ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 1, January 2014

Improving Message Authentication by Integrating Encryption with Hash function and its VLSI Implementation

M.Meenakumari[1], G.Athisha[2]

Asst. Prof (SG), Department of ECE, SNS College of Engineering, Coimbatore, India [1]

Prof, Department of ECE, PSNA College of Engineering, Dindigul, India [2]

Abstract: Presently more techniques are available for improving secure data communication. Public and private key encryption algorithms are available to provide confidentiality. Encryption techniques provide origin authenticity by using shared secret key. Advanced Encryption System (AES) is the specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST). Hash function is an important technique for implementing information integrity. In this paper MD5 hash function is used. But almost each and every technique faces one or security related issues. The main problem is the creation of forged hash value by intruder. In this paper the technique of combining encryption algorithm with hash function is given so that both data integrity and confidentiality can be realised while transmitting message between sender and receiver. The coding of combined algorithm is done in VHDL and implementation is done using Xilinx Spartan family.

Keywords: Encryption, MD5, Confidentiality, hash function, AES

I. INTRODUCTION

The standard techniques for providing privacy and security in data networks include encryption/decryption algorithms such as Advanced Encryption System (AES) and RSA [1] etc. Hashing serves the purpose of ensuring integrity, i.e. making it so that if something is changed you can know that it's changed. Hashing is used in conjunction with authentication to produce strong evidence that a given message has not been modified. This is accomplished by taking a given input, hashing it, and then encrypting the sent hash with the recipient's public key. When the recipient opens the message with their private key they then hash the message themselves and compare it to the hash that was given encrypted by the sender. If they match it is an unmodified message. Examples of hash functions are SHA-3,MD5 etc. Encryption is for maintaining data confidentiality and requires the use of a key (kept secret) in order to return to plaintext. Hashing is for validating the integrity of content by detecting all modification thereof via obvious changes to the hash output.

Given a hash it should be difficult to find any message m such that h = Hash (m).This property is known as one-way function. The other important property of hash is collision Resistance. It should be difficult to find two different messages m_1 and m_2 such that Hash (m_1) = Hash (m_2).A cryptographic hash function is a hash function that takes an arbitrary block of data and returns a fixed-size bit string, the cryptographic hash value, such that any change to the data will (with very high probability) change the hash value. The data to be encoded are often called the message, and the hash values are sometimes called the message digest or simply digest. The ideal cryptographic hash function has four main properties:

- It is easy to compute the hash value for any given message
- It is infeasible to generate a message that has a given hash
- It is infeasible to modify a message without changing the hash
- It is infeasible to find two different messages with the same hash.

Cryptographic hash functions have many information security applications, notably in digital signatures, message authentication codes (MACs), and other forms of authentication. They can also be used as ordinary hash functions, to index data in hash tables, for fingerprinting, to detect duplicate data or uniquely identify files, and as checksums to detect accidental data corruption. They can also be used for signature bit generation in digital IP core protection [2], [3]. Cryptographic hash values are sometimes called (digital) fingerprints, checksums, or just hash values.

Symmetric key encryption [4] is also known as shared-key, single-key, secret-key, and private-key or one-key encryption. In this type of message encryption, both sender and receiver share the same key which is used to both encrypt and decrypt messages. Sender and receiver only have to specify the shared key in the beginning and then they can begin to encrypt and decrypt messages between them using that key. Examples include AES (Advanced Encryption Standard) and Triple DES (Data Encryption Standard).Symmetric key encryption is much faster than asymmetric key encryption. Single-key encryption does not require a lot of computer resources when compared to public key encryption. Asymmetric Key encryption [5] method of encrypting messages makes use of two keys: a public key and a private key. The public

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 1, January 2014

key is made publicly available and is used to encrypt messages by anyone who wishes to send a message to the person that the key belongs to. The private key is kept secret and is used to decrypt received messages. This paper deals with VLSI implementation of integrated AES and hash algorithm.

