International Journal of Advanced Research in Engineering Technology & Sciences ISSN: 2394-2819
April-2016 Volume 3, Issue-4
Email: editor@ijarets.org www.ijarets.org

Review on MD5 Hash Function

Deepak Lathwal
M. Tech Scholar
Department of ECE
RIMT Rayat Bhahara Group
Chidana, Haryana

Parveen Khanchi
Assistant Professor
Department of ECE
RIMT Rayat Bhahara Group
Chidana, Haryana

ABSTRACT:
In information security, message authentication is an essential technique to verify that received messages come from the alleged source and have not been altered. A key element of authentication schemes is the use of a message authentication code (MAC). One technique to produce a MAC is based on using a hash function and is referred to as an HMAC. Message Digest 5 (MD5) is one of the algorithms, which has been specified for use in Internet Protocol Security (IPSEC), as the basis for an HMAC. The input message may be arbitrarily large and is processed in 512-bit blocks by executing 64 steps involving the manipulation of 128-bit blocks. There is an increasing interest in high-speed cryptographic accelerators for IPSEC applications such as Virtual Private Networks.

KEYWORDS: Hash Function, MD5, IPSEC

I. INTRODUCTION:

Data integrity assurance and data origin authentication are essential security services in financial transactions, electronic commerce, electronic mail, software distribution, data storage and so on. The broadest definition of authentication within computing systems encompasses identity verification, message origin authentication and message content authentication. In IPSEC, the technique of cryptographic hash functions is utilized to achieve these security services.

HASH FUNCTIONS:

Hash functions compress a string of arbitrary length to a string of fixed length. They provide a unique relationship between the input and the hash value and hence replace the authenticity of a large amount of information (message) by the authenticity of a much smaller hash value (authenticator)[1]. In recent years there has been an increased interest in developing a Message Authentication Code (MAC) derived from a hash code. Among the many reasons behind this are that cryptographic hash functions such as MD5 and SHA-1 generally execute faster in software than symmetric block ciphers such as DES. The software for hash functions is widely available and there are no export restrictions from the United States or other countries for cryptographic hash functions. Hence, there are many applications of MD5, SHA-1 and other hash functions to generate MACs. The method to implement the MAC for IP security has been chosen as hash-based MAC or HMAC, which uses an existing hash function in conjunction with a secret key. The HMAC algorithm is specified for an arbitrary FIPS-approved cryptographic hash function. With minor modification, HMAC can easily replace one hash function with another [2].

MESSAGES DIGEST 5 (MD5) ALGORITHMS:

MD5 [3] is a message digest algorithm developed by Ron Rivest at MIT. It is basically a secure version of his previous algorithm, MD4 which is a little faster than MD5. This has been the most widely used secure hash algorithm particularly in Internet-standard message authentication. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest of the input. This is mainly intended for digital signature applications where a large file must be compressed in a secure manner before being encrypted with a private (secret) key under a public key cryptosystem.

FPGA IMPLEMENTATION:

Re-configurable program such as FPGAs are a highly attractive option for software implementations as they provide the flexibility of dynamic system evolution as well as the ability to easily implement a broad range of algorithms.

Most hash functions are targeted at software implementations. The advantages of software implementations are ease of use, ease of upgrading, portability and flexibility. However a software implementation has more physical security by nature, as it cannot easily be modified by an attacker. On the other hand the speed of a software implementation is restricted to the speed of the computing platform and there are vulnerabilities for viruses and other complications due to system failures.

II. ITERATIVE LOOPING ARCHITECTURE:

By implementing a generic step of the MD5 algorithm,a looping architecture with 64 iterations would seem toprovide the greatest area optimized solution. The blockdiagram of the iterative design is shown in Figure 2.

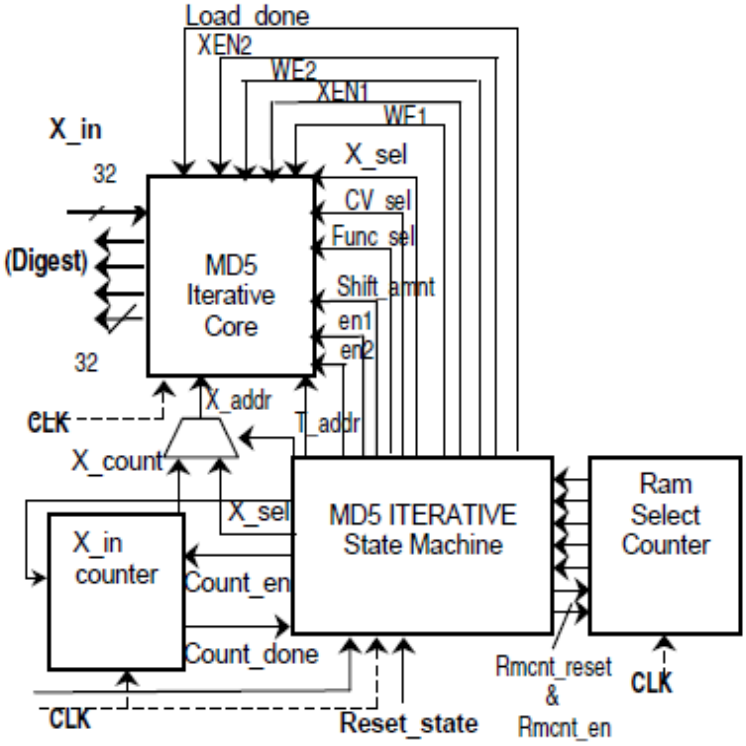

Figure 2 Block diagram of MD5 iterativedesign (Iterate_MD5).

