BUILDING SECURITY BARRIERS BY MODIFIED ALGORITHMS IN BLOCKCHAIN TO PREVENT SQL INJECTION AND XSS

Joshi Padma N

JNTUH Research Scholar, Associate Professor
Dept of CSE, Sreyas Institute of Engineering and Technology
Hyderabad, India
padmajoshi2015@gmail.com

Dr. N. Ravishankar

Professor, Dept of CSE
Geethanjali College of Engineering and Technology
Hyderabad, India
ravish00@yahoo.com

Dr. M. B. Raju

Professor, Dept of CSE
Pallavi Engineering College
Hyderabad, India
drrajucse@gmail.com

N.Ch. Ravi

Associate Professor,
Dept of CSE,
Pallavi Engineering College,
Hyderabad, India
ravi@saimail.com
N.Ch. Sai Vyuha
B.Tech student Dept of CSE,
BRECW, Hyderabad

Abstract
Now a days security of user credentials and confidential data of the user in database plays important role in web applications. Blockchain technology performs the hashing during the block generation process. MD5, SHA128, SHA256, and SHA512 are only few of numerous hashing algorithms. The use of SHA256 in Blockchain has been discovered. There are several aspects to consider while picking an appropriate hashing algorithm, including collision ratio, storage space, and time complexity. There are fewer collisions with SHA256 and it takes up less space. Modified MD5 is the hashing algorithm used in the suggested research. On the other hand, SHA256 takes more storage capacity, but traditional MD5 has a lower collision resistance. study's objective is to develop a faster and more collision-resistant version of the standard MD5 algorithm. Because of this, a system that is both secure and efficient is essential. This study tested the improved MD5 for storage capacity and collision probability, as well as a CPU clock cycle simulation. In terms of collision resistance, the improved MD5 results clearly exceed SHA256, MD5, and SHA1 while consuming less storage space and time. Moreover paper is also considering role of block chain to enhance security by preventing Sql Injection and XSS.

***Keywords*:** Blockchain, MD5, SHA1, SHA256, SHA512, collision resistance, SQL Injection, Cross Script

1. Introduction

Many people in the IT industry and beyond are interested in the "distributed ledger [1]" technology known as blockchain. By offering a safe, transparent, highly resistant to failure, auditable and efficient way to record transactions or any digital interaction, blockchain technology [2] has the potential to upend industries and create new business models. While the technology is still developing in its early phases, widespread commercialization is yet some years away. It's possible to use blockchain technology to store an ever-increasing volume of data [4]. Because it is decentralised, no one computer has complete command over the whole network. Rather, each of participating nodes receives a copy of chain from the others. New recordings are added to chain, making it ever-expanding as well.

There are two types of components that make up a blockchain:

- A transaction is a specific action taken by a member of the system.
- Blocks maintain track of these guarantee and transactions that they haven't been altered.

1.1 Features of Blockchain

There are two features of Blockchain

- Accessibility
  1. Private: Data can't be accessed anywhere if it's private. It permits entry to a legitimate site.
  2. Public: The term "public" denotes that anybody anywhere on the planet may see anything.
  3. Hybrid: Public and private accessibility are combined in this way.
- Decentralized: Transactions are not approved by a centralised authority in this situation.

All parties have access to the blocks and transactions included within. Even though everyone can see your public key, only you have access to the substance of the transaction.

The decentralised nature of Blockchain means that no one authority may authorise transactions or set certain requirements for the approval of transactions. In order to accept transactions, all network participants must come to an agreement. This requires a high degree of confidence.

The fact that it's secure is the most important consideration. Prior entries cannot be changed, only added to the database.

1.1.1 Working of Blockchain

If a new transaction has to be added to the chain, it will be verified by all of the network's members. According to the Blockchain [6] system, "valid" is defined in a way that differs from system to system. After then, a majority of the parties must agree to the deal. All authorised transactions are stored in a single block, which is sent to every node in the network as a copy of the whole block. They then check to see whether the new block is legitimate. A hash of the previous block is included in every subsequent block, resulting in a unique fingerprint.

Designed to be safe and fault-tolerant, blockchain are distributed computer architecture. Blockchains were created in order to achieve decentralised consensus. Blockchain may be used to store data of medical side and to do different record-keeping functions, like verifying identification, executing transactions, preserving provenance, and tracking food.

1.1.2 Hashing in Blockchain

When a string of letters and numbers is fed into a hash function, it is transformed into a fixed-length encoded output. As a result of a mathematical formula, it is a necessary part of running the cryptocurrency Blockchain network. Crypto currency's Hash Rate is the speed at which a computer completes a crypto currency [7] coding action. The result of a hash function is a hash value. Mining is extremely profitable because of the high rewards associated with discovering the next block and collecting the prize.

1.1.3 SQL Injection

In the worst-case scenario, the database may be breached and all of the company's sensitive data would be available to anybody with access to the database. Using SQL injection, an attacker may alter SQL queries and get access to the back-end database. The ability to alter the attacker's entry into the database allows the attacker to take use of SQL's syntax and capabilities. For the database, you may also utilize the OS functions.

SQL Injection is an attack that uses code and a SQL scan to get access to a database that isn't permitted. To help you understand, we'll use the example below. Banks have their own website where users may log in using a password and username. Authentication will take place and you will be logged in once you input the right username and password

1.1.4 SQL Injection Methods

Because user input is placed into existing SQL query without adequate validation, SQL injection is possible.

