Real-time Physical Layer Secure Key Generation in a mmWave Communication System

Navaneetha C. Manjappa[1,2], Lara Wimmer[1], Nebojsa Maletic[1] and Eckhard Grass[1,2]
[1]IHP – Leibniz-Institut für innovative Mikroelektronik, Im Technologiepark 25, D-15236, Frankfurt (Oder), Germany
[2]Humboldt University of Berlin, Rudower Chaussee 25, D-12489 Berlin, Germany
Emails: {manjappa, wimmer, maletic, grass}@ihp-microelectronics.com

***Abstract*—Physical layer security is one of the trending research areas in tackling data security issues of wireless networks. The main focus of this work is to explore practically the possibilities and performance of secret key generation solutions exploiting wireless channel randomness. This paper describes the implementation of a real-time secure key generation algorithm in a 60 GHz millimetre wave (mmWave) communication system. The performance of the algorithm is evaluated systematically for different parameter combinations to find the optimal parameter values for the system in an indoor static office environment.**

***Index Terms*—Physical Layer Security (PLS), 60 GHz mmWave communication system, Real-time secure key generation.**

I. Introduction

With the emerging fifth-generation (5G) and beyond wireless technologies, it has become more crucial to ensure privacy and data security. In this regard, physical layer security is a research area offering a solution with a potentially very high security level. A wireless channel is more vulnerable to security attacks than a wired connection due to its public accessibility. Conventional cryptography techniques have the disadvantages of infrastructure, key distribution, and management overheads. In addition, these techniques depend on the computational complexity, which eavesdroppers can easily attack with advanced infrastructure [1], [2]. Conversely, emerging physical layer secure key generation (PLSKG) techniques can be a good option to secure data transmission with less complexity and no infrastructure overhead. These techniques utilise the common randomness and the reciprocity of the wireless channel between legitimate users to generate secure keys [2].

Existing techniques for secure key generation in sub-6 GHz bands use channel characteristics such as Received Signal Strength (RSS), Channel State Information (CSI), and Angle of Arrival (AoA). The performance of existing key generation schemes is limited by high bit mismatch rates at a low signal-to-noise ratio (SNR), high reconciliation overhead, and the potential of co-located eavesdropping in sub-6 GHz systems [3]. Recent advances in mmWave cellular networks offer more opportunities to resolve the above-stated challenges. The transmission range of mmWave systems is short, restricting the ability of the eavesdropper to wiretap [3], [4]. Further, L. Jiao *et al.* [5] proposed virtual AoA and angle of departure (AoD) characteristics of mmWave massive MIMO channels and analysed generation of secret keys using these properties theoretically. Nasser A. *et al.* [6] presented induced randomness-based key generation in static environments. In the direction of physical layer authentication, an algorithm based on Kalman filtering and maximum-a-posteriori (MAP) estimation was proposed in [7]. Recently, a system-level solution using a time-frequency filter bank-based key generation method was put forward in [8]. However, it is important to implement and test the PLS key generation schemes practically to evaluate the performance in a real transmission environment [2]. In this regard, the implementation of real-time secret key generation in the mmWave communication systems has more scope for investigation and is the motivation for our contribution.

In this work, we present the implementation and testing of real-time CSI-based secure key generation in a 60 GHz mmWave communication system. Additionally, the effect of key generation parameters on the generated secret keys in a static indoor environment is analysed.

The following sections of the paper are organised as follows: the main properties of PLS and secure key generation algorithm are described in Section II. The implementation aspects of real-time PLS in a 60 GHz mmWave system are described in Section III. Experiments and results with accompanying discussions are provided in Section IV, whereas Section V outlines our conclusions.

II. Physical Layer Security: Theory and Algorithm

One of the fundamental theories of information-theoretic security is Shannon's pioneering theory of perfect secrecy [9]. According to this theory, if a legitimate sender encodes a message M with a codeword X, then the perfect secrecy can be achieved by satisfying the following condition:

$$H\left(M|X\right) = H\left(M\right), \tag{1}$$

where $H\left(M|X\right)$ is the conditional entropy of the message given the codeword, and $H\left(M\right)$ is the entropy or prior uncertainty that one could have about the message [10]. The operator $H(\cdot)$ is the Shannon's entropy which is the measure of the amount of information in bits [11]. Perfect secrecy can be achieved if the codeword X is statistically independent of the message M, such that no cryptographic algorithm will be able to decode the message.

