International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 6, December 2020, pp. 5844~5852
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i6.pp5844-5852

Software engineering based self-checking process for cyber security system in VANET

Muntadher Naeem Yasir[1], Muayad Sadik Croock[2]
[1]Department of Computer Science, Iraqi Commission for Computers and Informatics (ICCI), Informatics Institute for Postgraduate Studies, Iraq
[2]Department Computer Engineering, University of Technology, Iraq

Article Info

Article history:

Received Feb 22, 2020
Revised May 4, 2020
Accepted May 17, 2020

Keywords:

Cyber security
NIST
Self-checking process
Software engineering
VANET

ABSTRACT

Newly, the cyber security of vehicle ad hoc network (VANET) includes two practicable: vehicle to vehicle (V2V) and Vehicle to Infrastructure (V2I) that have been considered due to importance. It has become possible to keep pace with the development in the world. The people safety is a priority in the development of technology in general and particular in of VANET for police vehicles. In this paper, we propose a software engineering based self-checking process to ensure the high redundancy of the generated keys. These keys are used in underlying cyber security system for VANET. The proposed self-checking process emploies a set of NIST tests including frequency, block and runs as a threshold for accepting the generated keys. The introduced cyber security system includes three levels: Firstly, the registration phase that asks vehicles to register in the system, in which the network excludes the unregistered ones. In this phase, the proposed software engineeringbased self-checking process is adopted. Secondly, the authentication phase that checks of the vehicles after the registration phase. Thirdly, the proposed system that is able to detect the DOS attack. The obtained results show the efficient performance of the proposed system in managing the security of the VANET network. The self-checking process increased the randomness of the generated keys, in which the security factor is increased.

Corresponding Author:

Muntadher Naeem Yasir,
Department of Computer Science,
Iraqi Commission for Computers and Informatics (ICCI),
Informatic Institute for Postgraduate Studies,
Al-nidal Street, Baghdad, Iraq
Email: muntadher.naeem@yahoo.com; 120102@uotechnology.edu.iq

1. INTRODUCTION

The VANET has a significant influence in our modern era towards development and keeping pace with the developed countries that operate according to this type of network. VANETs operate on one of two nodes: either OBUs or RSUs. OBUs are devices onboard mobile vehicles. RSUs referes that the vehicles are connected to each other as well as to the server and work as the router inside the network [1, 2]. It is through the use of dedicated short range communication (DSRC) devices [3-7].

Different studies and research work in the field of security in VANET had presented to tackle the raised problems in terms of the self-checking process for keys. In [8], Researchers suggested an algorithm (ECDSA), where this algorithm mathematically derived from the digital signature algorithm. This algorithm uses a pair of different keys. The keys consist of a primary key is the public key and the second key is the private key. The primary key created based on multiples of the secondary key, where it is considered the random multiple of the primary point. The two keys used in the authentication process within the proposed

system. The researchers work problem is the reliability in building the primary key if a problem occurs in the secondary key that decreases the randomness of the primary key. In [9], the authors proposed an (ECMV) technology. This technology depends on the PKI infrastructure. The action of the mechanism is to give a short-term certificate for each vehicle, as it updated through the vehicles passage next each RSU. This mechanism works to generate the key for each digital certificate, which increases the load on the network.

In [10], the authors worked on a CMAP proposal to discover data sent from harmful compounds in VANETs. The mechanism of work of this protocol was to reduce the costs of Computational vehicles to verify received messages. Nevertheless, here the costs increased with the vehicles number increasing, because that the work of the protocol depends on the density of the presence of the vehicles. In [11], TESLA protocol uses similar keys instead of using different keys. According to the study, researchers find that the using of similar keys is much faster than digital signatures. This protocol avoided the denial of service attacks. Therefore, it was difficult to verify the lack of intrusion on the network data because the approved keys are the same. The problem here is in the case of knowing the key without making sure of increasing the randomness of the keys. In [12], the researcher used a method based on the groups signature for increased network security. Its mechanism of action is the association of a group's primary key with several private keys for another group. Here the attacker can easily find the message sent through the researcher's lack of interest in increasing the randomness of the keys which may lead to gaps in the network.

