World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 5.924

ENSURING DATA AND INFORMATION SECURITY WITH AN IMPROVED HASH-BASED MESSAGE AUTHENTICATION CODE (HMAC)

Sandeep Dhawan*

Senior IT Director, Food Authority New York.

Article Received on 30/01/2022 **Article Revised on 19/02/2022** **Article Accepted on 11/03/2022**

***Corresponding Author**
Sandeep Dhawan
Senior IT Director, Food Authority New York.

ABSTRACT

Data and information security are very important in the present digital world. There have been frequent cases of data leakages due to unauthorized access, data corruption, etc. Tampering messages between two or multiple parties is a common cybersecurity threat at present. Hackers often manage to break strong security measures nowadays. That's why this paper has proposed an improved Hash-based Message Authentication Code (HMAC) to prevent leakage of messages between two or multiple parties. The HMAC technique has been developed based on a hash function and secret key. The improved mapping algorithm used in the hash function and the secret key can make the HMAC technique very strong to prevent any type of data and information leakage. Relevant applications, future improvements, and limitations of HMAC have also been discussed in this paper.

KEYWORDS: HMAC; Hash function, Secret key; Data security; Information security.

1. INTRODUCTION

Information and data are key assets in the digital world. It's a primary objective of every person and organization to safeguard data. At present, social media, eCommerce solutions, financial solutions, etc., are playing a larger role in the global economy. There are possibilities of data leakage from communications, databases, transactions, etc. Now, it's quite a challenge to keep data and information safe. In recent years, there have been some major incidents of data hacking. Various types of cyberattacks like DNS spoofing[1], DoS

attacks, DDoS attacks[2], password cracking[3], etc., are responsible for severe data and information leakage.

In data transmission or data sharing, the verification of data integrity and data authenticity is important.[4] It's essential to ensure that the data was actually transferred by someone who is claiming to have transferred the data. Also, it's essential to ensure that the transfer of data takes place just like how it was set. Failing to ensure these things results in exposure, manipulation, and modification of data. There have been several techniques to strengthen data privacy. One of the effective techniques is the Hash-based Message Authentication Code (HMAC). Safe file transfer protocols, such as SFTP, FTPS, HTTPS, etc., use the HMAC technology for information and data security. This research proposes an effective HMAC technique to ensure data privacy and security.

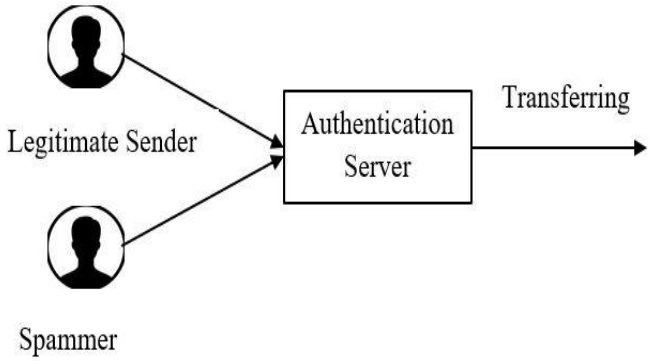

Figure 1: Threat to data security.

The next sections will discuss the significance and implementation of HMAC. Section 2 will discuss the related works of HMAC. Section 3 will discuss the implementation of HMAC to ensure data security. Section 4 will discuss the effectiveness, limitations, and possible future works of HMAC. Finally, section 5 will summarize the research work.

2. Related Works

Data security is a challenging task. So far, there have been some successful implementations of HMAC to ensure data authenticity and integrity. Still, there are huge potentials to improve data and information security.

HMAC has been effective in constructing a secure and reliable architecture for cloud servers. Here, users can safely transmit and store their data. The server and the client have secure communication during data transfer in this platform. File sharing and transferring between

persons is also safe on this platform. The architecture combines HMAC, three DNA cryptography, and a third-party auditor.[5] After investigating some novel and traditional cryptographic methods, the developed architecture was proved to be an effective data security system.

In optical burst switched (OBS) networks, it has been possible to overcome the challenges due to security vulnerabilities by implementing a robust algorithm of the HMAC authentication technology.[6] HMAC can handle security threats in OBS networks. The framework of HMAC authentication's robust algorithm has performed better than other conventional algorithms while handling security threats in OBS networks.

HMAC-SHA-1 is an iterated pseudorandom function that can be used for CPU-intensive operations. PBKDF2, a popular key derivation function based on passwords, can slow attackers down using HMAC-SHA-1. PBKDF2 can speed up by the exploitation of HMAC-SHA-1 optimization.[7]

While holding data in Cloud, it's still a big challenge to maintain the confidentiality of the data. Some of the Cloud's basic security threats are data confidentiality, authenticity, and integrity. Here, an improvement of data protection is possible by using the AES algorithm and RSA algorithm to encrypt data.[8] Hybridizing these two algorithms ensures better data protection before storing the data in Cloud. Here, HMAC computation takes place with SHA 512. The developed strategy based on these algorithms worked really well to ensure data protection.