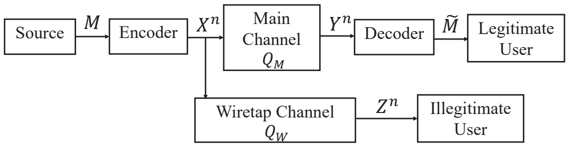

Fig. 1. Wyner's wiretap channel model.

Based on the above theory, in order to understand the effect of noise in the physical communication channel, the wiretap channel model (Fig. 1) was proposed by Wyner [12]. In this model, the legitimate transmitter encodes the message M with codeword X^n of symbol length n, and an eavesdropper receives the noisy version of the received signal Y^n denoted as Z^n. The condition for secrecy is given by the below equation:

$$\left(\frac{1}{n}\right) H\left(M|Z^n\right) \cong \left(\frac{1}{n}\right) H\left(M\right). \tag{2}$$

For a sufficiently large codeword length n, the rate of conditional entropy of the message given the codeword is arbitrarily close to the rate of entropy of the message. These codewords are known as wiretap codes, and the transmission rate achieved under these conditions is called secrecy capacity. The secrecy capacity is strictly positive if the signal received by eavesdropper Z^n is noisier than X^n [3]. Over the years, several PLS schemes have been developed and investigated for different applications. In general, PLS protocols can be classified into two categories: key-less and secret key-based [8], [11]. Key-less security methods are based on above mentioned fundamental theories, which are dependent on the performance of the legitimate channel when compared with the eavesdropper's channel. However, in practice, the implementation of these methods is complex due to the requirement of prior knowledge on eavesdropper's system capabilities and CSI [11]. In this regard, secret key-based techniques can be easily implemented, as only the key generation and key strength requirements need to be met. Key-based methods are mainly based on the following principles:

Temporal Variations: The movement of legitimate users or any objects in the wireless environment results in reflection, refraction, or scattering of the signal. These variations introduce the sufficient randomness required for the key generation [11].

Channel reciprocity: The wireless channel is reciprocal, such that the impulse responses in uplink and downlink direction between Alice and Bob are identical [14].

Channel coherence distance and spatial decorrelation: The channel impulse response remains the same within a coherence time, and a distance of half the wavelength called coherence distance [14], [8]. As Eve is at a distance greater than coherence distance, the channel is spatially decorrelated with Alice and Bob.

One way to achieve secret keys is to use the received signal strength indicator (RSSI). Due to high sensitivity to channel fluctuations and limited key quality, RSSI-based key generation methods are inefficient [13], [8]. In contrast, secret key generation using CSI is of more interest. In this work, we mainly focus on key-based security using CSI for key generation.

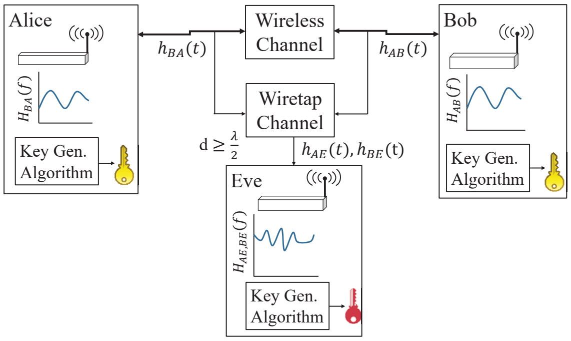

Fig. 2. PLS key generation in the presence of an eavesdropper.

A. PLSKG Algorithm

The PLSKG algorithm is realised in our system as depicted in Figure 2, which has two legitimate communicating devices and a passive wiretapping device represented by Alice, Bob, and Eve, respectively. Alice and Bob intend to generate a random secret key by using the channel impulse response to encrypt the communication. Whereas the eavesdropper Eve listens to the communication passively with the intention to retrieve all possible information about the secret key. In this work, Eve is considered to have knowledge of the key generation algorithm and is at a distance greater than the coherence distance. Hence this user should not be successful in obtaining the secret key due to the decorrelated channel [15].