II. AES ALGORITHM

AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. The algorithm is based on several substitutions, permutations and linear transformations, each executed on data blocks of 16 byte – therefore the term block cipher. Those operations are repeated several times, called "rounds". During each round, a unique roundkey is calculated out of the encryption key, and incorporated in the calculations. Based on this block structure of AES, the change of a single bit either in the key, or in the plaintext block results in a completely different ciphertext block.AES block diagram in shown in fig.1. AES [6] operates on a 4×4 column-major order matrix of bytes, termed the state. Most AES calculations are done in a special finite field.

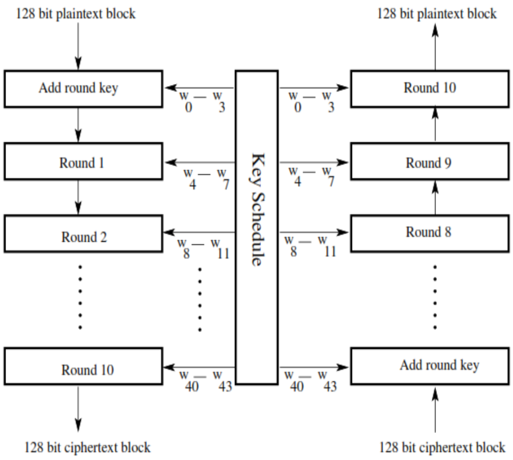

Fig. 1. AES Block diagram

A. Sub Bytes

In the SubBytes step, each byte in the state matrix is replaced with a SubByte using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The operation is shown in fig.2.

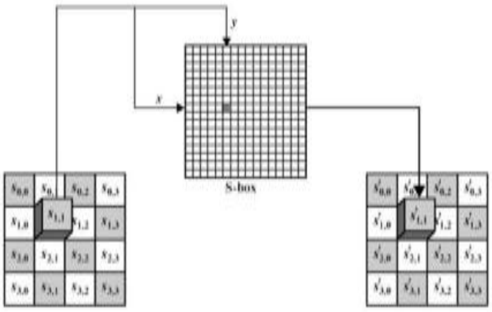

Fig. 2. Sub Bytes stage of the Algorithm

B. Shift Rows

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. It is shown in fig.3.For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by n-1 bytes. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks

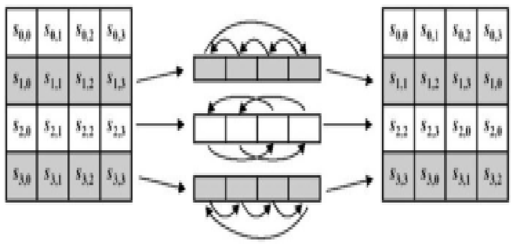

Fig.3. Shift row stages

C. Mix Columns

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. It is shown in fig.4. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.

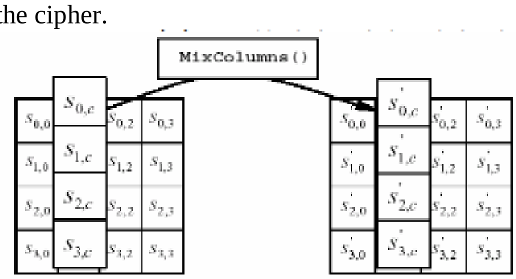

Fig. 4. Mixing of Columns state

D. Add Round Key

In the AddRoundKey step, the subkey is combined with the state. For each round, a sub key is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR. It is shown in fig.5

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 1, January 2014

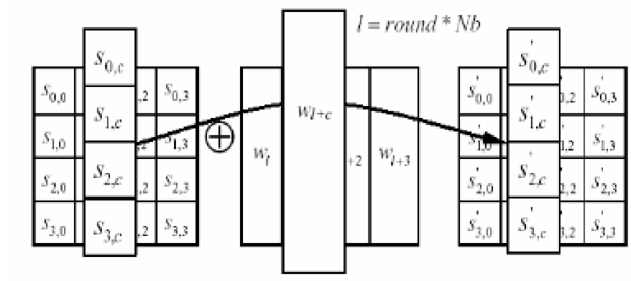

Fig.5. Add Round Key Transformation

III. MD5 ALGORITHM

The MD5[7] is a widely used algorithm to verify data integrity through the creation of a 128 bit message digest from data input (which may be a message of any arbitrary length) that is claimed to be as unique to that specific data as a fingerprint is to the specific individual.. It is used for digital signature applications. MD5 hashes are also used to ensure the data integrity of files. Because the MD5 hash algorithm always produces the same output for the same given input, users can compare a hash of the source file with a newly created hash of the destination file to check that it is intact and unmodified.