In [13], the authors proposed a basic group of key management system (CRT). The mechanism of the action of this protocol is to reduce the number of broadcast messages to allow the side road units to get the key. Yet, the researcher worked to increase the complexity of the primary server accounts without emphasizing the increasing complexity of the randomness of the keys. In [14], the authors suggested a system with a specific mechanism, which is to encrypt the public key to create an imaginary name. Through this name, exotic vehicles audited on the VANET network by obtaining a real combined identity. Whatever distinguishes the researchers work here is the ability of the system used to renew for use again which results in addition to improving security. The problem with researchers' work is the increase in the cost of storage.

In [15], The researchers suggested VANET's lightweight binary system to ensure the confidentiality of the network's work. The system used a double password based on the proposed authentication mechanism for the system. Nevertheless, network security was mostly dependent on the key given by CA. In [16], the authors worked on proposing a work technique called (3PAKE). This technology dealt with security attacks that cause increased cost and separation of service or request for unsafe service as well as the failure of the audit. Thus, they did not address the analysis of the rest of the types of attacks that fall within the work of the same basic framework for service interruptions within the network. In [17], the authors suggested a mechanism for maintaining the privacy of VANETs work. This mechanism was conditional upon the signature of the system efficiency increase. Consequently, the disadvantage of this system was that it did not suggest ways to increase the randomness of the encryption for the signature to increase efficiency.

As a result, the literary study of some researchers associated with the use of randomness of the key in the VANETs. The proposed cyber security system differs in terms of employing the software engineering based self-checking process, construction, phases and handling of DoS attacks. The proposed system supports two different types of communication, police vehicle to police vehicle (P_V2P_V) and police vehicle to infrastructure (P_V2I). Our work in the proposed protocol focused on the use of the self-check process during the registration phase. The self-check process uses NIST tests as thresholds to gurantee the validity of the generated keys in terms of rendomnass [18, 19].

2. PROPOSED SYSTEM SCHEMA

To establish a vehicular ad hoc network of police vehicles, we need a fast and secure system to complete the communication process. In Figure 1, we clarify the work of the system through the included chart that is proposed to indicate the work of the three phases of the system: Registration, Authentication and Detection of attacks. Each phase has a different work mechanism, but between all the phases there is a close association that depends on the results of the previous phase. In addition, the proposed system focuses on the use of a set of NIST tests in the registration phase specifically inside the server [18-22]. These tests work for ensuring the randomness for the key given to the vehicles after it is generated inside the server based on software engineering process (self-checking process). The aim of the proposed system is the urgent need to preserve the security and confidentiality of the data exchanged between the vehicles. It is also used to address the attacks that have become more prevalent in specified time that is mentioned in particular the DoS attacks that were designed to separate the vehicle from service.

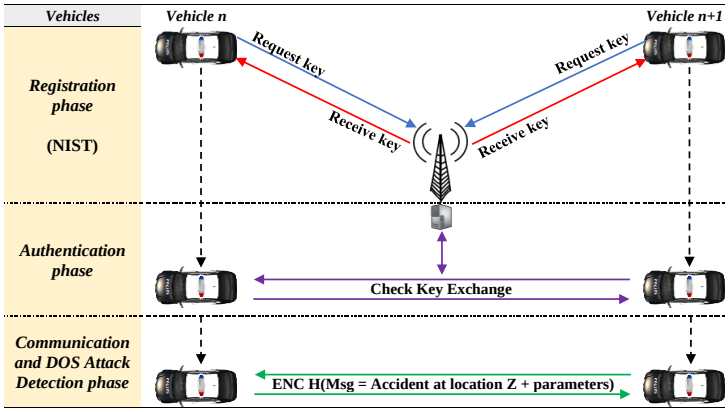

Figure 1. Proposed system schema

3. GUI OF THE PROPOSED SYSTEM

The discuss of the clarification regarding the proposed graphical iser interface (GUI) model, as shown in Figure 2. Both of C # and SQLServer were used in designing, building and programming the proposed system for operating a vehicular ad hoc network. We have worked on adding a group of vehicles, including what represents the police vehicles (number of vehicles: 11), vehicles attacking (number of vehicles: 3) and natural vehicles (number of vehicles: 6). The proposed model contains several parts: including what represents the environment of vehicle movement, the infrastructure that includes the server as well as the list of events that show us the results of the proposed system in all phases from the registration phase to the communication phase and detection of the attack.

