

ASH 512 Based Secured core banking using by Register Reusing Scheduling

¹K. Kannan and ²T. Radharani

¹Department of Mathematics, ²Department of Computer Science,
Srinivasa Ramanujan Centre, SASTRA University,
Kumbakonam-612001 India.

anbukkannan@rediffmail.com and raniradha@src.sastra.edu

November 8, 2017

Abstract

This paper discuss the secured communication in net banking using ASH512 hash function. Core banking is major demand part for the net banking users. Net banking customers may access their funds and perform the transaction from one account to another. The risk part of the core banking process is to store the customer ID, password and card numbers. There are various cryptographic algorithm available to encryption and decryption we propose hash function for both securing storage and authentication. The proposed system of the paper is to implement ASH 512 hash function using register reusing like parallel procession unit improves the execution speed. Also securing the password and authenticating the user. The algorithm accepts 1024-bits as input and produces 512-bits as output. And the algorithm and method can be implemented as web app for multi user by Angular Java Script.

AMS Subject Classification: 11T71, 00A69

Key Words and Phrases: Banking, security standards, short-time passwords.

1 Introduction

Today, most applications of net banking need to be secure as their core system. Since the design and algorithm of hash function has improved steadily, their detection is a critical problem. As a result, it is nearly impossible to be sure whether a pc connected to the internet can be considered risk and provide the safe guard. How to handle web applications that require a high level of security, such as net banking and core banking. In a net banking system, there is a chance of getting encounter by cookies, password, card information at transaction. And in internet banking system, the credentials of the customer may be misused and hacked. Thus security is still a risk in these applications. Here, we propose a technique to secure the customer information with hash function and to prevent the possible forgery of password hacking. The concept of image processing, an visual cryptography is used. Hash function technique processing is a technique of processing an input password and to get the output as either improved form of the same data and/or characteristics of the input data.

In traditional banking process is a threat of forgery during user transactions. In online banking, security begins with the authentication method, used to confirm the authenticate and security, and not someone who has stolen your identity. Authentication generally involves one or more basic factors: user knows (e.g., password, PIN, ATM card, smart card, biometric characteristic and fingerprint).

Single factor authentication uses one of these multi user authentication more than one, and thus is considered a stronger fraud deterrent. In case of using ATM, for example, you are utilizing multi-user authentication is risk. Factor number one is something you have, your ATM card; factor number two is something you know, your PIN. But in any of these methods, key elements like passwords can be misused and hacked. So here we propose a technique to shield the customer information and to defend the possible forgery.

2 Related Work

In [1], There are a continuous growth customers numbers has increase who use core banking its convenience make the user simple and better. But the security and privacy of Information may be one of the biggest concerns to the Online Banking users. Core banking is a set of services provided by a group of networked bank branches [2]. Bank customers may get their banking data and perform other transactions from one to another. Now a day banking system has meet features of banking for the user, which is a set of services, where authentication plays important role [3, 4, 5, 6]. Now a days, because of increase of realization and growth in the field of hacking it is not safe to rely on internet to store all the information.

In [7], there are continuously growing number of customers who use Internet banking because of its convenience. But the security and privacy of customer information may be one of the biggest concerns to the Online Banking users. Sending and receiving information over the internet is easy, fast and cost effective due to development of information [8]. Now a days most of the information either personal or official is communicate by internet only. Most of the paper based documents are replaced by electronic-documents.

3 System Requirements

Software Requirements

Front end: HTML5, CSS3

Backend: PHP, Mysql, Angular Java Script

Software Tools: Eclipse Mars

Tools: Chrome browser

4 Proposed Algorithm

Design Considerations:

- Securing the password, cvv, debit and credit card numbers, customer id
- Using Ash 512 algorithm
- Reusing the register scheduling
 - Register Reuse Scheduling (RRS), a novel approach to global register allocation that derives a register assignment while reducing code by locally re ordering independent block operations.
 - This new method aimed at minimizing spill code by attempting to find an set instruction schedule to allows for the less expensive register allocation.
- Parallel processing
 - A mode of operation in which a process is split into parts, which are executed simultaneously on different processors attached to the same computer.

