DATA SECURITY OF ENCRYPTED DATA USING HASH-BASED MESSAGE AUTHENTICATION CODE (HMAC) AND T9-BASED DATA CONVERSION

FRANCIS G. BALAZON

Batangas State University, Lipa City, Batangas, Philippines

Abstract: The researcher has recently been working on the authenticity of information particularly with the use of cryptographic hash functions such as MD5 and Secure Hash Algorithm (SHA). After critically analyzing the Hash-based Message Authentication Code (HMAC) algorithm, the researcher finds out that it can still be improved and strengthened by adding the T9 conversion, which also adds more complexity and security to the system. T9 keypad-based conversion is used in converting the data and this pre-converted data would then pass through the process of encryption using HMAC encryption algorithm. This work focuses on following the cryptosystem design, and includes the study of the activity process and procedures, as well as the method of the proposed system. Cryptography key, plaintext, message security, cipher, message authentication code, and hash function are the major activities that go with the cryptosystems. The study uses the incremental model approach to generate and illustrate the system functions and architecture of this project. The resulting system is a new algorithm developed using hypertext preprocessor (PHP) in a form of Library PHP file by the researcher for securing the integrity and authenticity of the data. It provides high security of text data with the use of authentication process integrated with the new researcher-developed T9 encryption decryption process.

Keywords: *encryption, decryption, cryptography, cipher, T9*

1. INTRODUCTION

Security of data to maintain its confidentiality, proper access control, integrity and availability have been among the major issues in data communication. When a sensitive data is placed on a communication pathways, it must then be foremost in the sender's mind that the information should not get intercepted and read or decrypted by others. Codes, hence, form an important part of our history, starting from the paintings of Da Vinci and Michelangelo to the ancient Roman stenographic practices, the necessity of data hiding is obvious (Pediapress, 2011).

Today in the electronic age, the need to protect communications from prying eyes is greater than ever before. Cryptography, the science of encryption, plays a central role in mobile phone communication, e-commerce, Pay-TV, sending private e-mails, transmitting financial information, etc. It touches on many aspects of daily life. Today's technology can be traced back to the earliest ciphers, and has grown as a result of evolution. Code breakers set to work on these and eventually find flaws, forcing cryptographers to invent better ciphers, and the cycle goes on. The significance of key is an enduring principle of cryptography (Mahapatra, 2007).

Providing confidentiality is not the only objective of cryptography. Cryptography is also used to provide solutions for other challenges like: data integrity, where the receiver of the message can check whether the message was modified during transmission, either accidentally or deliberately. No one should be able to substitute false message for the original message or parts of it (Abo-Elsoud, 2013); and authentication, where the receiver of a message should be able to verify its origin. No one should be able to send a message to the receiver and pretend to be the sender (data origin authentication). When initiating a communication, sender and receiver should be able to establish entity authentication (they should be able to identify each other), and non-repudiation (sender should not be able to later deny having sent the message).

The need for techniques providing data integrity and authentication has arisen due to the rapidly increasing significance of electronic communication (Delfs, 2015). System analysis is a detailed study of the different tasks or operations done by a process and the relationships within and outside of the system (Jawahar, 2014). The key issue involves the lack of the aspects in the existing process, and the solutions to solve this. Therefore, the analysis starts with the existing system.

The researcher analyzes the existing process of data encryption. Figure 1 shows the process involved in the existing data encryption process. The process starts when the sender sends a ’Plaintext’. This plaintext may contain alpha or numeric characters, and serves as the message of the sender. The plaintext goes to the process of encryption. In this stage the message/plaintext is converted to a secret code through the aid of the Hash-based Message Authentication Code (HMAC) algorithm. This process of algorithm uses a secret key together with the hash message. The sender’s HMAC and the plaintext will be sent to the receiver. Then the plaintext and receiver’s key will calculate the receiver’s HMAC. After the calculations, it will authenticate and compare the sender’s HMAC to the calculated receiver’s HMAC (Bruen, 2005).

But after critically analyzing the HMAC algorithm, the researcher finds out that it can still be improved and strengthened by adding the T9 conversion, which also adds more complexity and security to the system. T9 keypad-based conversion is used in converting the data and this pre-converted data would then pass through the process of encryption using HMAC encryption algorithm.

