International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

Enhancing SHA-512 by using Iterative Hashing and Swap Function

Sumita Tyagi[1], PG Scholar; **Yaduvir Singh**[2], Associate Professor, Department of CSE, Ideal Institute of Technology, Ghaziabad

Abstract

With the usage of web the web data is increasing at a tremendous rate. So is the concern of the data security. Cryptography has provided a new and secured way to store secret information in the database and most importantly the passwords. This was provided by hashing where the variable length input was converted into a fixed length output using various hash algorithms like MD5 and SHA. Despite various advances in the field of hashing there are still some weaknesses and many attacks on these algorithms have been discovered. In this paper we propose a new way of hashing applied to these algorithms. We enhance the SHA 512 algorithm by combining tree structure and swapping applied together in these algorithms. This helps to provide an output that is more secure and complicated making it more resistant to various collision attacks. Possibility of collision attacks, brute force, rainbow table attacks and birthday attacks is also mitigated by the complex structure of this new proposed algorithm.

Keywords

SHA-512
Iterative Hashing
Rainbow Table Attack
Brute Force Attack
Birthday Attack

Introduction

The Secure Hash Algorithm (SHA) was developed by National Institute of Standard and Technology (NIST) in 1993. SHA takes arbitrary length of input and processes it in fixed length blocks and then produces an output of fixed length. The first version of SHA called as SHA-1 produces an output of 160 bits [7]. This output is generally called as “message digest “or simply “digest”.SHA-1 was prone to many types of attacks [9].Then in 2002, NIST gave three new and revised versions of SHA called SHA-256, SHA-384 and SHA-512 which produces the digest of length 236, 384 and 512 bits respectively.SHA-512 is more popular amongst all the versions and is widely used now-a-days.

Properties of Hash Function

1 These hash functions can be applied to any size data producing a fixed-length output.
2 Hash functions H(x) are relatively easy to compute for any given message x
3 Follows one-way property where it is computationally infeasible to find x such that H(x) = h
4 Have weak collision resistance where it is computationally infeasible to find y ≠ x such that H(y) = H(x)
5 Hash functions are strong collision resistance computationally infeasible to find any pair (x, y) such that H(x) = H(y)

Algorithms at a glance

SHA 512 takes as input a message of any arbitrary length, then processes this input into fixed length blocks of 1024 bits and finally produces a hash value of 512 bits.

Step I: Padding: The message is appended with padding with a string of 1 followed by 0’s.Padding is done to make the variable length message a multiple of 1024 bits.

Step II: Append Length: A block of 128 bits is appended to the message after padding field. This block consists of actual length of message before padding is done.

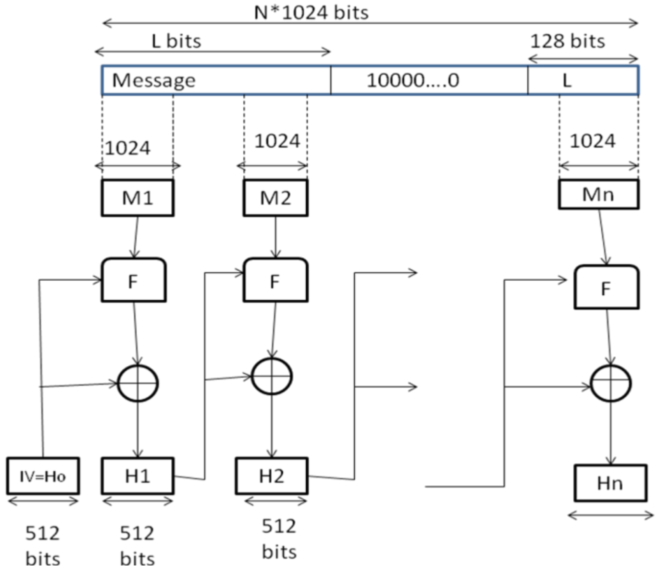


Figure 1: Basic Structure of SHA-512

Step III: Initialize Buffer: Eight 512 bit buffer is designed to hold intermediate and final result of hash function. Each buffer is represented as 64 bit register (a, b, c, d, e, f, g, h)

a=6A09E667F3BCC908
b=BB67AE8584CAA73B
c=3C6EF372FE94F82
d=A54FF53A5F1D36F1
e=510E527FADE682D1
f=9B05688C2B3E6C1F
g=1F83D9ABFB41BD6B
h=5BE0CD19137E2179

Step IV: Process message in 1024 bit blocks: The message is processed in blocks of 1024 bits. Each block goes through a compression function reducing it into 512 bits. This compression function consists of a total 80 rounds increasing the complexity of hash function produced.

