Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 6, No. 4, December 2018, pp. 385~392
ISSN: 2089-3272, DOI: 10.11591/ijeei.v6i1.494

QR Code Integrity Verification Based on Modified SHA-1 Algorithm

Rogel L. Quilala[1], Ariel M. Sison[2], Ruji P. Medina[3]
[1,3]Technological Institute of the Philippines, Philippines
[2]Emilio Aguinaldo College, Philippines

Article Info

Article history:

Received Mei 8, 2018
Revised Sep 7, 2018
Accepted Dec 2, 2018

Keyword:

Certificates
Fraud
Hash
Smartphone

ABSTRACT

The modified SHA-1 algorithm was applied in the data integrity verification process of certificates with QR code technology. This paper identified the requirements needed in the certificate verification that uses the modified SHA-1. The application was tested using legitimate and fraudulent certificates. Based on the results, the application successfully generated QR codes, printed certificates, and verified certificates with 100% accuracy. During the trial run of the application, four test cases were seen which involves correct names and QR codes, and three other possible test cases of faking certificates such as modification of the name, regeneration of QR codes using valid hash and a fake name, and modification of the QR code. Although these cases exist, the application successfully verified all thirty certificates correctly. Also, it is noticed that during the scanning, the smartphone camera should be in focus to capture the QR code clearly.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Rogel L. Quilala,
Technological Institute of the Philippines,
938 Aurora Blvd., Cubao, Quezon City, Philippines.
Email: rlquilala@gmail.com

1. INTRODUCTION

Quick Response (QR) codes are a low-cost tagging technology famous for its simple production and less difficulty in implementation [1]. A study of mobile phones reveal fast pace of changes paralleling spreading mobile usage, across different age groups, places, times and situations [2] this increased usage resulted in the utilization of QR codes in different services because of its speedy recognition and processing [3]. Today, many areas use QR codes such as in authentication of products [4], [5] student result mark sheets and their profile information [6], [7] and banking [8], [9].

Another application of QR codes includes the checking of the data integrity of certificates issued by institutions to prevent dissemination of fraudulent documents [10]. Documents now can be effortlessly forged by tampering names and can be submitted to whichever company or institution to gain employment, reduce costs, and other financial benefits [11], [12]. Multimedia security researchers recommend document verification and authentication due to the rise in the number of fake documents because of advances in printing and scanning [13].

Several studies have used QR code for authentication on printing document for fraud identification. A mobile app proposed to use student's information from the database to be encrypted and saved on the server which is then integrated into the QR code and printed on the document for verification purposes [14]. A different study combined QR Code, digital signature signed by university authorities, hashing and smartphone application [10] in the verification process. However, both studies need to Install the created mobile application separately to read the QR code. In another study, incorporating a significant amount of self-describing data in the QR protects paper-based documents, but this emphasizes the need for no shake on the camera and better focusing mechanisms of smartphones [12]. Another scheme embeds watermark object

with QR code to determine printed document validity, but this scheme needs the watermark image transparency set to 50% and prepare validation link in advance, and also set the size of QR to be not less than 2x2 cm2 to be able to read the watermarked QR code efficiently [15]. Another study implemented paper-based document authentication with the use of digital signature and QR code however with the inclusion of an optical character recognition (OCR) which requires human intervention when OCR fails which makes it inconvenient [16]. This paper will apply QR code technology in verifying the authenticity of certificates using a web application which doesn't require additional installation on the part of the user.

In information security, cryptographic hash algorithms form a significant part specifically in data integrity [17]. A hash is computed from data files to verify its integrity and identify duplicated data or files [18]. Through this, a small change done on the data during transit will produce a different hash value [19]. This way, a hash assures that the receiver obtained the same message sent by the sender without alteration during transmission [20].

Secure Hash Algorithm 1 (SHA-1) is a cryptographic hash function that produces a 160-bit hash value [21], [22]. Designed by the National Institute of Standards and Technology (NIST), SHA-1 is considered the most widely used hash algorithm in a vast range of applications [17], [23], [24] due to its time efficiency, robustness [25] and speed [26]. Currently, SHA-1 is still in use by 21% of websites in the world in signing certificates [27]. SHA-1 based fingerprint is still widely used and supported for verification [28].

