ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika

ISSN(p): 2338-8323 | ISSN(e): 2459-9638 | Vol. 10 | No. 1 | Halaman 1 - 15
DOI : http://dx.doi.org/10.26760/elkomika.v10i1.1 Januari 2022

Steganography Based on Data Mapping and LSB Substitution With RSS Key Generation

MUHAMMAD TAUFIQ SUMADI, AMANG SUDARSONO, MIKE YULIANA

Politeknik Elektronika Negeri Surabaya, Indonesia
Email: sumadi11895@gmail.com

Received 14 Juni 2021 **|** *Revised* 6 Juli 2021 **|** *Accepted* 23 Juli 2021

ABSTRAK

Untuk menghindari pihak ketiga yang tidak berwenang, berbagai solusi keamanan telah dibuat untuk mengamankan kerahasiaan pesan atau informasi, salah satunya adalah steganografi. Penelitian ini mengusulkan teknik pembangkitan kunci dalam proses steganografi. Secara umum steganografi menggunakan konsep kunci simetris, dimana metode ini membutuhkan pihak ketiga dalam proses pendistribusian kunci rahasia. Solusi dari permasalahan tersebut penulis mengusulkan metode baru dengan menggabungkan pembangkitan kunci rahasia menggunakan RSS pada jaringan nirkabel dengan steganografi untuk menyembunyikan pesan. Sistem yang kami usulkan diterapkan pada kondisi statis dan kondisi dinamis untuk menguji kinerja. Pada proses pembangkitan kunci didapatkan waktu rata-rata 110,52 detik untuk membangkitkan kunci. Pada proses embedding, waktu komputasi rata-rata untuk 65.536 karakter adalah 6,38 detik untuk menghasilkan citra stego. Pada proses ekstraksi didapatkan waktu komputasi rata-rata 0,63 detik untuk mendapatkan pesan rahasia.

Kata kunci*: RSS, Key Generation, Symmetric Cryptography, Steganography.*

ABSTRACT

To avoid unauthorised third parties, variety of security solutions have been created to secure the confidentiality of messages or information, one of which is steganography. This study proposes key generation technique in the steganography process. In general, steganography uses a symmetric key concept, where this method requires a third party in the secret key distribution process. The solution to this problem the authors propose a new method by combining secret key generation using RSS on a wireless network with steganography for hiding messages. To test performance, our proposed system is used in both static and dynamic condition. In the key generation process, an average time of 110.52 seconds was obtained to generate keys. In the embedding process, the average computation time for 65,536 characters is 6.38 seconds to produce a stego image. In the extraction process, an average computation time of 0.63 seconds is obtained to get a secret message.

Keywords*: RSS, Key Generation, Symmetric Cryptography, Steganography.*

1. INTRODUCTION

Confidentiality and security are important aspects needed in the process of exchanging messages or information via the internet. To avoid unauthorised third parties, a variety of security solutions have been created to secure the confidentiality of messages or information, one of which is steganography. The science of steganography is in line with cryptography, but the two have differences. Steganography aims to hide secret messages through a medium. Meanwhile, cryptography disguises a secret message but does not hide it. In steganography, information is hidden in digital media so that no one can see whether digital media has information in it or not **(El-Dairi & House, 2020)**.

Steganography techniques have been employed in a variety of digital media in recent years, namely, images, audio, and video, to achieve secret communication. Likewise, in a variety of applications, it is used in paper **(Liu et al., 2018)** using steganography to share secrets and authentication. Steganography is used to send data from the phone to the cloud and vice versa in paper **(Xiang et al., 2015)**. Paper **(Tondwalkar & Vinayakray-Jani, 2016)** uses a steganography process to secure node localisation in a wireless network.

In steganography, in increasing security against steganalysis, a key is needed that can be used to determine the position of the cover-image pixel to hide secret data. Key-based cryptographic techniques can be divided into symmetrical and asymmetrical. Symmetric cryptography has low computation but has problems with key distribution and key management **(Yassein et al., 2017)**. So there is a risk when the key distribution, third parties, can tapping into the transmission process. The asymmetric cryptography scheme uses two keys to communicate, namely, the public key and the private key. The public key can be seen by anyone on the network, which will be used for encryption. Meanwhile, the private key will be used to perform the decryption process. In asymmetric cryptographic schemes, the computation process takes longer when compared to symmetric cryptography schemes.

Currently, the physical layer of a wireless channel can be utilised to enhance conventional security **(Yuliana et al., 2019b)**. This can be done by taking advantage of the wireless channel's randomness and uncertainty. The benefit of extracting a shared symmetric key using physical layer information from a wireless channel is that it eliminates the need for a fixed key distribution infrastructure between two wireless devices within the transmission range. The Secret Key Generation (SKG) scheme is an alternative to symmetric cryptography that can be applied to wireless communication devices. Such schemes take advantage of the exchange, randomness, and uniqueness of wireless channels by extracting bit streams through information which may result in a high correlation between sender and receiver.

In paper **(Dewi et al., 2020) (Sudarsono et al., 2019) (Wei et al., 2013)** states that the SKG scheme consists of four stages which includes exchanging channel information between users to obtain RSS measurements via the channel probing stage, as well as converting the obtained RSS into binary bit form via the quantization stage, using an error correction scheme, the information reconciliation stage corrects the binary bit sequence obtained in the previous stage, and the privacy amplification stage increases the protection of the hidden key bit sequence to prevent it from being predicted by third parties.

Previous researchers merged other approaches with the Least Significant Bit (LSB) technique to increase the embedding potential even further. For example, research **(Hussain et al., 2018)** proposes using the Pixel Value Difference (PVD) a technique for increasing embedding ability based on LSB. High-value and low-value texture blocks are distinguished by the difference in the mean of the four-pixel blocks. The hidden message is then inserted using k-bit and an LSB substitution. In paper **(Khodaei et al., 2016)** with the PVD process and Modification Prediction Error (MPE), a recursive hybrid LSB approach was suggested (MPE). First, define an image's lower and higher textures, then use the substitution LSB and PVD methods to replace them. Recursively use the PVD shift and MPE to increase embedding capability. This approach offers more capacity while maintaining a reasonable PSNR. The embedding method can use up to four LSBs. As well as, paper **(Bai et al., 2017)** introducing adaptive LSB by utilising the characteristics of PVD. The cover-image is divided into two-pixel blocks, and the difference between the two pixels' values is calculated using this process. The difference is used to calculate the number of hidden bits. The hidden bits are then embedded in the selected pixel block using an adaptive LSB scheme. This approach also includes a readjustment step to keep the pixels within their ranges, resulting in increased embedding capability while preserving visual quality. During the embedding process, however, the maximum number of LSBs used is four.