To start with key generation, Alice and Bob estimate the CSI by transmitting data frames with each other. These simultaneous channel measurements should take place within the coherence time such that the random channel properties do not change. In this model, we consider an N-subcarrier OFDM system with a multipath fading channel. In Figure 2, the time domain channel impulse responses measured at Alice and Bob, are represented by $h_{BA}(t)$ and $h_{AB}(t)$ respectively. Eve passively listens to the channel and gets channel impulse responses denoted by $h_{AE}(t)$ and $h_{BE}(t)$ in time domain. Suppose Alice transmits a signal $x(t)$ through a multipath Rayleigh channel, then the received signal at Bob $y(t)$ in the frequency domain can be defined as:

$$Y_B(f) = X(f)H_{AB}(f) + W_B, \tag{3}$$

where $H_{AB}(f)$ denotes the channel frequency response, and W_A is the zero-mean additive white Gaussian noise (AWGN). Similarly, the received signal at Alice is represented by:

$$Y_A(f) = X(f)H_{BA}(f) + W_A. \tag{4}$$

The channel frequency responses H_{AB} and H_{BA} are estimated by decoding the signal field of the received packets on either side [15]. In practice, channel estimates are not completely reciprocal due to noise and hardware impairments. This results in key mismatches. In order to eliminate the mismatches and achieve a common secret key, a dedicated key generation algorithm is necessary. The algorithm used in this work is mainly inspired by the work described in [15]. The algorithm has the following three main stages:

1. Channel Quantization: Quantization is required to convert the channel measurements into binary sequences. Channel amplitudes are divided into blocks with k samples and classified as per adaptive thresholds. The adaptive threshold in the j-th block is calculated as [15], [16]:

$$t_j^+ = \tilde{X}_j(f) + \alpha \tilde{X}_j(f), \tag{5}$$

$$t_j^- = \tilde{X}_j(f) - \alpha \tilde{X}_j(f), \tag{6}$$

where $\tilde{X}_j(f)$ represents the median of the channel amplitudes in the j-th block and, α is an adaptive guard interval (AGI) constant. The i-th quantized bit in the j-th block is generated as proposed in [15]:

$$q_i = \begin{cases} 0, & X_j(f) < t_j^-, \\ 1, & X_j(f) > t_j^+, \\ \text{discard}, & t_j^- \leq X_j(f) \leq t_j^+. \end{cases} \tag{7}$$

The interval between two adaptive thresholds is known as guard interval. The bits in the guard interval are either not considered for the key or set to 1. The generated *n*-bit quantization sequence is represented as:

$$Q = \{q_1, q_2, q_3, ..., q_n\}. \tag{8}$$

For the purpose of reducing the mismatches between the generated Q at Alice and Bob, the indices of the bits in the guard interval are exchanged. In our case, $n = 828$ subcarrier amplitudes are considered for quantization.

2. Information Reconciliation: The exchange of guard band bit indices alone is not sufficient to generate the same keys at Alice and Bob. The reconciliation step helps to locate the exact mismatched bits and correct them while revealing minimal information to Eve. To do so, the reconciliation process can be further subdivided into the following two steps:

a. Parity sequence generation: The quantized binary sequence Q is rearranged in a random order known to both Alice and Bob. Parity sequences are generated by dividing Q into blocks of m bits. In our work, the quantization sequence of 828 bits is divided into blocks of $m = 4$ bits each, resulting in $828/m = 207$ blocks. The generated parity bits are exchanged between Alice and Bob to find bit mismatches. Both Alice and Bob set a mismatch threshold. If the number of mismatches is above this threshold, then the user is considered an eavesdropper. If there are no mismatches observed, then the privacy amplification to generate secret keys is carried out. Otherwise, bit mismatch correction, as explained in (b), is performed.

b. Correct bit mismatches: To further reduce the bit errors, the actual mismatched bit locations are sent from Alice to Bob. We iterate through the received sequence of mismatched indices E and for each $m/2$ bits in the interval $E(i)$ to $E(i) + (m/2) - 1$, parity bits are generated. This parity sequence generated is then sent to Alice. The mismatches between the newly generated parity sequences are computed, and mismatched indices are updated. These correction steps are repeated until the block size $m = 1$, then the bits corresponding to mismatched indices are flipped to correct the errors.