Although SHA-1 is popular, widely used, and accepted, it does not seem to offer adequate avalanche effect concerning the distribution of the input differences and unexpected weaknesses in the construction of all the step updating functions [29], [17]. This problem will lead to the chance of having two different input that will generate the same output value in the middle of algorithm or compression function [30], [31]. Hence, there is a need to devise a function with improved diffusion to distribute the output in each round and prevent the same in the next coming stages [32], [33], [30].

Some studies made proposed enhancements on SHA-1 aimed to attain additional diffusion [34], [35] but did not show the bit-difference on the simulation of result or have shown lower bit difference. Another incorporated MD5 to SHA-1 [32], but this approach will suffer from the same vulnerability [36], [37]. Another study has not included actual messages used during the experiment [38]. Therefore, the researcher has chosen to improve the algorithm of SHA-1 by increasing hash size output from 160 to 192 bits and provide a mixing mechanism to attain efficient diffusion.

This paper presents a certificate integrity verification process that can be quite convenient and quick by combining the modified SHA-1 hash and QR code technology in a windows and web application. The system will provide a module for printing the certificates with the generated QR code on paper. The study aims to 1) identify requirements in building the application, 2) design and determine the modules included, 3) integrate the modified SHA-1 in the certificate verification application, and 4) test the QR Code Integrity Verification application. This application will enable the verification of documents without a unique gadget or additional installation. The College of Computer Studies of Tarlac State University will test the QR code integrity verification application, but any institution that may require this service can use this application.

2. The Proposed Integrity Verification System

Figure 1 shows the block diagram applying the modified SHA-1 algorithm to the message integrity verification. The first task is to generate the QR code using the modified SHA-1 algorithm in the certificate. The input message M consists of the students' name and type of certificate for issuance along with the unique ID. The modified SHA-1 algorithm will be applied to the message and ID to create a hash value of 192 bits to be saved to the database. The QR code generator will generate the QR code from the save data in the database.

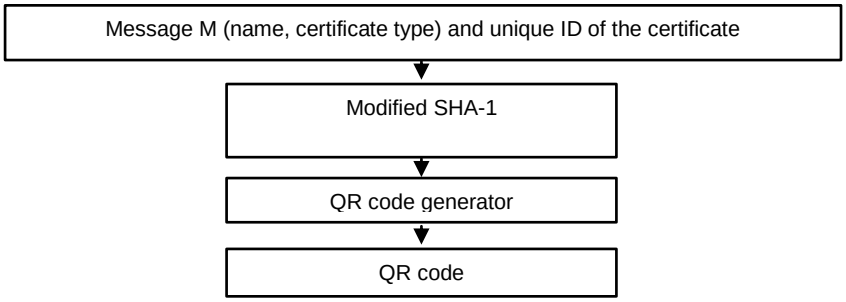


Figure 1. Block Diagram of QR code generation applying the modified SHA-1 algorithm

The second task is the verification of the QR code in the printed certificate with the modified SHA-1 algorithm. Figure 2 shows the block diagram of this process. Users who want to verify the certificate will use their smartphones running on Android with a camera to scan the QR code. After capturing the QR code, the web application computes the hash value and send the hash and the message to the web server. After receiving the information, the server search for the hash. If the value exists, the system will retrieve the unique ID of the certificate from the database. Using message M and ID, the modified SHA-1 compute for the hash value h'. If hash value h' = hash value h, the message is said to be authentic. Then, the system will display the resulting message for visual inspection and compare this to the details of the printed certificate. Otherwise, we can say that the QR code message M is modified.

