Reed Solomon Codes for the Reconciliation of Wireless PHY Layer based Secret Keys

Michelle Fernando*, Dhammika Jayalath*, Seyit Camtepe† and Ernest Foo*
Queensland University of Technology (QUT), Australia
michelle.fernando, dhammika.jayalath, e.foo @ qut.edu.au
Data61 / CSIRO , Australia
seyit.camtepe@data61.csiro.au

***Abstract*—This paper proposes a key reconciliation mechanism using Reed Solomon code to improve the effectiveness of secret key generation based on Received Signal Strength (RSS) variations in a wireless channel between two communicating parties. We use a two threshold quantization algorithm which can operate as a lossless system improving the secret bit extraction rate. We present our Reed Solomon (RS) code based algorithm for reconciling the independently derived secret keys through sharing of only the syndrome bits. We evaluate our reconciliation algorithm using simulated channel measurements and real environment data gathered between an unmanned aerial vehicle (UAV) and a controller in a semi mobile environment. We show that by selecting appropriate parameters of the RS code, it is possible to generate matching keys at the transmitter and the receiver which can be used to secure the communication at wireless PHY or upper layers.**

***Index Terms*—secret key reconciliation, wireless PHY layer, secret key generation, Reed-Solomon (RS) code, received signal strength (RSS)**

I. Introduction

Key management is one of the most challenging problems in wireless network security. While distributed key management means increased communication, computation and storage overhead, centralized one requires maintaining an additional infrastructure. From this view, extracting secret keys using unique randomness of a transmission channel is an appealing approach due to its simplicity. However, extracting perfect keys with minimal bit mismatch using PHY properties has its own challenges such as channel estimation, validity of channel reciprocity and correlation of secret bits. *Premnath et al.* [1] evaluated the effectiveness of secret key extraction from Received Signal Strength (RSS) variations on a wireless channel in a variety of real world environments. They proved that in dynamic scenarios where the two devices are mobile, and/or where there is a significant movement in the environment, high entropy bits can be obtained quickly. But, due to variations in the actual measurements, the extracted keys do not match at all times. In order to achieve a minimal bit mismatch between the generated keys, communicating parties have to exchange a large amount of information, and they require a large number of measurements due to lossy nature of the quantization methods [1, 2]. Therefore, a key reconciliation protocol must be employed to eliminate bit mismatches [3].

Reed-Solomon (RS) codes are widely used in wireless communication systems. Although RS codes are used for enhancing wireless PHY security, there is no published literature that investigates RS codes as a secret key reconciliation mechanism in wireless PHY security environments. Cascade is one of the most famous information reconciliation protocol. For example, an interactive cascade protocol for key reconciliation is proposed in [1] which requires the probability of the mismatch is known a priori to determine a suitable block size. Others include Low Density Parity Check (LDPC) [4], Cyclic Redundancy Check (CRC) [5] and Turbo Code [5] based key reconciliation protocols. All of the above mentioned protocols require repeated steps of interaction and information exchange between nodes resulting in information leakage relating to the derived secret key. Combination of advanced signal processing techniques and error control coding has also been used successfully for various physical layer security applications [6, 7].

Focus of this paper is to eliminate the bit mismatches in the keys extracted from RSS variations in a wireless channel between two communicating parties. We use two thresholds based on Adaptive Secret Bit Generation (ASBG) [1] and Level Crossing [8] quantization methods to improve the secret bit extraction rate. We present our Reed Solomon (RS) code based key reconciliation algorithm which only exchanges the syndrome bits. We analyze the relation between the key extracted versus information leaked to the adversary to ensure that the final reconciliated key will be sufficiently long for computational security. We evaluate our solution by using simulated channel measurements and real environment data gathered between an unmanned aerial vehicle (UAV) and a controller in a semi mobile environment. The rest of the paper is organized as follows. First, we describe the system and adversary models in Section II. Then in Section III, the key extraction and our RS code based key reconciliation algorithms are presented. In Section IV, we evaluate our algorithm, and Section V concludes the paper.

