ORIGINAL RESEARCH PAPER

Volume-8 | Issue-6 | June-2019 | PRINT ISSN No. 2277 - 8179

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

ALOGORITHM FOR DATA SECURITY AND ITS EFFICIENT PROCESSING

Computer Science

Atul Kumar Omar Department of Computer Science & Engineering Rama University, Uttar Pradesh, Kanpur, India

Dr. Hariom Sharan* Department of Computer Science & Engineering Rama University, Uttar Pradesh, Kanpur, India *Corresponding Author

ABSTRACT

Data security has become a necessity in present IT scenario. With the emergence of various technologies and transformation among various entities, data is to be secured from both the ends. Data is an important part of each organization. This is the base on which the functionality of an organization depends. At the time of data transformation, it is to be secured from both the ends so that it should not be misinterpreted by any third party. Here my observation is on end to end communication. Data is to be protected in the following manner.

KEYWORDS

Md5 Hash-Algorithm,SHA-1 Algorithm,Data Authenticity,Token Passing Method.

I.INTRODUCTION

a)When data is transferred it is to be protected at the time it is in transformation phase. Using algorithm it is to be encoded and decoded properly. Doing so it will not be prone to be accessible to other party.

b)The actual data should be received at the receiving points. These days due to some user friendly interfaces and modern browsers, the data can be manipulated too before submission. This manipulation takes place with the existing interfaces. So a level of protection is to be applied which can ensure the figures of data which is scanned before submission and with stored data, it can be matched for its accuracy before submission. If there is a change in the stored data and retrieved data at the end user, a message of error should come. For this a procedure is to be developed, tested and implemented so that it can be done easily as in a criterion, the volume of data can be heavy too, putting check on such a heavy data will not be feasible by taking each term in count. So there is need of mechanism to wrap the entire data in a string and it is to be matched at the other end.

c)To track the availability of data at one screen, it is to be tokenized to validate its authenticity at the receiving end. In many applications, data is validated at some interval. Here the possibility of validating the same can be multiple times at different intervals of time. Let's take case of email verification, an email is shot to the given id for the validation. But there is no surety for the user to open the link. There may be a possibility that user may open it later on or he may also request multiple times to send the link in email. Now each time, the link is to be accompanied with a token so that it can be identified uniquely. There is one more necessity of tokenization in query string. When data is sent using query string (using open string i.e. data is visible at URL) there is a case that he may enter various possible set of data to validate as per his guesses. If it is not tokenized then as per his guesses that data can be validated and the entire system can be befooled.

d) An effort to make data processing efficient, resourceful and instant.

The overall objective is to keep data safe and secured in the overall processing of data. The objective contains to develop an easy interface that can take care of data security at the priority. Implementation of handy functions to overcome hacking issues is also the part of the research.

II.LITRATURE REVIEW

The existing algorithms (message digest 5 and secured hash algorithm) are working effectively for the purpose of cryptography but their encoding and decoding patterns are available on internet. Seeing these patterns anyone can get the code and get to know the exact data. So it was problematic if this algorithm is followed in the whole project. The issues were based on availability of the pattern as they become accessible commonly. The expected solution to this problem and could be making the pattern hidden from accessibility. Since this is used by all so it was planned to provide it globally but was prone to error and malfunctioning.

About MD5 Hash

In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value. As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. An MD5 hash is typically expressed as a 32 digit hexadecimal number.

Md5 is a strengthened version of MD4. Like MD4, the MD5 hash was invented by Professor Ronald Rivest of MIT. Also, MD5 was obviously used as the model for SHA-1, since they share many common features. MD5 and SHA-1 are the two most widely used hash algorithms today, but use of MD5 will certainly decline over time, since it is now considered broken [2,3,4].

The MD5 hash should not be used for cryptographic purposes. To generate the hash of a piece of text, type below:

The Algorithm

The MD5 hash is described in RFC 1321 along with a C implementation. MD5 is similar to the MD4 hash. The padding and initialisation is identical.

MD5 operates on 32-bit words. Let M be the message to be hashed. The message M is padded so that its length (in bits) is equal to 448 modulo 512, that is, the padded message is 64 bits less than a multiple of 512. The padding consists of a single 1 bit, followed by enough zeros to pad the message to the required length. Padding is always used, even if the length of M happens to equal 448 mod 512. As a result, there is at least one bit of padding, and at most 512 bits of padding. Then the length (in bits) of the message (before padding) is appended as a 64-bit block.

The padded message is a multiple of 512 bits and, therefore, it is also a multiple of 32 bits. Let M be the message and N the number of 32-bit words in the (padded) message. Due to the padding, N is a multiple of 16.

III.SOLUTION TO THE PROBLEM:

The encoding and decoding pattern are to be decided by the user himself. The user may use the symbols, alphabet or numbers to make a certain letter encoded. There should be master for this and once these master entries are done, these symbols can be further utilized to encode and decode a message. Separate functions are to be made and should be kept in the service file so that easily they can be retrieved wherever they are required. The given string is exploded into letters, numbers and symbols separately and their respective symbol is brought from the master entry.

Although there is a checkpoint that while creating the master for alphabet, numbers and symbols, the allotted encoding pattern should be unique. Before saving this information into database, first of all it checks whether it has been allotted to other or not. If allotted, it displays the message of allotment and requests another symbol to be allotted. This maintains the uniformity of the system.

