International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 5, October 2021, pp. 4413~4422
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i5.pp4413-4422

Software engineering based secured E-payment system

Muayad Sadik Croock[1], Rawan Ali Taaban[2]
[1]Control and Systems Engineering Department, University of Technology-Iraq, Baghdad, Iraq
[2]Imam Ja'afar Al-Sadiq University, Baghdad, Iraq

Article Info

Article history:

Received Jan 8, 2021
Revised Mar 9, 2021
Accepted Mar 19, 2021

Keywords:

Back-propagation
E-payment
Neural network
NIST
Software engineering

ABSTRACT

Nowadays, the E-payment systems have been considered to be the safe way of money transfer in most of modern institutes and companies. Moreover, the security is important side of these systems to ensure that the money transfer is done safely. Software engineering techniques are used for guaranteeing the applying of security and privacy of such systems. In this paper, a secure E-payment system is proposed based on software engineering model and neural network technology. This system uses different proposed algorithms for applying authentication to the devices of users as mobile application. They are used to control the key management in the system. It uses the neural network back-propagation method for ensuring the security of generated keys that have sufficient random levels. The proposed system is tested over numerous cases and the obtained results show an efficient performance in terms of security and money transfer. Moreover, the generated keys are tested according to NIST standards.

Corresponding Author:

Muayad Sadik Croock
Department of Control and Systems Engineering
University of Technology
Baghdad, Iraq
Email: muayad.s.croock@uotechnology.edu.iq

1. INTRODUCTION

After the revolution of technologies, the modern information and management systems adopt the electronic copies of different applications. This is to reduce the labor efforts and speed up the process time. On the other side, theses electronic versions of systems consider the security as a main part to ensure the safety of implementation. The mentioned sides are designed using software engineering model to ensure reliability, flexibility, and extendibility [1]-[7].

Different researchers have considered the area of electronic (E)-systems with distinct subjects, such as security, management, and data exchange. In [8], the authors proposed E-wallet systems with high security using smart solid state drive (SSD). It could be used in different web applications. In [9], author presented a comprehensive analysis for threats that affect the E-wallet systems. This analysis tackled all possible types of threats and their effects on the work flow of money transfer systems. In [10], authors considered different types of security issues that face E-wallet systems and money transfer process. They also compared distinct kinds of E-wallet systems to conclude the best model that can be adopted. In [11], authors adopted a mobile application that used different E-wallet systems and managed the money transfer between them. The author of [12] presented a study on the importance of using security in the information system of Mu'th university. This study tool care of answering security questions, used for proposing a complete security system. In [13], risk and security consideration were introduced for information systems. They adopted the differences between attacks and the work procedure of them. The authors of [14] proposed information system security that tackled numerous types of threats that can attack these systems. While in [15], an information security

system was proposed for a bank. The security management was adopted in this system to adapt the change in the work direction. In [16], the authors presented an assessment method for expected risks in a banking system. The idea behind was to identify the risks that can face banking systems including security issues. In [17], a cloud computing was used for proposing an information security system in banking system. This system tackled the security threats and attacks to avoid data losing. The authors of [18]-[20] presented a security system for information based on the concepts of software engineering to ensure the completion of product. In these systems, different techniques were used, in which the attacks are tackled well in the information systems for banks and other institutes.

2. THE PROPOSED SYSTEM

As mentioned earlier, the proposed system is a secure money transfer application that is designed based on software engineering concepts and neural network. Generally, the proposed system uses a simple neural network that applied into a secured money transferring application between two parties. As shown in Figure 1, a user can download and install the proposed mobile application. Using this application, the user creates a new account using personal information (full name, phone number and password). Phone authentication has been adopted through the register process. Thus, the user can receive an SMS that contains a code to verify the phone number and activate the created account. In addition, a master key is generated and encrypted, explained in section 2.3, for the user to be securely used inside the mobile application when identity verification is needed.

