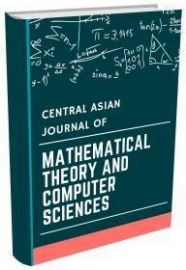

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS
Volume: 05 Issue: 03 | July 2024 ISSN: 2660-5309

Article

Security System of Data Transmission with MATLAB Modeling and Simulation the Hash Encryption

Aseel.A.Qasim[1]

1. Department of Information Technology, Faculty of Computer Science and Mathematics, University of Thi-Qar, Thi-Qar,64001, Iraq

* Correspondence: aseel.a.qasim@utq.edu.iq

Abstract: To put special emphasis on CMC network security (message encryption context), this paper is a design and simulation one. In this, I will demonstrate how to simulated using MATLAB encryption. The code is written in a way that any malicious attack will wipe the data out. Data will be auto protected by the system and get recovered at the end of the process. This aims to make security more efficient by providing an automated deep-learning system. Security can be improved with system training the system will validate the incoming data for any other discrepancies, or false data The defragmentation function is run the system will put the next stage which is to program the key length tracking messages; just to reinforcing that the mater message is conforming regarding formatting in the hash. If anything comes up which is not in hash format on system/panel, that will be ignored. This is the message that is being transmitted by transmitters as a sequence of blocks: What this is doing is that there is a receiver in a blocks series happening. Fragmentation too protects messages in the transmitter, and it arranges them in encrypted and unencrypted chunks.

Keywords: Cryptograph, Hash Function, Encryption, Data Security, Simulation, Network Security

1. Introduction

information security is the process of protecting data and communications from unwanted access, use, disclosure, disruption or alteration. In other words, we want to secure our system assets and data (wherever it is located) from common vulnerabilities that one can exploit. to the enemy side of that current are rude ass crackers cracking our networks to steal, hacking them and keeping up with antiviruses [1], scooping people buildings such as natural catastrophes like how it's snowing now having major power outages then firing the gas lines for so many they don't see lights breaking in trying to brazen off someone's place thinking it a joke their supposedly attempting policing better themselves till strained no longer redirect focus other way most apt types. [2].

Huge encryption schemes and data protection technologies have been designed to protect the message of today, or in other words you would probably call it as data Created programs to remove malware and viruses [3] If something is in a hurry, then it can either be spyware or collect user data and transaction history from the system.

Thereby, network security will be more and more advantageous. The protection of study outcomes will be shown by MATLAB. video message with user data from a corpus hash the data is then secured by an encryption using cryptography [4].

Using the CMS protocol means that encrypted data will become more solid and resistant to hacking attempts by computer hackers. The research study demonstrates [5].

Citation: Aseel.A.Qasim. Security System of Data Transmission with MATLAB Modeling and Simulation the Hash Encryption Central Asian Journal of Mathematical Theory and Computer Sciences 2024, 5(3), 239-250.

Received: 12th July 2024
Revised: 16th July 2024
Accepted: 25th July 2024
Published: 29th July 2024

2. Materials and Methods

Problem Statement.

There remains a lot of unexplored in the current level of understanding in the data protection. For example, one can include the following research areas:

- automatic algorithm updates for new attackers.
- real versus fake computer infections.
- a truly immune-to-worm network.

These challenges establish new opportunities for the new data protection systems or solutions. They may be needed for system experimentation and development. Therefore, the quality of software development also affects the degree of success. The device may also have an artificial intelligence system for additional protection.

Scope of Project

The modelling is using MATLAB coding. As mention before, the purpose of the modelling is to secure the data.

The general coding for hash message is:

- Define the key message
- Determine the key length
- Concatenate the message
- Present the message of hash in Hex

The very beginning before write the coding is to ensure the MATLAB supports the hash function. If the MATLAB does not support the hash function, user has to create one in the library.

After define the hash function, the next thing is to program the message follows the key length. This is to ensure the key message is follow format in the hash. Anything not follow the hash format will be ignored in the system or pad with zero.

Finally, the message is hashed and cryptography into a block so that the message is more secure and presentable in the network.

3. Results and Discussion

Hash Function Programming Coding in MATLAB

The Hash function must be defined before writing any hash security program. Below shows the Hash program written in MATLAB coding.

```
function hash = HMAC(key,message,method)

% key:      input secret key in char
% message:  input message in char
% method:   hash method, either:
%           'SHA-1', 'SHA-256', 'SHA-384', 'SHA-512'
```

Figure1: MATLAB coding method to hash

The hash function begins with "function hash = HMAC (key, message, method). Here the HMAC is a variable used to hold the key, message and method to hash .