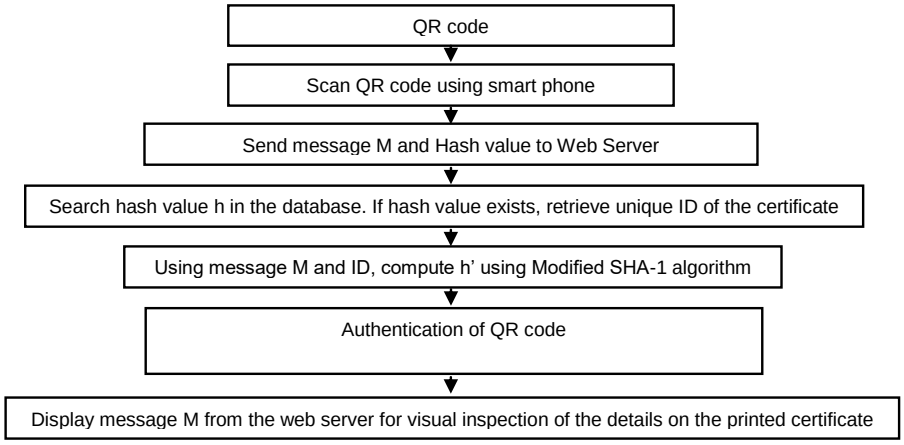


Figure 2. Block Diagram of QR code verification using modified SHA-1

3. Results and Discussion

3.1. Requirements for building the application

Hardware and software requirements include a camera with a smartphone running in Android 4.0.3 and up to access the web application. The web application will involve the capturing of the QR code printed on the certificate using the phone's camera and sending the obtained message and hash code to the web server. The web application is best viewed using Mozilla Firefox, a fast and free Android browser. A web server, where the QR code is printed on paper and checked against the database of valid hashes is needed. The software used Microsoft Web Server Internet Information Service (IIS) Version 7.5, a general-purpose web server developed by Microsoft running on Windows operating system. In this study, IIS run Active Server Page (ASP). The ASP.NET framework is a server-side script engine that creates interactive web pages. The application uses a server that has an Intel(R) Core(TM) i5-6500 CPU @3.20GHz 3.19 GHz processor, 8.0 GB RAM, running a 64-bit Windows Operating System for testing. On the client side, the required smartphone should have a camera running on an Android platform to be able to use the certificate integrity checker. The data or sample certificate used in this study will come from the College of Computer Studies Data of Tarlac State University. The college issues certifications to students enrolled in the field trips and seminars course. This course exposes the students to IT technologies being applied and adopted by companies. Students enrolled in this course are required to attend seminars to keep abreast of the current trends as far as hardware, software, and telecommunications are concerned. The student enrolled in the course are required to submit a compiled report containing reaction papers about the discussed topics which will be presented to and evaluated by their adviser. One of the requirements to be attached to the portfolio is the certificate of completion.

3.2. Design and modules

Three modules were identified and is explained in detail below:

a. QR code generation module: This module generates the QR code for printing on certificates. First, the system administrator will input message M which may consist of the name of the student and certificate type with the unique ID of the certificate. Next, the modified SHA-1 will be applied to message M and ID to create a hash value of 192 bits to be saved to the database. The QR code generator will generate the QR code from the save data in the database which consists of the hash value h(ID||M) and message. The QR code generated is to be printed on the certificates.

b. QR code printing module: This module prints the generated QR code on paper. Before the actual printing, the user provides the list of the names, QR code generated, and image for the design of the certificate. The certificates are set using a letter size (8.5"x11") paper. The user will arrange the design of the certificate on a windows application including the background, position of the name, and position of the QR code on paper. Once done, certificates are printed based on the list of students provided.

c. QR code scanning and verification module: This module scans the QR code and contacts the server for the verification process. After printing the code on the certificate, clients who want to verify the certificate will use their smartphones running on Android with a camera to scan the QR code. To do this, the user's smartphone WiFi should be connected to the same network as that of the server. The web server is located at the TSU-CCS Control room running on IIS version 7.5. After that, the user needs to access the web server by typing the URL http://193.168.1.28 using the phones' web browser. After typing the address of the server on the mobile browser, the user only needs to scan the QR code using the camera of their smartphone. After capturing the QR code, click the verify button to let the web application send the message M and the hash value to the web server for data integrity verification. The server receives the information stored in the QR code, and then it will search if the hash value exists in the database. If the value exists, the system will retrieve the unique ID of the certificate from the database. The modified SHA-1 will be applied to the message M and ID to create hash value h'. The system will verify if hash value h' = hash value h to check the integrity of the QR code and if successfully verified, the message is said to be authentic. If hash h' does not exist, the QR code message M is modified. The system displays the resulting message for visual inspection of the details from the printed certificate. Integration of the modified SHA-1 in the certificate verification.