Both steps (a) and (b) are repeated a specified number of times, denoted by the number of reconciling iterations (N_{ri}). At the end of reconciliation step, we reset the order of the bits and combine every $n/256$ (i.e. $828/256 \approxeq 3$) bits using logical operations and generate 256-bit keys [15].

3. Privacy amplification: In the previous stages of key generation, some information related to the secret key is leaked through public communication [15]. This is compensated by generating a smaller key using universal hash functions.

B. Key Evaluation Metrics

The following metrics are used to evaluate the key generation algorithm:

Bit Generation Rate (BGR): It is defined as the number of secret key bits extracted from the measured CSI in a second. The unit is measured in bits/second.

Bit Mismatch Rate (BMR): It is the ratio of mismatched bits between the generated keys at two wireless nodes to the total number of bits.

Randomness: The distribution of extracted secret key bits is known as key randomness. NIST statistical test suite is used to measure the key randomness.

Key Success Rate (KSR): It is the ratio of the number of matching keys generated between the two wireless nodes to the total number of times keys generated.

The key generation algorithm implemented in this work can achieve a BGR of up to 16 kbps and the generated 256-bit keys show high entropy in the NIST test [15]. In this work, we mainly focus on the evaluation of the algorithm in terms of KSR and BMR.

III. Implementation

The communication system used for the implementation is a system-on-chip (SoC) based software-defined radio (SDR) platform, with programmable logic (PL) hardware and a processing system (PS) firmware [17]. The programmable logic (PL) implements the digital signal processing and medium access controller (MAC) modules. The key generation algorithm is implemented in the bare-metal application of the PL firmware such that the secret keys are generated in real-time. The firmware application has its default mode of operation as a server and can be connected with a MATLAB client running on a computer through an Ethernet interface. A TCP client mode of operation is also implemented to connect the device Alice as a client to Bob and run the key generation. The implementation can be divided into two parts.

1. Channel estimation (CHE) module: Channel frequency response measurement is initiated by a command to device

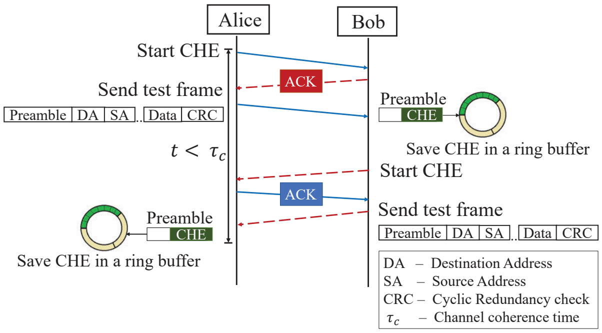
Alice
Bob
Start CHE
ACK
Send test frame
Preamble | DA | SA | Data | CRC
Preamble
CHE
Save CHE in a ring buffer
Start CHE
$t < \tau_c$
ACK
Send test frame
Preamble | DA | SA | Data | CRC
Preamble
CHE
Save CHE in a ring buffer
DA – Destination Address
SA – Source Address
CRC – Cyclic Redundancy check
τ_c – Channel coherence time

Fig. 4. The timing diagram of the Channel Estimation stage.

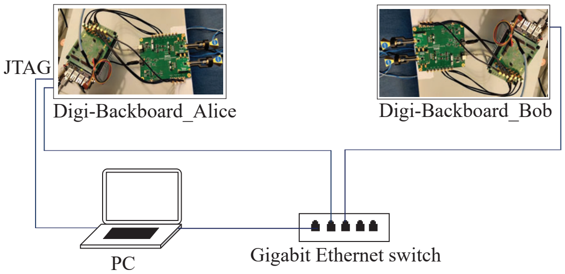

Fig. 5. Experimental setup.