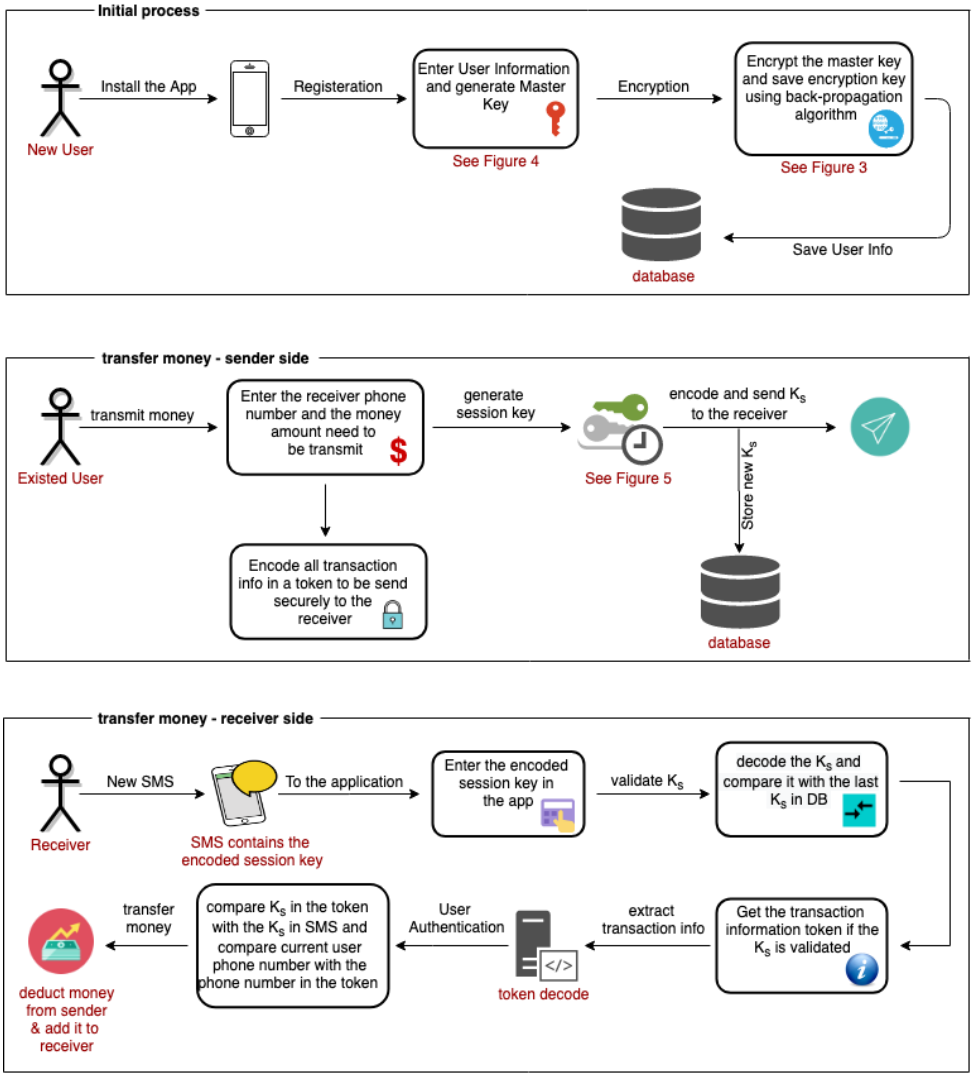


Figure 1. General block diagram of the proposed system

As depicted in Figure 1, transfer money process takes place when a user needs to transmit money to other user using the proposed application. In this case, the sender is prompted to enter the receiver phone number and the amount of money. After that, the system creates a new interim key, named session key K_s. An encrypted token contains all the transaction information in an encoded form (the K_s, receiver phone number, transaction date and the money amount).

The K_s and the token are used in the two-level authentication steps. The session key K_s is encoded and stored in the database and sent to the receiver as SMS. The receiver enters this key in the mobile application to be validated by the system. At the end of this step, the first level authentication is completed. In the second level of authentication, the token generated by the sender is activated to be usable by the receiver. At the receiver side, the token is decoded to extract the secured transaction information. The K_s, included in this token, is compared to the last generated K_s besides comparing the phone number in the token with the intended receiver phone number. If these two operations completed without any problem, the second authentication level is successfully completed. After that, the money can now be transferred from the sender to the receiver. To simplify the understanding of the full process and the used algorithms, the proposed system is divided into five parts as follows:

2.1. Software engineering model

It is important to note that the software engineering has a wide area of employment in the field of security and information systems. Figure 2 shows the designed software engineering model for the proposed system as a life-cycle. It is divided into four main phases: requirements, design, implementation and testing [21]-[23]. In the requirement phase, different points have been achieved including data collection, user information, and payment information. While in the design phase, the model contains designing the backpropagation, master and session key generation, and light-weight protocol for money transfer algorithms as well as designing the mobile application with related interfaces. At the other hand, the phase of implementation involves the implementation of the designed algorithms. In the testing phase, the proposed system is tested in terms of money transfer, key strong and neural implementation.