The SHA-3-HMAC SoC (System on a chip) module is a special design that can prevent common side-channel and error injection attacks. The design also supports many 256/384/512 digest values according to a configuration having an automatic padding function in SHA-3. During implementation, the maximum clock frequency can become 300MHz. With a time redundancy based circuit design and a clock randomizing module, a great improvement of the SoC module's reliability and security has been possible.[9]

Wireless communication based applications like web browsing, SMS, MMS, cellular telephony, video conferencing, etc., have some inherent problems. These problems include transmission error and security of transmitted data. By using HMAC, it's possible to ensure data integrity protection. There were some simulation-based tests on this proposed strategy.

The results of HMAC were compared to those of convolutional coding, maintaining similar coding rates. There was a coding gain for the HMAC based method for ensuring data integrity.[10]

Date Time Keyed HMAC or DTK-HMAC does the message hashing utilizing user-specific and communication details, such as time, date, and key info. Therefore, the DTK-HMAC scheme doesn't rely on only the message. Its focus remains on data integrity and assurance of data origin integrity. The specific details of communication are time info and data. However, the details of user-specific are the key shared secretly between the communicators.[11]

3. HMAC AND ITS EFFECTIVE IMPLEMENTATION FOR DATA SECURITY

3.1 HMAC and Its Suitability for Data Transfer

HMAC is a special type of message authentication code got by any cryptographic hash function (such as MD5, SHA256, SHA1, etc.) over the data (for authentication) and a secret shared key.[5] HMAC has similarities with digital signatures. Both technologies enforce authenticity and integrity, and both utilize cryptography keys. Also, both technologies employ hash functions. HMAC uses symmetric keys for data integration and authentication.

HMAC has a great ability to enable message authentication and data integrity. Hash functions can receive a message with an arbitrary length and turn it into a digest of fixed length. So, even if the messages are relatively long, their corresponding digests can stay short, allowing to maximize bandwidth.

The properties of HMAC highly rely on its hash function, so HMAC is identified mainly depending on its hash function. Some of the HMAC algorithms are HMAC-SHA1, HMAC-MD5, HMAC-SHA256, etc. MD5 has some collision-related vulnerabilities. SHA-1 is cryptographically more robust than MD5. However, SHA-2 and its various forms like SHA-512, SHA-256, SHA-224, etc., are cryptographically more robust than SHA-1. These facts have to be in consideration before implementing HMAC.

3.2 Implementation

The working of HMAC is linked with a hashed function that can be used for data integrity checking on data or file transfer.[12] Hashing transforms data into a constant-sized hash. Hashing is deterministic. For the same input, the algorithm generates the same hash. Its killer feature is that no way is possible to get the original data or information from the hash.

Suppose, from a remote server, a client application is downloading a file. Usually, the general assumption is that the server and the client already have an agreement on a common hashed function, such as SHA2.

The server uses the SHA2 hash function to obtain the file's hash before sending out the file. Then, it transmits the hash (such as a digest of the message) with the file. Upon receiving both items (the hash and the downloaded file), the client gets the downloaded file's SHA2 hash. Then, he checks this hash and also the downloaded hash. The two hashes have to match to prove that there wasn't any tampering of the file. If they don't match, then it has to be assumed that there must be some problems. Figure 2 shows the overall scenario of how HMAC implementation takes place.

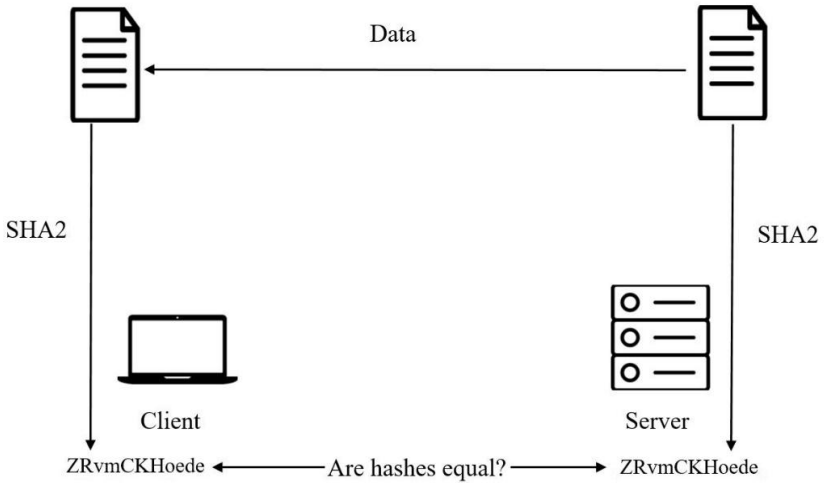

Figure 2: The working of HMAC during a data transfer.

In case an attacker succeeds in intercepting the file, then alters its contents, and transmits it to the recipient, the malicious activity won't stay unnoticed. It's because when the client checks the tampered file with the hash algorithm previously agreed, there won't be any match between the downloaded hash and the resulting hash. It will let the receiving side find out that there was file tampering during data transmission.