Figure 2. GUI of the proposed system

4. PROPOSED SYSTEM ALGORITHM

The algorithm of Figure 3 shows the work of the proposed system to ward off DoS attacks. The system contains more than one phase: which is registration, authentication, data transmission and attack detection. The registration phase between the vehicle and the server is to send a request as well as receive a key for each vehicle in the network. The authentication phase between two vehicles or between the vehicle and the server by exchanging the keys between the vehicles and also confirming them inside the server.

The phase of data transmission and attack detection. This phase is done after the completion of the previous two phases. When messages are sent between vehicles, the identity of the sending vehicle and its intentions at the receiving vehicle are identified if the vehicle is an attack or not. The following steps illustrate the work of the proposed algorithm to VANET.

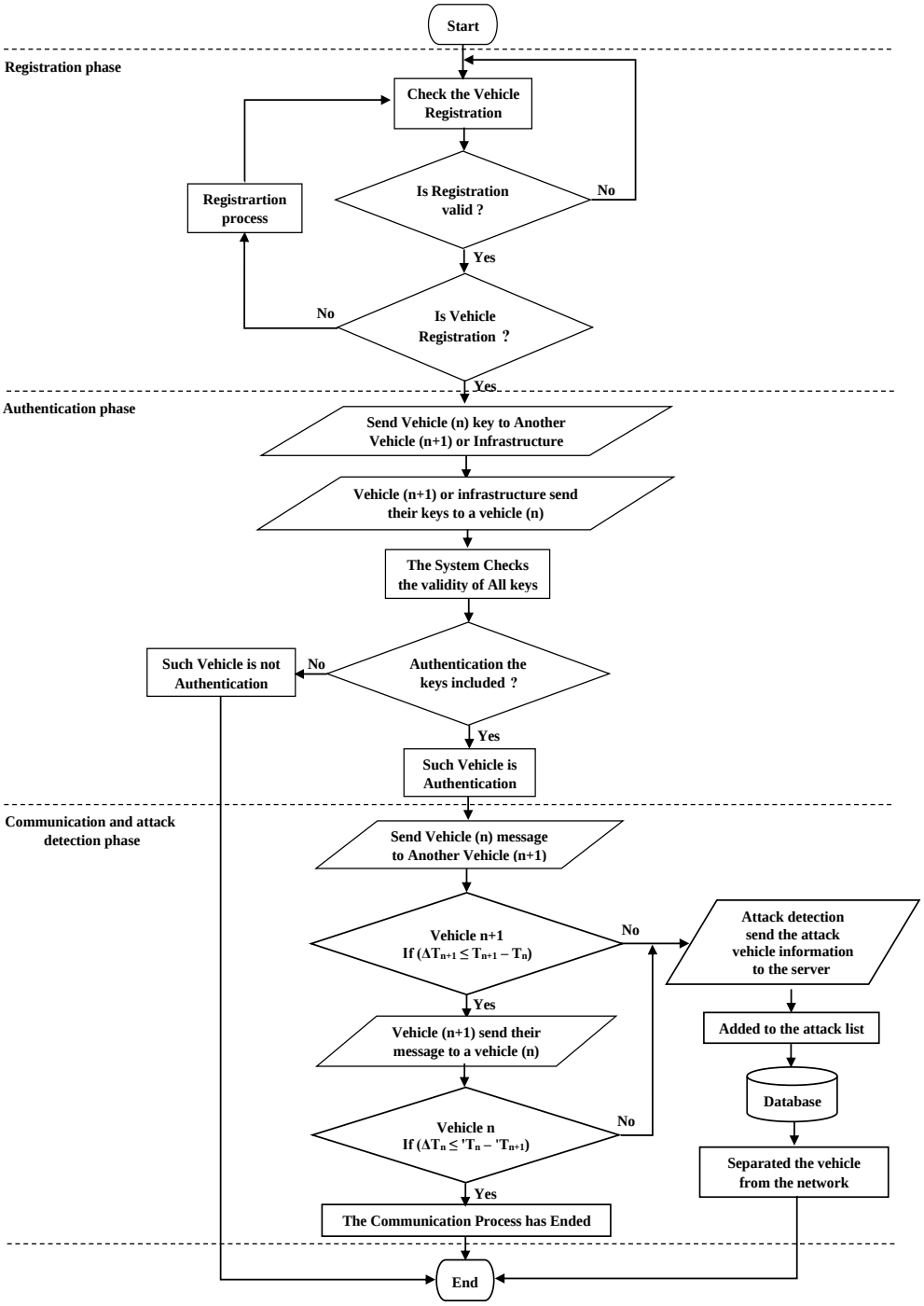

Figure 3. Proposed system algorithm

Algorithm:

```
Step 1  : Start.
Step 2  : Each vehicle has its information registered on the server.
Step 3  : The registration phase in order to complete the registration process within
          the network where each vehicle will send a key request to the server.
```

```
Step 4  : The server, Works to verify the request by knowing whether the vehicle has its
          information previously registered inside the server or not. As well as knowing
          whether the vehicle is already registered as a vehicle of attack. (Step 2)
Step 5  : The server, after checking the safety of the vehicle, it works to send the key to
          the vehicle.
Step 6  : After the registration phase is followed by the authentication phase.
          The authentication is done between the vehicles on the network as well as with
          RSU. This is done by exchanging the keys between the vehicles and then sending
          them to the server.
Step 7  : The server, matching the received keys with the database. If they are identical,
          the authentication process completes. Otherwise, the authentication process
          terminates and the vehicle is considered alien on the network. (Step 11)
Step 8  : The phase of data transmission or communication and the detection of attacks.
          Vehicle n will send a message to Vehicle n+1. The attacking vehicle is detected
          when the vehicle receives the harmful vehicle message, checking the time
          difference for messages received.
Step 9  : If the time difference is higher than usual. The vehicle is considered harmful and
          represents a DoS attack (Step 10). Otherwise, the receiving vehicle will send