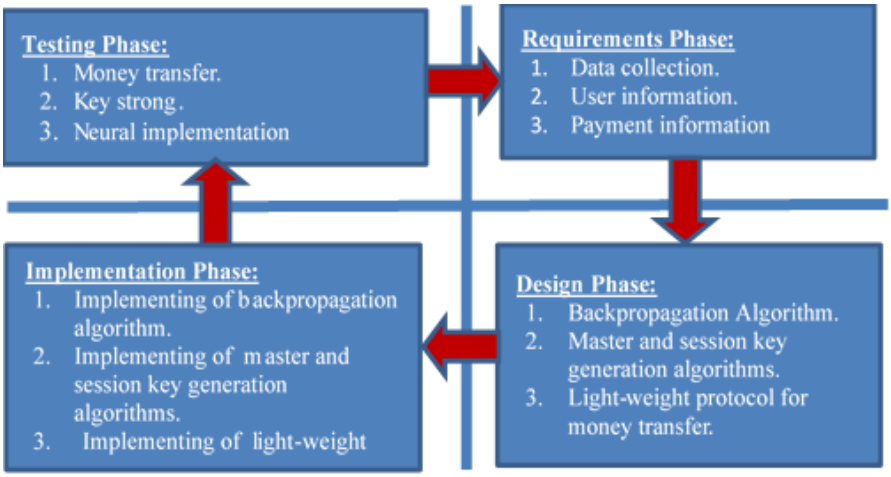


Figure 2. Designed software engineering model

2.2. Employed backpropagation algorithm

Back-propagation neural network consists of three layers: an input vector X, a hidden layer H and an output layer Y in addition to the weight's coefficients W. All layers consist of variable number of neurons that determined according to a proposed algorithm. The used structure of back-propagation is very simple as it will be explained nextly. The different stages of backpropagation neural network algorithm can be stated as [24]:

a. Initialization

1) Determination of neural network parameters: j is the number of hidden layer units, i is the input layer neuron's number for each hidden layer unit and y which is the output layer neuron's number.
2) Initialize the input and output neurons' values.
3) The network weights have been initialized in this step to small random numbers (input-hidden and hidden-output weights).

b. Feedforward stage

4) Sums weighted input and apply activation function to compute output of hidden layer.

$$H_j = ActFun\,(\sum_i X_i\ W_{ij}) \quad (1)$$

Where

H_j : The actual output of hidden neuron j for input signals x.

X_i : Input signal of input neuron (i).

W_{ij} : Weight between input neuron i and hidden neuron j

$ActFun$: The activation function (Sigmoid function).

5) Sums weighted output of hidden layer and apply activation function to compute output of output layer.

$$Y_k = ActFun\,(\sum_j H_j\ W_{jk}) \quad (2)$$

Where

Y_k : The actual output of output neuron k.

W_{jk} : Weight between hidden neuron j and output neuron k.

c. Backpropagation stage:

6) Computes back propagation error.

$$\delta_k = (D_k - Y_k)\, f'\,(\sum_j H_j\ W_{jk}) \quad (3)$$

Where

f' : The derivative of the activation function.

D_k : The desired of output neuron k.

7) Calculates weight correction term.

$$\Delta w_{jk} = \eta\, H_j\, \delta_k \quad (4)$$

Where, η is the learning rate.

8) Sums delta input of each hidden neuron and calculate the error term.

$$\delta_j = \sum_k \delta_k\, W_{jk}\ f'\,(\sum_i X_i\ W_{ij}) \quad (5)$$

9) Calculates weight correction term.

$$\Delta w_{ij} = \eta\, X_i\, \delta_j \quad (6)$$

Updates weights.

$$W_{jk}\ (\text{new}) = W_{jk}\ (\text{old}) + \Delta w_{jk} \quad (7)$$

$$W_{ij}\ (\text{new}) = W_{ij}\ (\text{old}) + \Delta w_{ij} \quad (8)$$

10) Repeat step (4) for a given number of training iterations.

2.3. Master-key generation algorithm

As aforementioned, a master key is generated to be used wherever user identity verification is required in the proposed mobile application. It is generated automatically depending on the user information and encrypted using back-propagation neural network algorithm explained earlier. The algorithm, used for generating and encrypting the master key, passes through number of steps as illustrated below:

a. *K_m generation:*

1) After user registration, a unique ID has been generated for the user.
2) The first letter of the first, second and third user name are extracted.
3) Convert these three letters into the equivalent ASCII numerical values.

4) Determine the master key before encryption.

K_m (before encryption) = (1st letter ASCII**.** 2nd letter ASCII**.** 3rd letter ASCII**.** User ID)

b. *K_m encryption*

1) Convert the 3 ASCII values in step (3) to 8-bit binary numbers.
2) Concatenate the 3 binary numbers from last step with user ID after convert it to 8-bit binary number.
3) Using the back-propagation explained in 5.1, a neural network is trained for 1000 iterations by using the 32-bit binary number in step (5) as an Input layer neurons values. 26 neurons are used in the output layer to extract 26-bit desired output.
4) The desired output (encrypted K_m) and the last iteration updated weights are stored to be used in verification process.

