtain the result of MD5. Each round uses different values of auxiliary equation. There are generally four different auxiliary equations F_1, F_2, F_3 and F_4 respectively. F_1, F_2, F_3 and F_4 are variable of *x*, *y* and z as given below.

$$F_1(x, y, z) = (x.y) + (\bar{x}.z) \quad (10)$$
$$F_2(x, y, z) = (x.z) + (y.\bar{z}) \quad (11)$$
$$F_3(x, y, z) = x \oplus y \oplus z \quad (12)$$
$$F_4(x, y, z) = y \oplus (x + \bar{z}) \quad (13)$$

Here **'.', '+', '-'**and **'⊕'** represent logical operations AND, OR, NOT and XOR respectively. Four rounds of operations are there and each round has sixteen sub steps. *T[n]* represents the value of table for n[th] sub step. The values of *T[n]* generated by taking the integer part of 2^{32} *(abs(sin(n))). 'abs' is denoted as absolute value. First 32 bit is considered for the evaluation of result. The input of every round is of 512 bit that is 'X_m' and initial variable of 128 bits. Each round uses different auxiliary function and after the end of round 4 the value of initial variable are added with the result obtained from last round.

5) *Step 5:* In the end when 'N' number of blocks of 512 bit are executed then the output N[th] stage is of 128 bit message digest of the input signal. There are four different circular shifts used in each round and value of circular shift is different for each round.

$$B_{temp} = B + (ROL((a + F_n(B, C, D) + IM_i^j + T[i]), k) \quad (14)$$

B_{temp} is the signal uses for storage of temporary value. $ROL(x,m)$ is denoted as circular shift right of variable 'x' with 'm' times. $F_n(B,C,D)$ is the auxiliary function of variable *B, C* and *D for* n^{th} round. *T[i]* is the value of table for i[th] step. After each step the value of variable register are altered. Four variable register are there ***A, B, C*** and ***D***.

$$A = D\ ,\ B = B_{temp},\ C = B\ ,\ D = C;$$

After the execution of 4[th] round the value of MD5 buffer are added with result obtained from last step.

$$A = A + buff\_A\ ,B = B + buff\_B$$
$$C = C + buff\_C\ ;D = D + buff\_D$$

In the end the message digest of 128 bit is obtained by concatenating the values as below:

$$md5_{output} = D\ \&\ C\ \&\ B\ \&\ A \quad (15)$$

This research work is successfully completed and verified on Xilinx ISim simulator. This work is successfully implemented on virtex-6 FPGA device XC6VLX240T.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

IV. RESULTS AND DISCUSSION

A. Simulation Result of Unified Hash Based Authentication Module

The unified hash based authentication module is the combination of both SHA-512 and MD5 modules. For the simulation, a signal select_algorithm is taken to select one of the above two algorithm. For '1' value of the select_algorithm signal MD5 will be selected and SHA-512 for value '0'.

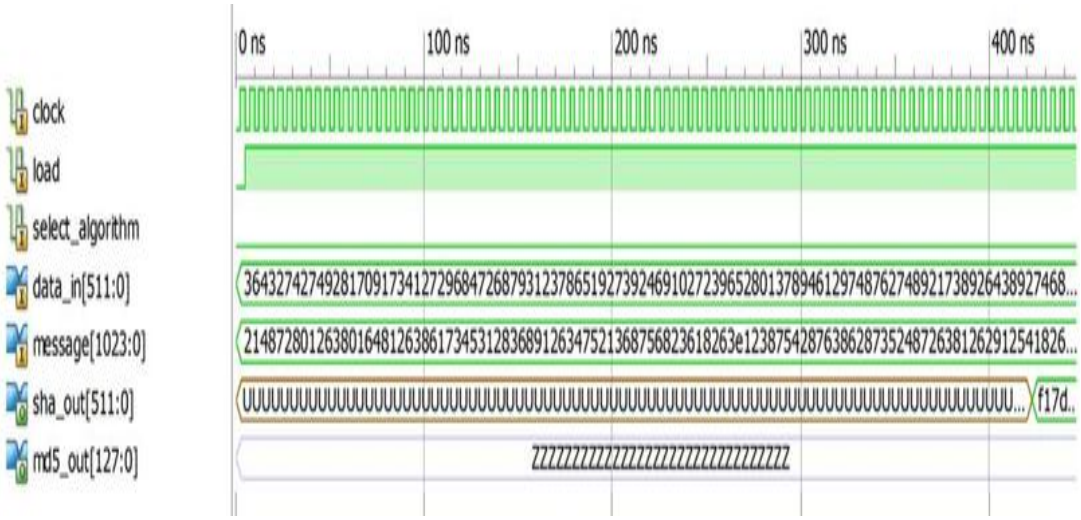


Fig. 2 Simulation waveform of SHA-512 in unified hash based authentication algorithm.