The following three key groups can be divided for SQL injection attacks:

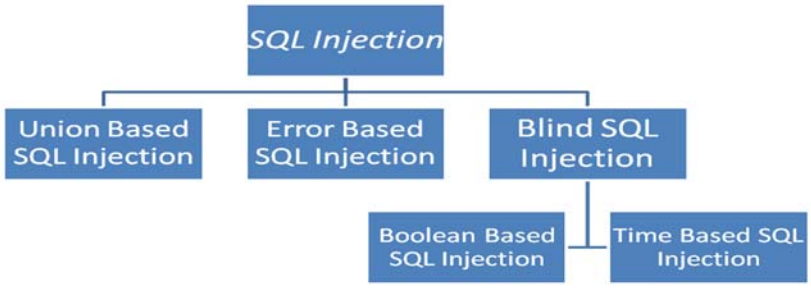

Fig.1. Division of SQL injection methods

1.1.5 Cross Script

Cross-site scripting allows an attacker to influence how users interact with an online programme. An attacker may evade the policy, which is supposed to keep websites different, by exploiting this weakness.

2. Literature Review

Sno.	Author (Year)	Objective	Research methodology	Limitation
1	Joshi Padma (2019)	Sql Injection and CSS in the Case of a Web Application Sql Injection and CSS in the Case of a Web Application Encountering a SQL Injection in Web Applications is being researched.	Sql injection	Research is limited to sql injection and ignored other attacks
2	Joshi Padma (2018)	Http's Security against Cross Scripting Consideration and SQL Injection	Sql injection	Did not provide solution for brute force attack.
3	Joshi Padma (2017)	The most effective defence against unauthorised access to a computer network Web server's influence on methods of risk reduction and SQL Injection	Sql injection	Threat of man in middle attack.
4	Parveen Sadotra (2017)	CSS and SQL Injection in Web Applications are being explored.	Sql injection	Research is taking lot of time during intrusion detection.
5	Nabeel Salih Ali (2016)	Web protection application for the detection of structured query injection attacks using real-time technology	Sql injection	Need to introduce more technical team.
6	Swathy Joseph (2016)	Evaluating Electiveness' Of Conventional Fixes For standardized query language Injection Vulnerability	Sql injection	Complex to implement.
7	Swathy Joseph (2016)	Evaluating Effectiveness of Conventional Fixes for standardized query language Injection Vulnerability	Sql injection	No solution is provided for cross script
8	Abirami J, (2015)	Injection attacks against standardised query languages are one of the most common types of web security vulnerabilities.	Sql injection	Did not provide solution for vulnerabilities.
9	Bharti Nagpal(2015)	XSS Mapping Podel for CSS and Standardized Web Injection Language Attacks: SECSIX security engine for XSS, CSRF, and sql injection	Sql injection	Research scope is limited.
10	Rathod Mahesh Pandering (2015)	Injection attacks against standardised query languages are one of the most common types of web security vulnerabilities.	Cross script	Does not provide security from intrusion detection.
11	Mukesh Kumar Gupta.(2015)	Security vulnerabilities in web applications Predicting Cross Site Scripting (XSS)	Cross script	Research did not considered sql injection
12	Sonewar, Piyush A. (2015)	A new way to identify attacks on SQL injection and site scripts	Sql injection and cross script	Need to do more work to improve the security
13	Wang, Rui, et al.(2015)	Enhanced N-gram method for on-line social network cross-site scripting detection	Cross script	No solution for sql injection

14	Habeeb Orotund (2014)	Mitigating standardized query language Injection Attacks Via Hybrid Threat Modelling	Sql injection	The implementation of hybrid model is complex.
15	Amirmohammad Sadeghian (2014)	Using Header Sanitization, Standardized Query language Injection Vulnerability General Patch	Sql injection	The limited security feature have been proposed.
16	Rocha, Thiago S., and Eduardo Souto. (2014)	Enhanced N-gram method for on-line social network cross-site scripting detection A programme to identify Cross-Site Scripting vulnerabilities automatically ETSS Detector	Cross script	Need to do work for sql injection
17	Yusof, Imran, and Al-Sakib Khan Pathan.(2014)	Pattern-based filtering prevents XSS attacks from persisting.	Cross script	Need to add on security features for sql injection.
18	Sonam Panda(2013)	Anti-Sql-Injection Security for Internet	Sql injection	Web application need more security from brute force and man in middle attack.
19	Kindy, D.A., & Path an, & A.K.:(2013)	Vulnerabilities in web application SQL injection are examined in detail.	Sql injection	Need to do work on intrusion detection.
20	Jane, P.Y. (2013)	Database Intrusion Prevention and Detection System: SQLIA.	Sql injection	Research is considering only prevention mechanism and providing limited solution.
21	Etienne Janot(2012)	Avoiding injections in online applications using a standardised query language: a research with suggestions and a Java solution prototype	Sql injection	There is lack of technical work.
22	Lwin Khin Shar (2012)	Exploiting vulnerabilities in standardised query languages and cross-site scripting OWASP	Sql injection, cross script	Scope of research is limited.
23	Pankaj Sharma (2012)	Implementing an integrated strategy to guard against CSS attacks and standardised query language attacks	Sql injection	Several attacks such as intrusion and man in middle have been ignored.
24	Selvamani, K. (2011)	New approach to SQL attack prevention using cryptography and control policies for access	Sql injection	Research is suffering due to limitation of Cryptography
25	David A. Shelly (2010)	Use Web Server Test Bed to analyse Web Application Vulnerability Scanner limitations	Vulnerability Scanners	Research ignored cross script and sql injection.
26	AppSec DC (2010)	This congress includes the security testing and vulnerabilities of online applications.	Vulnerability Scanners	Ignored sql injection and cross script
27	Bisht, P. (2010)	Using dynamic candidate assessments, SQL automates the mitigation of injection risks.	SQL injection	Research did not considered cross script
28	Van Gundy (2009)	On Thwart Noncespaces, randomization is used to impose information flow monitoring and to prevent cross-site scripting attacks.	Cross script	Did not considered sql injection
29	Swap nil Kharche(2007)	Preventing standardized query language based Injection Attack with the help of Pattern Matching Algorithm	SQL injection	There is need to introduce protection against man in middle and bruteforce attack.
30	Halfond (2005)	AMNESIA: mitigating SQL injection assaults analysis and monitoring.	SQL injection	Did not considered cross script and Vulnerability Scanning
31	StephenW. Boyd (2003)	SQL: Preventing SQL Injection Attacks	SQL injection	Ignored web based authentication
32	Sampada Gadgil	SQL attacks and methods of prevention	SQL injection	Did not considered cross script