The method to hash is giving user an option so that they can choose which type of hash convenient to secure their data.

Define the Hash Method

Once the hash function is defined, now user has to choose which hash they have to choose for cryptography [6]. Below shows the MATLAB coding written for user to choose the type of hash used in secure communication [7].

```
original_message=input('Please enter the message you want encrypted using
CMS:', 's') % Asking to key in the message
         number_message=double(original_message)
         key = original_message
      for k=1:length(original_message)
          if number_message(k)>=65 && number_message(k)<=90
              number_message(k)*number_message(k)+key
              if number_message(k)>=90
                  number_message(k)*number_message(k)-26
              end
          elseif number_message(k)>=97 && number_message(k)<=122
              number_message(k)*number_message(k)+key
              if number_message(k)>=122
                  number_message(k)*number_message(k)-26
              end
          end
      end
      ec = char(mod(original_message + key - number_message, 26) +
number_message)
      fprintf('The encrypted message is %s \n',number_message)
```

Figure2: MATLAB CMS coding

In the coding above, the first line is asking user to enter the message that he or she wants to encrypt using CMS algorithm [8]. The rest of the coding is a CMS algorithm where the message first being check with its length and content [9]. After checking the length of the message, a secret code is added. This is shown in the equation when the original message is added with key minus the number message [10]. This is the CMS algorithm. The for loop and if else function are used to checking the message content and ensure the message is valid for CMS algorithm [11] .

Checking and Select the Key Length

Checking and select the key length is second part for the hash function. This is important to calculate how to hash the message so that the message is within the network requirements .
The MATLAB coding to check and hash the message is shown below:

```
% if key length > Blocksize calculate Hash and format as binary
if length(key) > Blocksize
    Opt.Method = method;
    Opt.Format = 'uint8';
    Opt.Input = 'bin';

    Hash_key = DataHash(uint8(key),Opt)

    for i = length(Hash_key):Blocksize
        Hash_key(1,i) = 0;
    end
    key_bin = uint82bin8(Hash_key);
end

% if key length < Blocksize right pad with zeros and format as binary
if (length(key) > 0) && (length(key) < Blocksize)
    key_bin = str2bin8(key);
    L = length(key);
    for j = L+1:Blocksize
        key_bin{1,j} = [0 0 0 0 0 0 0 0];
    end
end
```

Figure 3: MATLAB coding Checking and Select the Key Length

Note that for every hash method selected, the block will generate in the form of matrix. The matrix consists of data that will be hashed.

Formatting the Hash Data

In order to arrange the hashed data so that they are presentable in the network, the following coding is writing.

```
% format inner and outer padding as binary cell arrays
i_pad = [0 0 1 1 0 1 1 0];
o_pad = [0 1 0 1 1 1 0 0];
for i = 1:Blocksize
    i_pad_bin{1,i} = i_pad;
    o_pad_bin{1,i} = o_pad;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculate key xor ipad and key xor opad
i_pad_key_bin = bit8xor(key_bin,i_pad_bin);
o_pad_key_bin = bit8xor(key_bin,o_pad_bin);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Change format to uint8
i_pad_key_hex = bin2hex(i_pad_key_bin);
i_pad_key_uint8 = hex2uint8(i_pad_key_hex);

o_pad_key_hex = bin2hex(o_pad_key_bin);
o_pad_key_uint8 = hex2uint8(o_pad_key_hex);
```