Step V: Output: After all N 1024 bit blocks have been processed the output obtained from nth stage is the message digests.

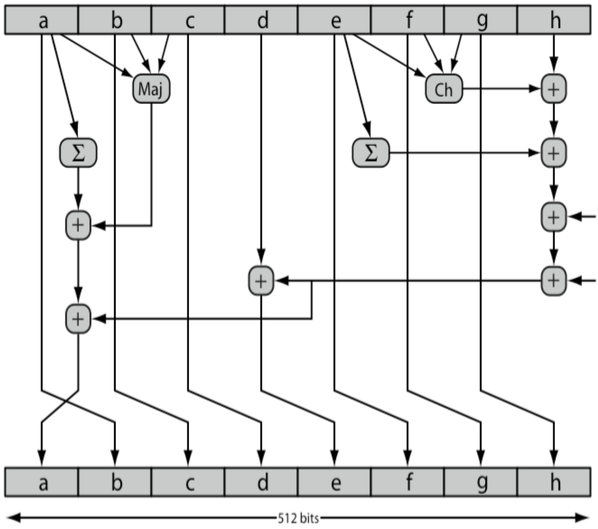


Figure 2: Single Round function on SHA-512

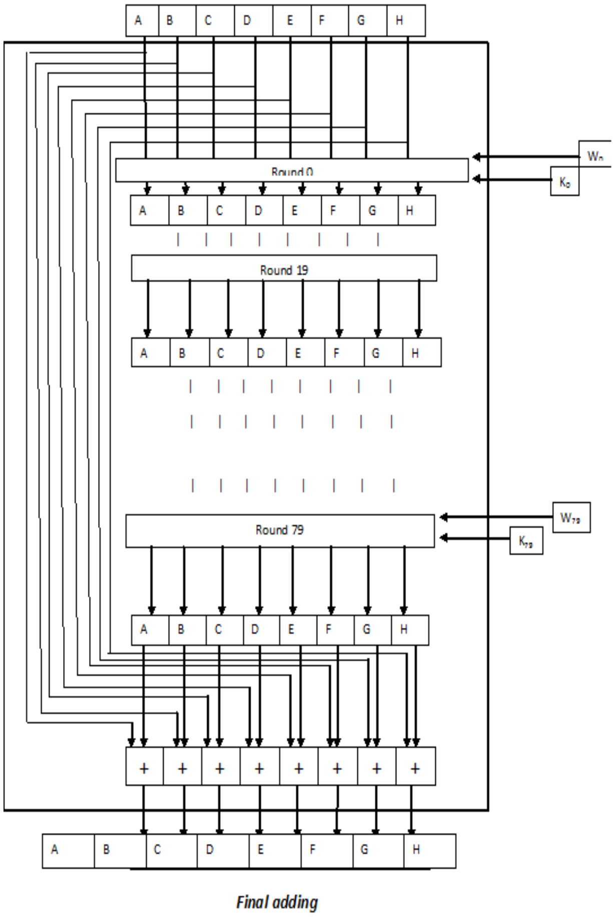
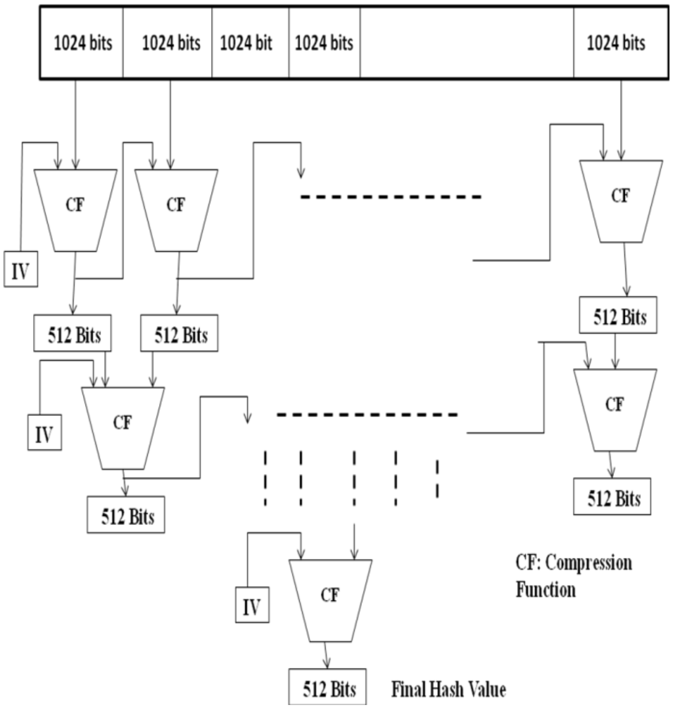