.

3. Hashing Techniques

It is the technique of converting any length input into a cryptographic fixed output by using a mathematical procedure. The following algorithms are used to determine the hash value of any Blockchain block:

3.1 MD5

In the cryptographic hash function known as MD5 (Message Digest), a 128-bit hash value is generated, which is written in text form. MD5 has been used in cryptography and is often used to check the integrity of data. [21]

3.2 SHA1

A cryptographic hashing algorithm devised by the National Security Agency (NSA) is called SHA1 (NSA). SHA1 generates a 40-digit hexadecimal hash value with a length of 160 bits (20 bytes). One of the most often used SHA hash functions is the SHA1 algorithm, which may be found in a number of popular applications and protocols. SHA1 should be avoided at all costs. [21]

3.3 SHA224

NSA developed cryptographic hash code Secure Hash Algorithm 224. SHA224 generates a 56-digit hexadecimal hash result with a 224-bit (28-byte) length. [20]

3.4 SHA256

NSA developed cryptographic hash function SHA256. Hexadecimal numbers are used to represent SHA256's 256-bit, which is 64 digits in length. [24]

3.5 SHA384

NSA developed cryptographic hash algorithm SHA384. Hexadecimal numbers of 96 digits are used to represent the 384-bit (48-byte) hash value generated by the SHA384 algorithm. [20]

3.6 SHA512

NSA developed cryptographic hash algorithm SHA512. SHA512 generates 128-bit hexadecimal value from a 512-bit (64-byte) hash value. [23]

A table summarising hashing algorithms has been provided.

Keys for comparison	MD5	SHA
Security	Less secure than SHA	More secure than MD5
Message Digest Length (bits)	128	160
Attacks needed to find out original message (bit Operation)	2128	2160
Attacks to try and find two messages producing the same MD (bit Operation)	264	280
Speed	Faster	Slow
Successful attacks so far	Yes	No
Collision Resistance	Less Collision resistance	More collision resistance

Table 1. Different types of Hashing Algorithms

As seen in the accompanying graph, MD5 is quicker and less secure than SHA. A malformed MD5 method that is more secure and less susceptible to collisions than the traditional MD5 algorithm is the goal.

4. Traditional MD5

The 64 operations performed by MD5 are divided into four 16-operation rounds, each of which performs a total of 64 operations.

A nonlinear function, F, has been explored. In each cycle, the function is used.

There's a 32-bit chunk of the message that Mi is pointing to Ki is a 32-bit constant that may change depending on the operation being performed. s is the number of places to rotate the left bit. The value of s changes depending on the procedure.

⊞ signifies the modulo 2^{32} addition of two numbers.

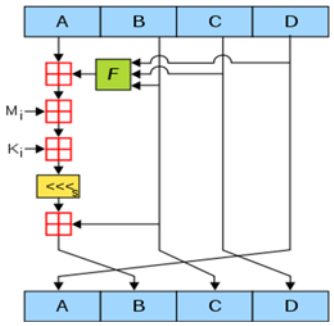

Fig.2. MD5 Model

This section contains code for a modified MD5, which concatenates a integer with the MD5 hash number to lessen the likelihood of a collision. In addition, this method was proven to be quicker than SHA256.

Hexadecimal values are often used to represent 128-bit MD5 hashes. An MD5 hash and 43-byte ASCII input are shown in following example.

MD5("The quick brown fox jumps over the lazy dog")
9e107d9d372bb6826bd81d3542a419d6

5. Proposed Work

In proposed work MD5 mechanism has been modified by integrating AES to use in block chain. And this block chain mechanism has been used to prevent Sql Injection.

5.1 *Proposed Modified MD5*

A 64-bit MD5 has been proposed as an improvement. Here, the efficiency of MD5 has been tested. However, extra bits were connected to boost the device's resilience to collisions. A 32-bit addition random number was used to achieve this. New algorithm models have been shown in Figure 3.