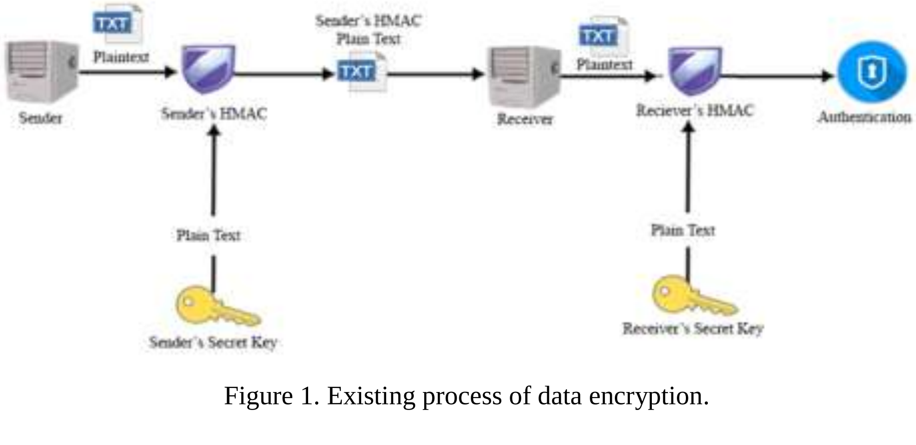

Figure 1. Existing process of data encryption.

This paper aims to improve the data security of encrypted data using HMAC and T9-based data conversion. This also seeks to increase the way the existing algorithms hold and encrypt the data, and implements new way of security for invulnerability of the cyphertexts. This process will cover the message security, message integrity, user authentication and key management of messages and data transfers. The process involved in encrypting data and information in this system is only dedicated in hypertext preprocessor spell-out language and cannot be implemented in other programming languages.

Adding T9 keypad-based conversion strengthens the security and adds protection that does not rely only on the default computer security. The built-in protections may be adequate in cases like hackers or intruders who have not learned how to get around a simple default mechanism or when no one is interested in stealing data from a particular computer. But many hackers do have the skills and resources to break various security systems.

One of the most important tools for protecting data from an unauthorized access is data encryption and in this system, it did not only rely on the process of encryption. The proponent proposed a new way of securing data. Even if hackers obtain the contents of the file, the accessed information is useless.

2. METHODOLOGY

Following the cryptosystem design is central in this work. A cryptosystem is a suite of algorithms which are needed for security service, particularly in achieving confidentiality or encryption (Menezes, 2014).

Three (3) algorithms comprise the cryptosystem. These are the key generation, the encryption, and the decryption algorithms. Cipher, also called cypher, refers to a pair of algorithms called encryption and decryption. Therefore, when the key algorithm is important, that cryptosystem is most used. Both “cipher” and “cryptosystem” are used for symmetric key techniques, thus cryptosystem is commonly used in public key techniques (Dietrich, 2007).

Incremental approach is used in this study in order to generate and illustrate the architecture and system functions. Incremental model is a process of software development where requirements are broken down into multiple stand alone modules in a software development cycle. Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented. During requirement analysis, the researcher collects the requirements and specifications of the software. Some high-end functions are designed during design stage while coding of the software is done during code stage. Once the system is deployed, it goes through the testing phase (Pressman, 2010).

Message security

Figure 2 shows the process of security services related to message or entity. This process of security provides four services: privacy (confidentiality), message authentication, message integrity, and non-repudiation (Chandra, 2015).

Message privacy

Sender and receiver of the encrypted data retain its confidentiality. The transmitted message must make sense only to the intended receiver. To all others, the message must be unintelligible. Encrypted information is virtually hidden from everyone who doesn't know how to decrypt it or doesn't have the appropriate key (Kumar, 2004). This makes it possible to share secret information over unsecure communication channels, such as the local area network (LAN), thus providing confidentiality even though the network itself is quite open. Encryption ensures confidentiality of stored data even if the computer itself gets compromised or stolen. To achieve privacy, the message must be encrypted. That is, the message must be rendered unintelligible to unauthorized parties (Forouzan, 2007).