3.3. Integration of the modified SHA-1 in the certificate verification application

The modified SHA-1 algorithm has been incorporated both in the generation of the QR and in the verification process. Figure 3 illustrates the proposed modification on the SHA-1 construction with the inclusion of a counter. The counter was XORed to the intermediary hash value to strengthen the M-D construction. The counter will have an initialized value of zero and is incremented by 1 for every message block until the last block thus changing the assigned number to the counter changes in every round**.**

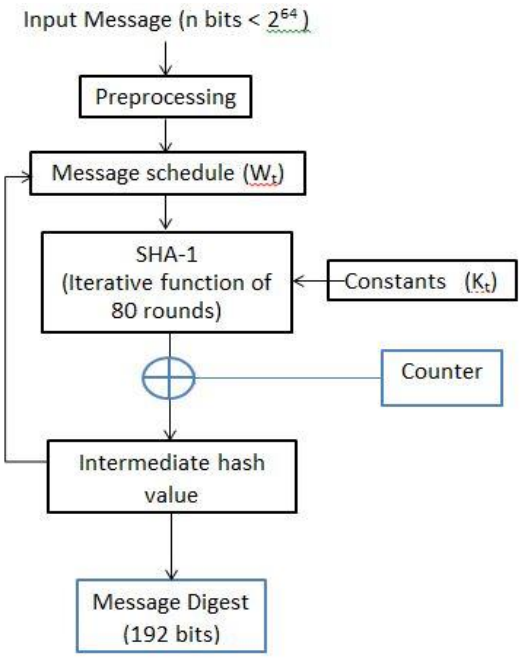
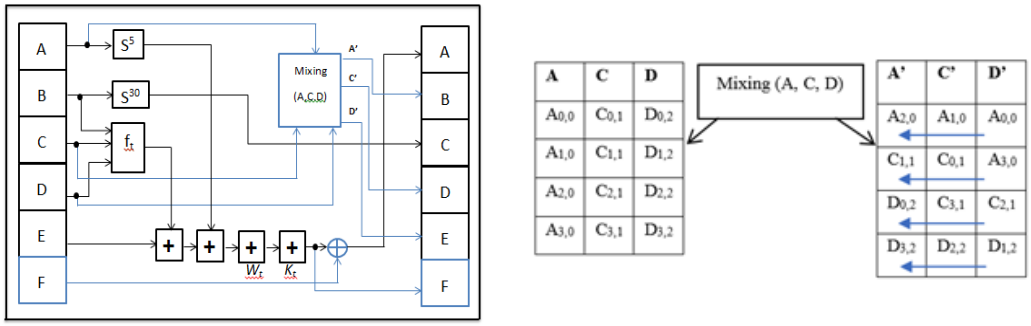


Figure 3. Modification on SHA-1 construction

Figure 4. (a) explains compression modification on SHA-1 including the added mixing method. The modified SHA-1 increased the hash output from 160-bits to 192-bits to strengthen the algorithm and kept the 80 rounds. The modification added one chaining variable F and XORed to the output of E.



(a) (b)

Figure 4. (a) Modification on SHA-1 compression with added mixing method (b) Mixing method

All studies did not change variables A, C, and D in every round. In the proposed algorithm, these variables entered the mixing function for better diffusion. The mixing function assures the distribution of the input values (A, C, and D) because the contents of the variables will not be the same in the coming rounds. Variable E goes to variable F after executing its addition operations. The difference of SHA-1 and modified SHA-1 lies in the computation of the message digest. The padded message is still used to compute for the message digest. The calculation makes use of two buffers (A, B, C, D, E, F and H0, H1, H2, H3, H4, H5). The first buffer uses five 32-bit words, and the second buffer comprises of eighty 32-bit words (W0, W1 ... W79). This process also applies TEMP1 and TEMP2 buffers. {Hj} are initialized before processing any blocks with values of 67452301, EFCDAB89, 98BADCFE, 10325476, C3D2E1F0, 40385172 (H1-H5). Let hash value length be m.

Modified SHA-1 steps to process the message in 16-word blocks:

1) Split Mi into 16 words starting from left to right, W0, ... W15
2) When t = 16 to 79, we do Wt =S1(Wt-3 XOR Wt-8 XOR Wt-14 XOR Wt-16).
3) Then let A=H0, B=H1, until F=H5, counter = m
4) When t = 0 upto 79 do
   mixedACD= mixingACD(A, C, D)
   A'=mixedACD;C'=mixedACD;D'=mixedACd
   TEMP1 = S5(A) + ft (B, C, D) + E + Wt + Kt;
   TEMP2 = F xor TEMP1
   E = D'; D = C'; C = S30(B); B = A'; A = TEMP2; F=TEMP1
5) counter+= m, then do H0 = (H0 + A) xor counter, H1 = (H1 + B) xor counter, H2 = (H2 + C) xor counter, H3 = (H3 + D) xor counter, H4 = (H4 + E) xor counter, H5 = (H5 + F) xor counter.