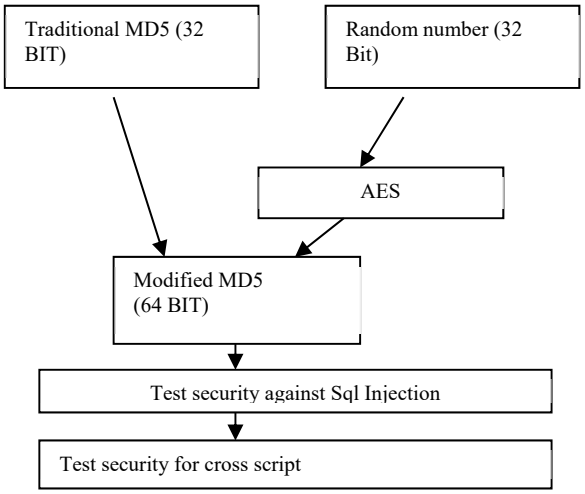

Fig. 3. Proposed MD5 to Prevent SQL Injection and Cross Script

In the proposed effort, a random key was established in order to reduce the likelihood of collisions in the present MD5. The traditional MD5 has been made more secure by including AES. SHA 386 and SHA 512 are not considered for testing since their file sizes are larger than SHA 256 and MD5. The number of clock cycles rises as the clock size grows.

The performance of a hashing method is measured by the total number of CPU clock cycles required during execution, storage space, and collision probability.

Performance of hashing algorithm

$$= f(\text{cpu clock cycle consumed}, \text{storage space}, \text{collision probability})$$

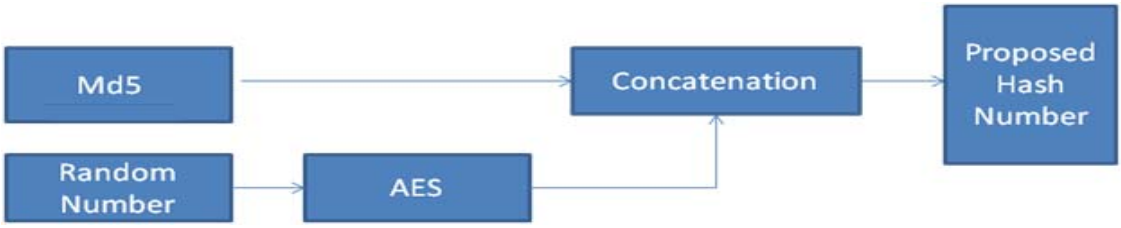

Fig.4. Proposed Model

5.2 Proposed Method and Framework

We will encounter both XSS and SQL injection attacks in this framework by employing

1) SQL injection attacks may be detected and prevented using a variety of techniques, including multiplicative inverse encryption, deep learning with Multi head LSTM like transformers, block chain technology, tokenization, WAF with deep learning approach and pattern locking.
2) To guard against cross-site scripting (XSS) secure session sanitization, a system employs, , Deep learning with transformers, block chain technology,escaping,filtering methods of XSS,WAF with deep learning approach and pattern locking following SQL injection detection and prevention and successful user log in.