Fig ure 3: SHA 512 compression function

Proposed Algorithm

Recently many new Hash algorithms had been discovered using block ciphers and for digital signatures [2] [6].In the proposed algorithm we modify the basic structure of the algorithm in order to prevent various attacks possible on hash functions[3].We modify the algorithm by introducing the concept of Iterative Hashing[1]. We implement the iterative hashing by implementing it through the tree structure.



Step I: Append Padding Bits: It is similar to step existing algorithm

Step II: Append message length: It is also similar to existing algorithm

Step III: Initialize Buffer: It is also similar to the already existing algorithm.

Step IV: Process message in 1024 bit blocks: Here we implement the change in the existing algorithm. We introduce the concept of swapping the buffer values.

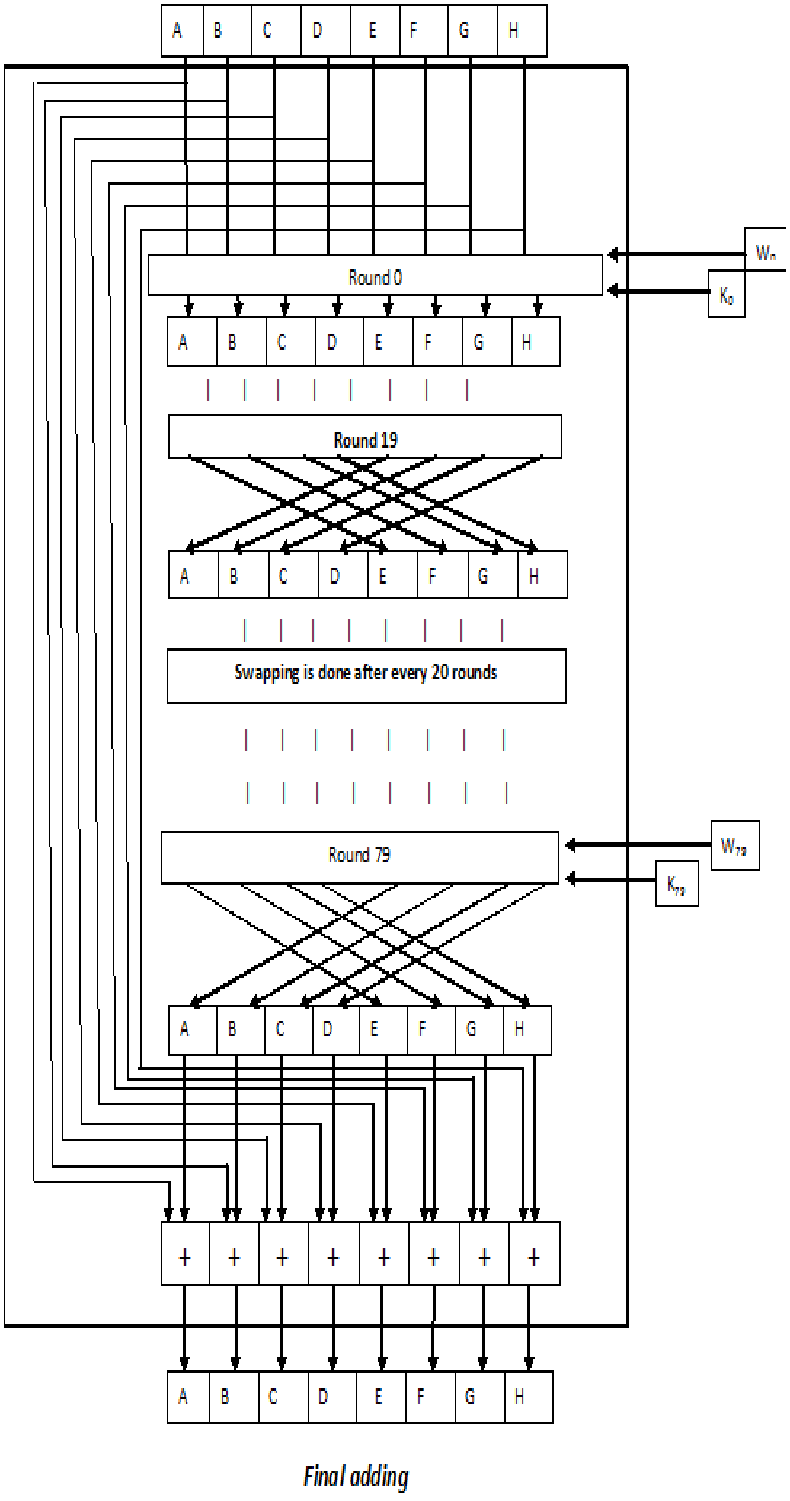


Figure 4: Modified SHA 512 compression function

Step V: Output: This algorithm outputs a more secure and complex Message Digest or we can say Hash value of 512 bits. Hash codes are generally used to save passwords in database. The hash value produced as output is saved in the database. Whenever user enters a password to unlock something then the hash value of the text entered is calculated and matched with the one saved in the database. If both the values match that means the person is authentic user.

Conclusions

SHA 512 was prone to various types of attacks [3].We have provided a more complex and secure version of the previous algorithm by modifying it. We introduce iterative hashing in the basic structure of SHA -512 which makes it more complex than the previous version. In this iterative hashing the intermediate results produced by each 1024 bit block is hashed again by combining two blocks of 512 intermediate results. The process is repeated until we get a single 512 bit hash code. Next change we introduce is in the compression function of the algorithm. This compression function is the heart and core of the SHA-512. We introduce a small change which is applied at every 20 rounds. The change is to simple swap the values of buffer a, b, c and d with the values of buffer e, f, g and h. This change makes the message digest very secure. Even if a single bit of message is changed then there is a lot of change in the message digest.