II. System Model

The system model consists of two legitimate wireless nodes, Alice and Bob, who wish to extract a secret key (S) based on the received signal strength (RSS) in the presence of a passive eavesdropper, Eve. Focusing primarily on the key reconciliation, we present a complete key generation scheme consisting of three components: (1) secret key extraction

through quantization, (2) key reconciliation and (3) privacy amplification, that will minimize information sharing and computational overhead while limiting the information leaked to Eve.

A. Physical Layer Model

To establish a shared key, Alice and Bob transmit probe packets to each other and measure the variations of the wireless channel RSS. These measurements are taken over different *coherence time* where the wireless channel remains correlated. The observed signals for Alice and Bob in coherence time can be represented by:

$$y_A(t) = (h_{BA}(t) * x_B(t)) + n_A(t) \quad (1)$$

$$y_B(t) = (h_{AB}(t) * x_A(t)) + n_B(t) \quad (2)$$

where $x_A(t)$ and $x_B(t)$ are signals transmitted by Alice and Bob, respectively. $n_A(t)$ and $n_B(t)$ represent noise in the channel. "*" denotes convolution. We denote Bob's estimate of the Alice-Bob channel as h_{AB}, Alice's estimate of the Bob-Alice channel as h_{BA}, Eve's estimate of the Alice-Eve channel as h_{AE}, and Eve's estimate of the Bob-Eve channel as h_{BE}. We assume $h_{AB}(t) \approx h_{BA}(t) = h(t)$ is the channel impulse response which is identical in both directions due to channel reciprocity property.

The measured RSS signal is then quantized to generate an initial secret bit sequence. The quantization is done based on specific thresholds. Different quantizers have been proposed in literature with different choices of thresholds [1, 2, 8]. We propose to use two quantization levels and extract a single bit from a single RSS measurement. Due to noise, interference, hardware differences, time variations in signal transmission, measurements and sampling, the extracted bit sequences at Alice and Bob may not be identical. Hence, a secret key reconciliation method must be used to correct the bits where the two bit streams differ. We propose Reed-Solomon (RS) code based encoder and decoder as our secret key reconciliation protocol. Because, RS codes are known to have robust nature for error detection and correction [9].

B. Adversarial Model

Eve is assumed to be a passive stationary eavesdropper. She can neither perform a man-in-the-middle attack nor cause any changes in the wireless environment which can lead to changes in the measured RSS values. Eve can measure both $h_{AE}(t)$ and $h_{BE}(t)$ at the same time as Alice and Bob measure $h(t)$, and attempt to derive $h(t)$ from her measurements. However, we assume that Eve is not close enough to Alice and Bob thus the channel observed by Eve is uncorrelated to the channel between Alice and Bob [8].

III. Secret Key Extraction and Secret Key Reconciliation Protocol

A. Secret Key Extraction

We assume that both Alice and Bob are time synchronized. They record RSS values while exchanging packets. Thus, for a given time interval, both Alice and Bob would have an equal number of RSS measurements. The secret key extraction algorithm used in this paper requires an adaptive secret key size *(L)*, scaling factor $\alpha \leq 1$ and a time-stamp marking the start of the measurements. Next, Alice and Bob quantize their recorded time series RSS measurements to generate their bit sequences. The quantization method used in this paper is based on the Adaptive Secret Bit Generation (ASBG) [1] and the Level Crossing Algorithm [8]. Both of these quantization methods use a threshold value and ignore RSS samples that lie above or below the threshold. Hence, they are called as lossy quantization methods. The method used in this paper is a lossless quantization method where each measured RSS value will produce a minimum of a single secret bit. To increase the secret bit rate, the secret extraction method can also be modified to extract multiple bits from a single RSS measurement.

Algorithm 1 Secret Key Extraction.