Bit inversion is another LSB-based technique that aims to improve the PSNR value. The LSB cover-image of a single pixel is modified in this inversion technique if it matches a specific pattern. As a consequence, while the visual quality improves, the embedding capability decreases. Another texture-adaptive LSB approach was recently suggested **(Bai et al., 2017)**. Pixels in the cover-image are divided into edge and non-edge areas using this form. The 3 Most Significant Bit (MSB) of the cover pixel evaluate edge information, while the remaining 5 LSBs are used adaptively through LSB substitution. By maintaining a reasonable visual quality, this method achieves a high embedding capability. The embedding method, however, uses a maximum of 5 LSBs.

This paper proposes a key generation technique in the steganography process. In general, steganography uses a symmetric key concept, where this method requires a third party in the secret key distribution process. To overcome this, the authors propose a new method by combining a secret key generator using RSS on a wireless network with steganography in hiding messages. Broadly speaking, the system is divided into two: SKG and steganography. SKG is divided into four stages, namely channel probing, quantization, information reconciliation, and privacy amplification. Meanwhile, steganography has two stages, namely embedding and extraction. By combining secret key generation techniques with steganography, it is hoped that the secret message will be sent more safely.

In this research, we'll talk about steganography technique with image media using the mapping-based Least Significant Bit (LSB) method to keep sensitive information hidden **(Zakaria et al., 2018)**. In addition, this study uses Received Signal Strength (RSS) as a secret key generation **(Sumadi et al., 2020)**. By combining these two methods, it can be a new method of information security.

2. SYSTEM DESIGN

We will outline the proposed system in this section. Alice (sender) and Bob (receiver) will be the two entities communicating. First, Alice and Bob each measure the RSS of their communication opponents. RSS collection is obtained by utilising requests and responses between communication opponents via the Internet Control Message Protocol (ICMP) protocol. After both of them have their respective RSS, then the SKG process is carried out. The output from the SKG process generates a secret key that will be used for the steganography process. In the steganography process, Alice inserts a secret message using a secret key into the cover-image to produce a stego-image. The resulting stego image through the steganography process is then sent to Bob as the recipient. Bob, who has received the stego image, then extracts it using a secret key so that he gets the secret message sent by Alice. Show the proposed system scenario in Figure 1.

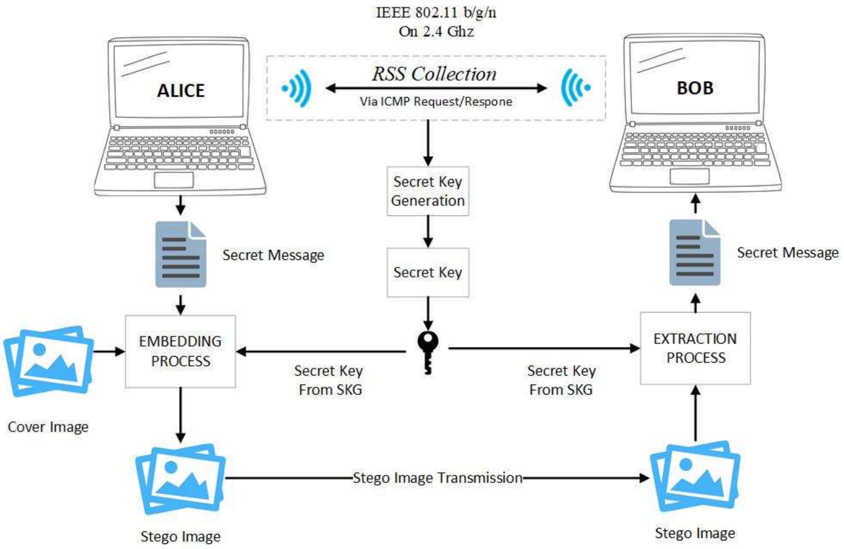


Figure 1. Proposed System Scenario

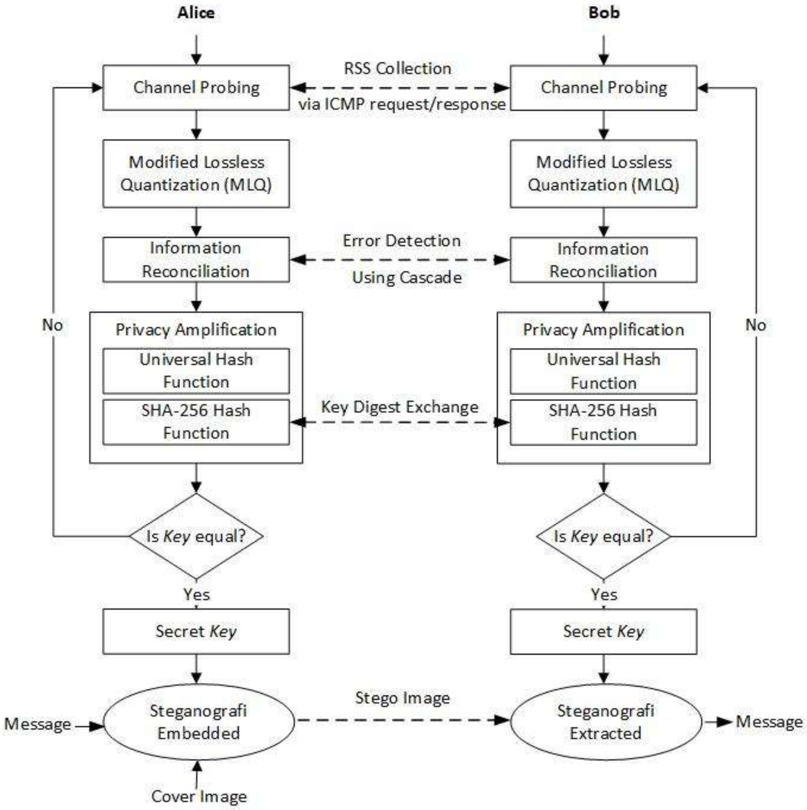


Figure 2. Proposed System Mechanism

This system has six stages in total, with four stages for SKG and two stages for steganography. The stages for the SKG process include channel probing, quantization, information reconciliation and privacy amplification. In contrast, the stages for the steganography process are embedding and extraction. Proposed system mechanism show in Figure 2.