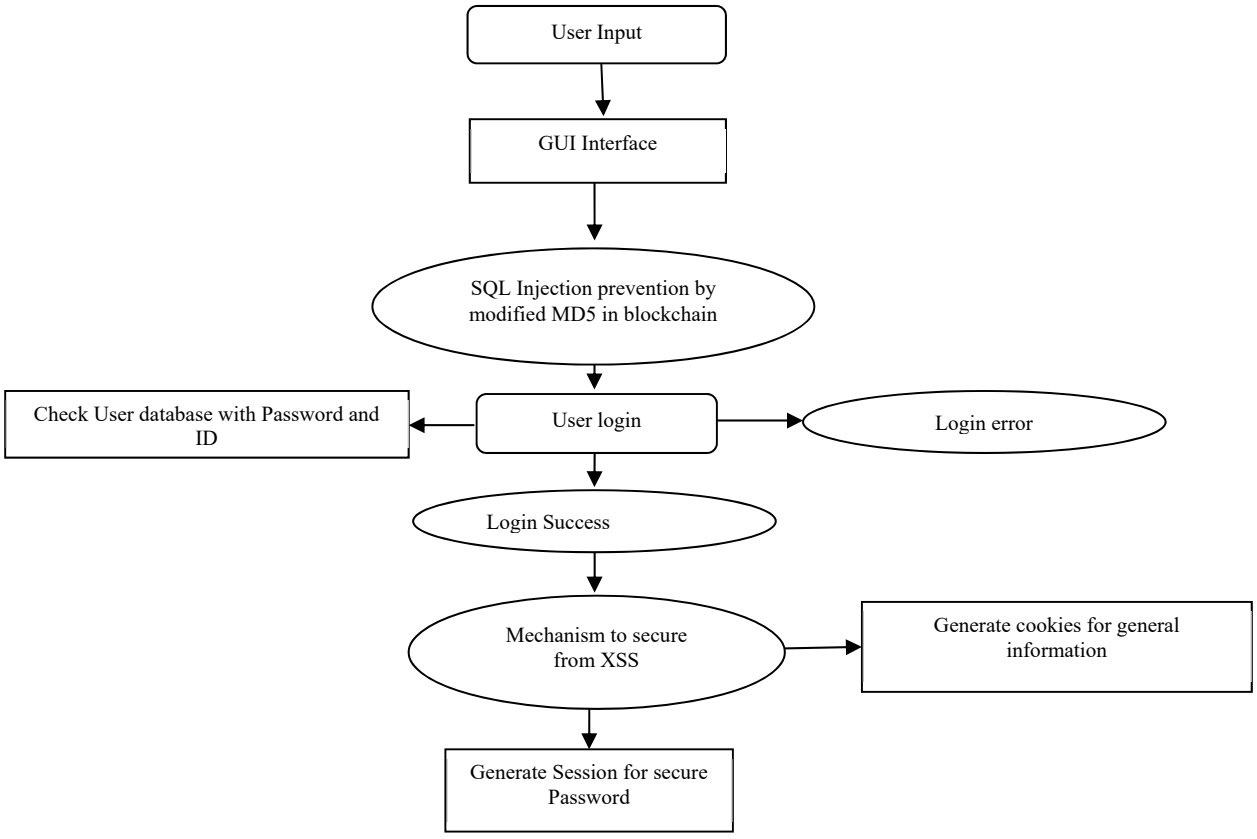

Fig.5.proposed framework

The construction of a realistic demonstration of the application of techniques is the major focus of this project. Defensive strategies and the formulation of a plan are creating a defense against them. The environment settings are the focus of the initial portion of the solution design. The next part is devoted to online testing applications of selected test subjects; this testing is assessed in the following piece of work, and a recommendation for safety measures is developed on the basis of this evaluation

6. Result & Discussion

	Total time by SHA 1	Total time by SHA 256	Total time by MD5	Total time by Modified MD5
Cpu clock 1	14.0907	115.2328	107.5259	54.4152
Cpu clock 2	13.5825	108.9021	109.2949	42.63945
Cpu clock 3	13.02653	110.917	109.1855	34.78608
Cpu clock 4	14.15034	112.4452	110.8115	32.86826
Cpu clock 5	15.24542	109.7804	109.3233	32.3393
Cpu clock 6	12.99839	110.3954	114.2044	33.93039
Cpu clock 7	14.45482	114.0698	107.8273	33.02711
Cpu clock 8	14.44014	102.479	107.5697	31.40688
Cpu clock 9	13.88749	112.3377	110.1986	31.3595
Cpu clock 10	13.30748	110.7437	105.5817	29.8179
Average	13.89147	110.6355	108.6881	35.5693

Table 2. CPU Clock Cycle V/S Time taken in the case of SHA1,SHA256, MD5 and modified MD5

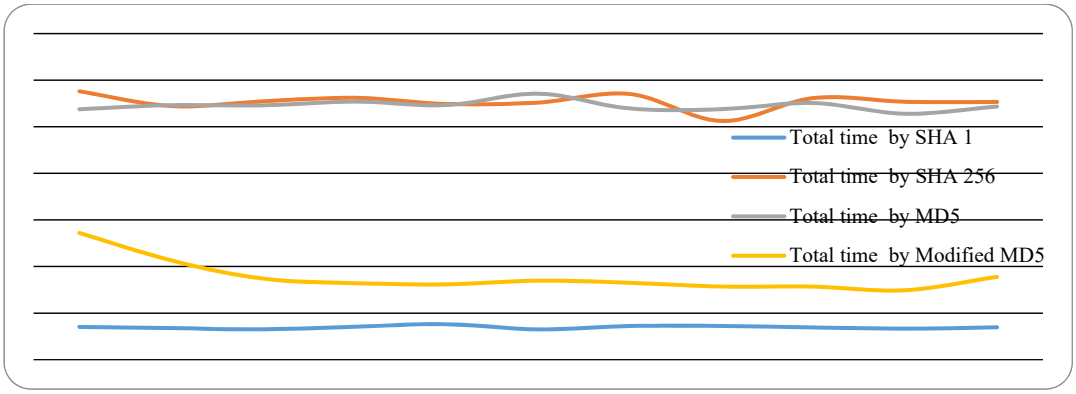

Fig. 6. Graphical representation clock cycle vs total time

6.1. *Simulation result for Traditional model*

	Total time	Clock Cycle	Time period of one cycle
MD5	216	107.5	1
SHA1	212	12.3125	15
SHA256	219	109	1
MODIFIED MD5	561	34.125	15