The verification process for the code mentioned in step (4) of master key generation can be summarized in the following steps:

c. *K_m verification:*

1) Split the entered code by the user into 4 parts: the three letters ASCII and the user ID.
2) Convert each part to the equivalent 8-bit binary number.
3) Concatenate all numbers together.
   Resulted code = (binary of first ASCII**.** binary of first ASCII**.** binary of first ASCII**.** binary of user ID)
4) Restore the saved master key and the updated weights in step (8) of K_m encryption.
5) Using only the feedforward part of back-propagation algorithm explained in 2.1. The 32-bit binary number obtained from step (3) is used as an input layer and the updated weights restored in step (4) are used as input-hidden and hidden-output weights values.
6) The obtained output from feedforward process is compared with the restored K_m in step (4).
7) If the comparison result is true, then the K_m is verified and is belong to that user.

2.4. Session-key generation algorithm

At the beginning of any money transferring process, a temporary key is automatically generated and being used between the two parties (sender and receiver). The main purpose of the session key (K_s) is to verify the receiver identity and to ensure that the money can transfer securely against any attacker tries to obtain that key and hack the transaction. K_s is updated regularly for each session. In addition, previous value of K_s is used when updating K_s to new value in a new session. Thus, the K_s value is cumulative and thus the new K_s depends on the previous one. It becomes more complicated for the attacker to keep up with the session key updates with every new session. At the first session, an initial value of session key (K_{is}) is randomly generated and be used for generating the next K_s. To simplify the process of generating K_s, an algorithm steps are illustrated as:

- Restoring the last used session key K_s.
- Using the back-propagation explained in section 5.1, a neural network is trained for 1000 iterations. The 24-bit binary number of last K_s is used as an Input layer neurons values and a randomly generated 24-bit binary number is used as output layer neurons values.
- The last updated weights between the input and hidden layers are extracted.
- Each number in the input-hidden weight values extracted in last step is rounded to the nearest integer number.
- Enter the resulted code through NIST frequency and serial tests.
- If the resulted key pass in randomness tests, use it as the new session key [25].
- Store the new generated session key in database to be used for next session.

2.5. Light-weight protocol for money transferring

When transfer money process takes a place over a network, all security considerations must be considered. Therefore, a light-weight mechanism has been proposed in this work. In this mechanism, while transferring process, every user has 2 keys: Master key K_m and Session key K_s. A simple interface is included in the designed application. Besides, a special activity page is particularized for transfer money. Thus, when a user needs to transfer money to another user all he/she needs is to enter the receiver phone number and the amount of money want to be transferred. All the rest of process steps are relied on the system itself. Figure 3 (see in appendix), the whole process begins with generating a new session key (explained in section 2.3) which is specified and dedicated only for the current session. At the end of this session the used K_s becomes idle and the only purpose of keeping it in database is to use it in the next session for the generation of new K_s. The K_s passes through number of steps to be ready for ensuring receiver authentication.