After processing Mn, these words represent the computed 192-bit hash value: H0 H1 H2 H3 H4 H5.

Figure 4 (b) shows the mixing function. The function accepts the working variables A, C, and D as the input column then disperses the bits to different arrangement from right to left in row-wise fashion in the output column A', C,' and D'.

3.4. Testing the QR Code Integrity Verification application

First, the QR code generation module will produce the 192-bit modified SHA-1 hash using the name of the student as the message combined with the ID number and is saved in the database. Figure 5 shows the QR code generation.

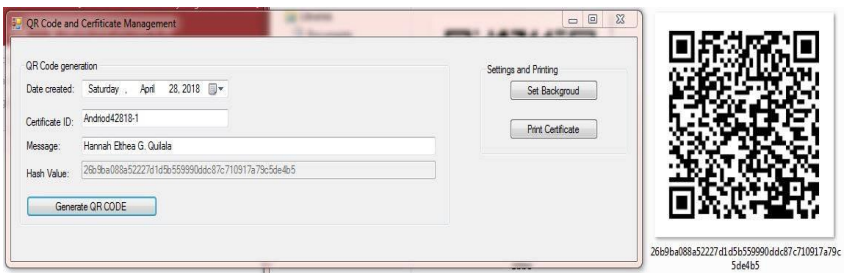


Figure 5. QR code generation using message and ID

Second, certificates are printed by batch with the background design integrated along with the name of the student and the QR code on paper. Figure 6 (a) shows the sample certificate.

(a) (b)

Figure 6. (a) Printing of certificates with QR code (b) Scanning of the QR code and verify button

Third and the last step is the verification process. To access the verification system, the user needs to type the URL http://193.168.1.28 on the web browser of their Android phones. The simulation program used the IP address 193.168.1.28 for the server situated in the TSU-CCS Control room. After typing the address of the server on the mobile browser, the user only needs to scan the QR code on the designated area as indicated by the red corners shown in Figure 6 (b) and then press the verify button. The generated hash extracted from the QR code will be compared against the hashes saved on the server as a way of verification. If the same hash is found on the server, the name of the student is displayed.

Thirty sample certificates were used in the trial run of the verification app. Four possible test cases were identified. Case 1 includes correct name of attendee and correct QR code; case 2 involves a fake name printed on certificate and valid QR code copied from another certificate; case 3 where the hash value from the QR code of another certificate is used to generate a new QR code along with a new fake name ; and case 4 comprises of fake QR code, this code is not valid because it is not registered on the list saved on the database. Out of the 30 samples, three counterfeit certificates were incorporated. Based on the result of the trial run, the app successfully verified all thirty certificates (27 being correct, and three being faked) achieving a 100% accuracy. Table 1 shows the result of the trial run recognizing all cases, messages displayed by the app for all instances, and the names listed on the certificate for manual inspection.

Table 1. Trial Run

Test Case	Certificate No.	Verified?	Verification Process Verification Message	Name on Certificate
1	1,2,3,4,5, … 27	Yes	This certificate belongs to: ANGELES, Jerwin	ANGELES, Jerwin
2	28	Yes	This certificate belongs to: ANGELES, Jerwin	XYZ, Fake1
3	29	Yes	Warning: The message has been modified. This certificate belongs to: ANGELES, Jerwin	XYZ, ModFake3
4	30	Yes	This certificate does not exist	XYZ, NotExFake4

For case 1, the name listed on the certificate and the verification message should match. For case 2, the name listed on the certificate will not match the message displayed in the verification prompt. This signifies that the certificate was modified. For case 3, the hash will be found in the database, but the name produced from the regeneration of QR code is not the same as that on the database therefore the message has been modified is displayed. Lastly, for case 4, the generated hash will not be found on the saved hashes on the database because the hash will not exist so the message "The certificate does not exist" is displayed. This only means that the certificate is also faked.

From the trial run, it is also observed that to scan QR codes efficiently, the smartphone camera should be in focus during the capturing.