          a response to the receiving vehicle that operates with the same mechanism for
          checking messages.
Step 10 : The victim vehicle: After knowing the harmful vehicles intentions. It sends its
          information to the server to store its information, add it to the list of
          attacking vehicles and separate it from the service.
Step 11 : End.
```

5. PROPOSED SOFTWARE ENGINEERING/SELF-CHECKING PROCESS ALGORITHM

In this work, the focused of the proposed system is on using a set of NIST tests in VANET as a conditional thresholds for accepting the keys. The purpose of using these tests inside the VANET is to increase the strength of key each vehicle and increase its randomness. Three tests were chosen namely: frequency, block and runs test through which the key is tests inside the server before sending to vehicles [18-22]. The proposed algorithm for key randomness tests is illustrated in Figure 4. The generation of the key is done through the two equations:
Server:

$$N_s=h[ID\_V_s||T_s]\oplus R\_V_s \quad (1)$$

$$Key\_V_i=h[ID\_V_s||Req_i||N_s]\oplus R\_V_s \quad (2)$$

where: ID_V_s server, time (T_s), generate values(R_V_s), Request the sending vehicle (Req_i). After that, the key converted to a binary number tested inside the three tests that work to know the arbitrary power of the key before sending it to the server.

5.1. Frequency test

This test obtained from the central limit theory for the number of random. This test aims to find out whether the frequencies of (1 & 0) across the entire key sequence are nearly equal, and the ratio of (1s & 0s) is close to half. If the number of (0s & 1s) is not the same, then this means knowing whether the difference falls within the randomness limit.

The primary test for randomness is the frequency test. If a pattern randomly generated, you would expect the number of (0s & 1s) to be almost the same. Also, many (0s | 1s) indicate no randomness. The Test of Frequency Test method estimates a sum where (0s) are encoded as a (-1) equivalent, and (1s) encoded as a (+1) equivalent. If the sum is equal to (0), there are similar numbers of (0s & 1s), but the sum varies from (0), whether it is very (-) or very (+), meaning a vast number of (0s | 1s).
Computes:
N : The length of the bit key.
Key_i : The key string.
Each bit 0 & 1 in the key is Serially by -1 and 1 alone by using the mathematical relationship:

$$X_i = 2key_i - 1 \quad (3)$$

where X_i represents a new value of the bit key_i at the i^{th} point.
The total of X_i represents S_n:

$$S_{obs} = |S_n|/\sqrt{n} \quad (4)$$

$$x = S_{obs} / \sqrt{2} \quad (5)$$

$$\text{P-value}=1\text{-}erf(x) \quad (6)$$

If (P-value < 0.01), then conclude that the key is non-random. Otherwise, conclude that the key is random.