Alice via the terminal. The program flow of CHE implementation is as shown in Figure 4. Alice operates in client mode and sends an initialization request to Bob. Next, the test frames are transmitted from Alice to Bob. The estimated data frame consists of 828 samples, whereby OFDM modulation with 768 data and 60 pilot subcarriers is used. CHE is measured (equations (3) and (4)) at Bob by decoding the signal field, and estimated subcarrier amplitudes and phases are stored in a ring buffer. Similarly, CHE is performed at Alice. It should be ensured that the time taken for CHE at both devices is within the channel coherence time.

2. *Secret key extraction module*: The channel frequency response data is retrieved from the ring buffer using the AXI stream interface and stored in a temporary buffer. Secret key extraction is initiated by providing serial input, and all the consecutive steps of quantization, reconciliation, and privacy amplification as outlined in Section II are executed one after the other. Figure 3 illustrates the program flow. At first, Alice sends a start request to Bob. At Bob, adaptive quantization (equations (5)-(8)) is executed, and indices of the discarded bits are shared with Alice. A similar process is followed by Alice, and both update the discarded bit indices to 1. Subsequently, Alice and Bob generate parity sequences by rearranging the quantized sequences in a known random order. The generated parity sequence is shared with Bob, and the mismatches are calculated. If the number of mismatches does not cross the Eve detect threshold, then the mismatch correction step will be executed at Bob. Alice will provide the required parity sequences for mismatch correction at Bob.

The final step in the key generation is privacy amplification. The secrecy of the generated 256-bit key at the end of the reconciliation step is increased using the SHA-1 hash function. Both Alice and Bob rearrange the reconciled bit sequence to initial order and use the hash function to generate 160-bit keys.

IV. Experiments and Results

A. Experimental Setup

The experimental setup is shown in Figure 5. It consists of two proprietary SoC-based SDR hardware platforms, the so-called Digi-Backboards, each equipped with a commercial 60 GHz RF transceivers from Analog Device and standard horn antennas of 20 dBi and 14° beam width each. The system uses signal bandwidth of 1.76 GHz and an OFDM modulation scheme with 1024 subcarriers [17].

We measure the channel by placing two 60 GHz mmWave devices (Alice and Bob) at a distance of 1.8 meters. For our experiments, we consider a static indoor office environment with the antennas in line of sight (LOS) condition. The devices are further connected to a computer using Ethernet cables and a Gigabit Ethernet switch in order to read the

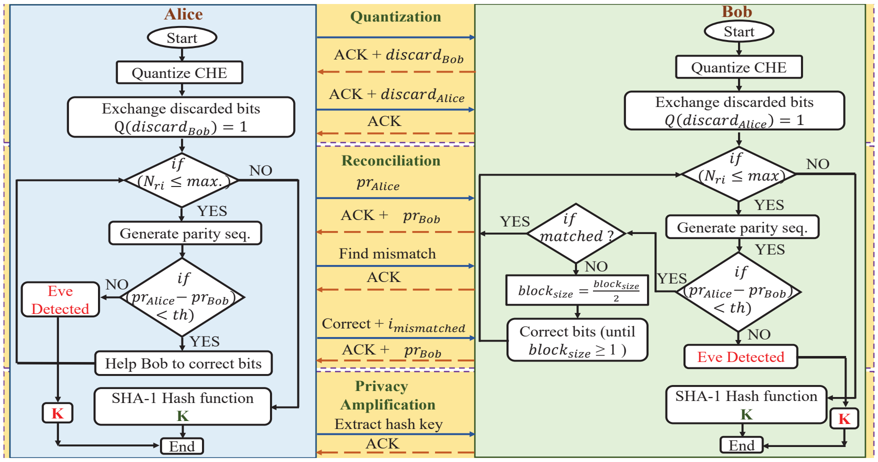

Fig. 3. Key Generation algorithm flow.