Figure 4: Formatting the Hash Data

After arrange the hash message, the next process before toward the end is the concatenate. The following program written in MATLAB shows concatenate the hash message.

```
% Calculate Hash of i_pad_key||message
Opt.Method = method;
Opt.Format = 'uint8';
Opt.Input = 'bin';

Hash_i_pad_uint8 = DataHash(concat_i_pad,Opt);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% concatenate (o_pad || Hash_i_pad)
concat_o_pad = [o_pad_key_uint8,Hash_i_pad_uint8];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate final Hash in HEX
Opt.Method = method;
Opt.Format = 'HEX';
Opt.Input = 'bin';

hash = DataHash(concat_o_pad,Opt)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```

Figure 5: Formatting the Hash Data

4. the Simulation and results

The settings are shown below

Step 1: Configure the upper layer of the transmitter

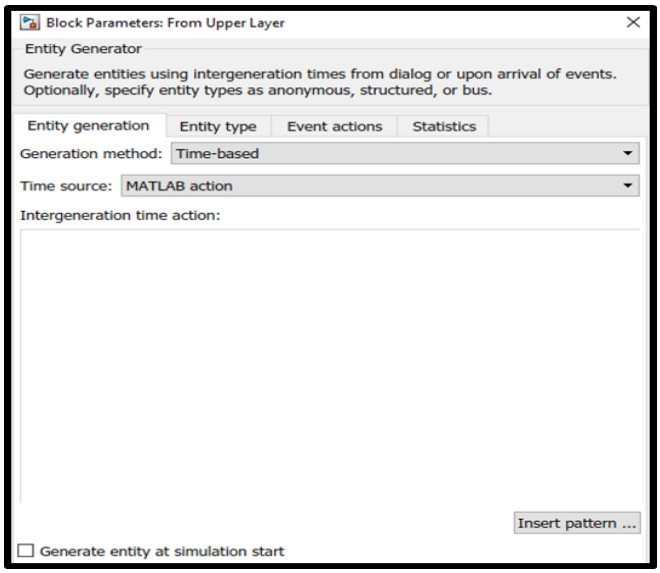

Figure 6: Configure the upper layer of the transmitter

Upper layer of the transmitter is the place where the user data is generated. This layer is important because it generates random user data before enter into the channels .

When double click on the upper layer block, the message block shown in Figure 6 appears. In the message block, do the following setting :

- Generation method: Time based
- Time Source: MATLAB action
- Entity type: Anonymous
- Entity priority: 300
- Data initial value: zeros (paramGobacknwithphy.numBitsPerPayload, 1)

The generation method in the setting is to tell the MATLAB what kinds of data generation is needed in the simulation. Here, we choose time based because the time is important to count the message being hashed and cryptograph into the network. Time based also allow the designer know what has happen to the message at a particular of time when it is attacked by the malicious. By setting in time based, one will know how efficient the data being protected .

The time source is also required in the simulation. The time source here is set to "MATLAB action". This is using internal library of the MATLAB and when choosing this setting, MATLAB will use its own way to compute and determine results .

The entity type is set to "Anonymous" so that the message being generated is unknown. Only this way, the message can be protected and hashed .

The entity priority is set to 300. This value can be changed and it will affect the performance of the message generated. 300 means 300 messages being targeted as priority in the study of hashed and cryptography .

The data initial value is the message source. This can be chosen from the file zeros (paramGobackwithphy…). This is a MATLAB library file where it helps to generate the messages for this CMS algorithm .