Message authentication

Message authentication means that the receiver needs to be sure of the sender’s identity and that an impostor has not sent the message (Hamid, 2008). In message authentication the identity of the user is verified once for each message or data sent across the network.

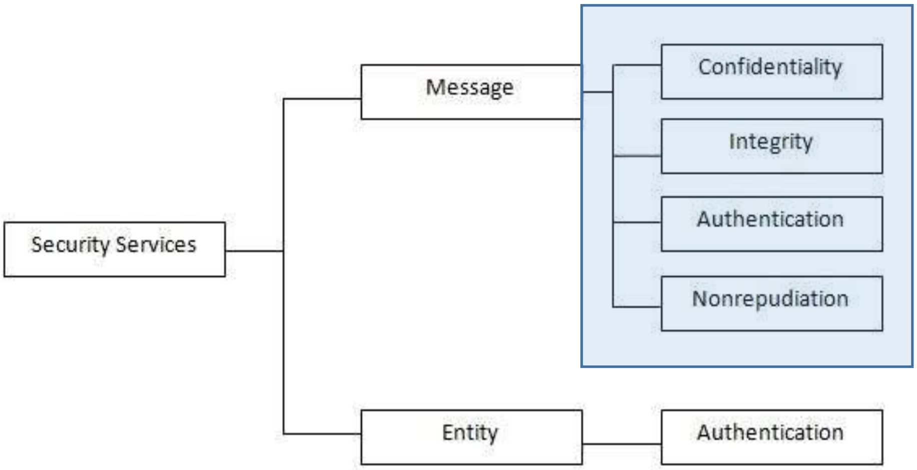

Figure 2. Security services related to message or entity.

Integrity

Integrity means that the data arrived at the receiver are exactly what are in the sender’s original plaintext (El-Bendary, 2015). There must be no changes during transmission, either accidental or malicious.

Non-repudiation of data origin

Non-repudiation means that the receiver must be able to prove that the received data are from a specific sender (Dixit, 2007). The sender must not be able to deny sending a message that he/she, in fact, did send. The burden of this proof falls on the receiver. For example, when a sender sends a message containing data, the receiver must be able to prove that the sender actually requested this transaction.

3. RESULTS AND DISCUSSION

The project entitled “T9-Based Conversion for Data Encryption/Decryption” is a new algorithm developed using hypertext preprocessor (PHP) in a form of Library PHP file by the researcher for securing the integrity and authenticity of data. It provides high security of text data with the use of authentication process integrated with the new researcher-developed T9 encryption decryption process. Authentication and encryption processes add an essential job in securing data. Security is needed in forwarding the protected data over the network; it is also compulsory in storing and retrieving information from the database. Mostly, all the information in the web is confidential and needs to be secured from intruders and hackers, and a wide range of applications and sites over the net demand high security.

Hash-based Message Authentication Code (HMAC)

A public and private key is provided to the server and the client in HMAC (Cuppens, 2014). Although the public key is identified, the private key remains known only to the specific server and client. The client then requests for a unique HMAC which hashes that data. After the server compares the two HMACs—one from the server and the other one from the client—and if they match, the process is called a handshake. The key and the message are hashed in separate steps which make HMAC more secure compared to Message Authentication Code (MAC).

There are two benefits of HMAC treatment of the hash function as a black box. The first is using it as a module in implementing HMAC with the code pre-packaged and ready to be used without any modification. Second is the ability of the HMAC hash function to be changed into a new module. This ensures the safety of the hash function once compromised.

In addition, this feature of the HMAC and its ability to provide a reasonable cryptographic strength is indeed its main advantage over other proposed hash-based schemes. Any MAC has the security feature of embedded hash functions. HMAC embeds both the strength of the embedded hash function and the strength of the HMAC. The strength of the MAC security feature is tested within a given amount of time by the forger, and the message is paired with the same key.

The proposed system

This work includes the study of the activity process, procedures, as well as the method, of the proposed system. Cryptography key, plaintext, message security, cipher, message authentication code, and hash function are the major activities that go with the cryptosystems.