Input: L , α
Output: A cryptographic key $K_a \neq K_b$

Alice: Calculate mean and standard deviation for $Q_a(x[i])$
  $q_{a+} = mean + (\alpha \times$*standard deviation*$)$
  $q_{a-} = mean - (\alpha \times$*standard deviation*$)$
for $i = 1$ to L **do**
  if $Q_a(x[i]) > q_{a+}$ or $Q_a(x[i]) < q_{a-}$ **then**
    $K_a[i] = 1$
  else
    $K_a[i] = 0$
  end if
end for

Bob: Calculate mean and standard deviation for $Q_b(x[i])$
  $q_{b+} = mean + (\alpha \times$*standard deviation*$)$
  $q_{b-} = mean - (\alpha \times$*standard deviation*$)$
for $i = 1$ to L **do**
  if $Q_b(x[i]) > q_{b+}$ or $Q_b(x[i]) < q_{b-}$ **then**
    $K_b[i] = 1$
  else
    $K_b[i] = 0$
  end if
end for

Algorithm 1 details the key extraction scheme for communicating parties, Alice and Bob. They collect RSS measurements, and the i^{th} measured element corresponds to the successive probes sent by Alice and Bob, respectively. Algorithm 1 consists of the following steps:

1) Alice and Bob will first decide the number of bits per key *(L)* based on the number of measurements, and next individually calculate the mean and standard deviation for the measured RSS values. The upper and lower threshold values (q_+ and q_-) will be calculated by using these mean and standard deviation and the chosen α.

2) Alice and Bob will independently evaluate the measured RSS value against the computed threshold values and assign either '1' to '0' to derive the secret key. Therefore, if $Q(x[i]) > q_+$ or $Q(x[i]) < q_-$ then $K[i] = 1$ else 0 will be assigned for each of $K[i]$ $(i = 1, \ldots, L)$.

In our experiments, the RSS measurement rate was one sample per second. In this case, the lossless quantization method of this paper can extract a single bit from each measured RSS value resulting a secret bit extraction rate of 1bit/s. Thereby, Alice and Bob will be independently extracting a secret key without exchanging information such as indices and statistical parameters. The secret bit extraction rate can be increased by use of higher sampling rate or multiple bit extraction.

B. Secret Key Reconciliation Protocol

One of the key assumptions of the key extraction based on wireless physical layer properties is the channel reciprocity principle. However, slightly varied channel estimates are obtained at two nodes in practice due to non-symmetric noise and channel interferences, variations in hardware at the nodes, and difficulties in simultaneous estimation of channel caused by the half-duplex operation. These lead to bit mismatches in the secret keys derived by Alice and Bob. Hence, a secret key reconciliation method should be used to correct these bit mismatches. Cascade protocol uses a lossy quantization method and exchanges the indices relating to the dropped RSS values and parity bits between Alice and Bob as the secret key reconciliation method [1]. We propose to use Reed Solomon (RS) codes as the secret key reconciliation protocol where only the syndrome bits are exchanged to reduce the communication overhead and the amount of information leaked to Eve.

Reed-Solomon (RS) codes are a family of block error correcting codes used in data communication, data storage and digital television. RS encoder takes a message M with k symbols and transforms it to a polynomial code consisting of $k+n$ symbols. The original message forms the first k symbols of the polynomial code. The remaining n symbols are called the syndromes. In the classical communication scenario, $k+n$ symbols are transmitted through an erroneous channel to the receiver. RS decoder at the receiver uses the received $k+n$ symbols to correct bit errors in the message and syndrome parts. The syndromes cannot be used alone to recover the original message (i.e., remaining coefficients of the polynomial code) and the received message should be used together with the syndromes to fix the bit errors. In the key reconciliation scenario, the message M corresponds to the keys K_a and K_b extracted by Alice and Bob, respectively. Since, Alice and Bob knows the message (i.e., keys K_a and K_b, respectively), they just need the syndrome to correct any discrepancies between their respective keys. Since our solution does not transmit keys or any other cryptographic transformation of the keys, it reduces the communication overhead. Syndrome alone is not sufficient to recover the remaining coefficients of the code polynomial. Thus, the information leaked to Eve is reduced substantially.