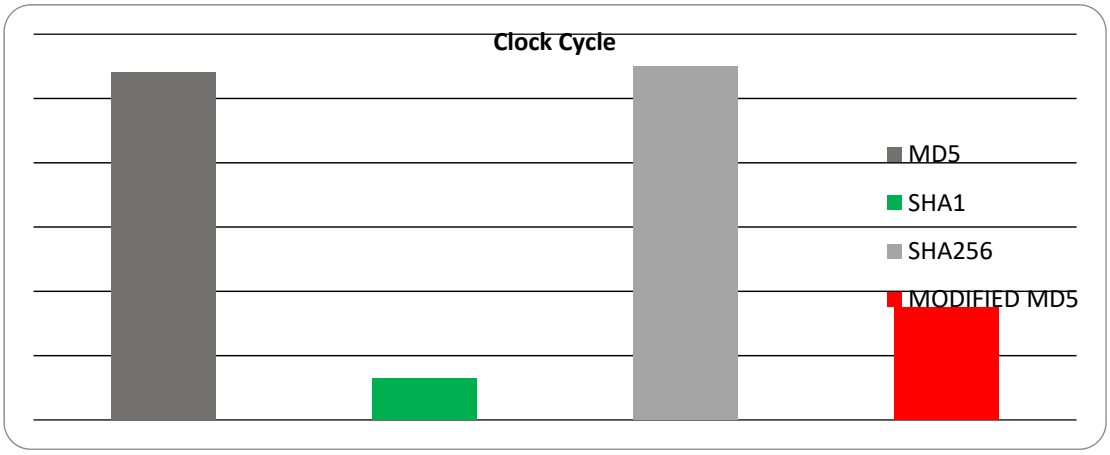

Table 3. Comparative analysis of simulation of average clock cycle

6.2 *Storage space*

Modified MD5's block size is less than SHA256's block size. In the following table, the two are compared side-by-side.

	SHA1 (in bits)	SHA256 (in bits)	MD5 (in bits)	MODIFIED MD5 (in bits)
Block1	75	99	67	95
Block2	77	100	69	96
Block3	77	101	68	96
Block4	78	102	70	98
Block5	79	103	71	98

Table 4. Storage space comparison

Graphically, this table appears as follows.

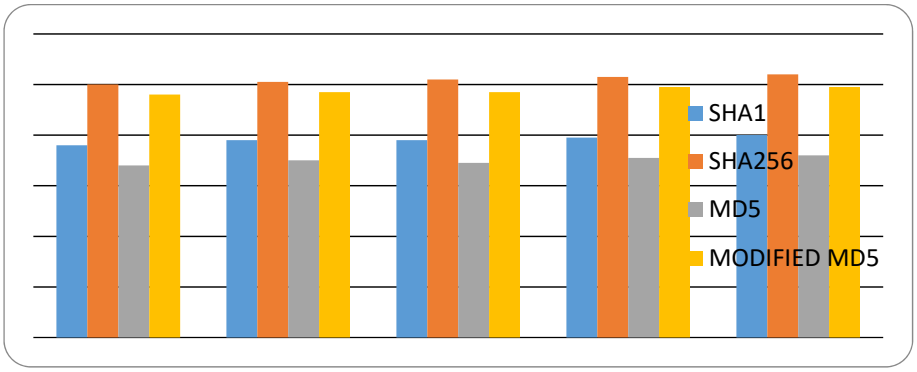

Fig. 7. Comparison chart of storage space occupied by SHA1, MD5, SHA256 and modified MD5

6.3 *Probability of Collision*

In the following table, collision rates for Modified MD5, SHA256 and MD5 are compared.

Technique	Probability Collision
SHA 1	5.59628687833139x10-63
SHA 256	2.49x10-100
MD5	1.58x10-50
MODIFIED MD5	1.58x10-78

Table 5. Probability Collision

Graphically, this table appears as follows.

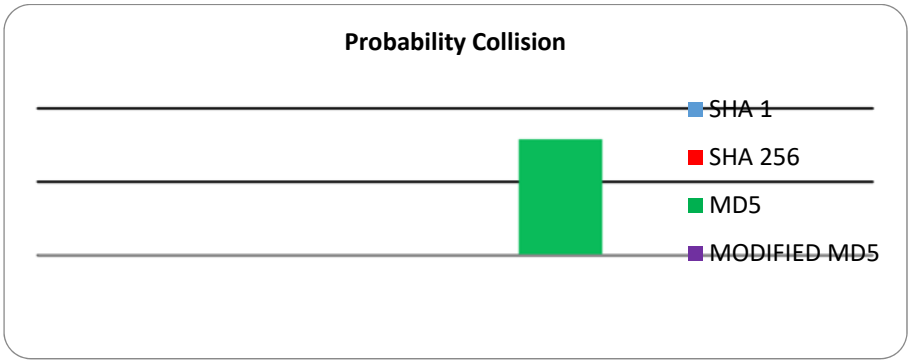

Fig. 8. Comparison chart of probability collision

7. Conclusion

The modified MD5 has been used in the research endeavour. For MD5, SHA1, and SHA 256, a CPU clock cycle was employed. A CPU clock cycle simulation with MD5 modifications was also performed. For security considerations, methods like MD5, SHA1, and SHA256 are used. Despite the fact that MD5 is more secure, it slows down the computer system. When it comes to hashing algorithms, SHA1 and SHA256 are faster, but they are also less safe. As a consequence, the system design proposed here is one that may provide enhanced security while also improving performance. Because the new hashing algorithm is more resistant to collisions, it consumes less storage space and time than standard MD5. Collision probabilities are lower in the proposed MD5 than in the traditional MD5. The recommended MD5 is less than SHA256 in terms of storage capacity. Sql Injection and cross-script attacks may now be prevented thanks to a modified MD5 algorithm.

8. Scope of Research

This research looks into the limitations of previously discovered methods and modules in order to protect the web-based app's database. This study is beneficial since it includes a multiplicative inverted module that prevents SQL injection and Cross script. It also includes modified MD5, which is used to model the results and outcomes of the proposed study. This article compares the planned work to existing safety procedures, which will aid in determining the study's efficacy and application. In future research work, Deep learning Multi head attention LSTM or transformers used for XSS detection which can integrate with sql injection prevention methods.