The given string is exploded into different literals and with respect to these literals it is encoded into a new string from the symbols defined in the database. Let's have an example to understand it in better manner. Suppose there is a string “welcome” now this string is divided into separate letters like w,e,l,c,o,m,e and in our database user has already defined unique symbols for all these letters. Now these symbols will be fetched as per the given letter and combining all we shall get the final string. Now this string will be sent as the part of communication not the exact string “welcome”. So if anyone hacks the data while it is being transmitted from one node to the other node, its symbolic formation is fetched not the exact word. Accessing that encoded message, it cannot work out as it becomes meaningless for a person. Now at the other node, the data is again decoded with the help of user defined function. Its decode brings it in the original form.

Benefits of proposed system

1. User defined codes are accessible to his project only. They are not accessible out of that project. Master entries are to be made once then the fed codes can be utilized via function to encode and decode the strings. This algorithm of cryptography becomes local to his system.
2. Even in online projects too there is a local conversion terminology. Earlier its conversion method was accessible globally which was prone to misuse of the data and its malfunctioning.
3. User has its own terminology which he understands well. Making a common hash code was out of his understanding level but this is something his own way so he recalls it well how he has defined the symbols. This brings user friendliness and makes the system smoother.
4. All master entries for symbolic code are to be made once. Once all entries are done, further editing to saved codes should not be possible as these codes would have been utilized somewhere.

Token passing method for data validation

Before submission of data, a token should be generated while the form for data feeding is opened and this token should be saved in the session. This random token is formed using randomization of the function. When the form is submitted the saved token is retrieved from one the submitted fields of the form. Now this retrieved value is compared with the value saved in the session. If this value gets matched, the important operation is carried out else it is denied. This makes the data completely secured from the mid interruptions. There are cases where hackers try to get the form by SQL Injections and other methods. Now from this form they pass the queries by hit and trial method. Now it will be processed only if it is a valid token. If not it will be made out of the processing feasibility. Session management plays an important role in it as the session is active on passing some authentication checks. This is an automatic process where user involvement is not much. There are service function which generates the random token and this random token is retrieved from the function in a string variable and kept it in one of the fields of the form, the name of this field is decided as a common name like tokenid or something. Now this field is submitted to the action of the form and there the comparison is made from the value stored in the session. Here a notable point is that one who is trying to access the page in between will not have that generated session value in hand. So it will not be passed to the operational function. This is all being done internally in data processing, there is no external effect. Thus it is summed up that it can resolve the issue of accessing intermittent page in a series of the other pages of a system.

DATAAUTHENTICITY:

There are some values retrieved in the form which assist in the formation of a formula and a calculation is carried out. Lets take a scenario of receipt generation where discount is brought from the master entry of the database. Now a user can go to the inspect element link of the browser and here in the hidden fields this data is available. Now manual changes can take place here and the changed data can be sent for operations. This results in fake results and wrong calculations. One can be benefitted a lot from such approaches. Now here one option is not to describe these calculation oriented values in the form as they have due importance in the calculation. But it becomes necessary to keep such values in the input field in order to carry out javascript operations. Here it is also to be optimized and authenticated by using checks before the final submission of the data. Taking these important values of calculation and concatenating them in the form of a string and then at the other end if these values are parsed and their authenticity is checked, this scenario can be handled easily.

In data authenticity, it is important to avoid the usage of special characters in databases. If we allow the special characters to be along with the original data, it might give corrupt information. If special characters are avoided, the issues of hacking or sql injection becomes less whereas for the required special characters we may allot a different notation like an alphabet or number and wherever it is used the same can be called. This way the required symbol will be used also and there will no harmful impact on the integrity of the data. On the basis of the description given above the following conclusion can be brought for data security.

User defined codes make the encoding and decoding local to the overall system. Its terminology is not shared globally so it is secured than using the algorithms whose hash codes are available on internet. Generating own code maintains uniqueness of the content and meaningless to the outsiders. Along with this, tokenization and prohibiting the usage of special characters make the data more authentic. Putting calculation oriented values in the session and then verifying them make calculation accurate and malfunctioning is least expected.

IV.CONCLUSION

User defined codes are accessible to his project only. They are not accessible out of that project. Master entries are to be made once then the fed codes can be utilized via function to encode and decode the strings. This algorithm of cryptography becomes local to his system.

Even in online projects too there is a local conversion terminology. Earlier its conversion method was accessible globally which was prone to misuse of the data and its malfunctioning.

User has its own terminology which he understands well. Making a common hash code was out of his understanding level but this is something his own way so he recalls it well how he has defined the symbols. This brings user friendliness and makes the system smoother. All master entries for symbolic code are to be made once. Once all entries are done, further editing to saved codes should not be possible as these codes would have been utilized somewhere.

REFERENCES

1. https://www.md5online.org/
2. https://md5hashing.net/hash/sha1
3. https://www.dcode.fr/sha256-hash
4. Base papers for this research section has been taken from IEEE Transactions on dependable and secure computing, data security and protection from IBM.
5. A survey paper on security issue with Big Data on association rule mining has also been referred.
6. R. Rivest, RFC 1321 - the MD5 message-digest algorithm, April 1992, at ftp.rfc-editor.org/in-notes/rfc332l.txt
7. X. Wang and H. Yu, How to break MD5 and other hash functions, athttp://www.infosec.sdu.edu.cn/paper/md5-attack.pdf
8. P. Hawkes, M. Paddon, arid G. G. Rose, Musings on the Wang et al. MD5 collision, at http://eprint.iacr.org/2004/264.pdf
9. M. Daum, Cryptanalysis of hash functions of the MD4-family, Dissertation zur Erlangurig des Grades eines Doktor der Naturwissenschaften der Ruhr-Universit at Bochum am Fachbereich Mathematik, atwww.cits.ruhr-uni-bochum.de/imperia/md/content/magnus/dissmd4.pdf