Figure 3 shows the T9 encryption process algorithm wherein $t9encrypt is the standard variable for converting plaintext through T9-based conversion, H is the hash function, II is the concatenate, Key is Keys 1-9 representing keypad numbers, I_number represents index number in keypad, $checksum holds the value of the generated HMAC, $secretKey holds the value of the key to be used in HMAC. The first priority are the space and dash, second are the numbers, third are the special characters and extended KEY ASCII KEY1, and the fourth priority are the alphabet, acute, accent and stressed letters KEY2-9. HMAC integrated by T9-based conversion is used for authentication of code.

Figure 4 shows the T9 decryption process algorithm wherein $t9encrypt is the standard variable for converting plaintext through T9-Based Conversion, H is the hash function, II is the concatenate, Key is Keys 1-9 representing Keypad numbers and I_number represents Index number in Keypad. The first priority are the Alphabet, Acute, Accent and Stressed Letters KEY2-9, second are the Special Characters and Extended KEY ASCII KEY1, third are the numbers, and the fourth priority are the space and dash.

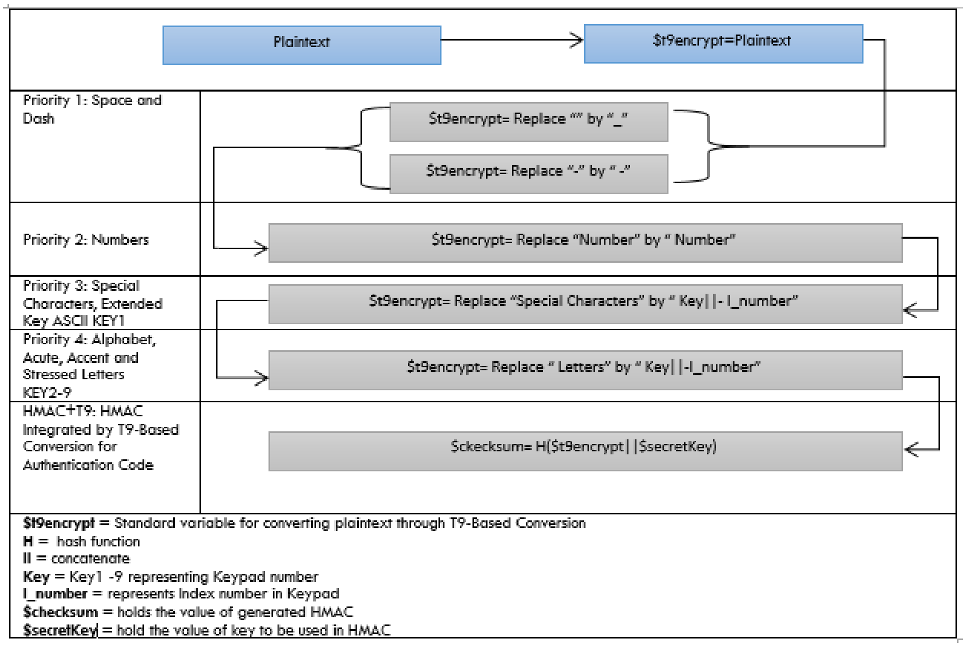
Plaintext
$t9encrypt=Plaintext
Priority 1: Space and Dash
$t9encrypt= Replace "" by "_"
$t9encrypt= Replace "-" by " -"
Priority 2: Numbers
$t9encrypt= Replace "Number" by " Number"
Priority 3: Special Characters, Extended Key ASCII KEY1
$t9encrypt= Replace "Special Characters" by " Key||- I_number"
Priority 4: Alphabet, Acute, Accent and Stressed Letters KEY2-9
$t9encrypt= Replace " Letters" by " Key||-I_number"
HMAC+T9: HMAC Integrated by T9-Based Conversion for Authentication Code
$ckecksum= H($t9encrypt||$secretKey)
$t9encrypt = Standard variable for converting plaintext through T9-Based Conversion
H = hash function
II = concatenate
Key = Key1 -9 representing Keypad number
I_number = represents Index number in Keypad
$checksum = holds the value of generated HMAC
$secretKey = hold the value of key to be used in HMAC

Figure 3. T9 encryption process algorithm.