2.1 Channel Probing

By utilising the ping command that uses the Internet Control Message Protocol (ICMP) protocol, we can record and measure the signal strength generated between the two communicating users (Alice and Bob). To calculate the interval time used between two communicating users, it depends on the coherence time (Tc). Based on **(Yuliana et al., 2019a)**, $Tc = \frac{1}{f_D}$ where f_D maximum droppler frequency. Meanwhile, $f_D \frac{v}{\lambda}$ where v denotes the speed at which legal users travel (i.e 1.1 m/s). $\lambda = \frac{c}{f_c}$, c is the speed of light, and f_c denotes carrier frequency of the channel. We using carrier frequency 2,4 GHz in our system, $\lambda = \frac{3 \, x \, 10^8}{2.4 \, x \, 10^9}$ = 0.125 m. As a consequence, Tc (coherence time) is 113.6 milliseconds. In our best estimation, the ping interval time was set to 110 milliseconds. In a semi-indoor environment, the experiment was carried out in both static and dynamic conditions.

2.2 Quantization

The result of the probing channel will get a series of strong signals in dBm. The signal strength obtained was then quantization. At this stage, a series of strong signals will be converted into bits (binary numbers). The strong signal row will be denoted T with T=[t_1, t_2 …, t_n] where n is the number of signal strengths. Calculate the value of the mode, max, and min of T. Mode is a value that occurs frequently. After obtaining the value, then calculate the value q^+ and q^-. T will be set to 0 if it is less than or equal to q^-. If T is more than q^+, it will be set to 1. T will be discarded if it is not inside the q^+ and q^- range. The quantization algorithm is shown in Algorithm 1 **(Sumadi et al., 2020).**

```
Algorithm 1: Quantization Algorithm
    Input: RSS measurements T = [$t_1$, $t_2$… $t_n$]
      n: number of RSS;
    Output: B = $Q_1$, $Q_2$, …, $Q_s$ the generated bits stream
    for j=1 to n do
      $M_o$ ← mode(t), max ← max(t), min ← min(t);
      $q^+ \leftarrow M_o$ + ɑ × (max + min), $q^- \leftarrow M_o$ - ɑ × (max + min);
      if $t_j \leq q^-$ then
          $t_j \leftarrow 0$;
      else if $t_j > q^+$ then
          $t_j \leftarrow 1$;
      else $t_j \leftarrow t_j$;
    B ← $Q_1$, $Q_2$, …, $Q_s$
```

2.3 Information Reconciliation

The communication between the two legitimate users (transmitter and receiver) is a half-duplex communication as a result of the channel probes that occur unable to be carried out a similar time. As a result, transmitter and receiver channel information is inconsistent. In addition, random disturbances that occur by the environment also affect the channel of information obtained. The channel data is gathered using both devices during the quantization stage, and the output is generated in the shape of a stream of bits. As a result, unequal bits are allowed in the bit streams received by the two devices. We use a cascade information reconciliation protocol **(Brassard & Salvail, 1994)** to detect dissimilar bits to eliminate bit differences between the two devices.

2.4 Privacy Amplification

The reconciliation information allows Eve to get a series of bits exchanged between two user. To avoid this, the set of bits is enhanced by using a universal hash **(Brassard & Salvail, 1994)**. This research, it was using a key with a length of 256 bits, resulting in more than one key. Some of the keys that are generated are then carried out by NIST test to determine which key is the best. After the best key is obtained from both users, the hashing process is carried out. Using SHA-256 will generate a key digest. This digest will be matched between the two users. If it matches, then the key will be used; if not, then reset the key is performed from the probing channel.

2.5 Stego Embedding

The secret key that has been generated from the SKG process is then used for the embedding process in steganography. Alice enters the secret message into the cover-image using the secret key. This is the stage of the embedding process.

a. Convert *K* (stego key) into *Sum* value using formula shown in Equation (1)

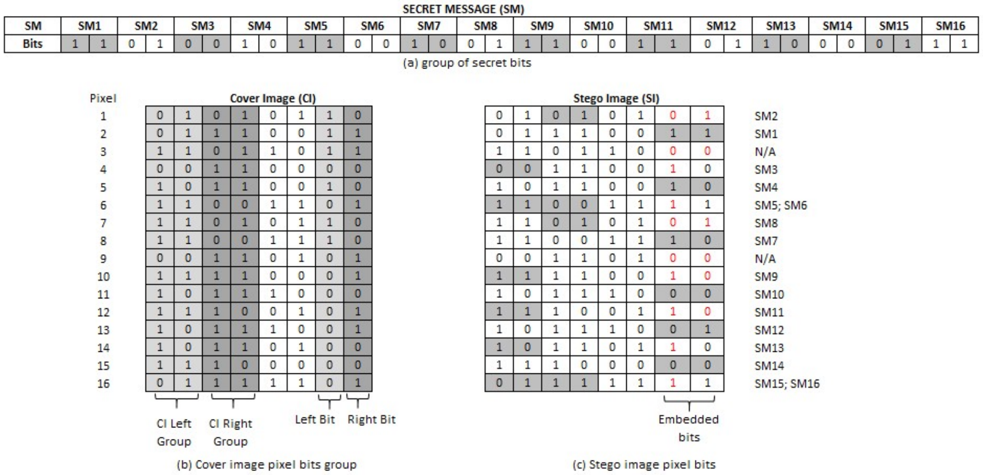
$$Sum = \sum_{i=1}^{k}(k_i \times i) \mod 256 \tag{1}$$