5.2. Block frequency test

We may notice that if the first half of the key chain filled with one and the other half with zero, then the test ends with a non-random key. The goal of this test is to ensure that the frequencies (0 & 1) are evenly distributed along with the key. Block testing means to tackle this randomness type. Block Test divides a key into blocks and checks the number of (1s) in each block. The random key expects to contain about 50 percent of (1) in each block. In short, the block test accepts the block length parameter, which is the number of bits per block. From this, the number of blocks can be calculated. Next, the mass test calculates the (1s) ratio in each block and then uses a magic formula to compute the chi-squared test statistic.

Computes:

M : The length of each block.

N : The length of the bit key.

Key_i : The key string.

The key (n-bit string) is divided into non-overlapping N blocks each of M-bit, where:

$$N=[n/M] \quad (7)$$

πi of 1s in each block is given by:

$$\pi i = \frac{1}{M}\sum_{j=1}^{M} key(i-1)M-j \quad (8)$$

where $1 \le i \le N$. Chi-square is:

$$\chi^2_{(obs)} = 4M\sum_{i=1}^{N}\left(\pi i - \frac{1}{2}\right)^2 \quad (9)$$

$$\text{P-value}=igamc(N/2,\chi^2_{(obs)}/2) \quad (10)$$

If (P-value < 0.01), then conclude that the key is non-random. Otherwise, conclude that the key is random.

5.3. Runs test

A key length runs test means whether the bits are the same, bound by bits with opposite values. The goal of this test is to find out if the operating frequencies of (0 & 1) are of different lengths within the randomness limits. In this test, it is possible to the key to passing the first and second test if there are equal numbers of (0s & 1s) may be in the following order 101010101010. Here each block will have about 50 percent from 0 bits and 50 percent from 1 bit if we assume that the key chain formed in the form. The following is 11000100 on four runs: 00,1,000,11. If any key generated, the expected number on operation tests calculated. This test decides whether the oscillation between such 0s and 1s is too fast or too slow.

Computes:

N : The length of the bit key.

Key_i : The key string.

Compute the test statistic:

$$V_{(obs)}= \sum_{k=1}^{n-1} r(k) + 1 \quad (11)$$

where r(k)=0 if $key_k=key_k+1$, and r(k)=1

$$\text{Compute P-value}=\boldsymbol{erfc}\left(\frac{|Vn(obs)-2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right) \quad (12)$$

If (P-value < 0.01), then conclude that the key is non-random. Otherwise, conclude that the key is random. Below the explaination of the proposed self-checking process algorithm is introduced:

```
Step 1 : Start.
Step 2 : After a request from the Vehicle n received
Step 3 : The server calculates the equation of number (1), (2) through which a key is
         generated
Step 4 : The key is converted to a binary number
```

Step 5 : After conversion, the randomness of the key is tested using the frequency test done by calculating equations (3), (4), (5), (6)
Step 6 : If the test process is successful, the key is passed to the next test. If the test process for the key fails, the key is neglected and back to Step 3 to generate another key
Step 7 : After the second test key has passed the test successfully, the key is tested using a frequency block test by calculating equations (7), (8), (9), (10)
Step 8 : Repeat Step 6
Step 9 : After passing the key the first test and the second test, the key is tested using a runs test through equations (11), (12)
Step 10 : Repeat Step 6
Step 11 : After the three tests are successfully completed, the key is ready for encryption using hash function MD5 [23-26]
Step 12 : Send the key to the Vehicle n
Step 13 : End