Table I
REED SOLOMON CODE PARAMETERS FOR EQN (3) WHEN x = 128.

bits/symbol	number of symbols				bits
m	n	k_{128}	t	s	L
3	7	31	-	-	-
4	15	27	-	-	-
5	31	28	1	3	140
6	63	36	13	27	216
7	127	55	36	72	385
8	255	96	79	159	768
9	511	180	165	331	1620
10	1023	350	336	673	3500

Specifically, RS encoder takes a message M (K_a in our scheme) with k symbols (m bits/symbol) and adds extra bits to extend it to $n = 2^m - 1$ symbols. The message M with k symbols is represented with the message polynomial $M(x) = M_{k-1}x^{k-1} + \ldots + M_1x + M_0$ where each coefficients are m-bit symbols from GF(2^m). RS encoder uses a generator polynomial $G(x) = (x+\alpha)(x+\alpha^2)\ldots(x+\alpha^{2t})$ with roots $\alpha, \ldots, \alpha^{2t}$. Transmitted code polynomial is calculated as $T(x) = M(x) \times x^{n-k} + R(x)$ where $R(X) = M(x) \times x^{n-k}\ mod\ G(x)$. Given $T(x) = M_{k-1}x^{n-1} + \ldots + M_0x^{n-k} + S_{n-k-1}x^{n-k-1} + \ldots + S_0$, RS codes require Alice to send the coefficients $(M_{k-1}, \ldots, M_0, S_{n-k-1}, \ldots, S_0)$ to Bob. On receiving these n symbols, Bob's RS decoder calculates the syndromes, finds error locations and magnitudes and finally corrects up to $t = (n-k)/2$ errors to recover the original data. However, in our reconciliation algorithm, Alice only sends the coefficients $(S_{n-k-1}, \ldots, S_0)$ (syndrome S_a in this work) to Bob. On receiving $n-k$ symbols, Bob's RS decoder uses its own version of the message M' (K_b in our scheme) to find and correct errors.

The message length *(k)* and error correcting capability *(t)* of the RS code would be dependent of the chosen secret key size *(L)*. Table I indicate the error correcting capability of the RS code based the number of bits per symbol *(m)*, the minimum size of message length in symbols *(k_x)* where $x = 128$ (recommended minimum key size in symmetric ciphers such as AES) and codeword length in symbols *(n)*. In selecting the value for k_x, it was assumed that the length of the secret key *(L)* chosen would be large enough to ensure that even if Eve has the knowledge of the full syndrome S_a, she would not be able to predict the secret key using a brute force attack. Therefore as shown in Fig. 1, it is assumed that $(k-t)$ would be the portion of the key which Eve could not predict. Equivalently, it is the portion of the key which cannot be corrected using RS code as the secret key reconciliation protocol.

The condition for the minimum message length in symbols *(k_x)* for which Eve could not be able to predict the secret key through brute-force attack is presented by Eq. 3:

$$m \times (k-t) \geq x, \quad t = \frac{(n-k)}{2}, \quad k_x \geq \frac{1}{3}\left(\frac{2x}{m} + n\right) \quad (3)$$

where $x > 80$ to assure computational hardness in brute force attack (in practice $x \geq 128$ is recommended) and $n = 2^m - 1$.

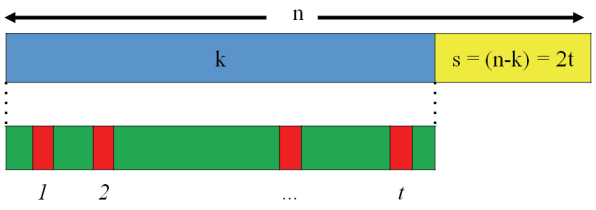

Figure 1. Codeword Length (n), Message Length (k) and Error Correcting Capacity(t).