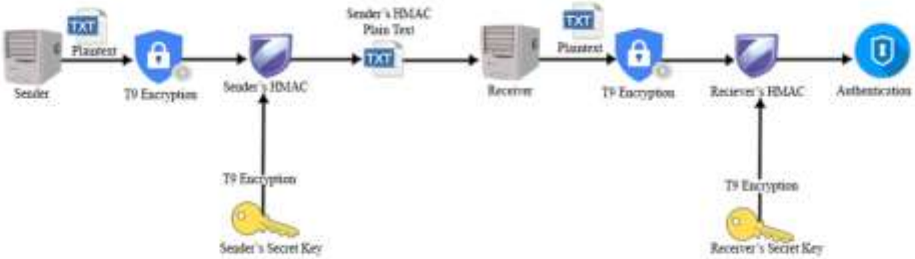

Figure 4. T9 decryption process algorithm.

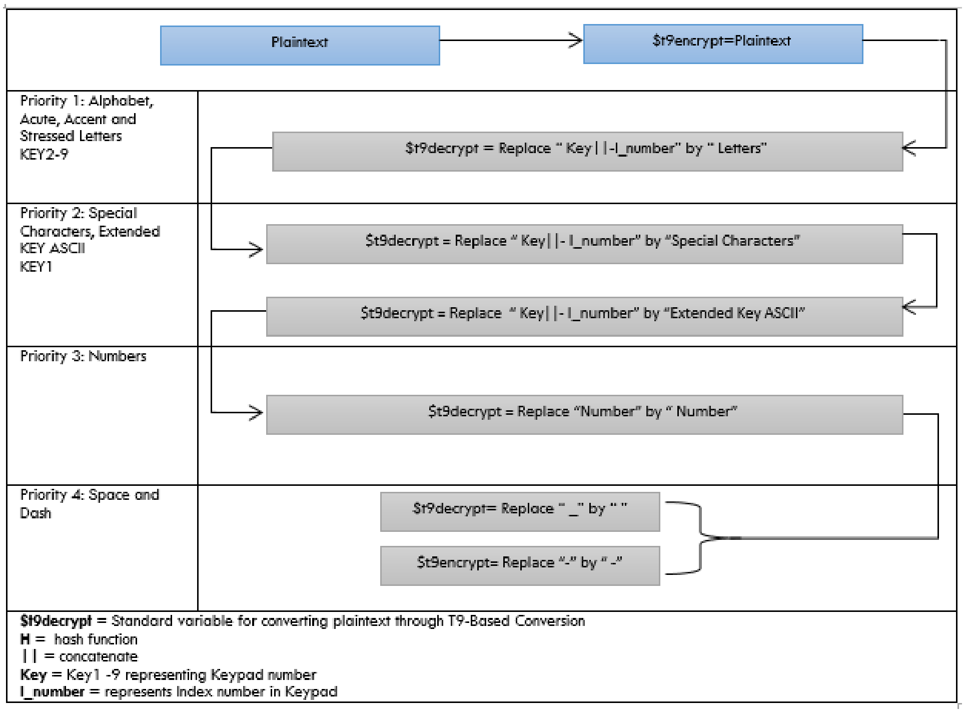
Plaintext
$t9encrypt=Plaintext
Priority 1: Alphabet, Acute, Accent and Stressed Letters KEY2-9
$t9decrypt = Replace " Key | |-I_number" by " Letters"
Priority 2: Special Characters, Extended KEY ASCII KEY1
$t9decrypt = Replace " Key| |- I_number" by "Special Characters"
$t9decrypt = Replace " Key| |- I_number" by "Extended Key ASCII"
Priority 3: Numbers
$t9decrypt = Replace "Number" by " Number"
Priority 4: Space and Dash
$t9decrypt= Replace " _" by " "
$t9encrypt= Replace "-" by " -"
$t9decrypt = Standard variable for converting plaintext through T9-Based Conversion
H = hash function
| | = concatenate
Key = Key1 -9 representing Keypad number
I_number = represents Index number in Keypad

Figure 5. Data encryption with T9-based conversion.