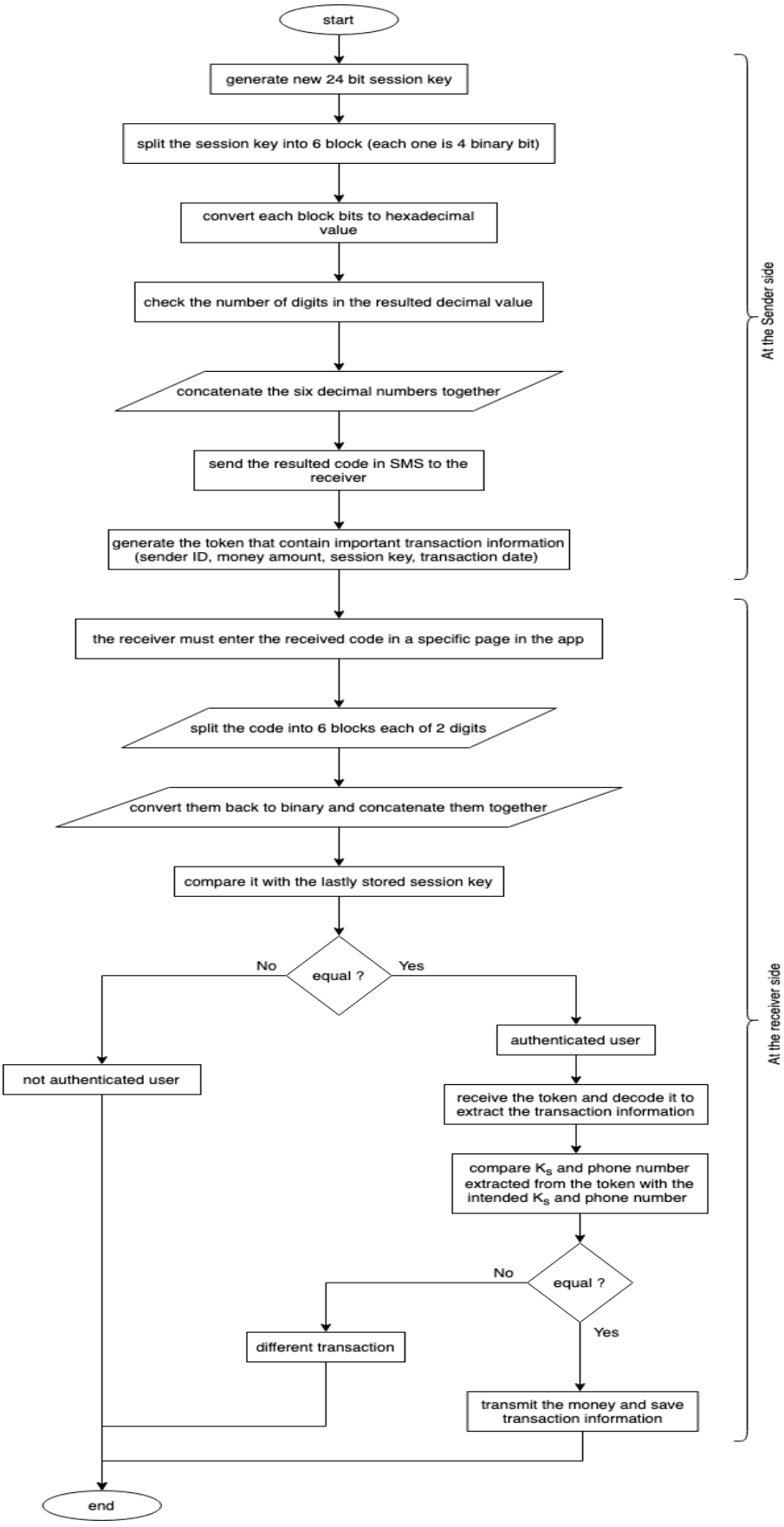


Figure 3. The whole money transferring system

First of all, the K_s is split into six blocks of 4-bit binary followed by converting each block separately to hexadecimal value. By concatenating these six resulted hexadecimal values, a new code is obtained to be sent in SMS to the receiver phone number to be used for further processes at the receiver side. In fact, the process at the sender side is not limited to only generating, processing handling and sending the K_s however, it exceeds that to hashing the transaction information and pass it to the receiver securely over network. Tokenization mechanism has been used for that purpose, which is the process of sending sensitive data via a mobile application call or batch file to a tokenization provider that then replaces that data with non-sensitive placeholders called tokens. Simply, it swaps out sensitive data-typically payment card or bank account numbers-with a randomized number in the same format but with no intrinsic value of its own.

It is worth mentioning that HS256 (HMAC with SHA256) algorithm has been used in obtaining token from our sensitive transaction information. Thus, the K_s, receiver phone number, transaction date and the money amount are transformed into a secured token. The receiver side process begins when the SMS that includes the encoded K_s is received. A special dialog box is appeared in the proposed mobile application to any user have a money to receive. This can prompt him/her to enter the received code in the SMS. After that, the system decodes the entered code to extract the original K_s. Decoding process is done by splitting the code back into six blocks and converting each block from hexadecimal to binary. Then, the algorithm concatenates all of them together in sequent manner, in which the original K_s can be obtained. To ensure receiver authenticity and dealing with the intended user, the lastly obtained K_s is compared with the stored K_s. If they are identical, the first level of authentication is successfully completed. The second level of authentication begins with activating the token (build at the sender side) at the receiver side. Thus, the token is decoded and the transaction information can be obtained which are: K_s, receiver phone number, transaction date and the money amount. The next step is to ensure the session activity besides, another level of authentication, done by comparing the phone number in the token with the user phone number and the K_s with the K_s in the SMS. After that, the final goal is achieved which is transferring the money from sender to receiver.