In figure 2 clock is an input to the module having period of 5 ns. When value of load is low then all the inputs are loaded in the variable registers and when load is active high all the processing under the control of clock signal. Here select_algorithm use to select authentication algorithm. In this figure select_algorithm is '0' mean SHA-512 is selected. The output of 512-bit is obtained after 80 clock pulses as shown in figure 3. In figure 3, *selected_algorithm* is '1' means MD5 is selected. The output of length 128-bit is obtained after 64 clock pulses as shown in figure 3

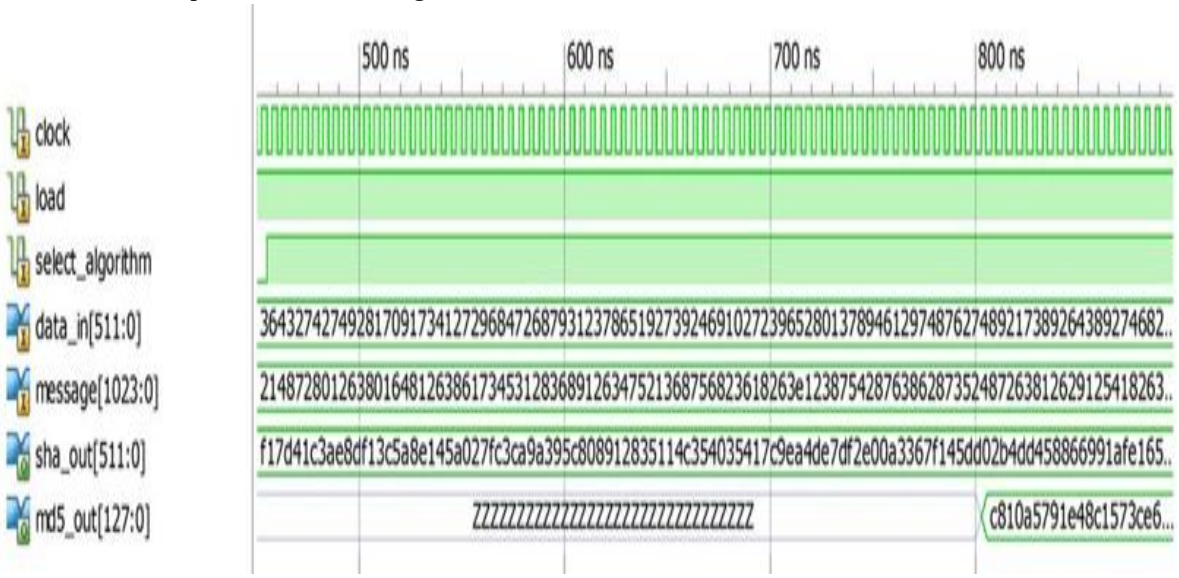


Fig. 3 Simulation waveform of MD5 in unified hash based authentication algorithm

B. Implementation of Unified Hash Based Authentication Algorithms on Virtex-6 FPGA Device XC6VLX240T

A unified top level module for hash based authentication algorithms (i.e. SHA-512 and MD5) has been successfully implemented on Virtex-6 FPGA device XC6VLX240T. Figure 4 shows the 8-bit result of SHA-512. 8-bit output result of SHA-512 is displayed on LCD of Virtex-6 as shown in figure 4. Displayed results are in decimal form.

Fig. 4 FPGA implementation of SHA-512

Figure 5 shows the 8-bit result of MD5. 8-bit output result of MD5 is displayed on LCD as shown in figure 6. Displayed results are in decimal form.

Fig. 5 FPGA implementation of MD5

1) *Synthesis Report:* The synthesis timing report of unified top level module for hash based authentication algorithms on virtex-6 FPGA device XC6VLX240T is shown in tables 1 and 2.
2) *Selected Device:* Virtex-6 device XC6VLX240T

TABLE I: TIMING REPORT OF UNIFIED HASH BASED AUTHENTICATION MODULE

Parameters	Values
Speed Grade	-1
Minimum period	5.849 ns
Maximum Frequency	170 MHz
Minimum input arrival time before clock	5.514 ns
Maximum output required time after clock	.823 ns

Table II shows the device utilization summary of Unified hash based authentication algorithm.