Step 2: Configure the transmitter

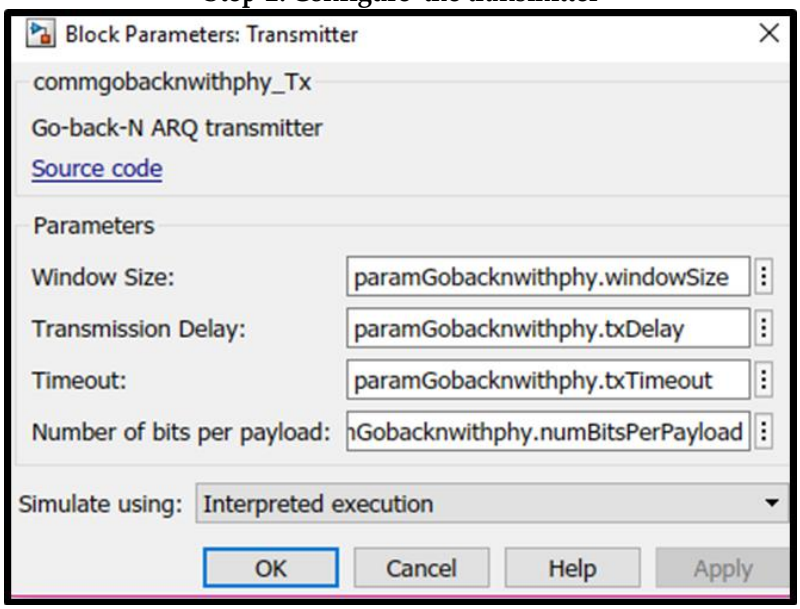

Figure 7: Configure the transmitter

This step 2 is very important because the code of hash and cryptography CMS is embedded. As can see from the diagram, there is a "source code" highlight in blue color. This source code is the code of security data protection using CMS with hashed message. Follow the window shown in Figure 7 to do the setting. All the setting should use the "paraGobacknwithphy…" files in the MATLAB Simulink library. After complete the setting, click "Apply" then "Ok" to close the window.

Step 3: Configure the channels

Double click on the channel 1 or channel 2 block, the following message window box appear. This message box shows the SNR parameter. We use this SNR setting to interfere the messages when they arrive in the channels. By doing interference, we will know how well the data being protected.

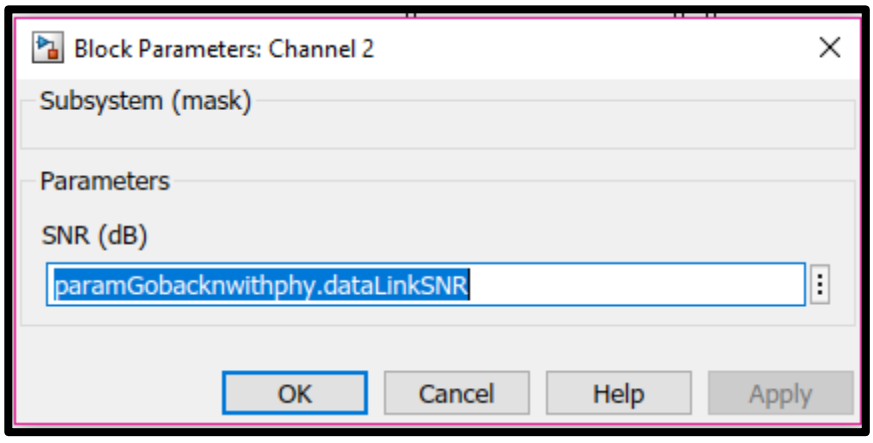

Figure 8: Using SNR represents the malicious attack in the network

Note that the SNR is the Signal to Noise ratio. This SNR can be low or high. When any attacks to the messages, this will consider as a noise. Thus, the messages will be corrupted when badly increase the SNR values.

Step 4: Configure the receiver block

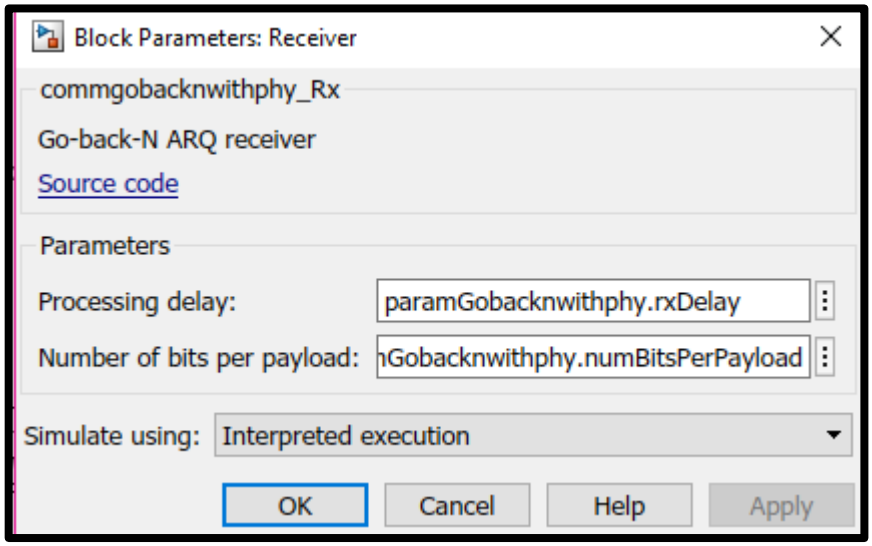

Figure 9: Configure the receiver block

The receiver block is the block where it should receive the message send from the transmitter without errors or missing messages. To ensure the messages being safe received at the receiver, this receiver will be programmed to remove the noise contents or correct the message when it is corrupted by the malicious.

Notice that the processing delay and the number of bits per payload are all using MATLAB library file. There is also the source code found in the receiver. This source code will show the hashed cryptography message being recover and avoid attack from malicious. The above setting is important ready to run the simulation.

Step 5: Configure the receiver upper block

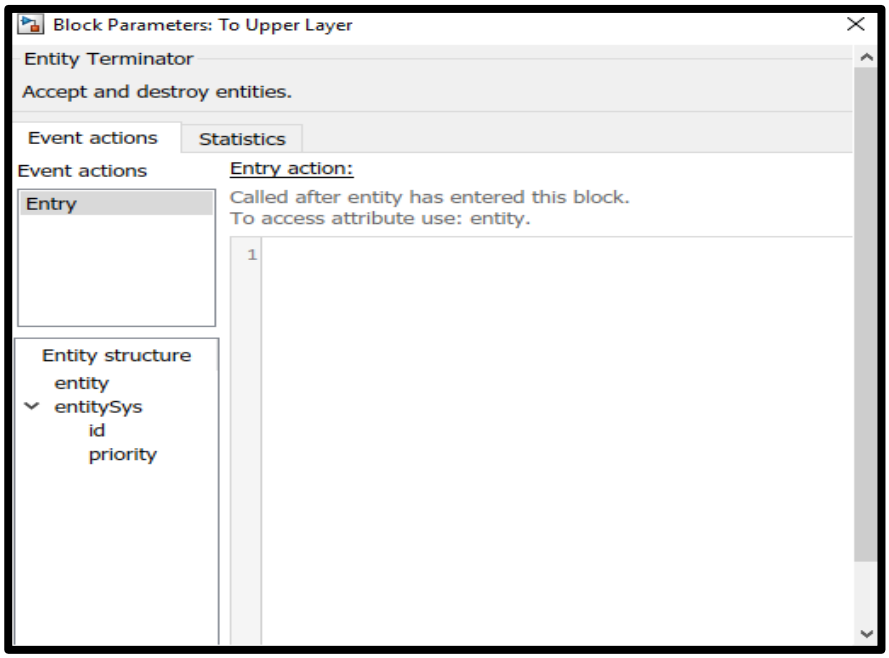

Figure 10: Receiver upper layer block

The receiver upper block is the block that should shows the message where they are the same like the one transmits from the transmitter. In a nutshell, this block basically used to recover the messages. For example, if transmitter transmit the message in green colors, then this receiver block should show all the green colors messages without errors

Source Code Analysis

The most important coding is at the transmitter and at the hashed message. This is shown below