5 Modules

A. Existing System

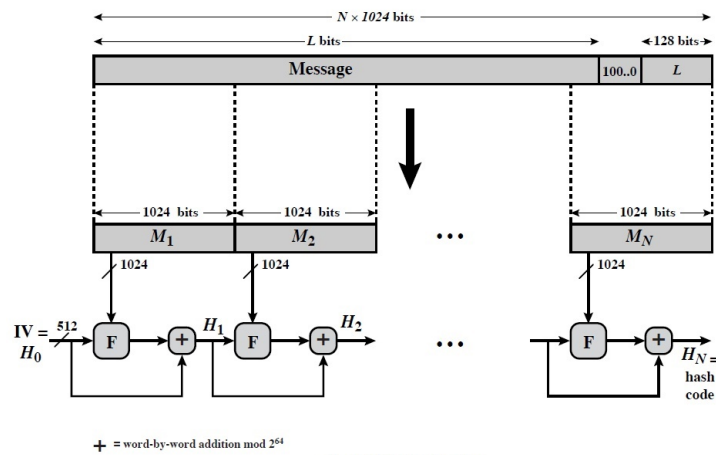


Figure 5.1. Existing Algorithm

B. Proposed Architecture

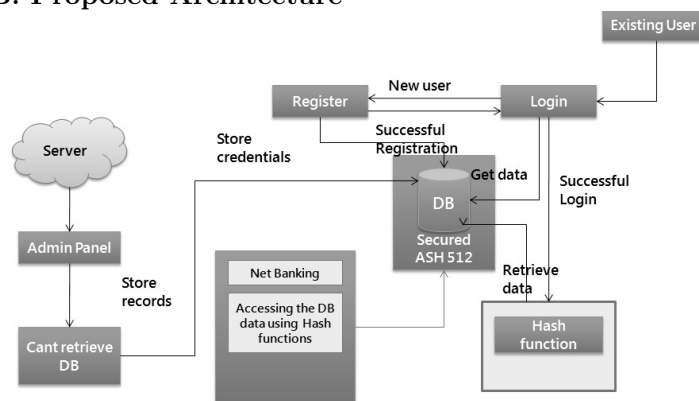


Figure 5.2. Proposed Architecture

6 Results and discussion

These two algorithms have been implemented in Java and run on Intel Dual CPU 1.86GHz, 1GB RAM and Windows-XP.

COMPARISON OF ASH-256 AND SHA2 (256)

Input	Hash code of ASH-256	Hash code of SHA2(256)
"" (empty string)	289caa11880507deb7c611 ef34bfe5bf3ba226ad5055 1d2ffe1fd1679a0b83ec3	e3b0c44298fc1c149af bf4c8996fb92427ae41 e4649b934ca495991b 7852b855
"a"	ed3ac8083dfecb651ce3 6aec84c478f53fd6455c 364fc8573dd60df83216 981c	ca978112ca1bbdcafacc 231b39a23dc4da786ef f8147c4e72b9807785a fee48bb
"abc"	af6192fede3cd581066b 1cf2ce22471d367cd5b8 0bb36e84c0c879c0099d b0b2	ba7816bf8f01cfea4141 40de5dae2223b00361a 396177a9cb410ff61f2 0015ad
"01234 56789"	6be2b83ad4bbd85d473 1f799b90db6cd3aa979e d94c5d85d5273ad55bc 7e0442	84d89877f0d4041efb6 bf91a16f0248f2fd573e 6af05c19f96bedb9f88 2f7882
"message digest"	e9526cad2e76607d4ab a1b6527d58da93b0d80 0573a79f4ed1ec48228a	f7846f55cf23e14eebea b5b4e1550cad5b509e3 348fbc4efa3a1413d39

Table 6.1. Comparison of ASH-256 and SHA2 (256)

Operation	ASH-256	SHA2(256)
Bitwise operations	704	1024
Addition	32	600
Subtraction	144	-
Multiplication	192	-
Division	16	-

Table 6.2. Comparison between number of operations of ASH-256 and SHA2 (256)

Outputs of ASH-256 for three sample inputs:

String : "The quick brown fox jumps over the lazy dog.

Output (Hash Code) : " 8802 012c 743d c2e1 94da f4b8 d58e c487 e789 2037 2449 6bd0 7cb7 c482 59d7 0236

String : The quick brown fox jumps over the lazy eog.

