On the Complexity Analysis of Message Authentication and Cryptographic Protocol Implementations in Low Cost Embedded Systems

Dimitrios A. Karras
Sterea Hellas Institute of Technology, Automation Dept, Psachna, Evoia, 34400, Greece,
dakarras@teiste.gr, dimitrios.karras@gmail.com, dimitrios.karras@ieee.org

Abstract-- This paper presents an experimental analysis and discussion on implementation issues with regards to the performance of message authentication algorithms in embedded applications. Two such algorithms, namely, MD5 and SHA-1 are investigated when implemented into two different platforms:

1. **32-Bit Processors (Intel processor (Pentium III processor at 1000 MHz))**
2. **8-bit microcontrollers (like PHYTEC miniMODUL-537)**

Computational Complexity as well as CPU-processing time results, regarding the performance of these algorithms, both in the 32-bit processors case as well as in the 8-bit processors case are presented. The CPU-processing time results show that such algorithms could be easily incorporated in embedded applications relying on 8-bit microcontrollers and requiring data integrity assurance and data origin authentication. Such applications could involve electronic mail, system monitoring, data mining, biometric security, secure data communications etc. Although NIST has presented similar analyses regarding cryptographic algorithms [1], algorithms aimed on the other hand at data authentication and security protocols implementation have not been investigated in this context. This is precisely the goal and the contribution of this paper.

Index Terms—Message Digests Complexity Analysis, Hash Functions, Message Authentication Complexity Analysis, Embedded Systems.

I. INTRODUCTION

This paper deals with message authentication algorithms and more specifically with one way hash functions, which are of primary importance for the security mechanisms of telecommunication systems.

One-way hash functions take as input strings of variable length and produce, according to their mapping algorithm, an output of fixed length, generally smaller than the input string, called digest or hash value. One-way hash functions are used in database and file system management and in security protocols such as authentication, data integrity and digital signature mechanisms in telecommunication applications [2,3,4,5,6,7,8]. Regarding their required characteristics, among others, we point out the easiness to compute the digests and the hardness to compute the message from a given digest and to find another message with the same digest value. However, these characteristics are not sufficient for the one-way hash functions to be applied in cryptographic protocols of communication systems. Therefore, collision resistance is required, which means that it is hard to find two or more random messages to have the same digest [6].

Although Message digests are so important in the security mechanisms of telecommunication systems there is no comparative report in the literature regarding their performance characteristics. More emphasis has been drawn in theoretical tests regarding their quality characteristics. Much less emphasis, however, has been put on practical evaluation schemes regarding these quality characteristics. Such practical evaluation methodologies and comparative reports exist with regards only to cryptographic algorithms [1]. The goal of this paper is to present an experimental report for the performance characteristics, regarding computational and processing time complexity of one way hash functions.

In this section, we discuss some aspects of well-known one-way hash functions. The second section is dedicated to the factors that characterize the computational complexity of one-way functions. In the third section, we describe the experimental methodology for evaluating complexity of the algorithms under consideration. Along with MD5 and SHA in this experimental section we include, for comparison reasons the well known CRC algorithms. Finally, we conclude the paper and outline future work on this subject.

Several one-way hash functions have been proposed and are in the meantime extensively used in security mechanisms, such as MD2, MD4, MD5, SHA and functions that rely on symmetric or asymmetric cryptographic systems. In the following, we shortly underline some relevant parameters of the MD5 and the SHA from the points of view under consideration.

The MD5 algorithm handles messages of arbitrary length and produces digests consisting of 128 bits [9, 10, 11]. Each message, after it has been appended by padding bits, is processed in blocks of 512 bits in length. A buffer of 128 bits holds the intermediate and the final results of the one-way function. Four rounds of processing comprise the algorithm. Each round consists of 16 processing steps. The processing relies on values constructed from the sine function, bit rotation and addition modulo 232. Four primitive logical functions are used, where each one of them is used for one out of the four rounds of processing. The logical functions perform a set of bit-wise logical operations and take three 32 bit words as input and produce a 32 bit word as output.