Figure 5 shows the encryption process of T9-based conversion. The process starts when the sender sends a 'Plaintext'. This plaintext may contain alpha or numeric characters that serve as the message of the sender. The plaintext goes through the process of T9-based conversion and is converted into a T9 value. The T9 value will now be used in the encryption process. At this stage, the T9 value will be converted into a secret code through the aid of the HMAC. This process of algorithm uses a secret key together with the hash message. The sender's HMAC and the plaintext will be sent to the receiver. The plaintext will now convert to T9. The receiver's key and the T9 will now concatenate for the receiver's HMAC calculation. After the calculations, it will authenticate and compare the sender's HMAC to the calculated receiver's HMAC.

Account verification and data decryption

This shows how an integrated HMAC and T9 is used for verification and data decryption. Figure 6 shows the process of account verification and data decryption. The user inputs a username and password. The data input will be converted into T9 encryption. For username HMAC calculation, the T9 encrypted username will concatenate with the secret key. And for the password HMAC calculation, the T9 encrypted password will concatenate with the secret key. The calculated username HMAC and password HMAC will be compared to the database for the authentication of decryption. After the authentication, the T9 encrypted data will now be decrypted.

Integration between Hash-based Message Authentication Code (HMAC) and T9-based conversion

Compared to other data authentication processes, the integration of T9 process makes the authentication more unbreakable. Following the common process of authenticating data using HMAC, the message or the data is concatenated with the shared public key between the two users, and uses a hash function. In the integration of T9 process with the HMAC, the common process of authenticating data has changed and covered up the real data. The new process has added a T9 process before executing the HMAC function. This new authentication process is more secure and efficient because it uses “256 bit” sha256 hash function, and covers up the real message with T9 process. It would take a year to break or extract the real data in the authentication process, and the integrated system makes this data possibly unbreakable due to the additional static key which is declared. The efficiency of the newly developed algorithm for authentication is from having been built as PHP file library of values of T9-based conversion, namely “T9Encrypt.php” as encryption for T9 process. The library file also includes the corresponding standard for the variable key used by the researchers to be employed in HMAC calculations. The “T9Encrypt.php” consists of the T9-based conversion process for T9 encryption, and automatic HMAC calculations for data authenticity. Along with the building of the “T9Encrypt.php”, a counterpart is also built, namely “T9Decrypt.php”, which will decrypt the T9-Based Conversion Process.

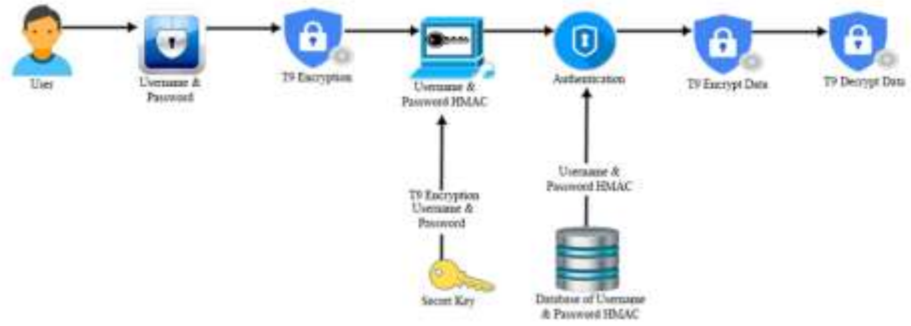

Figure 6. Account verification and data encryption process.

The developed algorithm can be used in different ways. One way is to use the "T9Encrypt.php" itself as the authentication process for checking the integrity and authenticity of data. Another is for the use of the authentication process from "T9Encrypt.php" for decrypting a T9-based conversion, and calling the "T9Decrypt.php" for decryption.

Table 1 presents the result of the unit testing done. This shows whether the library and variables have passed or failed during and after the procedure.

All the library and variable functions are working as intended. The encrypt library function can generate cipher texts as intended and the cipher texts can be decrypted using the decrypt library.

Table 2 shows the problem encountered during the execution of process in the integration testing. The failure of the initial testing of the registration module is due to the use of "varchar" data type to the fields in the database. This type can only handle a limited number of numeric data, characters, spaces or punctuation. Replacing the data type "varchar" to "longtext" is the key in debugging and avoiding errors in the execution of processes.