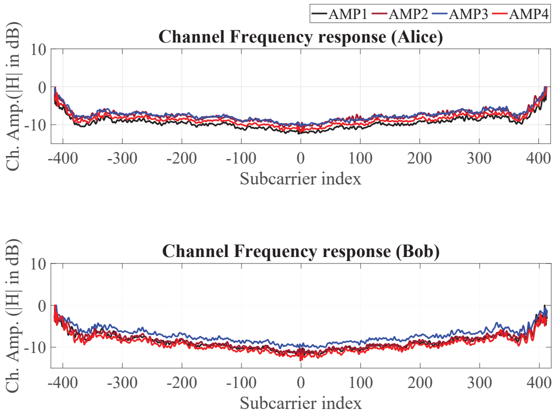

Fig. 6. Real-time channel frequency response measured at Alice and Bob.

results in MATLAB and perform the analysis. The devices are programmed via the JTAG interface to run the firmware.

For simplicity, we consider amplitudes of ten different channel frequency responses measured at each device. The channel coherence time of a 60 GHz indoor radio channel can vary between 1 ms to 32 ms [18], where the lower limit might be reached with omni-directional antennas. With directional antennas, the minimum channel coherence time would correspond to the upper limit [18]. The average round trip time latency of our system is around 250 μs. Owing to the timing diagram in Figure 4, the channel measurements would be within the coherence time. In Figure 6, four channel readings at Alice and Bob are shown. We can observe the channel reciprocity with some small dissimilarities due to noise.

After the channel estimation, key generation steps such as quantization, information reconciliation, and privacy amplification are executed. The resulting mismatches due to the differences in the channels are removed, and both devices successfully generate the same secret key. The generated secret key output is as shown below:

@Alice :129181528257029E054AB87506A5DDCF21BE5543,
@BoB :129181528257029E054AB87506A5DDCF21BE5543.

Key generation algorithm performance is influenced by parameters such as the AGI constant and the number of reconciliation iterations. Therefore, in the following, we analyze their impact on the performance of the algorithm.

B. Influence of the AGI constant on key generation algorithm's performance

In our evaluation, for the AGI constant (α), a value of 0.25 is selected as a minimum and is incremented in steps of 0.05. Corresponding key success rates and bit mismatch rates for all ten exemplary channels are derived. The error bar diagrams in Figures 7 and 8 show the estimation interval of the mean value of mismatched bits, discarded bits, and key success rate with respect to α. In Figure 7, the trade-off between the numbers of mismatched and dropped bits can be observed. It can be noted that the number of mismatched bits reduces with the increase in the mean value of α. On the contrary, the number of discarded bits increases, resulting in more 1's in the secret key. Next, we investigate the effect of the AGI constant on BMR and KSR.

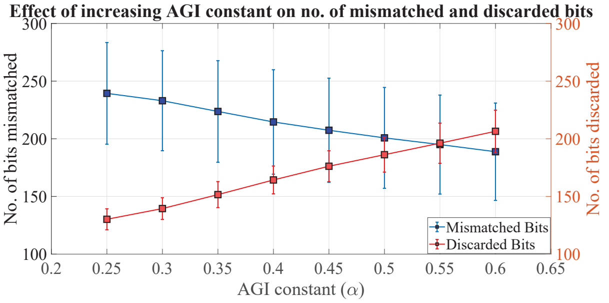

Fig. 7. Impact of AGI constant on number of mismatched and discarded bits.

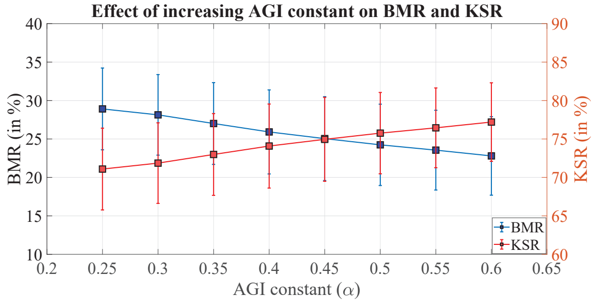

Fig. 8. Impact of AGI constant on BMR and KSR.

The results are as shown in Figure 8. The KSR can be improved above 75% with a choice of $\alpha > 0.45$. Respectively, BMR can be reduced to below 25%. But due to the trade-off between mismatched and discarded bits, for the large value of α, more discarded bit indices are exchanged between Alice and Bob, due to which Eve might be able to get more information about the secret key. Hence it is suitable to choose α between 0.45 and 0.55.