```
% Hashed message for protection
classdef (StrictDefaults)commgobacknwithphy_Tx < matlab.DiscreteEventSystem & ...
                    matlab.system.mixin.Propagates & ...
                    matlab.system.mixin.SampleTime

    properties(Nontunable)
        N = 5 % Window Size
        txDelay = 0.1 % Transmission Delay
        timeout = 2 % Timeout
        numBitsPerPayload=55 % Number of bits per payload
    end

    properties (DiscreteState)
        % Protocol CMS based properties
        slidingWindowSendBase % The starting index of the sliding window
        nextSeqNum % The sequence number of next sending packet

        % Implementation based properties
        newSeqNum % The new sequence number of next incoming packet from upper layer
        txSeqNumState % Buffer the sequence number
        txPayloadState% Buffer the payload
        rtt % Round Trip Time
        isTxing
    end
```

Fig11: MATLAB code CMS protocol

The first thing in designing the hashed message protection is to determine the window size (message size). Here the message size is 5, which come out like a square box. The box here indicates the message is already hashed.

The Tx-delay and timeout as well as the number Bitsperpayload are the important parameters to control the hashed message being enter into the channel. Here the Tx-delay for every message is 0.1 and time allowed is 2 seconds for every message block.

The next coding shows the properties of discrete state. This a discrete protocol that making the block of messages flow in discrete manner.

The real CMS (Cryptography Message Syntax) to protect the message is shown below. As seen in the coding, the methods define the access of message is protected. The protection is using CMS scheme named "getEntityTpesImpl(obj)". By adding the 'Ack", it helps the message being protected when reach the enter point of the network.