It can be seen from Table I that a suitable RS code could not be selected for $m = 3$ and $m = 4$ as they cause $n - k < 0$. Additionally, the error correcting capability (t) is only 1 symbols for $m = 5$. For $m = 6$, the corresponding value for n is 63 symbols per codeword. We have calculated that the minimum value for k should be 36 symbols per message resulting in a minimum of 216 bits in the derived key which satisfies $m \times (k - t) \geq 128$. Therefore the error correcting capability of the RS decoder for $L = 216$ bits would be 13 symbols or 78 bits. It should be noted that though the RS (63,36,13) can correct 78 bits in total, all these mismatched bits should occur inside 13 symbols. Algorithm 2 describes the secret key reconciliation protocol with the following steps:

1) Based on L, suitable values for m and k should be selected to satisfy $m \times (k - t) \geq x$ where $x = 128$ or 80 bits and $t \times m$ is the maximum number of bit mismatches between keys K_a and K_b.
2) The secret keys derived at both Alice and Bob will be appended with zero bits at the end to ensure that the new length of the key is divisible by m (i.e., $k = L/m$ symbols with m bits/symbol). RS encoder/decoder do not need to know the number of zero bits appended. Hence, this information is not disclosed to prevent Eve to gain any extra advantage.
3) Alice computes the syndrome bits S_a, by encoding the secret key K_a, through a RS encoder *RS (n, k, t)*.
4) Alice sends S_a to Bob.
5) Bob decodes his secret key K_b through a RS decoder *RS (n, k, t)* using the received S_a to derive K_a.

C. Privacy Amplification

Key extraction and reconciliation produces matching keys with sufficient number of bits for Alice and Bob. However, cryptographic keys should have high entropy meaning that bits, bit sequences or symbols in the key should be uncorrelated so that an adversary may not have any advantage in his/her attempts to discover the key. To remove any correlation, we recommend to use hash functions with low collision rates, i.e, $Secret = Hash(K_a|Salt) = Hash(K_b|Salt)$. The salt value can be either the previous key or some random value publicly negotiated along with the syndromes. Alternatively, symmetric encryption on keys can be used to increase the entropy, i.e., $Secret = Enc_{K_a}(Salt) = Enc_{K_b}(Salt)$.

Algorithm 2 Secret Key Reconciliation.

Input: k, m where $n = 2^m - 1$ and $t = (n - k)/2$
Output: A cryptographic key $K_a = K_b$

Alice: Select m and k such that $k = L/m$
$m \times \left[k - \frac{(n-k)}{2}\right] \geq x$ where *x= 128 or 80*
Append $K_a(i) = 0$ for $i = 0, \ldots, L$
Encode $K_a \longrightarrow$ *RS(n,k,t)* $\longrightarrow \mathbf{S_a}$

Bob: Select m and k such that $k = L/m$
$m \times \left[k - \frac{(n-k)}{2}\right] \geq x$ where *x= 128 or 80*
Append $K_b(i) = 0$ for $i = 0, \ldots, L$
Decode $K_b \longrightarrow$ *RS(n,k,t) with* $S_a \longrightarrow \mathbf{K_a}$

IV. Results and Discussion

In this section, we evaluate the performance of the key extraction algorithm and proposed key reconciliation protocol in terms of bit mismatch rate and error correcting capability, respectively. Although the bit extraction method is capable of using a moving average for calculating the statistical properties, we evaluate the threshold values based on the entire key length.

We simulated a random channel with Rayleigh fading and Doppler frequency 100 Hz to generate RSS values suitable for key extraction. In our simulation, $x_A(t) = x_B(t)$ but due to randomness of the channel $y_A(t) \neq y_B(t)$. The correlation coefficient between $y_A(t)$ and $y_B(t)$ was 60%.

We used real environment RSS measurement data between a UAV and a controller. There was little movement in the surround area and the UVAs were moving slow. The correlation coefficient between the measured RSS values was 87%.

The key extraction algorithm uses two thresholds, q_+ and q_-. Fig. 2 shows the variation of the bit mismatch rate with the Scaling Factor (α) for different values of L. It can bee seen from the graph that for α values of 0.2 to 0.4 and α greater than 0.7, we can achieve bit mismatch rates below 0.35. When comparing the bit mismatch rates of quantization methods stated in Section III, and with reference to the data presented in *Premnath et al.* [1], it can be clearly seen that the changes made in the key extraction algorithm have improved the bit mismatch rate in semi-mobile environment.