b. Compute SM (secret message) value using formula shown in Equation (2)

$$\text{For } 1 \leq i \leq n, M_i = (m_i + Sum)\, mod\, (n+1) \tag{2}$$

c. Embedding SM to cover-image

The secret key is used to determine the insertion position of the message bits into the pixels on the cover-image. The algorithm works by mapping the secret message to the cover-image, where the records of the mapping are managed by using LSB substitution on the cover-image **(Zakaria et al., 2018)**. In the process of embedding, secret messages are divided into groups, namely SM1 (11), SM2 (01), SM3 (00), ... SM$_n$ as shown in Figure 3. Likewise, the bits on the cover-image are also divided into groups. For example, the CI Left Group and CI Right Group is shown in Figure 3. CI Left Group is the 1st and 2nd bit on each pixel on the cover-image. Thus the CI Right Group is the 3rd and 4th bit on each pixel on the cover-image. CI Left Group and CI Right Group will be matched with the secret message group (SM). Left bits are the 7th bit, and Right bits are the 8th bit in each pixel on the cover-image. Left bits and Right bits are used as an indicator of insertion. The process of mapping secret bits to the cover-image follows the following rule: If the first group of bits (i.e., SM1) matches the CI Left Group of the cover-image pixels, then the Left bits are replaced with '1' if not replaced with '0'. If the second group of bits (i.e., SM2) matches the CI Right Group of the cover-image pixels, then the Right bits are replaced with '1'; otherwise, they are replaced by '0'. If either Left bits or Right bits are '0' then the 2 LSB for each cover-image pixel is then replaced by the secret message group value that did not match the previous one (i.e., SM1 or SM2). If both Left bits and Right bits have a value of '0', this indicates that no secret message group is inserted in the cover-image pixel.



SECRET MESSAGE (SM)

SM	SM1		SM2		SM3		SM4		SM5		SM6		SM7		SM8		SM9		SM10		SM11		SM12		SM13		SM14		SM15		SM16	
Bits	1	1	0	1	0	0	1	0	1	1	0	0	1	0	0	1	1	1	0	0	1	1	0	1	1	0	0	0	0	1	1	1

(a) group of secret bits

Pixel	Cover Image (CI)							
1	0	1	0	1	0	1	1	0
2	0	1	1	1	0	0	1	1
3	1	1	0	1	1	0	1	1
4	0	0	1	1	0	0	0	0
5	1	0	1	1	0	0	1	0
6	1	1	0	0	1	1	0	1
7	1	1	0	1	0	1	1	0
8	1	1	0	0	1	1	1	0
9	0	0	1	1	0	1	0	1
10	1	1	1	1	0	0	0	1
11	1	0	1	1	1	0	0	0
12	1	1	1	0	0	1	0	1
13	1	0	1	1	1	0	0	1
14	1	0	1	1	0	1	0	0
15	1	1	1	0	0	0	0	0
16	0	1	1	1	1	1	0	1

(b) Cover image pixel bits group

Stego Image (SI)								
0	1	0	1	0	1	0	1	SM2
0	1	1	1	0	0	1	1	SM1
1	1	0	1	1	0	0	0	N/A
0	0	1	1	0	0	1	0	SM3
1	0	1	1	0	0	1	0	SM4
1	1	0	0	1	1	1	1	SM5; SM6
1	1	0	1	0	1	0	1	SM8
1	1	0	0	1	1	1	0	SM7
0	0	1	1	0	1	0	0	N/A
1	1	1	1	0	0	1	0	SM9
1	0	1	1	1	0	0	0	SM10
1	1	1	0	0	1	1	0	SM11
1	0	1	1	1	0	0	1	SM12
1	0	1	1	0	1	1	0	SM13
1	1	1	0	0	0	0	0	SM14
0	1	1	1	1	1	1	1	SM15; SM16

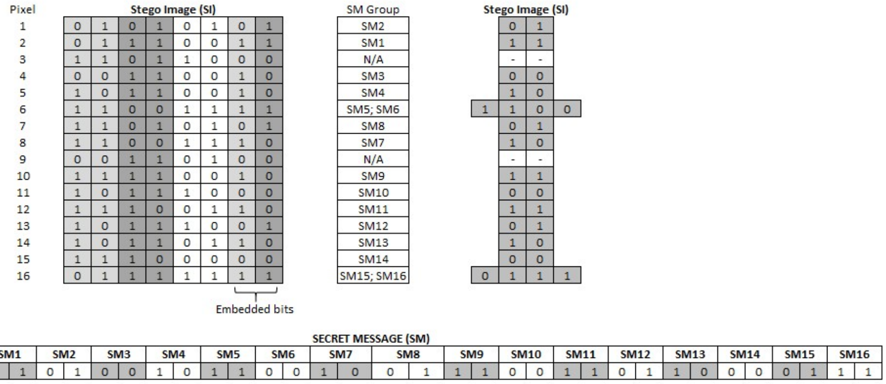
(c) Stego image pixel bits

Figure 3. Secret Message, Cover Image and Stego Image Pixel Bits

For example, in Figure 3. The 1st pixel has a value of $CI_{L,1}$ = 01, $CI_{R,1}$ = 01, $L_{B,1}$ = 1, $R_{B,1}$ = 0 and SM1 = 11, SM2 = 01. By following the applicable rules, SM1 ≠ $CI_{L,1}$ so that the value of $L_{B,1}$ is replaced by 0. Likewise, by following the rule of SM2 = $CI_{R,2}$ so that the value of $R_{B,1}$ is replaced by 1. Because the value of $L_{B,1}$ = 0, the value of left bit and right bit for the next pixel is SM1 (11). The same process is repeated by following the mapping rule so that the entire secret bits group is all inserted, for example, in Figure 3. The 1st pixel has a value of $CI_{L,1}$ = 01, $CI_{R,1}$ = 01, $L_{B,1}$ = 1, $R_{B,1}$ = 0 and SM1 = 11, SM2 = 01. By following the applicable rules, SM1 ≠ $CI_{L,1}$ so that the value of $L_{B,1}$ is replaced by 0. Likewise, by following the rule of SM2 = $CI_{R,2}$ so that the value of $R_{B,1}$ is replaced by 1. Because the value of $L_{B,1}$ = 0, the value of left bit and right bit for the next pixel is SM1 (11). The same process is repeated by following the mapping rule so that the entire secret bits group is all inserted.