References

[1] Joshi Padma, Dr.N.Ravishankar, Dr. M.B. Raju, N.CH.Ravi “Smart Algorithms to Secure Web Based Applications from Sql Injection Attacks” Journal of Xidian University,Feb,2022,https://doi.org/10.37896/jxu16.2/026
[2] Joshi Padma, Dr.N.Ravishankar, Dr. M.B. Raju, N.CH.Ravi “Surgical Striking Sql injection attacks using LSTM”Indian journal of Computer Science and Engineering,Feb,2022, doi :10.21817/indjcse/2022/v13i1/221301182
[3] Joshi Padma, Dr.N.Ravishankar,Dr. M.B. Raju,N.Ch.SaiVyuha”Secure Software Immune receptors from Sql injection and Cross site scripting attacks in Content delivery Network Web applications” 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO 2021) DOI: 10.1109/ICRITO51393.2021,Sept. 2021
[4] Joshi Padma, Dr.N.Ravishankar,Dr. M.B. Raju, N.CH.Ravi(2018) “Encountering SQL Injection in Web Applications” Proceedings of the Second International IEEE Conference on Computing Methodologies and Communication.
[5] N.CH.Ravi, Joshi Padma etal. "ADVANCED ACCESS CONTROL MECHANISM FOR CLOUD BASED E-WALLET" in Volume 31 of the Lecture Notes on Data Engineering and Communications Technologies series of Springer of Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI - 2018)
[6] N.CH.Ravi,Joshi Padma,etal., “Inspecting Access Controls in Cloud Based Web Application" Proceedings of the Second International IEEE Conference on Computing Methodologies and Communication.2018
[7] Joshi Padma, Dr.N.Ravishankar,Dr. M.B. Raju, N.CH.Ravi(2017) “Contemplating Security of Http From Sql Injection and Cross Script” 2017 IEEE International Conference on Computational Intelligence and Computing Research.
[8] Joshi Padma, Dr.N.Ravishankar, Dr. M.B. Raju“Defensive Walls for Detecting and Preventing SQL Injection and XSS attacks in Dynamic Content Delivery Network Web Applications” Design Engineering (Toronto) ,vol.2021,issue 7,2021,pp10019-10039
[9] Parveen Sadotra(2017) “SQL Injection Impact on Web Server & Their Risk Mitigation Policy Implementation Techniques: An Ultimate solution to Prevent Computer Network from Illegal Intrusion” Volume 8, No. 3, March – April 2017 International Journal of Advanced Research in Computer Science
[10] Nabeel Salih Ali (2016) “Protection Web Applications using Real-Time Technique to Detect Structured Query Language Injection Attacks” International Journal of Computer Applications (0975 – 8887) Volume 149 – No.6, September 2016
[11] Swathy Joseph (2016) “Evaluating Electiveness’ Of Conventional Fixes For standardized query language Injection Vulnerability” IEEE Computer Society, vol: 46, Issue: 3, pp.69 - 77, IEEE
[12] Swathy Joseph (2016) “Evaluating Effectiveness of Conventional Fixes for standardized query language Injection Vulnerability” vol. 5, no. 2, pp. 80–92 August 2016
[13] Abirami J, (2015) “A Top Web Security Vulnerability standardized query language Injection attack – Survey standardized query language Injections in Online Applications”
[14] Bharti Nagpal(2015) “SECSIX: security engine for CSRF, standardized query language injection & XSS”
[15] Rathod Mahesh Pandering (2015) “A Mapping-based Podel for Preventing Cross Site Scripting & standardized query language Injection Attacks on Web Application & its Impact Analysis” 2015
[16] Mukesh Kumar Gupta (2015) “Predicting Cross Site Scripting (XSS) Security Vulnerabilities in Web Applications”, International Joint Conference on Computer Science and Software Engineering (IJCSE), IEEE, pp. 40-52, 2015.
[17] Sonewar, Piyush A., and Nalini A. Mhetre(2015) "A novel approach for detection of SQL injection and cross site scripting attacks." Pervasive Computing (ICPC), 2015 International Conference on. IEEE, 2015.
[18] Wang, Rui, et al.(2015) "Improved N-gram approach for cross-site scripting detection in Online Social Network." Science and Information Conference (SAI), 2015. IEEE, 2015.
[19] Habeeb Orotund (2014) “Mitigating standardized query language Injection Attacks Via Hybrid Threat Modelling” International Symposium on Secure Software Engineering. IEEE, Conference Proceedings, pp. 65–81. 2014
[20] Amirmohammad Sadeghian (2014) “Standardized Query language Injection Vulnerability General Patch Using Header Sanitization” international conference on World Wide Web, pp. 396-407. ACM, 2014.
[21] Rocha, Thiago S., and Eduardo Souto. (2014) “ETSS Detector: a tool to automatically detect Cross-Site Scripting vulnerabilities.” Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on. IEEE, 2014.
[22] Yusof, Imran, and Al-Sakib Khan Pathan.(2014) "Preventing persistent Cross-Site Scripting (XSS) attack by applying pattern filtering approach." Information and Communication Technology for the Muslim World (ICT4M), 2014 the 5th International Conference on. IEEE, 2014.
[23] Sonam Panda(2013) “Protection of Web Application against Sql Injection Attacks” International Journal of Modern Engineering Research Vol.3, Issue.1, Jan-Feb. 2013 pp-166-168