III. IMPLEMENTATION:

MD5 algorithm is a block-chained hashing algorithm. The hash for a block depends on both the block data and the hash of its preceding block. As a result, blocks cannot be hashed in parallel. Each step consists of four additions, three component logical operations, two table lookups and one rotation. The tree of operations can be optimized by performing operations, which involve items not dependent on the previous step, early. The item that depends on the previous step is word B and hence the result of logical operation has a considerable delay. The optimized tree of operation (assuming each operation takes one unit time) will be as given in Figure 1. According to this one time unit step can be reduced [7].

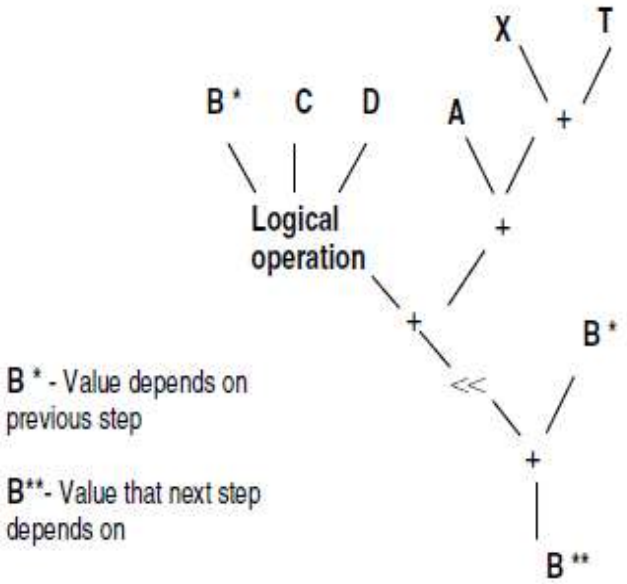

Figure 1 Optimized operation tree

IV. BACKGROUND THEORY:

MD5 is one of the most widely used cryptographic hash functions nowadays. It was designed in 1992 as an improvement of MD4, and its security was widely studied since then by several authors. The best known result so far was a semi free-start collision, in which the initial value of the hash function is replaced by a non-standard value, which is the result of the attack. In this paper we present a new powerful attack on MD5 which allows us to find collisions efficiently. We used this attack to find collisions of MD5 in about 15 minutes up to an hour computation time. The attack is a differential attack, which unlike most differential attacks, does not use the exclusive-or as a measure of difference, but instead uses modular integer subtraction as the measure [2]. We call this kind of differential a modular differential. An application of this attack to MD4 can find a collision in less than a fraction of a second. This attack is also applicable to other hash functions, such as RIPEMD and HAVAL.

MD5 is a well-known and widely-used cryptographic hash function. It has received renewed attention from researchers subsequent to the recent announcement of collisions found by Wang et al. To date, however, the method used by researchers in this work has been fairly difficult to grasp. In this paper we conduct a study of all attacks on MD5 starting from Wang. We explain the techniques used by her team, give insights on how to improve these techniques, and use these insights to produce an even faster attack on MD5. Additionally, we provide an \MD5 Toolkit" implementing these improvements that we hope will serve as an open-source platform for further research [3]. Our hope is that a better understanding of these attacks will lead to a better understanding of our current collection of hash functions, what their strengths and weaknesses are, and where we should direct future efforts in order to produce even stronger primitives.

Hash functions are tools used in integrity of messages, digital signatures and digital time stamping. Message digest algorithms started with public key cryptography for authentication. Digest algorithms compute some hash functions, which are message digest values based on a simple set of primitive operations of 32-bit words. Among the digest algorithms MD4 and MD5 are most popular. Both these algorithms perform a set of bitwise logical operations. They generate 128-bit digest values from a given message. Time complexity of MD5 is more than MD4 and hence somewhat slower to execute. The message digest algorithms MD4, MD5 have been discussed in detail [4]. A new method has been introduced for obtaining collisions for reduced number of rounds of MD4 and MD5 algorithms. The time complexity, performance and attacks of MD4 and MD5 algorithm have been computed using this method. The strength has been computed on change in message; the new method can prove its strength.

V. CONCLUSIONS:

The significance of the software implementation of the MD5 algorithm has been examined. Two programs have been studied for both area utilization and speed with FPGAs as the target program. It is clear that both programs can be easily fitted to a single program. Although the inherent nature of the MD5 structure does not allow parallel hash operations of blocks, software implementations can obtain a significant throughput to cater to some of currently available IP bandwidths.

REFERENCES:

1. B. Preneel, "Cryptographic Primitives for Information Authentication- State of the Art in Applied Cryptography", Lecture Notes in Computer Science vol. 1528, Springer-Verlag Berlin Heidelberg NY 1998.
2. National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication # HMAC, 2001.
3. R. Rivest, The MD5 Message-Digest Algorithm, RFC 1321, MIT LCS & RSA Data Security, Inc., April 1992.
4. W. Stallings, Cryptography and Network Security, Second edition. Prentice Hall, 1997
5. K. Gaj and P. Chodowiec, "Comparison of the Software Performance of the AES Candidates Using Configurable Software", http://csrc.nist.gov/encryption/aes/round2/.
6. Xilinx Inc., Virtex 2.5V Field Programmable Gate Arrays, 2000.
7. J. Touch, Report on MD5 Performance, RFC 1810, June 1995.