2.6 Stego Extraction

Similarly to the embedding procedure, the bits in a Stego Image (SI) are divided into SI Left (bit 1 and 2), SI Right (bit 3 and 4), Left bit (bit 7) and Right bit (bit 8) **(Zakaria et al., 2018)**. In the secret bit extraction process in the stego image, follow the following rules: If $L_{B,1}$ = 1 and $R_{B,1}$ = 0, then $SI_{L,1}$ is SM1, and the last 2 bits in the next pixel are SM2. If $L_{B,1}$ = 0 and $R_{B,1}$ = 1 then $SI_{R,1}$ is SM2 and the last 2 bits in the next pixel are SM1. If $L_{B,1}$ = 1 and $R_{B,1}$ = 1 then $SI_{L,1}$ is SM1 and $SI_{R,1}$ is SM2. If $L_{B,1}$ = 0 and $R_{B,1}$ = 0 then there is no secret message at this pixel.



Pixel	Stego Image (SI)								SM Group	Stego Image (SI)			
1	0	1	0	1	0	1	0	1	SM2		0	1	
2	0	1	1	1	0	0	1	1	SM1		1	1	
3	1	1	0	1	1	0	0	0	N/A		-	-	
4	0	0	1	1	0	0	1	0	SM3		0	0	
5	1	0	1	1	0	0	1	0	SM4		1	0	
6	1	1	0	0	1	1	1	1	SM5; SM6	1	1	0	0
7	1	1	0	1	0	1	0	1	SM8		0	1	
8	1	1	0	0	1	1	1	0	SM7		1	0	
9	0	0	1	1	0	1	0	0	N/A		-	-	
10	1	1	1	1	0	0	1	0	SM9		1	1	
11	1	0	1	1	1	0	0	0	SM10		0	0	
12	1	1	1	0	0	1	1	0	SM11		1	1	
13	1	0	1	1	1	0	0	1	SM12		0	1	
14	1	0	1	1	0	1	1	0	SM13		1	0	
15	1	1	1	0	0	0	0	0	SM14		0	0	
16	0	1	1	1	1	1	1	1	SM15; SM16	0	1	1	1

SECRET MESSAGE (SM)

SM	SM1		SM2		SM3		SM4		SM5		SM6		SM7		SM8		SM9		SM10		SM11		SM12		SM13		SM14		SM15		SM16	
Bits	1	1	0	1	0	0	1	0	1	1	0	0	1	0	0	1	1	1	0	0	1	1	0	1	1	0	0	0	0	1	1	1

Figure 4. Stego Image and Secret Message

For example, as shown in Figure 4, the bit value in the first pixel of the stego image is (010101101). $L_{B,1}$ = 0 and $R_{B,1}$ = 1 then the bit value in $SI_{R,1}$ = 01 is as SM2. Then the 2-LSB of the next pixel is SM1. For the 3rd pixel stego image, $SI_{L,3}$ = 0 and $SI_{R,3}$ = 0, which means that there is no secret bit in this block. The same process is repeated for the stego image to get the entire secret message.

3. EXPERIMENT AND ANALYSIS

The proposed system is described in this section. We have implemented using Intel(R) Core(TM) i5-7200u 2.50 GHz CPU 2 cores and 4 threads with 8 GB RAM. Experiments were conducted at the D4 Politeknik Elektronika Negeri Surabaya building. The wifi module used is a standard TL-WN722N wireless adapter with a frequency of 2.4 GHz. There are two scenarios used, namely the static scenario and the dynamic scenario shown in Figure 5. Alice is on the 1st floor in this scenario, while Bob and Eve are on the 2nd floor with the initial distance between Bob and Eve 1 meter. In static conditions, the three entities do not move, while in dynamic conditions, Bob moves in the hallway as far as 4 meters. The image used for this paper uses a standard test image (***The Standard Test Image Frequent Use in Literature*, n.d.)** with a grayscale base colour as the cover-image, i.e., cameraman, Lena, jetplane, and living room, shown in Figure 6.

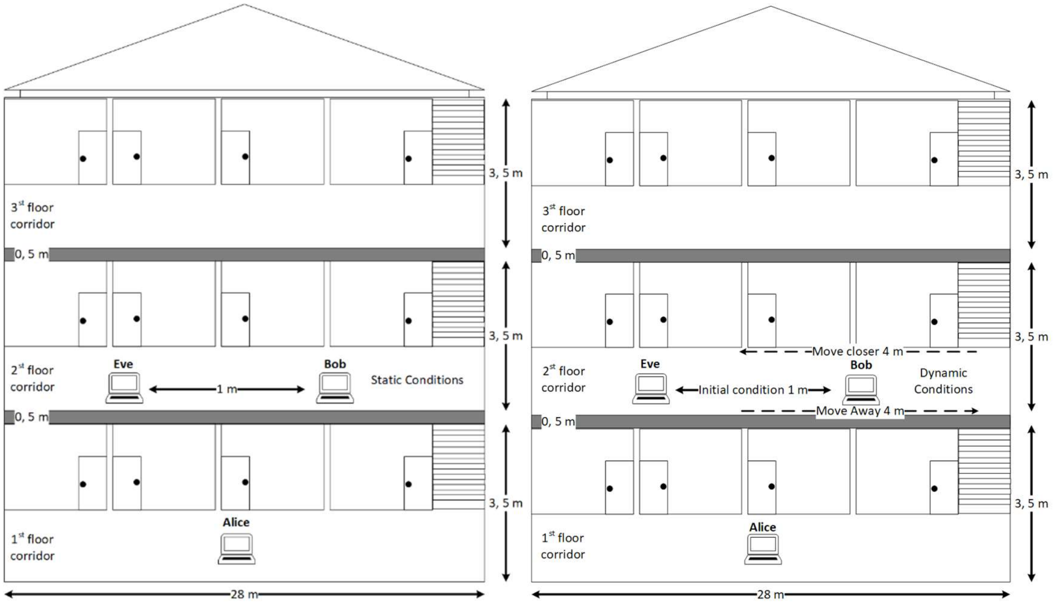


Figure 5. Static Condition (left) and Dynamic Condition (right)