```
methods (Access = protected)
    % Define entity types, packet, ack, payload In format of
    % ('Name','Data type');
    function entityTypes = getEntityTypesImpl(obj)
        entityTypes = [obj.entityType('payload') ...
                       obj.entityType('packet', 'Packet') ...
                       obj.entityType('ack', 'Ack')];
    end

    % Define the input/output ports based on the info obtained from
    % getEntityTypesImpl()
    function [inputTypes,outputTypes] = getEntityPortsImpl(~)
        inputTypes = {'payload', 'ack'};
        outputTypes = {'packet'};
    end
```

Figure 12: MATLAB cod CMS protocol protection and Acknowledge

Once the message is being hashed, a new message hashed message is appear and transparent to the user. The coding to do that is shown below:

```
    switch storage
        case 2 % New packet created
            seqNum = obj.newSeqNum;
            obj.newSeqNum = obj.newSeqNum + 1;
            commgobacknwithphy_UI(obj.getCurrentTime(), 'buffer', seqNum);
            % Generate packet: add sequence number, checksum,
            % timestamp to payload
            entity.sys.priority = 300; % lower priority
            entity.data.seqNum = seqNum;
            entity.data.payload = obj.txPayloadState;
            entity.data.timestamp = obj.getCurrentTime();
```

Figure 13: MATLAB code

Note that in the coding, the "obj.newSeqNum = obj.newSeqNum + 1" is the command to create a new hashed message. The hashed symbols are stored in case 2 and inside the "obj.newSeqNum". When open up this file, it will look like below:

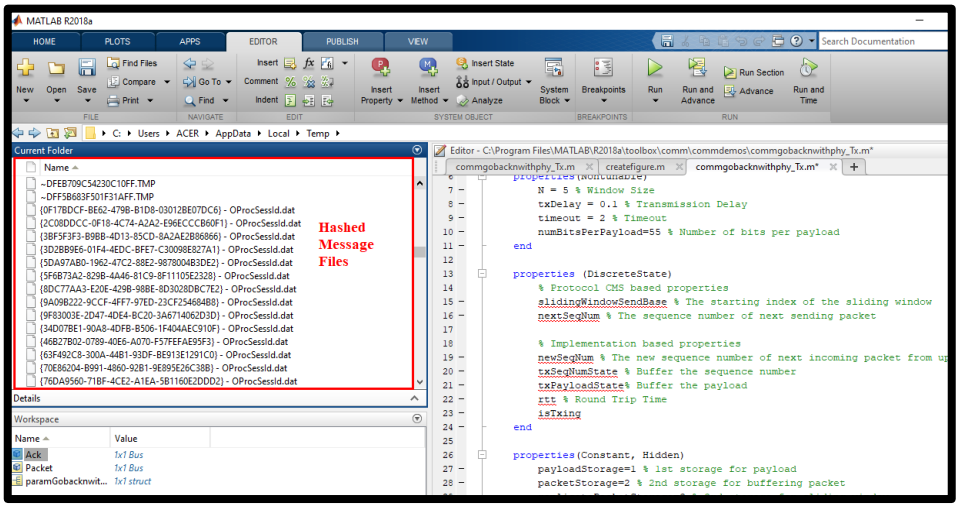

Figure 14: User message being hashed

The hashed messages are using unknown characters combined with numbers to protect the data.

5. Conclusion

From the observation, one can see that the receiver does the process of extracting the messages from the channels are able to avoid bad messages and avoid malicious attack. The simulation is using 100 second, thus all the messages should send or received within 100 seconds. The messages are randomly generated from the MATLAB library file. The messages are then sent to the receiver through two channels. The important of these two channels are to observe the messages collision Due to its orientation in the network or in the channel as in cryptography order, then this makes the malicious have hard time to get the data. Thus, the data is said being protected.

REFERENCES

1. "5.1 Security Considerations for Implementors". Retrieved 2011-07-21. Deriving a key from a password is as specified in [RFC1320] and [FIPS46-2].
2. "What are MD2, MD4, and MD5?". *Public-Key Cryptography Standards (PKCS): PKCS #7: Cryptographic Message Syntax Standard: 3.6 Other Cryptographic Techniques: 3.6.6 What are MD2, MD4, and MD5?. RSA Laboratories*. Retrieved 2011-04-29.
3. ABUGOUKH, T. M., AL SHARABY, A., ELSHAIKH, A. O., JODA, M., MADNI, A., AHMED, I., ... & ABDELRAHMAN, N. (2022). DOES VITAMIN D HAVE A ROLE IN DIABETES?. CUREUS, 14(10).