C. Influence of the number of reconciliation iterations (N_{ri}) on key generation algorithm's performance

For this analysis, we measure the key success rate by increasing the reconcile iterations for each value of α considering all the ten channel measurements. The results are shown in Figure 9 (a), (b) and (c). In Figure 9 (a), considering $\alpha = 0.25$, KSR is directly proportional to N_{ri}, and a 100% success rate is achieved for $N_{ri} > 4$. This linear proportionality of KSR with N_{ri} does not hold for other values of α. This is because the parity sequence generation begins with random rearrangement of quantized bits. If a certain rearrangement requires more N_{ri} to resolve all the bit mismatches, then the KSR can be reduced. For example, when $\alpha = 0.45$ and $N_{ri} = 7$, KSR is reduced to 90%. Nevertheless, the random rearrangement of bits is necessary to avoid information leakage during public communication between Alice and Bob for bit mismatch corrections.

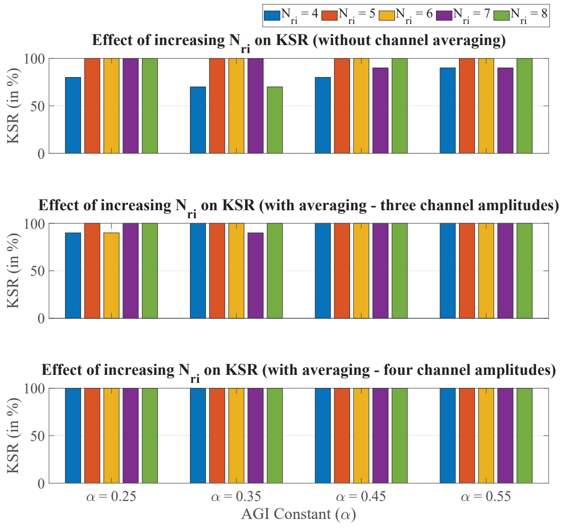

Fig. 9. a) Number of reconciling iterations vs. KSR without channel amplitude averaging, (b) with an average of three channel amplitudes, (c) with an average of four channel amplitudes.

If we consider taking the average of three successive CSI readings (Figure 9 (b)), essentially reducing uncorrelated noise, then similar results are obtained as explained above. However, the probability of achieving 100% KSR is high. For some channel amplitudes, with $\alpha = 0.25$ and 0.35, the performance with averaging is not improved because, at lower α, the BMR at the end of the quantization step is high. These bit mismatches combined with random reordering in the reconciliation phase require a higher number of iterations to generate the same keys. For $\alpha > 0.35$, the results show improvements over the approach without averaging. Further, considering the total of 20 channel readings and increasing the averaging to 4 channels, 100% KSR can be achieved irrespective of changes in α or N_{ri} as shown in Figure 9 (c).

V. Conclusion

In this paper, we have presented an implementation of a physical layer secure key generation algorithm in a 60 GHz mmWave wireless communication system. The algorithm is realised in the processing system firmware of the communicating devices, allowing real-time secure key generation. Secure keys between the two wireless devices are successfully generated in an indoor static environment. The performance of the algorithm is evaluated in terms of different parameters such as Key Success Rate (KSR) and Bit Mismatch Rate (BMR). The results have shown that to achieve low BMR in the quantization step, the preferable value of the Adaptive Guard Interval (AGI) constant is found to be $0.4 \leq \alpha \leq 0.55$. In addition, with the choice of the number of reconciling iterations $N_{ri} > 4$, the KSR of the algorithm can be improved to reach 100%. Further, the performance evaluation conducted by taking an average of multiple channel estimations, essentially reducing uncorrelated noise, indicates that the performance can be enhanced irrespective of other parameters of the algorithm.

In the future, we aspire to extend the work by taking different environmental conditions and multiple antenna setups with beamforming capabilities along with a realistic eavesdropper into consideration. The influence of hardware impairments, SNR (i.e. range), and antenna pointing errors on the algorithm performance will be investigated.

Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation project 5GENESIS, under grant agreement number: 815178.

References

[1] Li, G.; Sun, C.; Zhang, J.; Jorswieck, E.; Xiao, B.; Hu, A. Physical Layer Key Generation in 5G and Beyond Wireless Communications: Challenges and Opportunities. Entropy 2019, 21, 497.