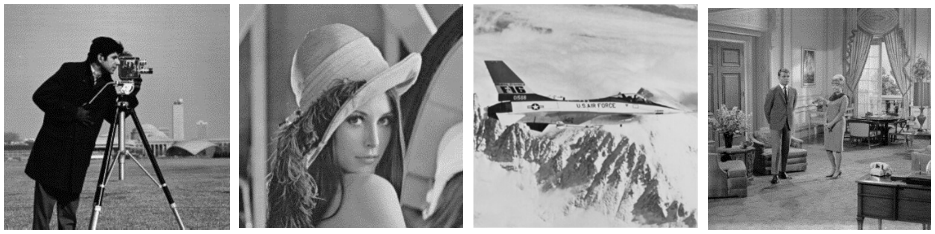


Figure 6. Cover Image Cameraman, Lena, Jetplane, and Livingroom From (*The* Standard *Test Image Frequent Use in Literature*, n.d.) For Experiment

3.1 Channel Probing Result

The results of the measurement of signal strength used in this research are 1400 RSS for static conditions and dynamic conditions shown in the Table 1. The signal strength obtained is in the range of -28 to -83 for static conditions, while in dynamic conditions, it is in the range of -25 to -80. 100 RSS data displayed in the graphical form shown in Figure 7. It can be observed that in dynamic conditions, the shape of the graph fluctuates when compared to static conditions due to moving Bob's measurement.

Table 1. RSS Obtained From Channel Probing

No	Static				Dynamic			
	Alice	Bob	Eve to Alice	Eve to Bob	Alice	Bob	Eve to Alice	Eve to Bob
1	-65	-58	-67	-39	-74	-64	-65	-36
2	-63	-59	-67	-39	-73	-65	-65	-36
3	-62	-57	-67	-39	-70	-64	-64	-35
4	-63	-57	-67	-39	-70	-65	-64	-35
5	-62	-55	-69	-38	-71	-65	-64	-35
6	-62	-57	-71	-38	-72	-64	-64	-35
7	-62	-57	-72	-40	-70	-66	-67	-35
8	-62	-57	-72	-39	-74	-66	-66	-34
…	…	…	…	…	…	…	…	…
1400	-73	-68	-69	-32	-72	-53	-53	-35

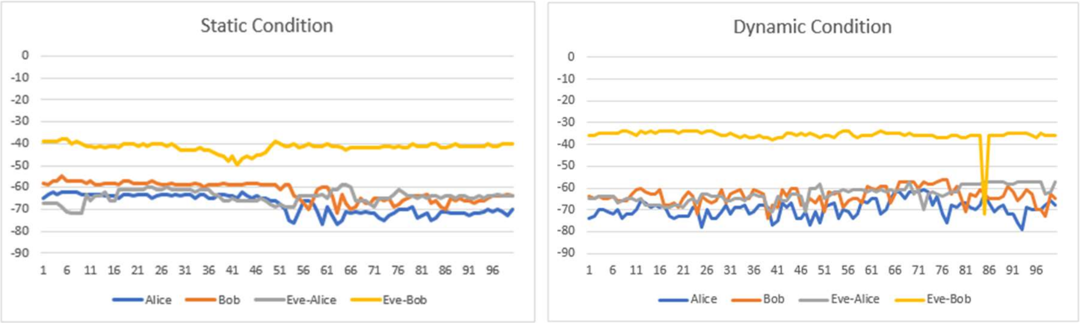


Figure 7. RSS Graphic in Static and Dynamic Condition

3.2 Performance SKG

SKG performance in this study was measured using Key Disagreement Rate (KDR), Key Generation Rate (KGR) and randomness **(Sumadi et al., 2020)**. The percentage of key inequalities generated between the two users (Alice and Bob) is known as the KDR. The number of mismatched bits between the two devices is denoted by M_{bits}, and the total number of RSS obtained is denoted by R_{bits}. The KDR formula shown in Equation (3). KGR is a calculation of the number of bits generated in seconds. The number of RSS sent and received between 2 users is denoted by R_{bits}, and the computation processing time is denoted by T. The KGR formula shown in Equation (4).

$$KDR = \frac{M_{bits}}{R_{bits}} \text{ x } 100 \tag{3}$$

$$KGR = \frac{R_{bits}}{T} \tag{4}$$

Table 2. The Result of KDR, KGR and Number of Key

Scenario	*KDR(%)*	*KGR(bits/s)*	*Number of Key*
Static	20.07	12.73	4
Dynamic	17.86	10.45	4

In static conditions, the KDR value is 20.07%, while in dynamic conditions, the KDR value is 17.86%. In static conditions, the KGR value was obtained 12.73, while in dynamic conditions, the KGR value was 10.45. The number of keys generated in static and dynamic conditions is four keys, with a length of each key is 256 bits. As shown in Table 2.

The total keys obtained were then performed a randomness test using the NIST test. The standard of randomness that must be passed is 0.01 tested with the seven parameters shown in the Table 3. The key with the highest approximate entropy is selected and used as a steganography key. The results of the NIST test are shown in Table 3. The key used in static conditions is Key 2, with an approximate entropy value of 0.96. The key used in dynamic conditions is Key 3, with an approximate entropy value of 0.79.