It has also been pointed out that the high error detection capability of RS codes makes these codes very useful to correct the bit mismatch between K_a and K_b. As discussed in Section III, when selecting the parameters for RS codes, the information leakage to Eve should be considered. In the worst case scenario, it can be assumed that Eve has the knowledge of full S_a. Thus, we defined the lower bound for *k* as k_x in Table I with reference to Eq. 3. As the value of *m* increases, it can be seen that k_x is becoming smaller compared to *n*; thus, resulting in a higher error correcting capability and a larger syndrome. Fig. 2 shows that the required error correcting

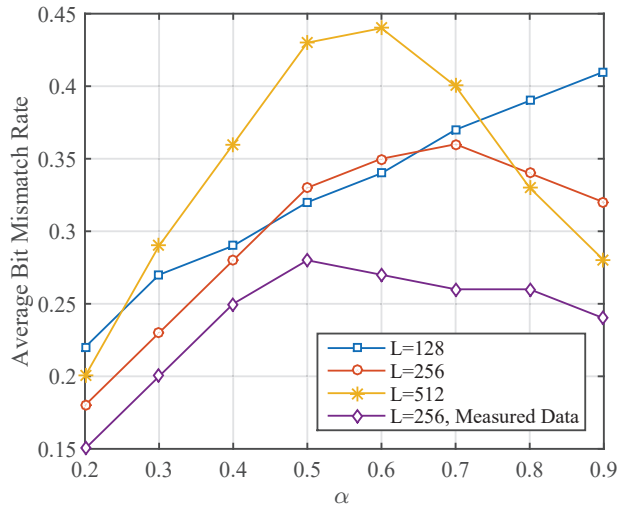

Figure 2. Average Bit Mismatch rate vs Scaling Factor for different values of L.

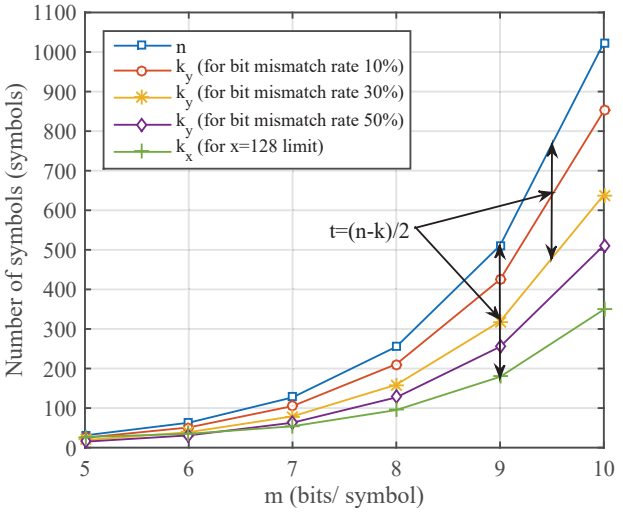

Figure 3. Comparison of Codeword Length (n) and Message Length limits (k_x and k_y) for different m.

capability based on actual bit mismatch rates are much lower than that corresponding to k_x. Therefore, we can select a higher value for k to minimize the syndrome size and to lower the amount of information shared between Alice and Bob. This way, we can arbitrarily choose a k_y value based on bit mismatch rate as stated in Table II where y = *bit mismatch rate*. In Fig. 3, we define a range for k based on k_x and k_y for different values of m. Therefore by looking at Fig. 3 and Table II, it can be noted that (1) Although k_x is not practical for values of m = 3, 4, and 5, based on the y, a suitable k_y can be defined, and (2) for other values of m, depending on k_y, a suitable k value could be selected which lies between k_x and k_y.