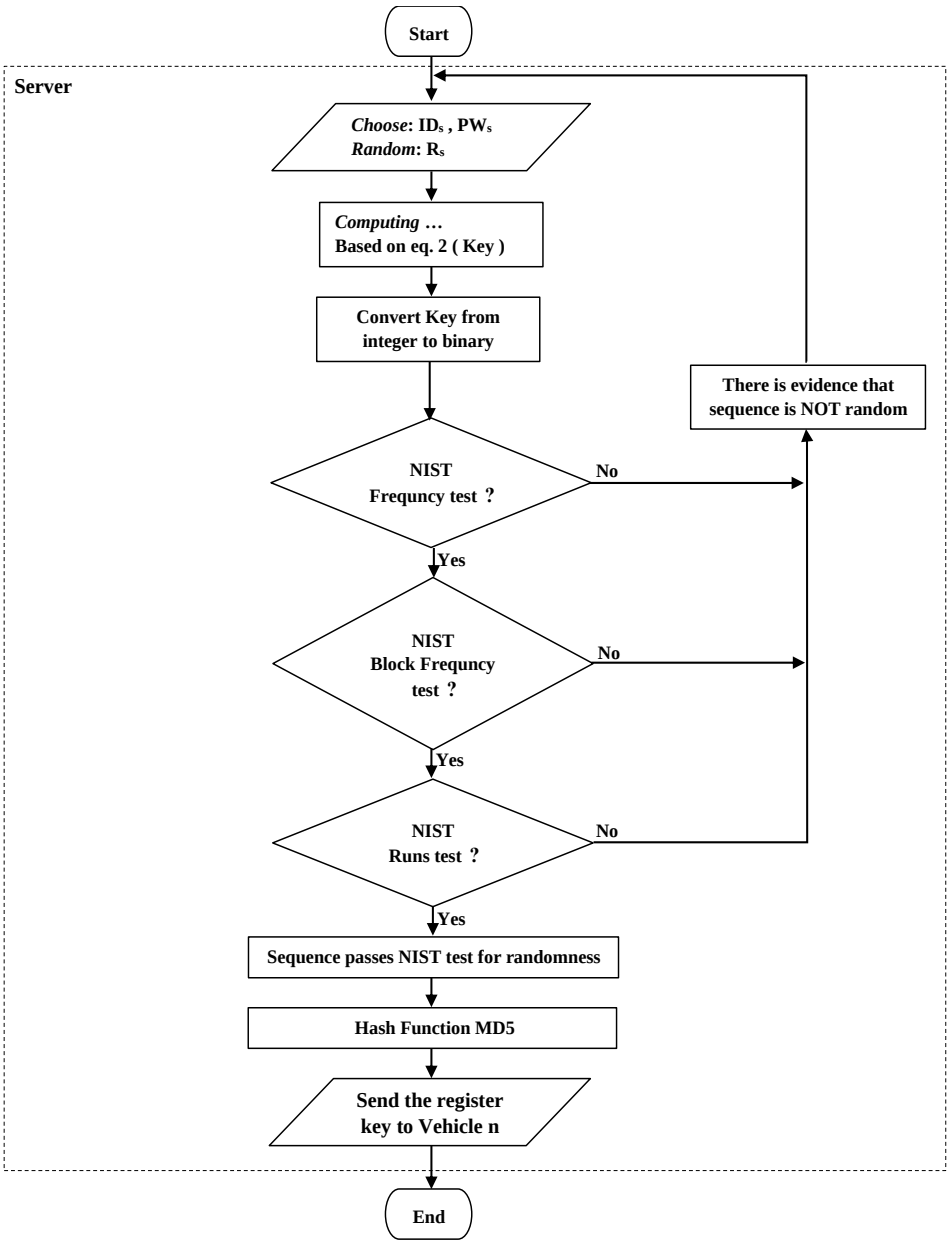

Figure 4. Proposed self-checking process algorithm

6. TEST RESULTS

In this part, we provide a set of tests for a set of keys to 20 vehicles. Some of them are passes, and some of them fail, depending on the randomness of the key. As shown in the Table 1, if the key is random, it is validated. Otherwise, it is not passed. All actions depend on the mechanism of making the three tests used in our proposed network system. In Table 2, we show solutions to a set of keys that did not pass the three tests by returning them to create a new key. This process is done automatically when each key is given. This means that no non-random key passed to vehicles, so it is difficult to know which keys are given to vehicles by the server.