4. Andress, J. (2014). The basics of information security: understanding the fundamentals of InfoSec Angellotti, E., & Pittas, A. G. (2017). The role of vitamin D in the prevention of type 2 diabetes: to D or not to D?. Endocrinology, 158, 2013-2021.

5. BALLANTYNE, J. C., ET AL. (1998). THE COMPARATIVE EFFECTS OF THORACIC EPIDURAL ANESTHESIA AND GENERAL ANESTHESIA ON PERIOPERATIVE PULMONARY FUNCTION. ANESTHESIA & ANALGESIA, 86, 598-612.
6. BENUMOF, J. L. (1975). LOCAL EFFECTS OF ANESTHETICS ON HYPOXIC PULMONARY VASOCONSTRICTION. ANESTHESIOLOGY, 43, 525-532.
7. BERBUDI, A., RAHMADIKA, N., TJAHJADI, A. I., & RUSLAMI, R. (2020). TYPE 2 DIABETES AND ITS IMPACT ON THE IMMUNE SYSTEM. CURRENT DIABETES REVIEWS, 16(5), 442-449.
8. BIKLE, D. D. (2014). VITAMIN D METABOLISM MECHANISM OF ACTION AND CLINICAL APPLICATIONS. CHEMICAL BIOLOGY, 21, 319-329.
9. BLOMBERG, S., ET AL. (1989). THORACIC EPIDURAL ANESTHESIA AND PULMONARY FUNCTION DURING ONE-LUNG VENTILATION. ANESTHESIA & ANALGESIA, 69, 558-562.
10. BOLDT, J., ET AL. (1996). CARDIORESPIRATORY CHANGES IN PATIENTS UNDERGOING LUNG RESECTIONS: COMPARISON OF THORACIC EPIDURAL AND GENERAL ANESTHESIA. JOURNAL OF CARDIOTHORACIC AND VASCULAR ANESTHESIA, 10, 854-859.
11. BOUILLON, R., MARCOCCI, C., CARMELIET, G., BIKLE, D., WHITE, J. H., DAWSON-HUGHES, B., ET AL. (2019). SKELETAL AND EXTRASKELETAL ACTIONS OF VITAMIN D: CURRENT EVIDENCE AND OUTSTANDING QUESTIONS. ENDOCRINE REVIEWS, 40(4), 1109-1151.
12. BRIMIOULLE, S., ET AL. (1997). SYMPATHETIC MODULATION OF HYPOXIC PULMONARY VASOCONSTRICTION. CARDIOVASCULAR RESEARCH, 34, 384-392.
13. BRODNER, G., ET AL. (1998). A MULTIMODAL APPROACH TO FAST-TRACK CARDIAC ANESTHESIA. ANESTHESIA & ANALGESIA, 86, 228-234.
14. BUTLER, A. E., DARGHAM, S. R., LATIF, A., MOKHTAR, H. R., ROBAY, A., CHIDIAC, O. M., ... & ATKIN, S. L. (2020). ASSOCIATION OF VITAMIN D3 AND ITS METABOLITES IN PATIENTS WITH AND WITHOUT TYPE 2 DIABETES AND THEIR RELATIONSHIP TO DIABETES COMPLICATIONS. THERAPEUTIC ADVANCES IN CHRONIC DISEASE, 11, 2040622320924159.
15. CHENEY, F. W. (1980). THE EFFECT OF CARDIAC OUTPUT ON ARTERIAL OXYGENATION DURING ONE-LUNG VENTILATION. ANESTHESIOLOGY, 52, 496-503.
16. Daemen, J. (1995). Cipher and hash function design strategies based on linear and differential cryptanalysis (Doctoral dissertation, Doctoral Dissertation, March 1995, KU Leuven).
17. GARUTTI, I., ET AL. (1999). ARTERIAL OXYGENATION DURING ONE-LUNG VENTILATION. ANESTHESIA & ANALGESIA, 88, 494-499.
18. HACHENBERG, T., ET AL. (1997). EFFECT OF THORACIC EPIDURAL ANESTHESIA ON PULMONARY GAS EXCHANGE. ACTA ANAESTHESIOLOGICA SCANDINAVICA, 41, 1142-1148.
19. HAWA, M. I., ET AL. (2013). ADULT-ONSET AUTOIMMUNE DIABETES IN EUROPE IS PREVALENT WITH A BROAD CLINICAL PHENOTYPE: ACTION LADA 7. DIABETES CARE, 36, 908-913.
20. HUGHES, M., ET AL. (1992). CONTEXT-SENSITIVE HALF TIME IN MULTIPLE INFUSIONS OF INTRAVENOUS ANESTHETICS. ANESTHESIOLOGY, 76, 334-341.