[24] Kindy, D.A., & Path an, & A.K.:(2013) "A Detailed survey on various aspects of SQL injection in web applications: vulnerabilities, innovative attacks & remedies." In: International Journal of Communication Networks & Information Security, vol. 5, no. 2, pp. 80–92 August 2013
[25] Jane, P.Y., Chaudhari, M.S. (2013) "SQLIA: Detection & prevention techniques: a survey." IOSR J. Comput. Eng. 2, 56–60. IOSR J.
[26] Etienne Janot(2012) "Preventing standardized query language Injections in Online Applications: Study, Recommendations & Java Solution Prototype Based on standardized query language" Document Object Model Power, R.: CSI/FBI Computer Crime & Security Survey. Computer Security Issues & Trends, 8, 1, 1--22
[27] Lwin Khin Shar (2012) "Mining standardized query language Injection & Cross Site Scripting ,Vulnerabilities using Hybrid Program Analysis OWASP. "The open web application security project," http://www.owasp.org, accessed January 2012
[28] Pankaj Sharma (2012) "Integrated approach to prevent standardized query language injection attack & reflected cross site scripting attacking" (Oct-Dec 2012) 3(4):343–351
[29] Selvamani, K. and A. Kannan.(2011) "A Novel Approach for Prevention of SQL Injection Attacks Using Cryptography and Access Control Policies" Advances in Power Electronics and Instrumentation Engineering. Springer Berlin Heidelberg, 2011.
[30] David A. Shelly (2010) "Make use of Web Server Test Bed to Analyze Limitations of Web Application Vulnerability Scanners" International Conference on System Sciences, 0:479, 2010
[31] AppSec DC (2010) "Such conference consists of tracks for web application security testing, vulnerabilities"
[32] 26-33. Bisht, P., Madhusudan, P., Venkatakrishnan, & V.N.(2010)"Dynamic candidate evaluations for automatic prevention of SQL injection attacks." In: ACM Transactions on Information & System Security, vol. 13, no. 2, p. 139. ACM (2010)
[33] Van Gundy, Matthew, and Hao Chen.(2009) "Noncespaces: Using Randomization to Enforce Information Flow Tracking and Thwart Cross-Site Scripting Attacks." NDSS. 2009.
[34] Swap nil Kharche(2007) "Preventing standardized query language based Injection Attack with the help of Pattern Matching Algorithm" International Conference on Computer & Information Technology Cit, 2007
[35] Halfond, W.G.J., Orso, A.: (2005) "AMNESIA: analysis & monitoring for neutralizing SQL injection attacks." In: Proceedings of 20th IEEE/ACM International Conference on Automated Software Engineering, pp. 174–183. ACM, New York
[36] StephenW. Boyd (2003) "SQL: Preventing SQL Injection Attacks" 2003
[37] Sampada Gadgil "SQL injection attacks & prevention techniques" International Journal on Recent & Innovation Trends in Computing & Communication ISSN 2321 – 8169 Volume: 1 Issue: 4

Authors Profile

Joshi Padma N is Associate Professor in Computer Science and Engineering Department in Sreyas Institute of Engineering and Technology, Nagole, Hyderabad in Telangana, India. She worked as Head of department for CSE in Sreyas for 2.5 years. She did B.E in CSE, M.Tech CSE and pursuing Ph.D from JNTUH has 19 years teaching experience and Consultant for projects in I.T industry. She has publications in Springer book chapter and IEEE. Her research interest areas are Web and Network Security, Deep learning, Block Chain Technology and Cognitive Security.

Dr. N. Ravishankar is working as Professor in Dept of CSE *in* Geethanjali College Of Engineering and Technology, Keesara, Hyderabad.He is also Controller of Examinations in GCET. He has vast experience in academics as well as administration. He previously worked as HoD in CSE in Sreenidhi Institute of Science and Technology, KMIT and Lakki Reddy Bal Reddy Engineering College. He has publications in IEEE, Springer. His research Interest is Network and Information Security.

Dr.M.B.Raju completed his B. E from Osmania University, M. Tech & Ph.D from JNTUH, Hyderabad. He has vast experience of 28yrs in academics as well as administration. He has authored 20 National and International research papers, 70 Journals, 02 books and also owns a patent.His research interest areas are Network security, Image processing, Machine and Deep learning and Data mining.

N.Ch.Ravi is Associate Professor in Computer Science and Engineering Department in Pallavi Engineering College, Nagole, Hyderabad in Telangana, India. He is also Dean R&D with 20 years experience in which 5 years in I.T Industry and 15 years in teaching. He has publications in Springer book chapter and IEEE. His research interest areas are Network Security, Deep learning, Block Chain Technology and Cognitive Security.

N.Ch. Sai Vyuha is B.Tech student of 3-2 Sem. of Department of CSE, Bhoj Reddy Engineering College for Women, Hyderabad. She published one paper in IEEE .She has certifications in Course era, Code Ninja, Infosys, Google for Python, java etc.,She has strong logical and programming skills in Rest API, Spring boot, Micro services, Devops, Angular JS,Unity C#, Flutter,Maven, Cordava, Less, Saas, Material design.