Table 1. Test results for random and nonrandom keys sets

NO	Key Generation	Statistical Test					
		Frequency P-value		Block Frequency P-value		Runs P-value	
1.	11011010101011111111111010000101110000 1	0.1495	PASS	0.1117	PASS	0.8555	PASS
2.	11011010101011101000010111000000000000 1	0.2623	PASS	0.4779	PASS	0.9662	PASS
3.	11011010101011101000010111001111111000 1	0.2623	PASS	0.4779	PASS	0.4813	PASS
4.	11111111011001110111111010000101110000 1	0.0023	FAIL	0.0003	FAIL	0.0137	PASS
5.	10000000000000000000000000000000101110000 1	0.0000	FAIL	0.0002	FAIL	0.0524	PASS
6.	11001101011101110010101110000101110110 1	0.2623	PASS	0.5578	PASS	0.1719	PASS
7.	11001100110011001100110011001100110011	0.6310	PASS	0.9735	PASS	0.9014	PASS
8.	10000000000000000000000000000000000000 1	0.0000	FAIL	0.0000	FAIL	0.1908	PASS
9.	10011101111011101110111011111101111001	0.0023	FAIL	0.0611	PASS	0.3715	PASS
10.	10011101111011101111110000000000000000	0.6310	PASS	0.0047	FAIL	0.0025	FAIL
11.	11000011000111101011111110001111001111	0.0782	PASS	0.5578	PASS	0.0851	PASS
12.	11011111111111111001101010100111100011	0.0065	FAIL	0.0113	PASS	0.7533	PASS
13.	10101101101111111100110101010100111100011	0.0782	PASS	0.2397	PASS	0.2884	PASS
14.	10000111100011000111001100110011100011	0.8728	PASS	0.9098	PASS	0.1504	PASS
15.	10000000000010100000110100110011100011	0.0782	PASS	0.1359	PASS	0.3049	PASS
16.	11110111111000101001100111100011000000	0.8728	PASS	0.2873	PASS	0.0787	PASS
17.	11110110001000101001100111100011101110	0.4233	PASS	0.3425	PASS	0.7009	PASS
18.	11110110001000101001000000000000000000	0.0023	FAIL	0.0073	FAIL	0.2278	PASS
19.	11110110001111111111000000000111000000	0.8728	PASS	0.1991	PASS	0.0002	FAIL
20.	11110110001110111110000111000111000100	0.4233	PASS	0.5578	PASS	0.0917	PASS

Table 2. Test solutions for nonrandom keys sets

NO	Key Generation	Result	Solutions	Statistical Test					
				Frequency P-value		Block Frequency P-value		Runs P-value	
1.	11011010101011111111110 100001011100001	PASS							
2.	11011010101011101000010 1110000000000001	PASS							
3.	11011010101011101000010 1100111111100 01	PASS							
4.	11111111011001110111111 0100001011100001	FAIL	11111111011001110111 1110100001011100001	0.0782	PASS	0.0266	PASS	0.3049	PASS
5.	10000000000000000000000 0000001011100001	FAIL	11100101001101110010 1011100001011100001	0.8728	PASS	0.8266	PASS	0.6278	PASS
6.	11001101011101110010101 1 100001011101101	PASS							
7.	11001100110011001110011 0 011001100110011	PASS							
8.	10000000000000000000000 0000000000000001	FAIL	10011110111011101110 1111110010011110001	0.0163	PASS	0.0497	PASS	0.5437	PASS
9.	10011101111011101110111 0111111011110 01	FAIL	10011101101011101110 0111011011011110001	0.0163	PASS	0.2873	PASS	0.0994	PASS
10.	10011101111011101111110 0000000000000000	FAIL	10011110111011101110 1110000000011100000	0.6310	PASS	0.0215	PASS	0.0174	FAIL
11.	11000011000111101011111 1100011110011 11	PASS							
12.	11011111111111111001101 0101001111000 11	FAIL	11011110001000111100 1101010100111100010	0.6310	PASS	0.5578	PASS	0.8428	PASS
13.	10101101101111111001101 0101001111000 11	PASS							
14.	10000111100011000111001 1 001100111100011	PASS							
15.	10000000000010100000110 1 001100111100011	PASS							
16.	11110111111000101001100 1 111000110000000	PASS							
17.	11110110001000101001100 1 111000111011110	PASS							
18.	11110110001000101001000 0000000000000000	FAIL	11110110001000101001 0000000111100000010	0.1495	PASS	0.1991	PASS	0.4050	PASS
19.	11110110001111111111000 0000000111000000	FAIL	11110110001111011111 0000010000111001000	0.8728	PASS	0.5578	PASS	0.0787	PASS
20.	11110110001111011111000 0 111000111000100	PASS							

7. CONCLUSION

In this paper, we hed proposed a software engineering/self-checking process based cyber security system for VANET. The lightweight protocol was adopted for managing VANET. The proposed protocol consisted of three levels, each of which works to maintain network security from attacks that are related to DoS attacks to reach the required safety. The technology of self-checking process checked the generated keys inside the server to ensure the randomness before sending it to all vehicles. It was based on the use of three types of NIST tests. These tests worked on computing the randomness of the keys. If they fulfilled the conditions, they sent to the vehicle. The obtained results showed high efficiency in the performance of the proposed system in detecting the randomness failure and finding the solutions in case.