Table 3. The Result of NIST Test in Static and Dynamic Condition

Parameters	*Static Condition*				*Dynamic Condition*			
	Key1	*Key2*	*Key3*	*Key4*	*Key1*	*Key2*	*Key3*	*Key4*
Frequency	0.32	0.89	0.62	0.12	0.26	0.62	0.32	0.45
Block Frequency	0.93	0.96	0.89	0.02	0.43	0.72	0.84	0.83
Runs	0.35	0.80	0.70	0.49	0.56	0.79	0.57	0.51
Longest Runs	0.37	0.80	0.58	0.16	0.89	0.03	0.18	0.39
Cumulative Sum (Forward)	0.63	0.86	0.75	0.52	0.30	0.69	0.52	0.86
Cumulative Sum (Reversed)	0.30	0.86	0.95	0.08	0.42	0.91	0.52	0.75
Approximate Entropy	0.27	0.96	0.95	0.17	0.20	0.48	0.79	0.26

3.3 Performance Steganography

In this section, steganography performance is calculated using Mean Square Error (MSE), Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR), and computation time. PSNR is used to assess how good is the visual distortion or the quality value of the cover-image that has been inserted with secret data. The PSNR formula shown in Equation (5). The higher the resulting PSNR value, the lower the image distortion level. Conversely, if the lower PSNR value is generated, the cover-image and stego image will have large distortion.

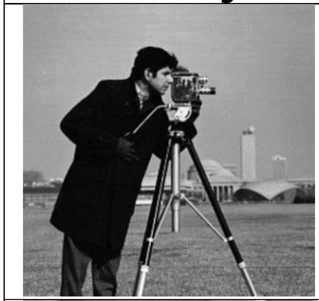
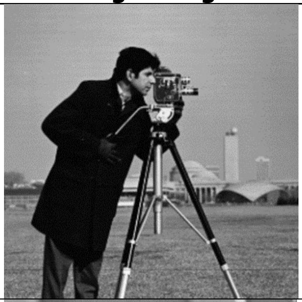
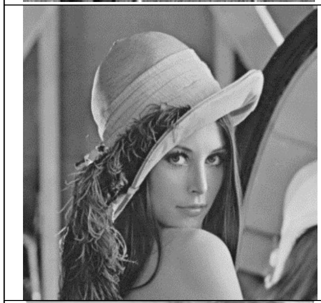
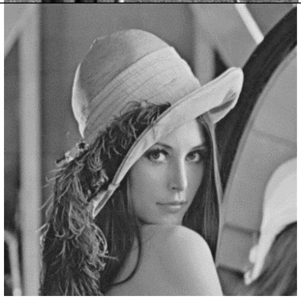
$$PSNR = 10\ \log_{10} \frac{255^2}{MSE} \tag{5}$$

$$MSE = \sum_{i=1}^{w \text{ x } h} \frac{(p'_i - p_i)^2}{W \text{ x } H} \tag{6}$$

$$RMSE = \sqrt{MSE} \tag{7}$$

MSE is the difference in average size squared from the pixel intensity of cover-image and stego image. The MSE formula shown in Equation (6), where p'ᵢ dan p is the pixel value in the cover-image and the stego image, respectively. *W* is the width of the grayscale image and *h* is the height of the grayscale image. The image quality improves as the MSE value decreases, while a high MSE value causes image distortion. The MSE value is inversely proportional to the PSNR value, meaning that the higher the PSNR value, the lower the MSE value. RMSE is the square root of the mean of all the errors squared, RMSE formula shown in Equation (7). In Table 4. shows the MSE and RMSE obtained on cover-image and stego image. As a result, a higher PSNR is regarded as desirable or indicative of low image distortion. In Figure 8 shows the PSNR results from images of Cameraman, Lena, Jetplane, and Livingroom by inserting 2,097,152 bits of messages or 262,144 characters. From the experiments conducted on four different images, it was obtained an average PSNR with a value of 43.07 dB.

Table 4. The Result of MSE Cover Image and Stego Image

Cover Image	Stego Image	MSE Value	RMSE Value
		3.24	1.80
		3.16	1.77
		3.31	1.81
		3.11	1.76

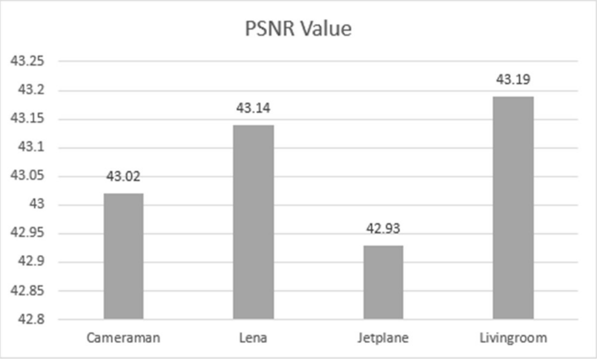


Figure 8. PSNR Value of Cover and Stego Image

In this research, the computation time was tested using a standard image (***The Standard Test Image Frequent Use in Literature*, n.d.)** is grayscale with a size 512x512. An experiment was conducted by changing the size of the message inserted with a variation of the size of a quarter of cover-image size (65.536 characters), half the size of the cover-image (131.072 characters), three-quarters of the cover-image size (196.608 characters), and full-size cover-image (262.144 characters). Obtained the variable computation time is shown in Table 5.

Table 5. The Result of Time Computation

Image	Length Message (Char)	Length Bit (bits)	Time Computation (second)			Total Time Computation (second)
			Key Generation	Embedding	Extraction	
Cameraman	65.536	524.288	110.49	5.51	0.68	116.68
	131.072	1.048.576	110.5	16.36	0.65	127.51
	196.608	1.572.864	110.52	30.96	0.67	142.15
	262.144	2.097.152	110.54	42.05	0.62	153.21
Lena	65.536	524.288	110.5	6.71	0.58	117.79
	131.072	1.048.576	110.52	23.5	0.64	134.66
	196.608	1.572.864	110.54	44.18	0.81	155.53
	262.144	2.097.152	110.56	63.94	0.62	175.12
Jetplane	65.536	524.288	110.49	5.39	0.56	116.44
	131.072	1.048.576	110.51	15.3	0.52	126.33
	196.608	1.572.864	110.52	28.23	0.53	139.28
	262.144	2.097.152	110.55	39.56	0.53	150.64
Living Room	65.536	524.288	110.5	7.9	0.61	119.01
	131.072	1.048.576	110.51	30.11	0.72	141.34
	196.608	1.572.864	110.53	57.78	0.7	169.01
	262.144	2.097.152	110.55	89.26	0.64	200.45

In the embedding process, the greater the size of the inserted message, the longer the computation time. In the key generation process, an average time of 110.52 seconds was obtained to generate keys. In the embedding process, the average computation time for 65,536 characters is 6.38 seconds, for 131,072 characters, it is 21.32 seconds, for 196,608 characters, it is 40.29 seconds, and for 262,144 characters, it is 58.70 seconds to produce a stego image. In the extraction process, an average computation time of 0.63 seconds is obtained to get secret message. By using a 512x512 image in grayscale, it can insert messages up to 2,097,152 bits.