The failure of the initial testing of the login module is due to the wrong use of the syntax in T9Encrypt.php library. If the registration module can't handle the data that is stored in the database, the encryption of data may have conflict or will result to data deficiency. The solution is to change the data type "varchar" to "longtext".

The failure of the initial testing for viewing information is due to the wrong type of data in the fields of database the encryption and decryption relies on for the process. T9Decrypt.php uses authentication of data by means of the HMAC algorithm combined with the concatenation of the T9encrypt data and the secret key. To fix this failure, the right use of data type in every variable declared in the T9Decrypt.php and T9Ecrypt.php must apply.

Table 1. Summary of results after conducting unit testing.

Library	Variables	Passed	Failed
T9Encrypt (PHP file library)	$t9encrypt (global variable)	✓	
	$t9encrypt=str_replace (string function)	✓	
	$secretKey (global string variable)	✓	
	$checksum= hash_hmac('sha256',$t9encrypt,$secretKey); (global string variable)	✓	
T9Decrypt (PHP file library)	$t9decrypt (global variable)	✓	
	$t9decrypt=str_replace (string function)	✓	
	global $key1 to $key9 (global variable)	✓	

Table 2. Summary of results after conducting integration testing.

Module	Connected	Failure	New	Modified
Registration	T9Encrypt.php	- the encryption of data can only handle a limited number of characters - there is error in saving data in database when the field text data is set to" varchar".	**F**	**P**
Log In	T9Encrypt.php	- if the data that was sent was wrong due to syntax errors in the database, the user would not be able to log in.	**F**	**P**
View Information	T9Decrypt.php	- the information that the user views might be incomprehensible text if data sent was invalid to the database due to the syntax error, and the decryption process would not work as intended.	**F**	**P**

Legend: F = failed; P = passed

4. CONCLUSIONS

To protect people's privacy, encryption and decryption technology is becoming more and more important in the communication area. It is an excellent tool for protecting data and information from unauthorized access, use, disruption and disclosure. The following conclusions can be drawn from this study: (1) Hash-based Message Authentication Code provides web developers with technology that ensures data and information availability, security, integrity and confidentiality when being sent and saved; (2) T9-based conversion features flexibility of values of every character. It gives strength and power to algorithms. It adds another layer of security and it has a self-generated unique signature for the authentication process that can be used to decrypt T9. This algorithm used in the encryption and decryption opens up new possibilities for developing new algorithm. (3) In order to process and decipher the encrypted messages, the cracker or hacker would have to figure the structure of the library which composes the T9 conversion, secret key and hash function used in creating the unique signature in decrypting T9. It is special because it is so different in that it doesn't rely on the basic processes and computation that computers basically do, like binary, hexadecimal and more. It would be especially useful for protecting and securing data wherein the basic processes and computation is revealed. (4) The developed system is working as intended based on the result of the unit and integration testing.

5. RECOMMENDATIONS

The researcher believes that this new encryption algorithm can still be enhanced and made more complex. The following areas can be improved: (1) the library can be enhanced by adding more features suitable for the algorithm; (2) random values should be generated for each character to decryption key to strengthen the data; and (3) this project can be implemented to other programming languages by rebuilding the process concept of T9 encryption so that it can be used not only in web development but also for many other purposes.

6. REFERENCES

Abo-Elsoud, M., Taki, E., Ali E., & Saif, S. M. (2013). Text and biomedical images disguising using advanced encryption standard. *International Journal of Engineering Research & Technology*, 2 (12), 3580 – 3582.

Aman, J. M. & Ali, B.Y. (2008). Image encryption using block-based transformation algorithm. *IAENG International Journal of Computer Science*, 35 (1), 15 - 23.

Bruen , A.A. & Forcinito , M. (2005). *Cryptography, information theory, and error-correction: a handbook for the 21st Century*. Wiley-Interscience, ISBN: 978-0-471-65317-2, John Wiley & Sons, Inc.

Cuppens, F., Garcia-Alfaro, J., Heywood, N. Z., & Fong, W. L. (2014). *Foundations and practice of security.* Springer, Montreal, QC, Canada.