TABLE II: IDEVICE UTILIZATION SUMMARY

Logic utilization	Used	Available	Utilization
Number of Slices	7,642	301,440	2%
Number of Slice LUT	15,054	150,720	9%
Number of occupied Slices	4,530	37,680	12%
Number of fully used LUT-FF pairs	7,065	15,325	46%
Number of bonded IOBs	13	600	2%

V. CONCLUSION

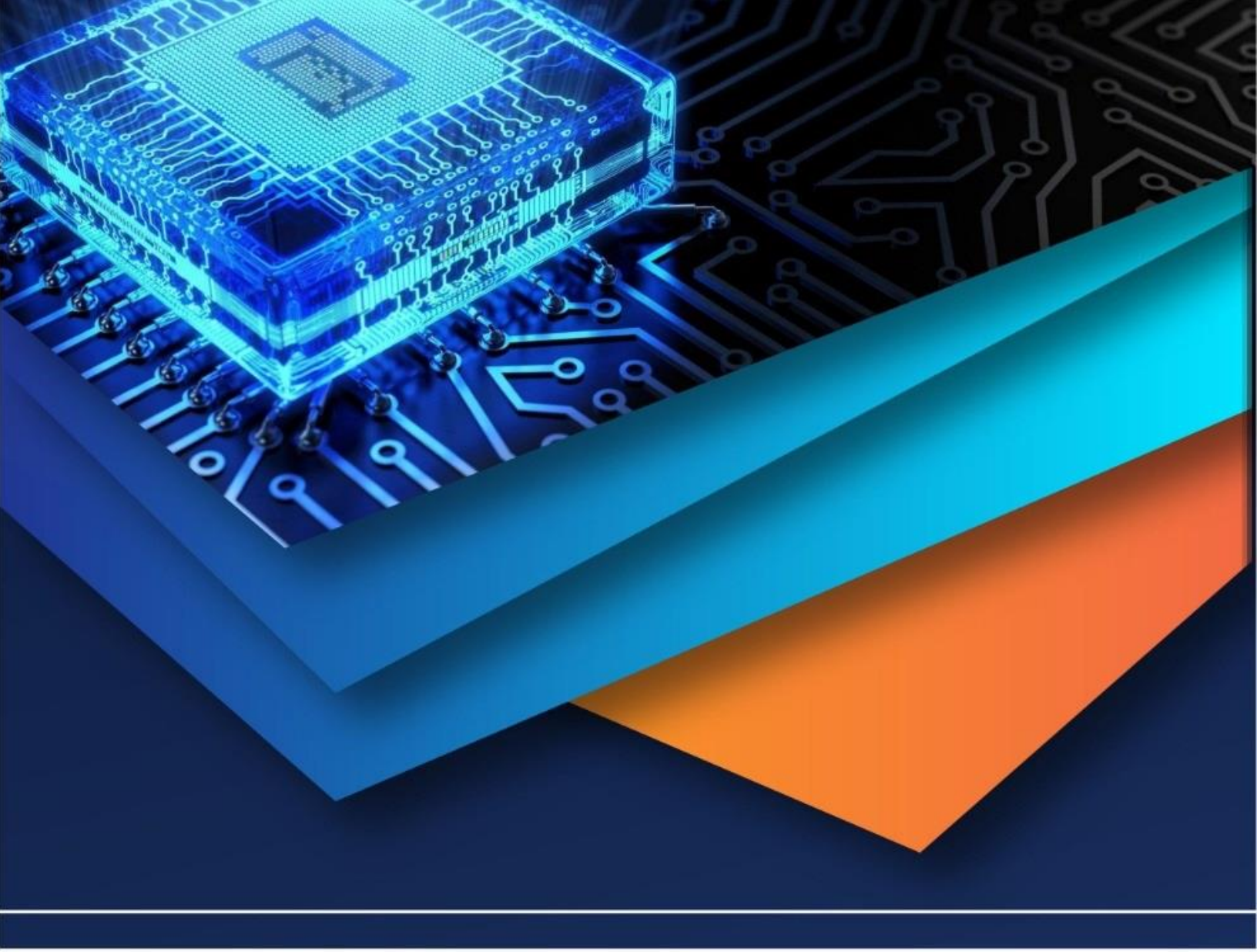
In the growing field of e-applications security issues are of major concern. To provide authentication, privacy and confidentiality to e-applications many algorithms have been developed. This thesis work unifies two important authentication algorithms that are hash function based SHA-512 and MD5. MD5 and SHA-512 provide the message digest of 128 bits and 512 bits respectively, indicating the larger memory requirement and larger CPU time for SHA-512 than MD5. Combining these two into a single hash function make it easier to choose any one depending on the user requirement. For the applications requiring high security and where we can compromise with the speed and memory, SHA-512 will be preferred and where memory and CPU time is of major concern than security there MD5 will be preferred. Three stage and five stage pipelined architecture has been developed in MD5 and SHA512 respectively is quite beneficial for optimizing the speed i.e. CPU time and throughput at expense of small increase in area. The obtained throughput is 185.05 Mbps for SHA-512 and 308.9 Mbps for MD5 resulting into a combined throughput for designed hash unit of 297.3Mbps. The maximum frequency of designed SHA512 is 277MHz and of MD5 is 162 MHz combining into a single hash unit with frequency of 162MHz.