The Secure Hash Algorithm (SHA, [9, 10, 11, 12]) is a variant of the MD4 algorithm, like the MD5 algorithm. However, the algorithm produces digests of 160 bits in length, although it takes as input messages of arbitrary length

Timing results for 8-bit microcontrollers

A.4 SHA-1 calculation complexity –processing time – in 8 Bit microcontrollers

Blocks	*sha1-process messa-ge block*	*sha1-inializ ation of first 16 words*	*sha1-circul ar shifts AND operat ions*	*sha1-circul ar shifts-XOR operat ions*	*sha1-circul ar shifts - AND operat ions*	*sha1-circul ar shifts - XOR operat ions*	*Messa ge digest-XOR and AND opera-tions*	*sha1-pad messa ge block*
sha-1 Algo-rithm CPU Cycles	695000 00 cycles	45800 0000 cycles	34400 0000 cycles	47400 0000 cycles	34400 0000 cycles	34400 0000 cycles	34400 0000 cycles	69500 000 cycles
Repeti tion cycles	10^5	10^5	10^5	10^5	10^5	10^5	10^5	10^5

A.5 MD5 calculation complexity –processing time – in 8 Bit microcontrollers

Blocks	MD5 Block Update operation	MD5 Finali-zation	MD5 basic transfo-rmation	MD5 Encode input
MD5 Algorithm CPU Cycles	80800000 cycles	58700000 cycles	80800000 cycles	34400000 cycles
Repetition cycles	100000	100000	100000	100000

IV. Conclusions

In this paper two very significant message digests algorithms were described in terms of computational /processing time complexity, namely, MD5 and SHA-1. The goal is to understand the requirements of the implementation of such algorithms when they are to be used for embedded applications in secure communication networks. Therefore, the experimental analysis conducted was performed in two computing platforms,

32-Bit Processors (Intel processor (Pentium III processor at 1000 MHz)) and

8-bit microcontrollers (like PHYTEC miniMODUL-537)

The performance of the algorithms to the 32-bit processors was excellent. The speed of calculating the output was very fast. For any input message the message digest of that message was calculated and that message digest was unique for every input. The same performance was also reported to 8-bit microcontrollers. For the SHA-1 algorithm the message digest output was calculated for any multiple of 512-bits because the sha-1 algorithm sequentially processes blocks of 512-bits. The same results were found for the MD5 and CRC algorithms. The timing results from the tables outlined above shows that small times duration were needed for performing any calculation for each algorithm. Finally we can say that these two algorithms can be used in electronic mail, electronic funds transfer, software distribution, data storage, and other applications which require data integrity assurance and data origin authentication even when all these is required to be embedded applications in low cost controllers [13].

References

[1] J. Nechvatal, Report on the Development of the Advanced EWncryption Standard (AES), October 2000, http://csrc.nist.gov/CryptoToolkit/aes/round2/r2report.pdf.
[2] I.B. Damgard, 'Collision Free Hash Functions and Public Key Signature Schemes', Advances in Cryptography, Eurocrypt '87, Lecture Notes in Computer Science 304, Springer Verlag, 1987, pp. 203-216.
[3] I.B. Damgard, 'A Designe Principle for Hash Functions', Advances in Cryptography, Crypto '89, Lecture Notes in Computer Science 435, Springer Verlag, 1989, pp. 416-427.
[4] B. Preenel, 'Cryptographic Hash Functions', Transactions on Telecommunications, vol. 5, 1994, pp. 431-448.
[5] M.N. Wegman, J.L. Carter, 'New Hash Functions and Their Use in Authentication and Set Quality', J. of Computer and System Sciences, vol. 22, 1981, pp. 265-279.
[6] B. Schneier, Applied Cryptography, John Willey and Sons, 1996.
[7] M.Peyravian, A.Roginsky, A.Kshemkalyani, 'On Probabilities of Hash Value Matches', J. Computers & Security, Vol. 17, No. 2, 1998, pp. 171-176.
[8] D. Stinson, 'Combinatorial Techniques for Universal Hashing', J. of Computer and System Sciences, vol. 48, 1994, pp. 337-346.
[9] W. Stallings, Network and Internetwork Security, Prentice Hall, 1995.
[10] C. P. Pfleeger, Security in Computing, Prentice Hal, 1997.
[11] G. J. Simmons (editor), Contemporary Cryptology, The Science of Information Integrity, IEEE Press, 1992.
[12] FIPS PUB 180-1, Secure Hash Standard, 1995.
[13] The 8051 Microcontroller Architecture, Programming and applications by Kenneth j. Ayala