4. Conclusion and Future Works

The use of QR codes in smartphones is becoming famous because of its simplicity and low-cost of production. One of the services that make use of QR code is data integrity verification of tampered certificates. Requirements were gathered and identified to build the simulation application. Modified SHA-1 applied a mixing method to attain better diffusion in the hash value and increased the hash value output to 192-bits for added strength. The QR code integrity verification application was tested using sample names and ID number to generate the QR code and print the certificates on paper. The fraudulent and legitimate certificates were included in the testing. Results indicate 100% accuracy in the verification process for thirty printed certificates. Although there were four cases noted including one correct and three other possible cases of faking certificates, the app successfully verified them all. Also, it is noticed during scanning that smartphone camera should be in focus to capture the QR code clearly.

As for future works, the application can be further applied to degree certificate verification and may use optical character recognition algorithms as a supplement to the manual inspection of the certificate.

REFERENCES

[1] M. Pérez-Sanagustín, D. Parra, R. Verdugo, G. García-Galleguillos, and M. Nussbaum, "Using QR codes to increase user engagement in museum-like spaces," *Comput. Human Behav*, vol. 60, pp. 73–85, Jul 2016.

[2] Z. S. Ali, "Mobile Phone and Pakistani Youth : A Gender Perspective," *J. Telemat. Informatics*, vol. 1(2), pp. 59–68, 2013.

[3] D.-S. Oh, B.-H. Kim, and J.-K. Lee, "A Study on Authentication System Using QR Code for Mobile Cloud Computing Environment," J. J. Park, Y. Pan, C.-S. Kim, and Y. Yang, Eds. Berlin, Heidelberg: *Springer* Berlin Heidelberg, vol. 309, pp. 500–507, 2011.

[4] H. Keni, M. Earle, and M. Min, "Product authentication using hash chains and printed QR codes," in *14th IEEE Annual Consumer Communications & Networking Conference (CCNC) 2017*, pp. 319–324, 2017

[5] A. Hole, M. Jadhav, S. Kad, and S. Shinde, "*Encryption and Decryption of Data Using QR Authentication System*," vol. 3(4), pp. 488–496, 2014.

[6] S. Goyal, S. Yadav, and M. Mathuria, "Exploring concept of QR code and its benefits in digital education system," *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 1141–1147, 2016.

[7] S. Dey, A. Nath, and S. Agarwal, "Confidential Encrypted Data Hiding and Retrieval Using QR Authentication System," in *2013 International Conference on Communication Systems and Network Technologies*, pp. 512–517, 2013.

[8] V. Malathi, B. Balamurugan, and S. Eshwar, "Achieving Privacy and Security Using QR Code by Means of Encryption Technique in ATM," *2017 Second Int. Conf. Recent Trends Challenges Comput. Model*, pp. 281–285, 2017.

[9] J. Murkute, H. Nagpure, H. Kute, N. Mohadikar, and C. Devade, "Online Banking Authentication System Using QR-code and Mobile OTP," *Int. J. ...,* vol. 3(2), pp. 1810–1815, 2013.

[10] A. Singhal and R. . Pavithr, "Degree Certificate Authentication using QR Code and Smartphone," *Int. J. Comput. Appl*, vol. 120(16), pp. 38–43, 2015.

[11] N. Gupta, N. Mokashe, and M. Parihar, "QR code : A safe and secure method of authenticating legal documents," *Int. J. Eng. Res. Gen. Sci*, vol. 3, no. 1, pp. 951–954, 2015.

[12] C. M. Li, P. Hu, and W. C. Lau, "AuthPaper: Protecting paper-based documents and credentials using Authenticated 2D barcodes," *IEEE Int. Conf. Commun*, vol. 2015, pp. 7400–7406, 2015.

[13] I. Tkachenko, W. Puech, O. Strauss, C. Destruel, and J.-M. Gaudin, "Printed document authentication using two level or code," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2149–2153, 2016

[14] Z. Yahya et al., "A New Academic Certificate Authentication Using Leading Edge Technology," in *Proceedings of the 2017 International Conference on E-commerce, E-Business and E-Government - ICEEG 2017*, pp.82–85, 2017

[15] T. Mantoro, M. I. Wahyudi, M. A. Ayu, and W. Usino, "Real-Time Printed Document Authentication Using Watermarked QR Code," *Proc. - 4th Int. Conf. Cyber Secur. Cyber Warf. Digit. Forensics, CyberSec 2015*, pp. 68–72, 2016.

[16] M. Warasart and P. Kuacharoen, "Paper-based Document Authentication using Digital Signature and QR Code," *4th Int. Conf. Comput. Eng. Technol. (ICCET 2012)*, no. Iccet, 2012.