The MD5 algorithm is an extension of MD4. MD5 is not quite as fast as the MD4 algorithm, but offers much more data security. MD5 consists in a number of evaluations of a "compression function" over data blocks. The input message is first padded, so that its length becomes a multiple of 512 bits. It is then split into 512-bit blocks. A 128-bit running state is initialized to a conventional value and then processed with the compression function. The compression function takes the running state and one 512-bit message block, and mixes them into a new value for the running state. When all message blocks have been thus processed, the final value of the running state is the hash output. The important point is that there are 64 rounds, but only 16 message words. This means that each message word enters the processing four times. MD5 is a one-way transaction such that it is almost impossible to reverse engineer an MD5 hash to retrieve the original string.

IV. INTEGRATION OF ENCRYPTION WITH MD5

The combined diagram of AES and MD5 is shown infig.6. 128 bit message and 128 bit key is selected for AES. The 128 bit output of AES is given to MD5 hash function. MD5 generates 128 bit fixed output. FPGA implementation of cryptographic algorithm is discussed in [8], [9].FPGA is integrated circuit that can be reconfigured by customers. FPGA consists of thousands of universal building blocks called CLBs which are interconnected by programmable interconnect.

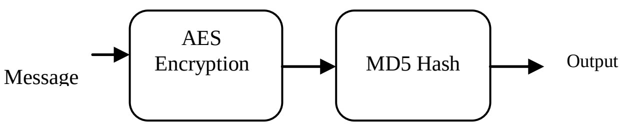

Fig.6. Block diagram of Integration Technique

V. RESULT

The overall structure of Encryption and hashing is simulated by Modelsim 5.7 SE and it is implemented using Xilinx Spartan 3E FPGA.Fig.7 shows the simulation of ASCII.

The simulation result of AES is shown in fig 8. The overall simulation of AES with MD5 hash function is shown in fig 9.