Output (Hash Code): "2001 bd07 230a d1c9 175d 7ae0 58d4 4710 fa53 e79e 0034 1238 1aad 6bd3 0cc6 6bd8

Outputs of SHA2 (256) for three sample inputs:

String : "The quick brown fox jumps over the lazy dog.

Output (Hash Code): ef53 7f25 c895 bfa7 8252 6529 a9b6 3d97 aa63 1564 d5d7 89c2 b765 448c 8635 fb6c

String : "The quick brown fox jumps over the lazy cog.

Output (Hash Code): 35e4 3eb1 9676 8d53 338a 7ea9 24c7 1e50
36c3 5763 f3f1 1347 63e3 4dd5 d019 76e9

Input String : "The quick brown fox jumps over the lazy eog.

Output (Hash Code): bd0c f2c3 ee82 be88 9a1b 5b75 4d2e 4a2a
e07d 22da 40c3 9380 da06 1092 ca16 38d7

In the above example, changing dog to cog produces a hash with different values for 117 of the 256 bits and changing dog to eog produces a hash with different values for 125 of the 256 bits (Fig. 5.2). These two algorithms have been implemented in Java and run on Intel Dual CPU 1.86GHz, 1GB RAM and Windows-XP.

Outputs of ASH-256 for three sample inputs:

Input String : "The quick brown fox jumps over the lazy dog.

Output (Hash Code) : " 8802 012c 743d c2e1 94da f4b8 d58e
c487 e789 2037 2449 6bd0 7cb7 c482 59d7 0236

Even a small change in the message will, with overwhelming probability, result in a completely different hash due to the avalanche effect. For example, changing dog to cog produces a hash with different values for 126 of the 256 bits and changing dog to eog produces a hash with different values for 126 of the 256 bits (Fig. 8).

Input String : "The quick brown fox jumps over the lazy cog.

Output (Hash Code): 84cf 499f 4fa1 2035 2b84 e68e 303b a06c
1cc0 312f db51 b29e 4d97 85d1 002d 1aa0

Input String : "The quick brown fox jumps over the lazy eog.

Output (Hash Code): 2001 bd07 230a d1c9 175d 7ae0 58d4 4710
fa53 e79e 0034 1238 1aad 6bd3 0cc6 6bd8

7 Conclusion and Future Work

The simulation results showed that the proposed algorithm performs better with reusing register scheduling and hash function. The proposed algorithm provides security, authentication and high speed. As the performance of the proposed algorithm is analyzed between two speed and security in future with some modifications in design considerations the performance of the proposed algorithm can be compared with other security factors. We have used for 100 data set and algorithm verified. We can increase the number of nodes and analyze the performance.

Acknowledgment

This study was supported by the Department of Science and Technology, Government of India (Reference number: SR/FST/MSI-107/2015(C)). The authors also thank DST(FIST) for providing financial support.

References

- [1] Upadhyay Anamika Atulkumar, Secure Internet Banking Based On Cryptography, *ASMs INTERNATIONAL E-Journal on Ongoing Research in Management and IT E-ISSN-2320-0065*, 1-5.
- [2] Chandrasekhara and Jagadisha, Secure banking application using visual cryptography against fake website authenticity theft, *International Journal of Advanced Computer Engineering and Communication Technology* **2(2)** (2013), 1-5.
- [3] N. Anusha, P. SubbaRao, Visual Cryptography Schemes for Secret Image, *International Journal of Engineering Research & Technology* **1(5)** (2012), 1-5.
- [4] B. Borchert, Segment Based Visual Cryptography, *WSI Press, Germany, 2007*.
- [5] W-Q Yan, D. Jin and M. S. Kanakanahalli, Visual Cryptography for Print and Scan Applications, *IEEE Transactions, ISCAS-2004*, 572-575.
- [6] H. Yan, Z. Gan and K. Chen, A Cheater Detectable Visual Cryptography Scheme, *Journal of Shanghai University*, **38(1)** (2004).
- [7] S. Priya, M. Malathi and G. Jayaseelan, Secured internet banking using image captcha technique, *International Journal of Emerging Technology and Innovative Engineering* **1(11)** (2015), 1-5.

- [8] Pallipamu Venkateswara Raoa, K. Thammi Reddyb and P. Suresh Varma, ASH-512: Design and implementation of cryptographic hash algorithm using co-ordinate geometry concepts, *Journal of Information Security and Applications* **20** (2015), 47–60.