Chandra, N. (2015). *What is message confidentication, integrity, authentication and non-repudiation?. Blogger Templates.* Retrieved from http://nishichandra.blogspot. com/2015/09/message-confidentiality-integrity-authentication-nonrepudiation-entity-identification.html.

Delfs, H. & Knebl, H. (2015). *Introduction to cryptography: principles and applications* 3rd Edition. Springer, Verlag Berlin Heidelberg.

Dietrich , S. & Dhamija , R. (2007). *Financial cryptography and data security*. Retrieved from http://www.springer.com/gp/book/9783540773658.

Dixit, J.B. (2007). *Computer programming.* 2nd Edition, Laxmi Publications (P) LTD, 113, Golden House, Daryaganj, New Delhi.

El-Bendary, M. A. M. (2015). *Developing security tools of WSN and WBAN networks applications*. Springer, Tokyo Heidelberg New York Dordrecht London.

Forouzan, B. A. (2007). *Data communications and networking*. 4th Edition. McGraw-Hill Forouzan Networking Series.

Gurvinder, S.S., Vinay, V. & Kumar, R. (2013).Comparing popular symmetric key algorithms using various performance metrics. *International Journal of Advance Research in Computer Science and Management Studies,* 1(7), 394 - 399.

Hamid, J, Revett, K. & Palmer-Brown, D. (2008). Global E-Security. *4th International Conference, ICGeS 2008*, London, UK, June 23-25, 2008, Proceedings.

Hans, D. & Helmut, K. (2007). *Introduction to cryptography: principles and applications*. 2nd Edition. Springer ISBN: 9783540492436, Hasan Mirjalili EPFL, Switzerland.

Jahankhani, H., Revett, K. & Palmer-Brown, D. (2008). Global E-security. Retrieved from https://downloaddigitaldevelopingdesign8.files.

Jawahar, N. (2014). Overview of system analysis and design. *System analysis and design information systems components.* Retrieved from https://eternal sunshineoftheismind. wordpress.com/2013/02/06/system-analysis-and-design-information-systems components.

Kessler, G.C. (2006). Types of cryptographic algorithms. *An overview of cryptography*. Retrieve from http://www.garykessler.net/library/crypto.html.

Kumar, R. (2004). *J2EE security for servlets, EJBs and web services: applying theory and web services*. Prentice Hall Press, Upper Saddle River, NJ, USA.

Mahapatra, A. & Rajballav, D. (2007). *Data encryption and decryption by using hill cipher technique and self repetitive matrix*. Retrieved from https://pdfs. semanticscholar.org/0580/7cd29b50cfbe160a0f299c79eb7e5e60026c.pdf.

Menezes, A. J., van Oorschot, P.C. & Vanstone, S.A. (2014*). Overview of cryptography*. Retrieved from http://cacr.uwaterloo.ca/hac/.

Miller , B. & Ranum , D. (2013). *Problem solving with algorithms and data structures release 3.0*. Retrieved from https://www.cs.auckland.ac.nz/courses/compsci 105ssc/resources/ProblemSolvingwithAlgorithmsandDataStructures.pdf

Pediapress GmbH. (2011). *Computer science: an overview.* Mainz, PediaPress GmbH, Boppstrasse 64, Mainz, Germany.

Ponemon, L. (2012). Encryption in the cloud. *Encryption in the cloud: who is responsible for data protection in the cloud?* Retrieved from https://www.ponemon.org/ local/upload/file/Encryption_in_the_Cloud%20FINAL_6_2.pdf.

Pressman, R. *(2010). Software Engineering: A Practitioner's Approach. Boston:* McGraw Hill, 41–42. ISBN 9780073375977.

Rouse, M. (2010). *Hash-based-Message-Authentication-Code-HMAC*. Retrieved from from http://searchsecurity.techtarget.com/definition/Hash-based-Message-Authentication-Code-HMAC

Shahd, ARH. (2013). Image Integrity based on HMAC Structure. *International Journal of Computer Science and Information Security Publication*, 11 (8), 19 - 24.

Vishwanath, U. (2012). Development of data encryption algorithms for secure communication using public images. *The University of Toledo Digital Repository*. Retrieved from http://utdr.utoledo.edu/cgi/viewcontent.cgi? article=1478&context=theses-dissertations.