[17] N. Kishore and B. Kapoor, "Attacks on and advances in secure hash algorithms," *IAENG Int. J. Comput. Sci*, vol. 43,(3), pp. 326–335, 2016.

[18] Sam Farisa Chaerul Haviana and D. Kurniadi, "Average Hashing for Perceptual Image Similarity in Mobile Phone Application," *J. Telemat. Informatics*, vol. 4(1), pp. 12–18, 2016.

[19] I. Alsmadi and M. Zarour, "Online integrity and authentication checking for Quran electronic versions," *Appl. Comput. Informatics*, vol. 13(1), pp. 38–46, 2017.

[20] R. P. Arya, U. Mishra, and A. Bansa, "A Survey on Recent Cryptographic Hash Function Designs," *Int. J. Emerg. Trends Technol. Comput. Sci*, vol. 2(1), pp. 117–122, 2013.
[21] NIST, "FIPS 180-1 - Secure Hash Standard," *FIPS PUB 180-1*, no.17, 1995.
[22] Q. H. Dang, "Secure Hash Standard," *Gaithersburg*, MD, Jul, 2015.
[23] S. Rao, "Advanced SHA-1 Algorithm Ensuring Stronger Data Integrity," *Int. J. Comput. Appl*, vol. 130(8), pp. 25–27, 2015.
[24] R. A. N. Karthik, A.K. Parvathy, "Non-convex Economic Load Dispatch using Cuckoo Search Algorith," *Indones. J. Electr. Eng. Comput. Sci*, vol. 5(1), pp. 48–57, 2017.
[25] K. Saravanan and A. Senthilkumar, "Theoretical Survey on Secure Hash Functions and issues," *Int. J. Eng. Res. Technol*, vol. 2(10), pp. 1150–1153, 2013.
[26] P. Garg and N. Tiwari, "Performance Analysis of SHA Algorithms (SHA-1 and SHA-192): A Review," *Int. J*, vol. 2(3), pp. 130–132, 2012.
[27] Venafi, "Venafi Research: Twenty-One Percent of Websites Are Still Using Insecure SHA-1 Certificates and Putting Users at Risk," *Venafi Press* Release, 2017.
[28] M. Stevens and D. Shumow, "Speeding up detection of SHA-1 collision attacks using unavoidable attack conditions.," *USENIX Secur*, vol. 2017, p. 173, 2017.
[29] X. Wang, Y. L. Yin, and H. Yu, "Finding Collisions in the Full SHA-1," *Adv. Cryptol. – CRYPTO 2005*, no. 90304009, pp. 17–36, 2005.
[30] A. Kumarkasgar, J. Agrawal, and S. Shahu, "New modified 256-bit MD5 Algorithm with SHA Compression Function," *Int. J. Comput. Appl*, vol. 42(12), pp. 47–51, Mar 2012.
[31] P. Karpman, T. Peyrin, and M. Stevens, "Practical Free-Start Collision Attacks on 76-step SHA-1," vol. 2012, 2015.
[32] G. Gupta and S. Sharma, "Enhanced SHA-192 algorithm with larger bit difference," *Proc. - 2013 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2013*, pp. 152–156, 2013.
[33] X. Xu, Q. Zhao, and C. Li, "Advanced framework for iterative hash functions," *Proc. - 2012 Int. Conf. Comput. Sci. Electron. Eng. ICCSEE 2012*, vol. 2, pp. 599–602, 2012.
[34] C. C. G. San Jose, B. T. Tanguilig III, and B. D. Gerardo, "Enhanced SHA-1 on Parsing Method and Message Digest Formula," pp. 1–9, 2015.
[35] L. Thulasmani and M. Madheswaran, "Security and Robustness Enhancement of Existing Hash Algorithm," *2009 Int. Conf. Signal Process. Syst*, pp. 253–257, 2009.
[36] X. Wang, D. Feng, X. Lai, and H. Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD," *IACR Cryptol. ePrint Arch*, vol. 5, pp. 5–8, 2004.
[37] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions," *Adv. Cryptol. – EUROCRYPT 2005*, pp. 19–35, 2005.
[38] S. Verma and G. S. Prajapati, "Robustness and security enhancement of SHA with modified message digest and larger bit difference," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pp. 1–5, 2016