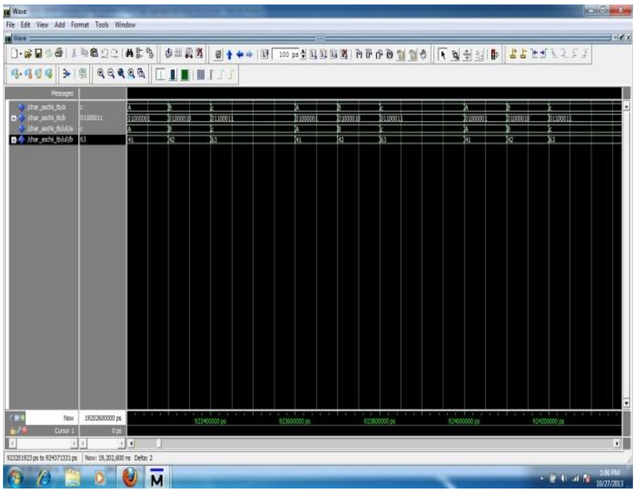

Fig.7. Simulation output of. ASCII Converter

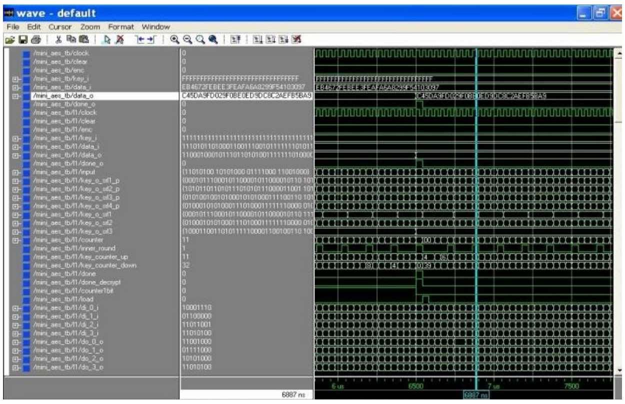

Fig.8. Simulation output of. AES

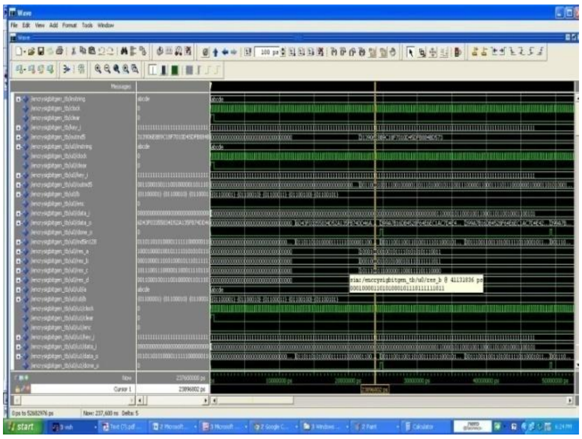

Fig.9. Simulation output of AES with MD5

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 1, January 2014

VI. CONCLUSION

The Advanced Encryption Standard algorithm is a symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128, 192, and 256 bits. An efficient FPGA implementation of 128 bit block and 128 bit key AES algorithm with MD5 hash algorithm has been presented in this paper. The design is implemented on Xilinx using Spartan 3E FPGA.

REFERENCES

[1] Rivest R.L., A. Shamir and L. Adleman, 1978. "A method for obtaining Digital Signatures & Public key Cryptosystems" ,*Communication ACM* Vol.21, No.2, pp. 120-126.
[2] M.Meenakumari and G.Athisha, "A Survey on Protection of FPGA based IP Designs", *International Journal of Advanced Electrical and Electronics Engineering,* Vol.2 ,Issue.2, pp 93-99, 2013.
[3] Cox I.J., M.L. Miller and J.A. Bloom, *Digital Watermarking*, Morgan Kaufmann publishers,1998.
[4] William Stallings,. *Cryptography and Network Security: M. Principles and Practices*, Third Edition, Pearson Education, 2003.
[5] W.Diffie and M.Hellman,"New Directions in Cryptography*", IEEE Transaction on Information Theory*,Vol.22, Issue.6, pp 644-654 ,1976.
[6] Daemon J., and Rijimen. V," Rijindael: The Advanced Encryption Standard", *Dr.Dobb's Journal* , March 2001.
[7] Alfred J.Menezes ,Paul.C.Van Oorschot and Scott A.Vanstone," *Handbook of Applied Cryptography*", CRC Press,1997.
[8] Samir El Adib and Naoufal Raissouni ," AES Encryption Algorithm Hardware Implementation: Throughput and area Comparison of 128 , 192 and 256- bits key", *International Journal of Reconfigurable and Embedded Systems*, Vol. 1,No.2, July 2012, pp 67-74.
[9] Prof.S. Venkatarajulu , Deepa G.M and G.Sriteja ,"Implementation of Cryptographic Algorithm on FPGA ", *International Journal of Computer science and mobile computing* , Vol.2, Issue.4 , April 2013 , pp 604-609.

.

BIOGRAPHIES

M. Meenakumari completed her B.E degree in Electronics & Communication Engineering from Madurai Kamaraj University, Madurai in 1989 and M.E Degree in Applied Electronics from Anna University, Chennai in 2004. She is pursuing Research in Anna University, Chennai. Currently she is working as Assistant Professor (Selection Grade) at SNS college of Engineering, Coimbatore, Tamilnadu, India. Her research interests include protection of VLSI design and digital system design . She has published 5 papers in reputed journals and 10 papers in conferences.

.

G.Athisha is currently serving as the Professor and Head, Department of Electronics and Communication Engineering, P.S.N.A. College of Engineering and Technology, Dindigul, Tamilnadu,India. She received her Ph.D. in Information and Communication Engineering from Anna University, Chennai in the year 2006.She completed her M.E.in Applied Electronics from Bharathiyar University, Coimbatore in 1999 and B.E. in Electronics and Communication Engineering from Madurai Kamaraj University, Madurai in 1997.Her research interests include Information Security, Wireless Sensor Networks, Nanocomputing, Reconfigurable Architectures, She has published about 20 papers in reputed journals and 50 papers in conferences.