REFERENCES

[1] H. Hasrouny, et al., "VANET security challenges and solutions: A survey," *Vehicular Communications*, vol. 7, pp. 7-20, 2017.

[2] M. S. Anwer and C. Guy, "A survey of VANET technologies," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 5, no. 9, pp. 661-671, 2014.

[3] Q. Xu, et al., "Vehicle-to-vehicle safety messaging in DSRC," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pp. 19-28, 2004.

[4] V. D. Kumar, et al., "Data transmission between dedicated short range communication and WiMAX for Efficient vehicular communication," *J. of Computational & Theorical Nanoscience*, vol. 15, no. 8, pp. 2649-2654, 2018.

[5] B. Jia, et al., "Performance Analysis for the Coexistence of Radar and Communication in VANETs," in *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 979-983, 2019.

[6] M. S. Gurmani and D. P. F. Möller, "Mechanism Protecting Vehicle-to-Vehicle Communication," in *Smart Technologies*, Springer, pp. 335-343, 2020.

[7] U. Shaikh and N. Thalkar, "Vehicle Communication Systems: Technology and Review," in *Proceedings 2019: Conference on Technologies for Future Cities (CTFC)*, 2018.

[8] D. Johnson, et al., "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36-63, 2001.

[9] A. Wasef, et al., "ECMV: efficient certificate management scheme for vehicular networks," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pp. 1-5, 2008.

[10] W. Shen, et al., "Cooperative message authentication in vehicular cyber-physical systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 84-97, 2013.

[11] A. Perrig, et al., "The TESLA broadcast authentication protocol," *Rsa Cryptobytes*, vol. 5, no. 2, pp. 2-13, 2002.

[12] J. Guo, et al., "A group signature based secure and privacy-preserving vehicular communication framework," in *2007 Mobile Networking for Vehicular Environments*, pp. 103-108, 2007.

[13] X. Zheng, et al., "Chinese remainder theorem based group key management," in *Proceedings of the 45th annual southeast regional conference*, pp. 266-271, 2007.

[14] J. Li, et al., "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938-948, 2015.

[15] F. Wang, et al., "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 896-911, 2016.

[16] R. Muthumeenakshi, et al., "Extended 3PAKE authentication scheme for value-added services in VANETs," *Computers and Electrical Engineering*, vol. 59, pp. 27-38, 2017.

[17] C. Sun, et al., "A privacy-preserving mutual authentication resisting DoS attacks in VANETs," *IEEE Access*, vol. 5, pp. 24012–24022, 2017.

[18] L. Bassham, et al., "Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications," *National Institute of Standards & Technology*, 2010.

[19] I. V Chugunkov, et al., "Parallelization of test for assessing pseudorandom number generators using CUDA technology," in *2015 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW)*, pp. 60-64, 2015.

[20] A. Suciu, et al., "Parallel implementation of the NIST statistical test suite," in *Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing*, pp. 363-368, 2010.

[21] J. K. M. S. Uz Zaman and R. Ghosh, "Review on fifteen Statistical Tests proposed by NIST," *Journal of Theoretical Physics and Cryptography*, vol. 1, pp. 18-31, 2012.

[22] J. K. M. S. Uz Zaman and R. Ghosh, "A review study of NIST Statistical Test Suite: Development of an indigenous computer package," *arXiv Prepr. arXiv1208.5740*, 2012.

[23] M. Erritali, et al., "A Contribution to Secure the Routing Protocol" Greedy Perimeter Stateless Routing" Using a Symmetric Signature-Based AES and MD5 Hash," *International Journal of Distributed and Parallel Systems*, vol. 2, no. 5, pp. 95-103, 2011.

[24] R. Shaikh and D. Deotale, "A survey on VANET security using ECC, RSA & MD5," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 167-172, 2015.

[25] X. Wang and H. Yu, "How to break MD5 and other hash functions," in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 19-35, 2005.

[26] R. D. Ardy, et al., "Digital image signature using triple protection cryptosystem (RSA, Vigenere, and MD5)," in *2017 Int. Conf. on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, pp. 87-92, 2017.