[2] Y. Wu, A. Khisti, C. Xiao, G. Caire, K. Wong and X. Gao, "A Survey of Physical Layer Security Techniques for 5G Wireless Networks and Challenges Ahead," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 4, pp. 679-695, April 2018, doi: 10.1109/JSAC.2018.2825560.

[3] Jiao, L., Wang, N., Wang, P., Alipour-Fanid, A., Tang, J., and Zeng, K. (2019). Physical layer key generation in 5G wireless networks. IEEE Wireless Communications, 26(5), 48–54. https://doi.org/10.1109/MWC.001.1900061

[4] Wang, C., and Wang, H. M. (2016). Physical layer security in millimeter wave cellular networks. IEEE Transactions on Wireless Communications, 15(8),5569–5585.

[5] L. Jiao, J. Tang and K. Zeng, "Physical Layer Key Generation Using Virtual AoA and AoD of mmWave Massive MIMO Channel," 2018 IEEE Conference on Communications and Network Security (CNS), 2018, pp. 1-9, doi: 10.1109/CNS.2018.8433175.

[6] N. Aldaghri and H. Mahdavifar, "Physical Layer Secret Key Generation in Static Environments," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2692-2705, 2020, doi: 10.1109/TIFS.2020.2974621.

[7] H. Vogt, C. Li, A. Sezgin and C. Zenger, "On the Precise Phase Recovery for Physical-Layer Authentication in Dynamic Channels," 2019 IEEE International Workshop on Information Forensics and Security (WIFS), 2019, pp. 1-6, doi: 10.1109/WIFS47025.2019.9034987.

[8] Zoli, M., Barreto, A. N., Köpsell, S., Sen, P., and Fettweis, G. (2020). Physical-Layer-Security Box: a concept for time-frequency channel-reciprocity key generation. Eurasip Journal on Wireless Communications and Networking, 2020(1).

[9] Shannon, C. E. (1949). Communication Theory of Secrecy Systems. Bell System Technical Journal, 28(4), 656–715.

[10] Sun, L.; Du, Q. A Review of Physical Layer Security Techniques for Internet of Things: Challenges and Solutions. Entropy 2018, 20, 730.

[11] Zhang, J., Duong, T. Q., Marshall, A., and Woods, R. (2016). Key Generation from Wireless Channels: A Review. IEEE Access, 4, 614–626.

[12] A. D. Wyner, "The Wire-tap Channel", The Bell System Technical Journal, vol. 54, Issue: 8 , Oct. 1975 , pp.1355–13.

[13] Liu, Y., Chen, H. H., and Wang, L. (2017). Physical Layer Security for Next Generation Wireless Networks: Theories, Technologies, and Challenges. IEEE Communications Surveys and Tutorials, 19(1), 347–376.

[14] Wang L. (2018) Existing Techniques in Physical Layer Security. In: Physical Layer Security in Wireless Cooperative Networks. Wireless Networks. Springer, Cham.

[15] N. Felkaroski and M. Petri, "Secret Key Generation Based on Channel State Information in a mmWave Communication System," SCC 2019; 12th International ITG Conference on Systems, Communications and Coding, 2019, pp. 1-6, doi: 10.30420/454862049.

[16] Xi, W., Li, X. Y., Qian, C., Han, J., Tang, S., Zhao, J., and Zhao, K. (2014). KEEP: Fast secret key extraction protocol for D2D communication. In IEEE International Workshop on Quality of Service, IWQoS (pp. 350–359). Institute of Electrical and Electronics Engineers Inc.

[17] Petri, M. and Ehrig, M. (2019). A SoC-based SDR platform for ultra-high data rate broadband communication, radar and localization systems. In IFIP Wireless Days (Vol. 2019-April). IEEE Computer Society.

[18] P. F. M. Smulders, "Statistical Characterization of 60-GHz Indoor Radio Channels," in IEEE Transactions on Antennas and Propagation, vol. 57, no. 10, pp. 2820-2829, Oct. 2009, doi: 10.1109/TAP.2009.2030524.