In this thesis work, we designed MD5 (message digest) and SHA-512 (standard hash algorithm) hashing algorithms in a single hash unit. To optimize the speed of algorithms, pipelining technique is used. The tool used for coding is Xilinx ISE design suite 12.4. The HDL design entry has been done using ISE text editor and synthesis tool used is Xilinx Synthesis Tool (XST) with Virtex-6 FPGA device XC5VLX30. Function verification has been done using ISim simulator.

REFERENCES

[1] J Deepakumara, H.M Heys, R Venkatesan, “FPGA implementation of MD5 hash algorithm”, Canadian Conference on Electrical and Computer Engineering, vol.2, pp.919-924, 2001.
[2] A.A Putri Ratna, P Dewi Purnamasari , A Shaugi, M Salman, “Analysis and comparison of MD5 and SHA-1 algorithm implementation in Simple-O authentication based security system”, International Conference on QiR (Quality in Research), pp.99-104, 25-28 June 2013.
[3] Xiaoling Zheng, Jidong Jin, “Research for the application and safety of MD5 algorithm in password authentication”, International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp.2216-2219, 29-31 May 2012.
[4] E Sediyono, K.I Santoso, Suhartono, “Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS”, International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp.1604-608, 22-25 August 2013.
[5] I Algredo-Badillo, M Morales-Sandoval, C Feregrino-Uribe, R.Cumplido, “Throughput and Efficiency Analysis of Unrolled Hardware Architectures for the SHA-512 Hash Algorithm”, Computer Society Annual Symposium on VLSI (ISVLSI), pp.63-68, 19-21 August 2012.
[6] Chen Fatang, Yuan Jinlong, “Enhanced Key Derivation Function of HMAC-SHA-256 Algorithm in LTE Network”, Fourth International Conference on Multimedia Information Networking and Security (MINES), pp.15-18, 2-4 November 2012.

[7] S Gueron, "Speeding Up SHA-1, SHA-256 and SHA-512 on the 2nd Generation Intel® Core™ Processors", Ninth International Conference on Information Technology New Generations (ITNG), pp.824-826, 16-18 April 2012.
[8] G.S Athanasiou, H.E Michail, G Theodoridis, C.E Goutis, "Optimising the SHA-512 cryptographic hash function on FPGAs", IET Computers & Digital Techniques, vol.8, no.2, pp.70-82, March 2014.
[9] N Sklavos, "Towards to SHA-3 Hashing Standard for Secure Communications: On the Hardware Evaluation Development", IEEE (Revista IEEE America Latina) Latin America Transactions, vol.10, pp.1433-1434, January 2012.
[10] S.I Naqvi, A Akram, "Pseudo-random key generation for secure HMAC-MD5", IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp.573-577, 27-29 May 2011.
[11] S.Gueron, S.Johnson, J.Walker, "SHA-512/256", Eighth International Conference on Information Technology New Generations (ITNG), pp.354-358, 11-13 April 2011.
[12] N Sklavos, O Koufopavlou, "On the hardware implementations of the SHA-2 (256, 384, 512) hash functions", International Symposium on Circuits and Systems (ISCAS) vol.5, pp.153-156, 25-28 May 2003.
[13] Mohammed A.Noaman, "A VHDL Model for Implementation of MD5 Hash Algorithm",Eng & Tech. Journal, Vol.31, June 2013.
[14] Xie Nan-bin, Huang Xiang-dan, "The Mixed Encryption Algorithm Based on MD5 and XOR Transformation", Second International Workshop on Education Technology and Computer Science (ETCS), pp.394-396, 6-7 March 2010.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY