Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi
Vol.6 No.2 July 2021, P-ISSN : 2502-3470, E-ISSN : 2581-0367

Secret Key Establishment Using Modified Quantization Log For Vehicular Ad-Hoc Network

M. Cahyo Kriswantoro[1], Amang Sudarsono[2], Mike Yuliana[3]

[1,2,3]*Informatics Engineering and Computer Department, Politeknik Elektronika Negeri Surabaya, Indonesia*

[1]cahyo.krizt@gmail.com, mcahyokriswantoro@pasca.student.pens.ac.id*

[2,3][amang, mieke]@pens.ac.id

Received: 2021-07-11; Accepted: 2021-07-25; Published: 2021-07-30

Abstract— Traditional cryptographic approaches such as symmetric and asymmetric cryptography are commonly employed to solve network security issues. The Secret Key Generation (SKG) system has the advantage of extracting secret keys from a wireless channel's physical layer information. It allows two wireless devices within the transmission range to extract a shared symmetric key without the use of a fixed key distribution infrastructure, allowing vehicular ad hoc networks to exchange information (VANET). This study aims to create a secure data communication system on the Vehicular Ad-Hoc Network using RSS Key Generation. Starting from the Modified Quantization Log, the results of the Modified Quantization Log show that the average KDRM between Alice and Bob is the average KDRM between Alice and Bob is 9.4%; meanwhile, the average KGR is 71.4 bps. This shows that the number of bit mismatches after the Modified Quantization Log process between the two valid users is already small, because they have used the pre-processing process in front of them, namely using the Kalman Filter and from the results of the BCH Code to be matched again so that it becomes the key. The next process is Universal Hash which is tested with the NIST test. The NIST Test parameters used are approximately entropy, frequency, block frequency, longest run, cumulative sum forward, and cumulative sum reverse. The existing results are appropriate; namely, the threshold in p whose value is above 0.01 is achieved. From the results of the Average Approximate Entropy, it is found that the largest value is obtained by the 40k10ms scheme, which is 0.7352.

Keywords— SKG, VANET, Modified Quantization Log, NIST Test, Kalman Filter

I. INTRODUCTION

One of the cornerstone pillars for the forthcoming 5G technological revolution is Intelligent Transportation Systems (ITS) [1]. Autonomous vehicles, in particular, are attracting a lot of attention because they don't require any human interaction. They can intelligently communicate with each other and with roadside units (RSU) to share real-time information thanks to the integration of in-vehicle sensing, communication, and networking capabilities. Vehicular ad hoc Networks (VANET), an important aspect of ITS, provide vehicle information exchange capabilities. For the targeted customers, this urban vehicle network ensures a variety of car-based services, such as road safety applications, smart traffic control, entertainment services, and so on. The notion of a linked car has been developed to give a practical autonomous system [2][3].

Symmetric and asymmetric cryptographic techniques are conventional mechanisms that are widely used to overcome network security problems. Symmetric cryptographic schemes require a symmetric secret key distribution between two valid users before encrypting and decrypting data [4]. This scheme has low computation, but the problem lies in key distribution and key management. Key distribution risks include eavesdropping during the transmission process by third parties. The problem with key management is that it requires the generation of different keys for each communicating user. Asymmetric cryptography schemes, also known as public-key cryptography, require exchanging public keys and secret keys between users before sending information [5].

The Secret Key Generation (SKG) system is a symmetric cryptography option used on wireless communication devices with limited compute and power. The benefit of extracting a shared symmetric key using physical layer information from a wireless channel eliminates the need for a fixed key distribution infrastructure between two wireless devices within the transmission range [4][6]. Received Signal Strength (RSS) [6][7], Channel Impulse Response (CIR) [7][8], and Channel State Information (CSI) [9] are all metrics that can be used as information from the wireless network for symmetric secret key extraction at the physical layer. Wireless communications are susceptible to eavesdropping attempts due to the nature of broadcasting. Exploiting the Randomness of Wireless Channels for Communication Security has sparked a lot of research interest in this area. Alice and Bob, two genuine users, exchange encrypted communications using secret keys generated by Received Signal Strength (RSS). Part of the secret key is based on reciprocity, which claims that without interference or non-linear components, the channel impulse responses from Alice to Bob and Bob to Alice are nearly identical [9].

Three metrics can be used to assess the key generation system: key generation rate (KGR), key disagreement rate (KDR), and unpredictability [5]. KGR is the maximum number of bits that can be created in a given amount of time. A high KGR value is necessary for the key generation process in a cryptographic technique that requires a specific key

DOI : http://dx.doi.org/10.25139/inform.v6i2.4037

length. KDR compares the total number of bits generated during the quantization process against the number of mismatched bits between Alice and Bob. The National Institute of Standard Technology (NIST) was employed to conduct the randomization test, which used the P-value parameter to evaluate confidence level. If the P-value is equal to 1, the resulting bit key will have complete Randomness. The parameter used in cryptography has a value of 0.01 [5]. If the P-value is high enough, the resulting key bit will meet the randomization requirement. [5].

The quantization scheme converts analog values into a binary sequence by comparing them with a qi reference threshold. For example, the quantization of sequence-1 with a gap, qg, is represented as where q1 is the threshold for sequence-1 quantization; km is binary quantified. When high-order quantization is adopted, several thresholds [q1, q2, •••] can be designed based on the HLog uv(m) dynamic range. A higher quantization sequence will increase the secret key generation rate but will result in serious key contention. Order and quantization gaps must be chosen carefully to balance the level of secret key generation and key disagreement [12].

Abhijit Ambekar and Hans D. Schotten [13], in their research on Enhancing Channel Reciprocity for Effective Key Management in Wireless Ad-hoc Networks, propose several methods to improve channel reciprocity. The proposed methods are l1 norm minimization, polynomial regression, and Kalman Filtering. These three methods are carried out before the quantization process (preprocess) in the SKG scheme with the aim of generating an effective secret key. The parameter used as information is the received signal strength indicator (RSSI). From the research results, this method can increase the reciprocity of RSSI so that the performance of the SKG scheme increases. This is because KDR decreased and KGR increased. In this case, the polynomial regression method produces the smallest percentage, while the l1 norm minimization produces the largest KGR value among the other three methods.

II. Research Methodology

The proposed system is a Vehicular Ad-Hoc Network (VANET) SKG (Secret Key Generation) scheme, which is a secret key generation process based on Received Signal Strength (RSS). The starting with channel probing of RSS, pre-processing, quantization, Key Agreement, and Key Confirmation. If these steps have been done, then the key is obtained in the bitstream between Alice and Bob.

The quantization scheme converts analog values into a binary sequence by comparing them with a qi reference threshold. For example, sequence-1 quantization with a gap, qg, where q1 is the threshold for sequence-1 quantization; km is binary quantified. When high-order quantization is adopted, several thresholds [q1, q2, •••] can be designed based on the HLog uv(m) dynamic range. Order and quantization gaps must be chosen carefully to balance the level of secret key generation and key disagreement.

The performance of the proposed scheme is assessed based on several parameters, namely KDR, KGR, and Randomness. KDR is defined as the percentage difference between Alice's generated key and Bob's generated key. KGR is defined as the number of bitstreams generated in each measurement per second. Randomness is a stream of key bits generated by a schema that normally must pass a NIST statistical test.

III. Result and Discussion

We outline our proposed system in this section. RSS channel probing, pre-processing, quantization, Key Agreement, and Key Confirmation are the four stages of our proposed method. Figure 1 depicts the proposed scheme in greater detail.

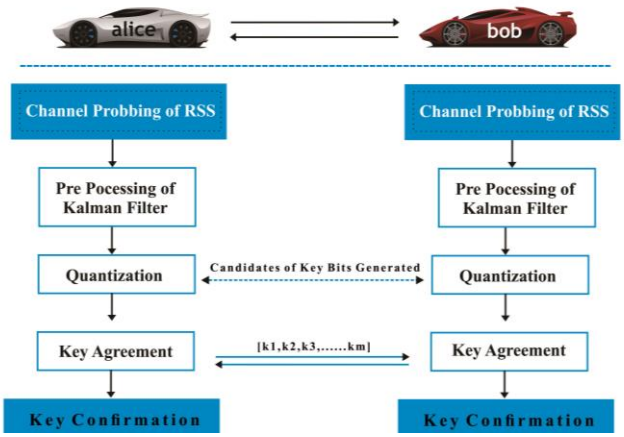

Figure 1. System design

A. Channel Probing Of RSS

The first stage is channel probing [4], where Alice and Bob, as legitimate users, take advantage of the wireless environment to generate RSS estimates. Meanwhile, Eve, who intercepts all communication information on the wireless channel between Alice and Bob, intercepts all communication information on the wireless channel. Therefore, Eve can record the secret key used by Alice and Bob to exchange messages during transmission. The channel probing scenario is shown in Figure 2.

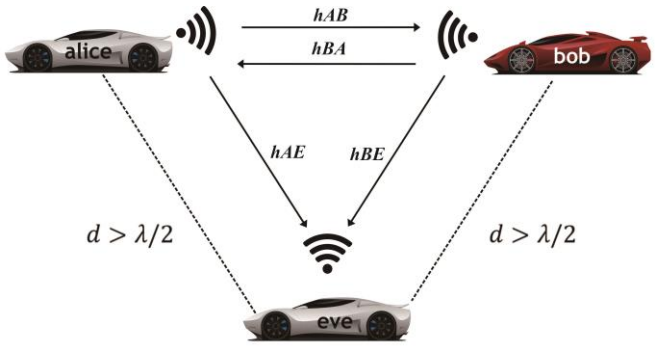

Figure 2. RSS collection scenario

The ping command, which employs the ICMP protocol, is used to collect RSS data between Alice and Bob. The channel

DOI : http://dx.doi.org/10.25139/inform.v6i2.4037

information measured by Bob from Alice is assumed to be, whereas the channel information measured by Alice from Bob is assumed to be. On the other hand, Eve received channel information from Alice as well as bob because the distance between Eve, Alice, and Bob is greater than half the wavelength broadcast and [10]. Alice and Bob exchange information within a specific amount of time, as defined by the equation below (1).

$$hAB = \{hAB(t1), hAB(t2),...,hAB(tn)\}$$
$$hBA = \{hBA(t1'), hBA(t2'),...,hBA(tn')\} \quad (1)$$

The time interval for probing between these two users is dependent on the coherence time required for Alice and Bob to have the same RSS value. Coherence time is defined as the time limit for a channel's impulse response time that remains constant or does not change in wireless communication systems. This parameter must be addressed in the secret key generation technique in V2V communication to get a high correlation value. The effect of the Doppler shift must be taken into account when V2V communication between Alice and Bob involves speed, especially for fast movement. The coherence time is inversely proportional to the maximum Doppler frequency [11].

The performance of the system is tested by looking at the reliability of the system in generating keys. Where the key has a match level and a level of confidentiality. The test scenario is outdoor or being on the road. So that through the process of generating keys between nodes that move at different speeds for each measurement. The initial goal for the current implementation is to employ the Received Signal Strength (RSS) scheme by taking it directly into the field. However, data collection is not possible, and existing data will be used to complete the next procedure, with all data obtained being RSS data.

System measurements were carried out using a wireless USB adapter TL-WN722N, an IEEE802.11b/g/n standard wireless network with a frequency of 2.4 GHz used for communication. Alice acts as the initiator while Bob acts as the responder. Between Alice and Bob, a 3-meter space was maintained. On the other hand, Eve is a tapper around 3 meters away and moves in the same direction as Alice and Bob and the circumstances at the time of collecting, which included a non-congested route.

The speeds are 0, 20, 40, and 60 kilometers per hour. So that it may be determined quickly after the secret key generation procedure has been analyzed. The second parameter, which is based on different ping intervals, also affects the measurement data between the two nodes. In this case, the ping interval is set based on the calculation of the coherence time, which must be above the coherence time value within a certain speed with an interval value of 7 ms, 10 ms, and 20 ms.

Vehicle 1 and Vehicle 2 communicate in an Ad-hoc (peer-to-peer) configuration and in the same channel. The RSS (Received Signal Strength Indicator) measurement process is carried out using the tcpdump command in monitor mode. Vehicle 1 will measure Vehicle 2's RSS signal strength, and Vehicle 2 will measure Vehicle 1's RSS signal strength during probing. When Vehicle 2 and Vehicle 1 get the same RSS signal strength, and the randomness condition is met. The key will be generated by Vehicle 1 to send a text message from Vehicle 1 to Vehicle 2 using a symmetric key where the key used for the text message encryption process will be the same as the key used to decrypt the message.

There are nine scenarios (A-I) that will be tested in secret key generation in Table I. In this scenario, there are two parameters, namely vehicle speed, and time interval.

TABLE I
MEASUREMENT SCENARIOS

Scenario	Speed	Ping Interval
A	20 km/hour	7 ms
B		10 ms
C		20 ms
D	40 km/hour	7 ms
E		10 ms
F		20 ms
G	60 km/hour	7 ms
H		10 ms
I		20 ms

In the channel probing process, Alice and Bob send probing. The channel probing mechanism is recording RSS data using Wireshark or by using a data retrieval program with the python program, with periodic ping requests from Alice to Bob, as shown in Figure 3.

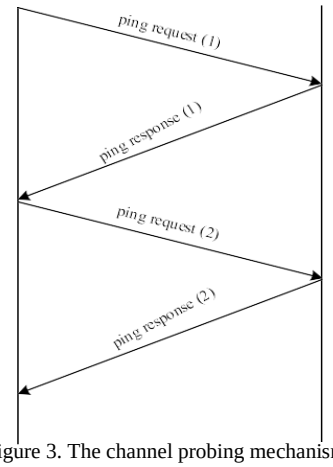

Figure 3. The channel probing mechanism

Figure 3 shows the channel probing mechanism that generates 3000 Alice and Bob's RSS measurement data. From the 3000 RSS, the next process is carried out, namely the quantization log in accordance with the quantization of the data above.

B. Kalman Filter Pre-Processing Test

The Kalman Filter algorithm will make the RSS value between the two nodes have a higher level of reciprocity so that the keys that both nodes will generate are the same. The way the Kalman Filter works is by estimating and using error

DOI : http://dx.doi.org/10.25139/inform.v6i2.4037

covariance. The parameters that become the reference for the Kalman Filter are a priori estimate and posterior estimate.

In Python, the Kalman Filter process runs by updating and estimating for as many as aa blocks (in this example, 800). This process occurs up to 10 times. There is a program to show the level of correlation when measurement data is given a Kalman Filter and without a Kalman Filter to see the performance of this Kalman Filter.

In this test, the correlation between Alice, Bob, and Eve will be calculated. 2 parameters were used, namely speed and ping interval. This first test is carried out in conditions where Alice and Bob move at a certain speed, then Alice pings with a certain ping interval. The RSS of each node will be stored in a stamp file, and then Eve will enter the Ad-Hoc network and participate in capturing the RSS of the two nodes. The results of the RSS correlation when measuring and after the estimation is carried out in the Kalman Filter program are located in Table II.

TABLE II
RSS CORRELATION VALUE FOR EACH SCENARIO

Scenario	Alice Bob Measurement	Eve Alice Measurement	Eve Bob Measurement	Kalman Filter Alice Bob
A	0.214726	-0.030299	0.047996	0.94969
B	0.611265	-0.134261	0.055011	0.98816
C	0.468336	0.006525	-0.114585	0.95611
D	0.378911	-0.135524	-0.156346	0.97993
E	0.646678	0.228607	0.376563	0.96338
F	0.214804	0.000932	0.026160	0.91245
G	0.544626	0.051117	0.368495	0.91414
H	0.300479	-0.000551	0.042370	0.93375
I	0.073022	0.214090	0.095378	0.88506

The data in Table II is obtained from the python program made by the author for the Kalman filter pre-processing program by calling the np.corrcoef command in python programming. The existing results are used as a comparison to see the differences in Figure 4.

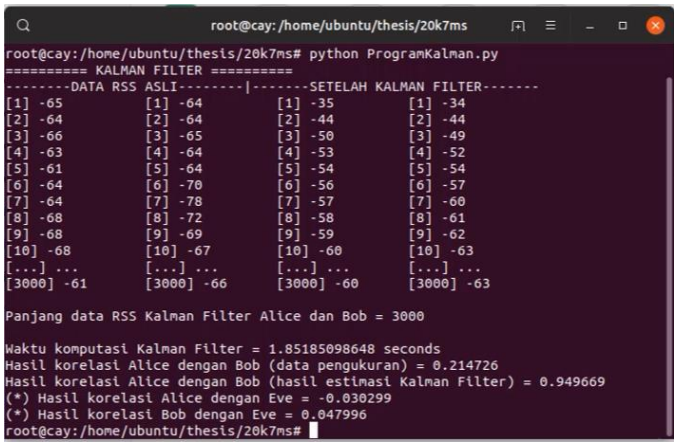

Figure 4. Kalman filter program

Based on Table II, the average correlation coefficient between Alice and Bob after using the Kalman Filter is 0.5936. Meanwhile, Eve's correlation coefficient with Alice and Bob is quite low, so it is very difficult for Eve to get the same RSS as Alice and Bob.

But indeed, there is also the Eve-Alice measurement result which is greater in RSS correlation than Alice-Bob correlation. It can be seen in measurement I (60 Km/hour 20 ms) the Eve-Alice correlation 0.214090 is greater than Alice-Bob, but this correlation does not stop at the measurement data, but there is a process that is carried out, namely the Kalman Filter process, the way it works is to estimate and use error covariance. So that in the first measurement, we get a good correlation between Alice-Bob, which is 0.885036, bigger than Eve-Alice.

C. Testing Modified Quantization Log

The quantization scheme converts analog values into a binary sequence by comparing them with a qi reference threshold. For example, sequence-1 quantization with a gap, qg, where q1 is the threshold for sequence-1 quantization; km is binary quantified. When high-order quantization is adopted, several thresholds *[q1, q2, ...]* can be designed based on the HLog uv(m) dynamic range. Order and quantization gaps must be chosen carefully to balance the level of secret key generation and key disagreement. But this quantity is not suitable for RSS data which incidentally has minus (-) data, so that the existing RSS data must be made absolute so that it produces a key with poor Randomness. It is necessary to modify the existing algorithm.

The modified algorithm in the calculation of 10log10RSS, the existing results will be calculated the mean to be compared if the results of every 10log10RSS more than the mean will be converted to 1, and vice versa if less than the mean will be converted to 0. Modified Quantization Log Algorithm (1).

Modified Quantization Log Algorithm (1)

Input RSS (H_{uv})

$H_{uv}^{log} = 10 \log_{10} |RSS|$

$l = 10$; $w = H_{uv}^{log}/l$;

For i = 0 : m

1. if , $H_{uv}^{log} > j$ then

   $H_{uv}^{log} = 1$

2. else if

   $H_{uv}^{log} < j$ then

   $H_{uv}^{log} = 0$

end for

$H_{uv}^{log} = (H_{uv}^{log}(1), H_{uv}^{log}(2), \ldots H_{uv}^{log}(m))$

Although the number of bit mismatches in the existing alterations is small, they still exist and must be processed again. This mismatch should not be good in practice, but it can still be treated at the following stage, namely the key agreement using BCH Code.

DOI : http://dx.doi.org/10.25139/inform.v6i2.4037

KDR quantization results (denoted as KDRM) vary for each scenario, as well as KGR results from the Modified Quantization Log as shown in Table III.

TABLE III
QUANTIZATION PERFORMANCE LOG MODIFICATION IN TERMS OF KDRM

Scenario	KGR (bps) Alice – Bob	KDR_M (%) Alice – Bob
A	71,43	11.61
B	71,43	5.55
C	71,43	11.7
D	71,43	8.51
E	71,43	9.53
F	71,43	9,38
G	71,43	9,66
H	71,43	10,40
I	71,43	8.85

Based on Table III show that the average KDRM between Alice and Bob is 9.4%. In comparison, the average KGR is 71.4bps. This shows that the number of bit mismatches after the Modified Quantization Log process between two legitimate users is small because they have used the pre-processing process in front of them before carrying out the Modified Quantization Log process. This data will be used in the next process; namely, Key Aggregation, using BCH Code to find out how many matched and equal bits are generated as keys.

D. BCH error correction test

After the Modified Quantization Log quantization process, then enter the Key Agreement, namely the BCH error correction test. The aim is to correct the remaining bit errors after the Modified Quantization Log quantization process in the BCH error correction test. This correction processing is done by checking each block (k), and in BCH, there is a limit for error correction that can be done for a certain length of k. In this case, the correction process is carried out with BCH (31.6), which means that the data will be checked per block with a block size of 6 bits. For n, k is 31.6, the error limit (t) that can be corrected is 7. The selection of the values of n and k is based on the error correction ability between the bit codewords Alice and Bob. Although the block size that is processed is too small, namely 6 bits per block, by using a Raspberry device, the computational time for such mathematical processing does not take a very long time. And the maximum correction size is also the highest value of various combinations of BCH codes when n is 31. The results of this BCH test can be seen in Table IV.

TABLE IV
BCH TEST RESULTS

Scenario	Total Error	Number of Bits After BCH	KDR
A	1686	1314	0%
B	1194	1806	0%
C	1752	1248	0%
D	1470	1530	0%
E	1512	1488	0%
F	1578	1422	0%
G	1482	1518	0%
H	1542	1458	0%
I	1398	1602	0%

Based on Table IV, it can be analyzed that the average error that still exists after the quantization process is more than 1512 errors. The error is an error when the input bit has become a 31-bit codeword. So by using BCH 31.6 and by checking through parity sent by Alice to Bob, the correction process is carried out on one of the nodes, namely Bob. Then Bob will send the index of the deleted block to Alice so that the bits generated after the BCH process become more equal with no errors generated.

The results obtained are that BCH is able to correct all processed blocks. For scenarios with a speed of 20 km/hour, the highest error is at an interval of 20 ms with a total of 1752 bits, and the lowest error is at an interval of 10 ms. When using correction with the BCH code, the BCH process is able to make corrections for all blocks that contain errors. And when the number of errors in the block is more than t or the maximum correction limit, it will be deleted.

In the scenario with a 40 km/hour speed, the 20 ms interval also has the most errors, but BCH can make corrections. And the lowest error is at the 7 ms interval of 1470. Likewise, for the 60 km/hour scenario with the lowest error being at the 20 ms interval, BCH can make corrections for all errors from the total block. In the correction process in this final project, blocks that have errors above t, and cannot be corrected by BCH will be deleted. So the error after BCH will be 0 or KDR 0%.

E. Universal Hash

After the BCH Code is processed, the resulting bits do not meet the randomness requirements. The entropy level of the bits generated by BCH will not always be high; consequently, a Universal Hash process is required, which will create a matrix or hash table that will be multiplied by bits sized according to the length of the key, resulting in high-entropy bits. In this case, the key length used is 256 bits.

After this is entered in NIST, NIST Test is used to determine the key to be used in the encryption and decryption process in symmetric cryptography for communication. There are several parameters in the NIST Test with a threshold of p whose value must be above 0.01. As a reference for determining the key, only the highest parameter is used from several keys that have been generated, or using a ranking of 1-3 keys is used. Only after that using sha-256.

To test the Randomness of the keys, used NIST test. The NIST Test parameters used are approximately entropy, frequency, block frequency, longest run, cumulative sum forward, and cumulative sum reverse. In order to pass the NIST test, the p-value must be greater than 0.01. Because there are 15 keys, using the NIST test can also determine which key will be used for the cryptographic process or a key winner. The results of the NIST Test can be seen in Table V.

DOI : http://dx.doi.org/10.25139/inform.v6i2.4037

TABLE V
NIST SCENARIO

Key	Approx. Entropy	Freq.	Block Freq.	Cusum Forward	Cusum Reverse	Runs	Long Runs
NIST Scenario 20K7MS							
1	0.532	0.453	0.183	0.338	0.857	0.274	0.827
2	0.405	0.707	0.972	0.803	0.990	0.537	0.683
3	0.197	0.211	0.183	0.160	0.422	0.685	0.264
4	0.924	0.900	0.785	0.945	0.990	0.899	0.941
5	0.891	0.802	0.466	0.519	0.338	0.704	0.760
NIST Scenario 20K10MS							
1	0.659	0.104	0.442	0.208	0.067	0.735	0.125
2	0.620	0.617	0.214	0.803	0.573	0.814	0.781
3	0.563	0.531	0.693	0.519	0.857	0.126	0.793
4	0.978	0.531	0.394	0.629	0.906	0.725	0.633
5	0.847	1	0.418	0.857	0.857	0.260	0.873
NIST Scenario 20K20MS							
1	0.869	1	0.249	0.573	0.573	0.317	0.310
2	0.671	0.707	0.084	0.573	0.857	0.458	0.772
3	0.925	0.531	0.920	0.573	0.629	0.689	0.873
4	0.300	0.707	0.101	0.422	0.745	0.264	0.509
5	0.359	0.381	0.741	0.629	0.745	0.938	0.629
NIST Scenario 40K7MS							
1	0.035	0.381	0.491	0.301	0.687	0.029	0.423
2	0.354	0.260	0.144	0.378	0.236	0.648	0.486
3	0.021	0.260	0.893	0.338	0.469	0.153	0.095
4	0.296	0.169	0.030	0.301	0.267	0.076	0.816
5	0.160	0.008	0.394	0.009	0.004	0.841	0.133
NIST Scenario 40K10MS							
1	0.943	0.211	0.214	0.267	0.378	0.685	0.719
2	0.984	0.802	0.214	0.906	0.857	0.897	0.645
3	0.587	0.900	0.306	0.629	0.745	0.901	0.933
4	0.202	0.31	0.286	0.469	0.208	0.116	0.991
5	0.960	0.900	0.267	0.422	0.338	0.708	0.889
NIST Scenario 40K20MS							
1	0.352	0.531	0.825	0.945	0.573	0.689	0.189
2	0.804	0.707	0.693	0.422	0.745	0.256	0.673
3	0.384	0.211	0.806	0.208	0.338	0.685	0.584
4	0.105	0.060	0.007	0.121	0.041	0.120	0.038
5	0.020	0.531	0.593	0.378	0.906	0.042	0.146
NIST Scenario 60K7MS							
1	0.629	0.381	0.306	0.236	0.301	0.354	0.152
2	0.438	0.017	0.825	0.035	0.035	0.209	0.582
3	0.399	0.045	0.517	0.091	0.091	0.897	0.510
4	0.854	0.707	0.942	0.945	0.629	0.893	0.381
5	0.904	0.453	0.878	0.803	0.301	0.379	0.575
NIST Scenario 60K10MS							
1	0.118	0.133	0.051	0.236	0.267	0.385	0.097
2	0.549	0.317	0.951	0.573	0.183	0.346	0.669
3	0.751	0.133	0.418	0.208	0.139	0.912	0.831
4	0.660	0.617	0.327	0.803	0.573	0.521	0.493
5	0.563	0.900	0.015	0.857	0.745	0.803	0.836
NIST Scenario 60K20MS							
1	0.161	0.133	0.063	0.208	0.183	0.788	0.842
2	0.916	0.381	0.893	0.573	0.687	0.938	0.738
3	0.456	0.317	0.092	0.208	0.629	0.090	0.477
4	0.842	0.169	0.693	0.105	0.267	0.710	0.430
5	0.928	0.802	0.327	0.857	0.974	0.788	0.944

From the results of the Average Approximate Entropy in Figure 9, it is found that the largest value obtained by the 40k10ms scheme is 0.7352

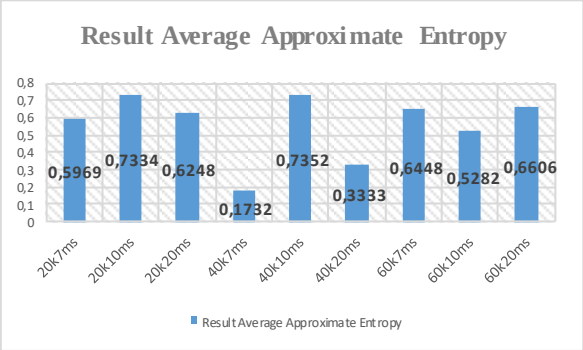

Figure 5. Graph of approximate entropy average results

IV. CONCLUSION

The results of the Modified Quantization Log show that the average KDRM between Alice and Bob is the average KDRM between Alice and Bob is 9.4%. In comparison, the average KGR is 71.4bps. This demonstrates that the number of mismatched bits following the Modified Quantization Log process between two genuine users is already low, as they used the pre-processing process in front of them, namely the Kalman Filter, before performing the Modified Quantization Log process. The combination of Modified Quantization Log and BCH Code shows that KDR has earned 0% between Alice and Bob. The bit results obtained on average are above 1512 bits, which means that the previous quantization process can fully generate a good key because when viewed, almost half more bits are suitable for quantization, from the results of the BCH code that can be processed to continue to become a key. The existing Universal Hash results have been tested with the NIST test. The expected results are appropriate, and namely, at the threshold in the form of p whose value must be above 0.01, it is achieved.

ACKNOWLEDGMENT

The author would like to thank profusely the author's supervisor and those who helped to carry out this research well.

REFERENCE

[1] C. Wei, “V2X communication in Europe - From research projects towards standardization and field testing of vehicle communication technology,” vol. 55, pp. 3103–3119, 10 2011.

[2] R. Coppola and M. Morisio, “Connected car: Technologies, issues, future trends,” ACM Comput. Surv., vol. 49, no. 3, pp. 46:1–46:36, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2971482

[3] S. Garg and G. S. Aujla, “Accessing risk priority of SSL SYN attack using game theoretic attack defense tree model for VANETs,” in 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India. IEEE, March 2016.

[4] A. Sudarsono, M. Yuliana, and P. Kristalina, “A Reciprocity Approach for Shared Secret Key Generation Extracted from Received

DOI : http://dx.doi.org/10.25139/inform.v6i2.4037

Signal Strength in The Wireless Networks", IEEE Access, pp. 177-182, 2018.

[5] M. Yuliana, Wirawan, and Suwadi, "Performance Evaluation of the Key Extraction Schemes in Wireless Indoor Environment", International Conference on Signals and Systems (ICSigSys), pp. 138–144, 2017.

[6] L. Cheng, L. Zhou, B.C. Seet, W. Li, D. Ma, and J. Wei, "Efficient Physical-Layer Secret Key Generation and Authentication Schemes Based on Wireless Channel-Phase", Journal of Hindawi Mobile Information Systems, Vol. 2017, Article ID 7393526, pp. 1–13, 2017.

[7] M. Yuliana, Wirawan, and Suwadi, "Performance Evaluation of the Key Extraction Schemes in Wireless Indoor Environment," International Conference on Signals and Systems (ICSigSys), pp. 138–144, 2017.

[8] M.A. Forman, D. Young, and D.R. Dowdle, "The Generation of Shared Cryptographic Keys through Channel Impulse Response Estimation at 60 GHz, Sand Report", Sandia National Laboratories, SAND2010-6662, Unlimited Release, Printed September, 2010.

[9] A. Ambekar and H.D. Schotten, Enhancing Channel Reciprocity for Effective Key Management in Wireless Ad-Hoc Networks, IEEE 79th Vehicular Technology Conference (VTC Spring), Electronic ISBN: 978- 1-4799-4482-8, 2014.

[10] J.L. Carter and M.N. Wegman, Universal Classes of Hash Functions, Journal of Computer and System Sciences 18, pp. 143–154, 1979.

[11] A. Sudarsono, M. Yuliana, P. Kristalina, and A. R. Barakbah, An Implementation of Shared Key Generation Extracted from Received Signal Strength in Vehicular Ad-Hoc Communication, The Sixth International Symposium on Computing and Networking, 2018.

[12] Peng, L., Li, G., Zhang, J., Woods, R., Liu, M., & Hu, A, An Investigation of Using Loop-back Mechanism for Channel Reciprocity Enhancement in Secret Key Generation, EEE Transactions on Mobile Computing. DOI: 10.1109/TMC.2018.2842215, 2018

[13] A. Ambekar and H.D. Schotten, Enhancing Channel Reciprocity for Effective Key Management in Wireless Ad-Hoc Networks, IEEE 79th Vehicular Technology Conference (VTC Spring), Electronic ISBN: 978- 1-4799-4482-8, 2014.

DOI : http://dx.doi.org/10.25139/inform.v6i2.4037