3. EXPERIMENTAL RESULTS

In order to test the operation and performance of the proposed system, different experiments have been considered. For easing the reading flow, the obtained results can be divided into three parts as follow:

3.1. Back-propagation results

The proposed ANN is very simple and absolutely works with no errors in the output and this is proved by testing it for different numbers and every time it shows very code results. Figure 4 shows a brief description about the final results of testing the proposed back-propagation algorithm separately from the application process. The used network includes 24-bit input matrix (the orange rectangle) that represents the input neurons layer. It also involves 26-bit desired output matrix (the blue rectangle) which considered to be the output layer neurons values. the input-hidden and hidden-output weights (the green rectangles) matrices are randomly generated values between 1 and -1. After 1000 training iterations and by using learning rate of about 1.0, the actual output (the red rectangle) obtained at the last training stage is almost the same as the desired output (the blue rectangle). The yellow rectangle represents the output matrix after rounded its values to the nearest integer number which is the identical of the desired output.

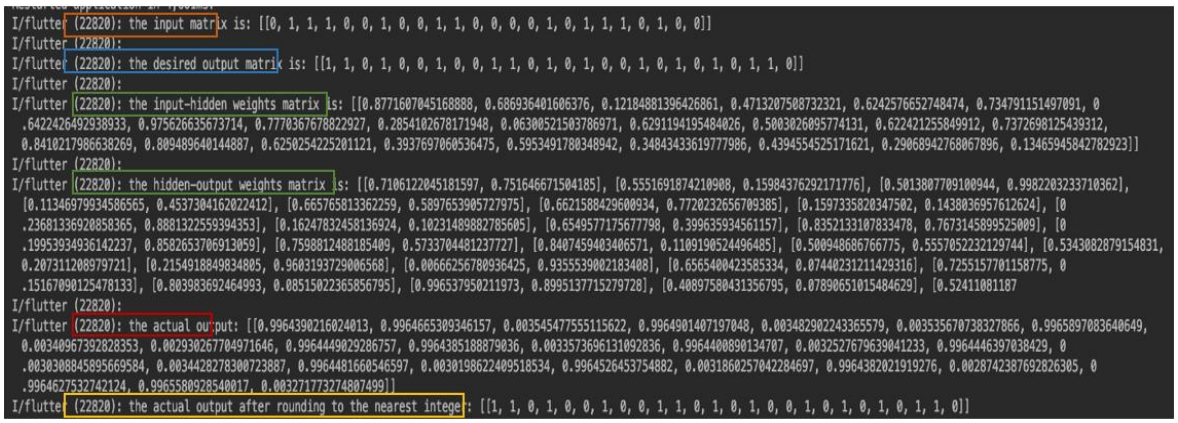
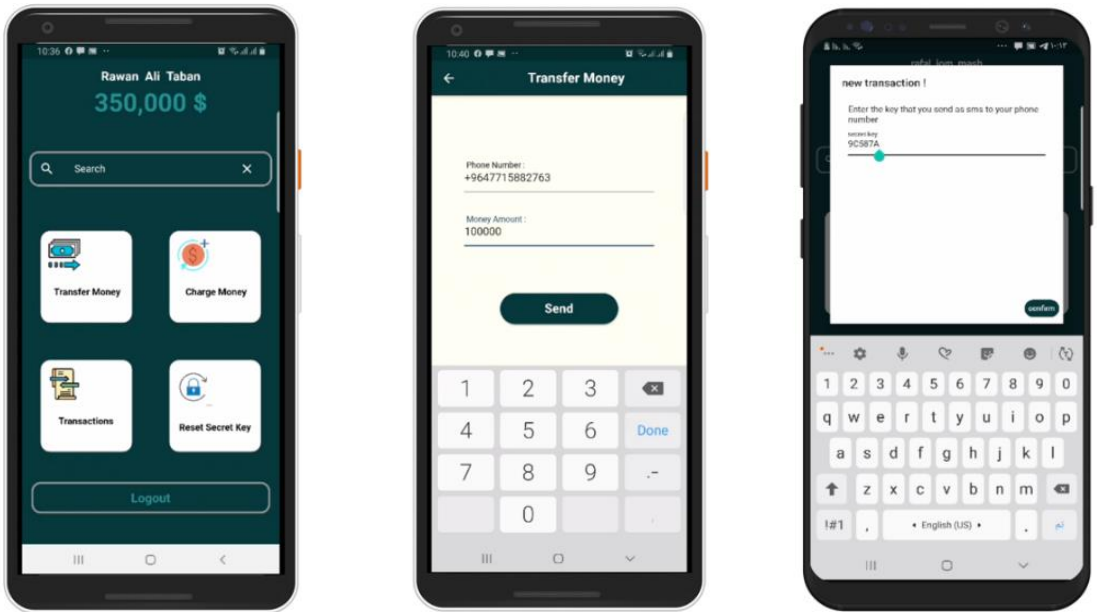