4. CONCLUSION

In this research, we propose a key generation technique in the steganography process by utilising RSS in wireless communication. The key generation uses 1400 RSS data obtained on the probing channel so that it is able to generate four keys. From the four keys, the best one is obtained with an approximate entropy value of 0.96 for static conditions and 0.79 for dynamic conditions. In the key generation process, an average time of 110.52 seconds was obtained to generate keys. In the embedding process, the average computation time for 65,536 characters is 6.38 seconds to produce a stego image. In the extraction process, an average computation time of 0.63 seconds is obtained to get a secret message. By using a 512x512 image size, it can insert messages up to 2,097,152 bits. For future research, we plan to implement the system into IoT devices with the aim of securing the sensor data that is sent.

ACKNOWLEDGEMENTS

The author would like to thank the Telecommunication Laboratory Politeknik Elektronika Negeri Surabaya for the assistance of the tools to carry out this research.

REFERENCES

Bai, J., Chang, C. C., Nguyen, T. S., Zhu, C., & Liu, Y. (2017). A high payload steganographic algorithm based on edge detection. *Displays*, *46*, 42–51. https://doi.org/10.1016/j.displa.2016.12.004

Brassard, G., & Salvail, L. (1994). Secret-key reconciliation by public discussion. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *765 LNCS*, 410–423. https://doi.org/10.1007/3-540-48285-7_35

Dewi, I. T., Sudarsono, A., Kristalina, P., & Yuliana, M. (2020). Higher Rate Secret Key Formation (HRKF) based on Physical Layer for Securing Vehicle-to-Vehicle Communication. *EMITTER International Journal of Engineering Technology*, *8*(1), 140–160. https://doi.org/10.24003/emitter.v8i1.493

El-Dairi, M., & House, R. J. (2020). Optic Nerve Hypoplasia. In *Handbook of Pediatric Retinal OCT and the Eye-Brain Connection,* (pp. 285–287). Elsevier. https://doi.org/10.1016/B978-0-323-60984-5.00062-7

Hussain, M., Wahab, A. W. A., Javed, N., & Jung, K. H. (2018). Recursive Information Hiding Scheme Through LSB, PVD Shift, and MPE. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, *35*(1), 53–63.

https://doi.org/10.1080/02564602.2016.1244496

Khodaei, M., Sadeghi Bigham, B., & Faez, K. (2016). Adaptive Data Hiding, Using Pixel-Value-Differencing and LSB Substitution. *Cybernetics and Systems*, *47*(8), 617–628. https://doi.org/10.1080/01969722.2016.1214459

Liu, Y. N., Zhong, Q., Xie, M., & Chen, Z. Bin. (2018). A novel multiple-level secret image sharing scheme. *Multimedia Tools and Applications,* *77*(5), 6017–6031. https://doi.org/10.1007/s11042-017-4512-5

Sudarsono, A., Yuliana, M., & Kristalina, P. (2019). A Reciprocity Approach for Shared Secret Key Generation Extracted from Received Signal Strength in the Wireless Networks. *2018 International Electronics Symposium on Engineering Technology and Applications, IES-ETA*, (pp. 170–175). https://doi.org/10.1109/ELECSYM.2018.8615568

Sumadi, M. T., Yuliana, M., & Sudarsono, A. (2020). Performance Improvement Based on Modified Lossless Quantization (MLQ) for Secret Key Generation Extracted from Received Signal Strength. *2020 International Electronics Symposium (IES)*, (pp. 190–194). https://doi.org/10.1109/IES50839.2020.9231640

The Standard Test Image Frequent Use in Literature. (n.d.). http://www.imageprocessingplace.com/root_files_V3/image_databases.htm (access Apr. 17, 2021)

Tondwalkar, A., & Vinayakray-Jani, P. (2016). Secure Localisation of Wireless Devices with Application to Sensor Networks using Steganography. *Procedia Computer Science*, *78*, 610–616. https://doi.org/10.1016/j.procs.2016.02.107

Wei, Y., Zeng, K., & Mohapatra, P. (2013). Adaptive Wireless Channel Probing for Shared Key Generation Based on PID Controller. *IEEE Transactions on Mobile Computing*, *12*(9), 1842–1852. https://doi.org/10.1109/TMC.2012.144

Xiang, T., Hu, J., & Sun, J. (2015). Outsourcing chaotic selective image encryption to the cloud with steganography. *Digital Signal Processing*, *43*, 28–37. https://doi.org/10.1016/j.dsp.2015.05.006

Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017). Comprehensive study of symmetric key and asymmetric key encryption algorithms. *2017 International Conference on Engineering and Technology (ICET)*, *2018-Janua*, (pp. 1–7). https://doi.org/10.1109/ICEngTechnol.2017.8308215

Yuliana, M., Wirawan, & Suwadi. (2019a). An Efficient Key Generation for the Internet of Things Based Synchronized Quantization. *Sensors,* *19*(12), 2674. https://doi.org/10.3390/s19122674

Yuliana, M., Wirawan, W., & Suwadi, S. (2019b). Performance Analysis of Loss Multilevel Quantization on the Secret Key Generation Scheme in Indoor Wireless Environment. *International Journal on Advanced Science, Engineering and Information Technology*, *9*(1), 100. https://doi.org/10.18517/ijaseit.9.1.7583

Zakaria, A., Hussain, M., Wahab, A., Idris, M., Abdullah, N., & Jung, K.-H. (2018). High-Capacity Image Steganography with Minimum Modified Bits Based on Data Mapping and LSB Substitution. *Applied Sciences*, *8*(11), 2199. https://doi.org/10.3390/app8112199