We now describe our validation efforts in typical semi-

Table II
CODEWORD LENGTH (n) AND MESSAGE LENGTH (k_x AND k_y) FOR DIFFERENT m.

m	***5***	***6***	***7***	***8***	***9***	***10***
n	31	63	127	255	511	1023
k(y= 10%)	25	51	105	211	425	851
k(y= 30%)	19	39	79	159	319	639
k(y= 50%)	15	31	63	127	255	511
$k_{x=128}$	28	36	55	96	180	350

The values for n, k_x and k_y above are given in terms of number of symbols. The values for m is given in terms of number of bits per symbol. For $m < 5$, the $(n-k) \times m \geq 128$ does not hold.

mobile environment. We derived a secret key with $L = 378$ for $\alpha = 0.8$ from the measured data. There were 87 bit errors between K_a and K_b resulting a bit mismatch rate of 0.23. First, we used RS code (127,55,36) for $m = 7$ to reconcile the bit mismatch between K_a and K_b. The errors of the analyzed K_a and K_b were distributed among 35 symbols. The total syndrome bits exchanged for key reconciliation were $S_a = 504$ bits. Then, we used the same K_a and K_b in cascade algorithm with a $block - size = 7$. $block - size$ is defined such that each block is likely to have no more than one error. It can be shown that for correcting all the errors, it could take 5 passes and with a maximum bit exchange of 160 bits (1 bit for each correct block + $(1 + log_2(7))$ bits for error blocks). When selecting the block size for cascade, for the block with maximum number of iterations, 5 parity bits would be exchanged including the last 2 bits. In summary, Cascade protocol is effective but it suffers from a high communication complexity and results in low throughput. Even with conservative estimates on packet size and network latency, this communication adds up. Whereas in RS code, we only have to exchange information only once and it is more effective in correcting burst errors than the cascade protocol.

V. CONCLUSION

In this paper, we presented a framework for the use of RS codes as a secret key reconciliation protocol in RSS based wireless PHY secret key generation. We evaluated the effectiveness of using RS codes as a secret key reconciliation protocol in secret key extraction method developed by us using two levels of threshold values. The key extraction technique was implemented as a lossless quantization method to improve the secret bit extraction rate using RSS variation in a wireless channel using simulated channel measurements. We evaluated the secret bit extraction using real measurements between two UAVs. Our experimental results showed that the higher error correcting capability of RS codes is effective in correcting the bit mismatch between independently derived secret keys between two communicating devices. By selecting appropriate k values, it is possible to secure the information of the derived key even if the adversary is capable of intercepting the syndrome bits exchanged between Alice and Bob. We could further improve the rate of secret bit extraction by extracting multiple bits from each RSS measurement and also

eliminate the predictable statistical properties of the channel by introducing a suitable block size $< L$ for mean and standard deviation values to be used in calculating the threshold values.

REFERENCES

[1] S. Premnath, S. Jana, J. Croft, P. Gowda, M. Clark, S. Kasera, N. Patwari, and S. Krishnamurthy, "Secret key extraction from wireless signal strength in real environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 917 – 30, 2013.

[2] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *14th ACM Conference on Computer and Communications Security*, vol. 1, 2007, pp. 401–410.

[3] Z. Rezki, M. Zorgui, B. Alomair, and M. S. Alouini, "Secret key agreement: Fundamental limits and practical challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 72–79, 2017.

[4] J. Etesami and W. Henkel, "Ldpc code construction for wireless physical-layer key reconciliation," in *Communications in China (ICCC), 1st IEEE International Conference on*, Aug 2012, pp. 208–213.

[5] A. Ambekar, N. Kuruvatti, and H. Schotten, "Improved method of secret key generation based on variations in wireless channel," in *Int. Conf. on Systems, Signals and Image Processing (IWSSIP),*, April 2012, pp. 60–63.

[6] W. K. Harrison, J. Almeida, M. R. Bloch, S. W. McLaughlin, and J. Barros, "Coding for secrecy: An overview of error-control coding techniques for physical-layer security," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 41–50, Sept 2013.

[7] Y. Liu, H. H. Chen, and L. Wang, "Physical layer security for next generation wireless networks: Theories, technologies, and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 347–376, Firstquarter 2017.

[8] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *ACM Int. Conf. on Mobile Computing and Networking*, NY, USA, 2008, pp. 128 – 39.

[9] I. Shakeel, M. Hasna, A. Abu-Dayya, and A. Ghrayeb, "Reed-solomon coding for cooperative wireless communication," in *IEEE Int. Symp. on Personal Indoor and Mobile Radio Commun. (PIMRC)*, Sept 2010, pp. 1021–1026.