Figure 4. The back-propagation process

3.2. Registration and master-key generation

As mentioned before, the master key is generated at the registration stage to be used for user identity verification. The registration process simply prompts the user to enter the full name, password and the phone number. Phone number authentication has been used in this stage to verify user phone number. At the other hand, the master key (K_m) is generated after registration process using of full user name. The generated master key length, before encryption, is more than 9 digits whereas the encrypted version of it is 26-bit length. For instance, the full name of the, user registered in Figure 5, is "Rawan Ali Taban", by taking the ASCII of first letters of each name (R=114, A=97, T=116) and by concatenating them with each other in addition to the user unique ID (Ex:55 for this user) in database, the resulted K_m is (1149711655). Figure 6 shows the structure of database after the user successfully registered in the application. The master key included in Figure 5 is the encrypted version of (1149711655) after apply backpropagation algorithm.



(a) (b) (c)

Figure 5. Money transmitting activities; (a) and (b) at sender side and (c) at receiver side

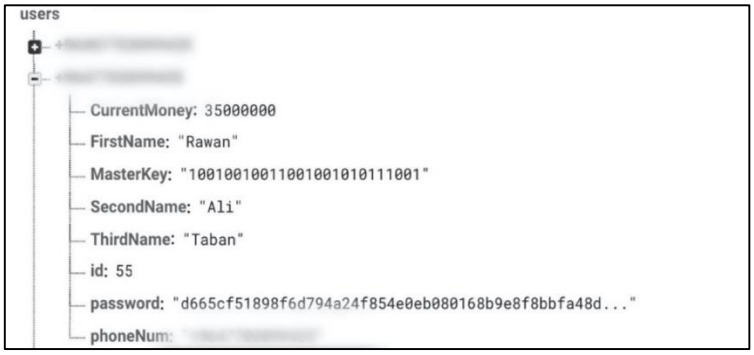


Figure 6. The database structure after registration

3.3. Money transmitting results

We have done a careful, secure and reliable technique for sensitive data exchange like money transfer. Depending on both master and session keys in the transaction provides better security as it increases the difficulty for the attackers to derive both keys. Moreover, every transaction is done with a new session key different from previous keys and generated with the help of last session key. Therefore, if the attacker gets the session key one time, he/she cannot be able to use it for next sessions.

Figure 5(a) shows the home page of the user that contains different activities like transfer money, charge money and current transactions. The most important page is the "transfer money" page. The page in Figure 5(b) is a simple designed only prompt the user to enter the receiver phone number and the amount of money. Figure 5(c), is a screenshot at the receiver side when the sender begins a transaction with him/her which prompt the user to enter the transaction K_s received in SMS.

3.4. NIST tests

All Session keys, generated in the application, are passed through NIST frequency and serial tests. This is done by checking the randomness of the key. If the key is valid, then it passes the test. Otherwise, it is failed and not ready for use. In this case, another key is generated and the whole test process is repeated [25]. Figure 7 shows the results of testing two keys. One of them is pass in frequency and failed in serial test and the other is failed in both tests. This is because the obtained test value is larger than the specified test threshold.

```
I/flutter (21710): the first code: 0101001001101010010101011010101010101011
I/flutter (21710): the frequency test: PASS
I/flutter (21710): the serial test: FAILD
I/flutter (21710): ------------------------------------------------------------
I/flutter (21710): the second code: 1111111111111111111101011011111111011110
I/flutter (21710): the frequency test: FALID
I/flutter (21710): the serial test: FAILD
```

Figure 7. NIST tests results

4. CONCLUSION

This paper proposed a secured money transfer system based on software engineering technology and neural network. The system was presented in a mobile application and a server side. Software engineering model was adopted to guarantee the reliability, availability and authentication phases for the proposed system. Neural network was used in key generation for increasing the security of the system. The obtained experiment results showed the high efficiency of the proposed system in terms of security and money transfer.

REFERENCES

[1] François Chollet, "Deep Learning with Python," Manning, 2018.

[2] M. Thoma, "Analysis and Optimization of Convolutional Neural Network Architectures," *arXiv preprint arXiv:1707.09725*, 2017.

[3] J. Ayuba, H. Safwana, and Y. Abdulazeez, and D. Ma'azu, "Software Development of Integrated Wireless Sensor Networks For RealTime Monitoring of Oil and Gas Flow Rate Metering Infrastructure," *Journal of Information Technology and Software Engineering*, vol. 8, no. 2, pp. 1-10, 2018, doi: 10.4172/2165-7866.1000233.

[4] A. Munem, and M. Croock, "Smart Traffic Light Control System for Emergency Ambulance," *Int. Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, vol. 5, no. 8, pp. 2247-2255, 2016.