These are the main roles of hash functions. But it's not enough. A hash function establishes data integrity, but data authenticity is yet to be ensured. The client needs to know that the message he received arrived from the correct source. For this, HMAC is required along with the hash functions. As two parties exchange files or messages through file transfer protocols like FTPS, HTTPS, SFTP, etc., these messages will be transferred with HMACs instead of only plain hashes. However, an HMAC employs a secret shared key and a hash function.

A secret shared key provides the message-exchanging parties a solution to ensure the message's authenticity. That is, it gives the exchanging parties a solution of verifying whether the message and HMAC they receive actually arrived from the correct party. The secret shared key strengthens this ability because it's created during key exchange, which is a preliminary process requiring the two parties' participation. Only the two parties participating in the exchange of key would know about the secret shared key. In turn, only they would have the ability to reach the same result by computing the corresponding MAC of the message by using the secret shared key.

4. DISCUSSION

Data security is a huge concern during data transmission. Data can be manipulated by spammers and hackers anytime. That's why the importance of data integrity and authenticity is too high. Therefore, the implementation of HMAC for data transmission is very helpful to ensure data integrity and authenticity. HMAC greatly reduces the possibility of data manipulation as the hash function helps the message or file receiver to detect if the data was tampered with. HMAC ensures data security mainly by using the hash function and the secret shared key. The hash function ensures data integrity, and the secret key ensures data authenticity. The combined security feature of the hash function and shared secret key makes HMAC a robust data security system.

However, there are still scopes for the improvement of HMAC. The performance of each of the hash algorithms, such as HMAC-MD5, HMAC-SHA2, HMAC-SHA256, etc., can be improved. These hash algorithms are strong, but they can still be broken. Also, if there's a compromising in the shared key, the attackers will find it easier to create unauthorized messages. So, future works can overcome these limitations of HMAC.

5. CONCLUSION

The use of HMAC has brought great improvement in secure data transmission. It efficiently provides a data authenticity and integrity check. Improved hash algorithms and the shared secret key concept have enhanced the data protection capability of HMAC. Even if a cyber-attacker tampers a message, the receiver can detect it by checking the hashes. So, unless the attacker is exceptionally skilled in breaking hash algorithms and the secret key is somehow compromised, HMAC is strong enough to ensure information and data security. However, further research works on HMAC can keep improving the protection level of HMAC in data transmission.

6. REFERENCES

1. Maksutov, I. A. Cherepanov and M. S. Alekseev, "Detection and prevention of DNS spoofing attacks," in 2017 Siberian Symposium on Data Science and Engineering (SSDSE), Novosibirsk, Russia, 2017.
2. M. A. Saleh and A. A. Manaf, "Optimal specifications for a protective framework against HTTP-based DoS and DDoS attacks," in *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, Kuala Lumpur, Malaysia, 2014.
3. S. Houshmand, S. Aggarwal and R. Flood, "Next Gen PCFG Password Cracking," *IEEE Transactions on Information Forensics and Security*, 2015; 10(8): 1776-1791.
4. H. Michail, A. Kakarountas, A. Milidonis and C. Goutis, "Efficient implementation of the keyed-hash message authentication code (HMAC) using the SHA-1 hash function," in *Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems, 2004. ICECS 2004.*, Tel Aviv, Israel, 2004.
5. Kumar, "Framework for Data Security Using DNA Cryptography and HMAC Technique in Cloud Computing," in *Proceedings of the Second International Conference on Electronics and Sustainable Communication Systems (ICESC-2021)*, 2021.
6. G. V. J. Balamurugan A.M, "Secured Header Authentication Design using Time Competent HMAC for Optical Burst Switched Networks," in *Proceedings of the 6th International Conference on Communication and Electronics Systems (ICCES-2021)*, 2021.
7. Visconti and F. Gorla, "Exploiting an HMAC-SHA-1 Optimization to Speed up PBKDF2," *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, 2020; 17: 4.
8. Kumar, "A Novel Privacy Preserving HMAC Algorithm Based on Homomorphic Encryption and Auditing for Cloud," in *Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2020.
9. Y. Z. N. J. T. N. W. L. B. P. Tao Zhou, "Reliable SoC Design and Implementation of SHA-3-HMAC Algorithm with Attack Protection," in *2020 IEEE International Conference on Smart Cloud (SmartCloud)*, 2020.
10. O. U. R. C. R. Nataša ŽIVIû, "Using HMAC for error correction over a Wireless Channel," in *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, 2009.

11. P. V. Mahima Mary Mathews, "Date Time Keyed - HMAC," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, 2016.
12. M. W. E.-K. F. G. M. A.-E.-B. Esam Khan, "Design and Performance Analysis of a Unified, Reconfigurable HMAC-Hash Unit," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS*, 2007; 54: 12.
13. "A Novel Privacy Preserving HMAC Algorithm Based on Homomorphic Encryption and Auditing for Cloud".