References

[1] E. Biham O. Dunkelman " A framework for Iterative Hash functions" Proceeding second NIST workshop2006, Santa Barbara, USA, August 2006.

[2] Vlastimil KLIMA "A new concept of Hash functions SNMAC using a special block cipher and NMAc/HMAC construction", Eurocypt 2007

[3] Marc Martinus Jacobus Stevens "attacks on hash functions and application"Centrum Wiskunde & Informatica March 2010.

[4] Mohammad Abu Taha, Mousa Farajallah, Radwan Tahboub: A Practical One Way Hash Algorithm based on Matrix Multiplication, International Journal of Computer Applications (0975 – 8887)Volume 23– No.2, June 2011

[5] Erika Batista, Gaël Canal, Karim Ziadeh: The Birthday paradox Operational Research and Optimization, December 2012.

[6] Thulasimani Lakshmanan and Madheswaran Muthusamy, A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes, The International Arab Journal of Information Technology, Vol. 9, No. 3, May 2012

[7] Dai Zubin,Zhou Ning:FPGA Implementation of SHA-1 Algorithm,IEEE-2003,pp 971-975.

[8] Rajeev Sobti, G. Geetha "Cryptographic Hash Functions: A Review" IJCSI, Volume 9, issue 2, March 2012, ISSN (online) : 1694-0814

[9] X. Wang, Y.L. Yin, H. Yu, "Finding collisions in the full SHA-1," Advances in Cryptology, Proceedings Crypto'05, LNCS 3621, V. Shoup, Ed., Springer-Verlag, 2005, pp. 1–16.

Biographies

Sumita Tyagi received B.Tech Degree in Information Technology from Ideal Institute of technology affiliated from Uttar Pradesh Technical University in the year 2006. Currently she is pursuing her M.Tech in computer science under Uttar Pradesh Technical University. Her interest area includes Cryptography and Network Security, Algorithms and Networking. Sumita Tyagi may be reached at **sumita.tyagi12@gmail.com**

Dr. Yaduvir Singh is P.hD in Computer Science and is presently working as Associate Professor in CSE Department of Ideal Institute of Technology. He has a total experience of around 10yrs in academics. Dr. Yaduvir Singh has published about 4 international and 7 national papers. His areas of interest include Networking, Security and Algorithms. Dr. Singh may be reached at **yaduyash@gmail.com**