[5] L. D. Nguyen, D. Lin, Z. Lin, J. Cao, "Deep CNNs for Microscopic Image Classification by Exploiting Transfer Learning and Feature Concatenation," *IEEE International Symposium on Circuitsand Systems (ISCAS)*, pp. 1–5, 2018, doi: 10.1109/ISCAS.2018.8351550.

[6] J. Mas and C. Matue, "Introduction to Web Application Development," Eureca Media, 2015.

[7] M. S. Croock, S. Al-Qaraawi and R. A. Taban, "Gaze Direction based Mobile Application for Quadriplegia Wheelchair Control System," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, pp. 415-426, 2018.

[8] I. Y. Jung, and G. J. Jang, "A Secure and Reliable e-Wallet using a Smart SSD," *Life Science Journal*, vol. 11, no. 7, pp. 117-121, 2014.

[9] M. P. Bosamia, "Mobile Wallet Payments Recent Potential Threats and Vulnerabilities with its possible security Measures," CMPICA, 2017.

[10] G. Kanimozhi, and K.S. Kamatchi, "Security Aspects of Mobile Based E Wallet," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 6, pp. 1223-1228, 2017.

[11] D. Kumar, and U. Sharma, "Design E-Wallet as a Centralized E-wallet," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 2, pp. 2249-8958, 2019, doi: 10.35940/ijeat.B2684.129219.

[12] A. J. Alnawaiseh, "Security Information System of The Computer Center In Mu'tah University," *European Scientific Journal*, vol. 10, no. 27, P.P. 292-302, 2014.

[13] K. Subrahmanyam, M. Haritha, V. Tejaswini, Ch. Balaram, and C. Dheeraj, "Information Security and Risk Management for Banking System," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 10, no. 3, pp. 171-176, 2014.
[14] S. Al-Dhahri, M. Al-Sarti, and A. Abdul Aziz, "Information Security Management System," *International Journal of Computer Applications*, vol. 158, no. 7, pp. 29-33, 2017, doi: 10.5120/ijca2017912851.
[15] Xiaofei Tang, "Development of Commercial Bank Information System Based on Risk Management Research," *4th International Conference on Education, Management and Computing Technology*, vol. 158, no. 7, pp. 710-714, 2017, doi: 10.2991/icemct-17.2017.152.
[16] S. Shokouhyar, F. Panahifar, A. Karimisefat and M. Nezafatbakhsh, "An information system risk assessment model: a case study in online banking system," *International Journal Electronic Security and Digital Forensics*, vol. 10, no. 1, pp. 39-59, 2018, doi: 10.1504/IJESDF.2018.089205.
[17] A. Mahalle, J. Yong, X. Tao, and J. Shen, "Data Privacy and System Security for Banking and Financial Services Industry based on Cloud Computing Infrastructure," *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, pp. 407-413, 2018, doi: 10.1109/CSCWD.2018.8465318.
[18] P. Giorgini1 and F. M. J. Mylopoulos, "Requirement Engineering Meets Security: A Case Study on Modelling Secure Electronic Transactions by VISA and Mastercard," *22nd International Conference on Conceptual Modeling*, Springer, 2003, pp. 263-276.
[19] P. T. Devanbu, and S. Stubblebine, "Software Engineering for Security: a Roadmap," *Conference on The Future of Software Engineering*, 2000, pp. 227–239, doi: 10.1145/336512.336559.
[20] S. U. Rehman, and V. Gruhn, "An Effective Security Requirements Engineering Framework for Cyber-Physical Systems," *Technologies*, vol. 6, no. 3, 2018, Art. No. 63, doi: 10.3390/technologies6030065.
[21] Ian Sommerville, "Software engineering," 10th Edition, Pearson Education, Inc, 2017.
[22] M. Kuutilaa, M. Mantylaa, U. Farooqa, and M. Claes, "Time Pressure in Software Engineering: A Systematic Review," *Elsevier*, vol. 121, 2020, doi: 10.1016/j.infsof.2020.106257.
[23] V. V. Phoha and S. Phoha, "Situation-Aware Software Engineering for Sensor Networks," *2nd International Conf. on Communication Systems Software and Middleware*, pp. 1-7, 2007, doi: 10.1109/COMSWA.2007.382426.
[24] C. Charu Aggarwal, "Neural Networks and Deep Learning," Springer, 1st Edition, 2018.
[25] M. S. Croock, Z. A. Hassan and S. D. Khuder, "Adaptive key generation algorithm based on software engineering methodology," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 589-595, 2021, doi: 10.11591/ijece.v11i1.pp589-595.