21. ISHIBE, Y., ET AL. (1996). THE EFFECT OF THORACIC EPIDURAL ANESTHESIA ON HYPOXIC PULMONARY VASOCONSTRICTION. ANESTHESIA & ANALGESIA, 86, 1049-1055.
22. Ja Shau Kok and Hui Ying, "Introduction to Hash Cryptography System", *IEEE Trans on Computer Science and Technology*, Vol. 10, Issue 10, 2016.
23. Jie Liang and Xuejia Lai Department of Computer Science and Engineering Shanghai Jiao Tong University Shanghai 200240.
24. Joux, A. (2004, August). Mult collisions in iterated hash functions. Application to cascaded constructions. In Annual International Cryptology Conference (pp. 306-316). Springer, Berlin, Heidelberg

25. Kasaba, T., et al. (1998). Hemodynamic effects of thoracic epidural anesthesia in patients with coronary artery disease. Canadian Journal of Anesthesia, 45, 1061-1065.
26. Kellow, N. H., et al. (1995). Comparison of the effects of propofol and isoflurane on pulmonary gas exchange during one-lung ventilation. British Journal of Anaesthesia, 75, 578-582.
27. Knekt, P., Laaksonen, M., Mattila, C., Harkanen, T., Marniemi, J., Heliovaara, M., et al. (2008). Serum vitamin D and subsequent occurrence of type 2 diabetes. Epidemiology, 19, 666-671.
28. Larsen, R., et al. (1998). Effects of propofol on hemodynamics and myocardial contractility. Anesthesia, 43, 25-31.
29. Ley, S. H., Hamdy, O., Mohan, V., & Hu, F. B. (2014). Prevention and management of type 2 diabetes: dietary components and nutritional strategies. Lancet, 383, 1999-2007.
30. McNeel, R. L., et al. (1999). Distribution and quantification of myoglobin and other proteins in beef muscles. Journal of Animal Science, 77, 611-621.
31. Pagel, P. S., & Hudetz, J. A. (1998). Desflurane and isoflurane. Anesthesia & Analgesia, 87, 800-807.
32. R. P. Arya, "Design and Analysis of a New Hash Algorithm with Key Integration," vol. 81, no. November, pp. 33–38, 2013.
33. Ronald L. Rivest Massachusetts Institute of Technology Laboratory for Computer Science NE43-324 545 Technology Square Cambridge
34. S. Al-Kuwari. Engineering Aspects of Hash Functions. In International Conference on Security and Management (SAM '11), 2011.
35. Sandra.L, Maggie. C and Huang Xia, "Advanced Hash Cryptography Encryption", International *Journal on Data Communications*, Vol. 99, Issue 17, pp. 8 – 23, 2017.
36. SPIES, C., BREUER, G., MAASSEN, V., & HANNAPPEL, E. (1991). COMPARISON OF ENFLURANE AND ISOFLURANE EFFECTS ON HEPATOSPLANCHNIC CIRCULATION. DER ANAESTHESIST, 40, 14-18.
37. STEEGERS, P. A., ET AL. (1990). PROPOFOL AND ALFENTANIL DURING ONE-LUNG VENTILATION. JOURNAL OF CARDIOTHORACIC ANESTHESIA, 4, 194-199.
38. TANAKA, K., ET AL. (1991). LOW-DOSE THORACIC EPIDURAL ANESTHESIA IN COMBINATION WITH GENERAL ANESTHESIA FOR THORACOTOMY. REGIONAL ANESTHESIA, 16, 318-321.
39. VAN KEER, L., ET AL. (1989). PROPOFOL AND VENTILATION. JOURNAL OF CLINICAL ANESTHESIA, 1, 284-288.
40. VANSAL, S. S., ET AL. (1999). DIRECT EFFECT OF EPHEDRINE ISOMERS ON THE SYSTEMIC VASCULATURE. BIOCHEMICAL PHARMACOLOGY, 58, 807-810.