Neural Network Based PHY-Layer Key Exchange for Wireless Communications

Dimitrios S. Karas*, George K. Karagiannidis*, and Robert Schober†
*Department of Electrical & Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece, E-mails: {dkaras, geokarag}@auth.gr
†Department of Electrical & Computer Engineering, University of British Columbia, Vancouver, Canada, E-mail: rschober@ece.ubc.ca

***Abstract*—Several key exchange methods for wireless channels have been proposed in the literature. They are referred to as PHY-layer security techniques and are usually based on the channel's fading characteristics and the principle of reciprocity. In this paper, we present a novel PHY-layer security algorithm whose function is based on neural networks. Specifically, we present a full key exchange scheme which includes channel sampling and thresholding and neural network based error reconciliation. The proposed method's performance and offered security are studied through simulations and interesting conclusions are drawn about its overall utility.**

I. Introduction

Cryptographic algorithms require a method to securely exchange cryptographic keys which are subsequently used to encrypt and decrypt the information messages. For example, symmetric-key encryption algorithms use the same key or trivially related keys for both encryption and decryption. In most of these methods, it is required that both communicating parties are privy to the cryptographic key. This is achieved by using a secure key exchange algorithm which must be designed in such a way that the information received by an eavesdropper, who is wiretapping the communication channel, cannot be used to deduce the key or it would require an extremely long amount of time to do so.

A. Related Literature

Numerous key exchange schemes have been developed for wired and wireless communications systems, including the Diffie-Hellman technique [1]. Over the past few years, a new class of key exchange methods over wireless channels has been proposed, which exploits the channel's properties and time response characteristics in order to facilitate the key exchange process [2] – [19]. These methods are referred in the literature as Physical (PHY)-layer key exchange algorithms.

The concept of combining key exchange and physical layer characteristics was first presented in 1995 in [2]. Since then, several methods have been proposed in this field. In [3], factors such as noise and interference were taken into consideration and a key exchange protocol was presented and tested in a real-world setting. Another PHY-layer key exchange method, based on a level-crossing algorithm, was presented in [4], and the issue of user authentication - meaning the process of validating the legitimacy of a communicating node and the prevention of a spoofing attack - was also addressed. Transmitter authentication in wireless channels was also explored in [5] and [6], while in [7] and [8] several key exchange methods were proposed with emphasis on the design of a high bit rate implementation. In another work [9], the concept of using multiple-antenna diversity for key generation purposes was explored.

Other research papers in the field deal with the PHY-layer security in Orthogonal frequency-division multiplexing (OFDM) systems [10], applications in static environments such as indoor networks [11], use of an adaptive quantization algorithm for key exchange [12], and error reconciliation based on randomness extractors [13]. A practical implementation of a key sharing platform at 60 GHz was presented in [14], while the performance of several key exchange methods and their practical feasibility is discussed in [15]– [17]. Furthermore, in [18] PHY-layer security was studied in the presence of an adversary who is performing a jamming (Denial-of-Service) attack, reducing the efficiency of the channel. In contrast, [19] proposes a method where jamming is used by the legitimate communicating nodes to hinder the adversary's activity.

B. Contributions and Organization of the Paper

In this work, we propose a novel PHY-layer key exchange protocol for a wireless link between two transceivers. Thereby, the principle of reciprocity is utilized so that the transceivers extract two highly correlated channel magnitude envelopes. Furthermore, a novel thresholding process is proposed, where the channel magnitude envelope is sampled by the transceivers over a predetermined time duration, and a least square curve is calculated by using a number of these samples as a data set. Each transceiver generates a bit string by comparing each sample of the magnitude envelope with the value of the least square curve at the corresponding position. Thus, the transceivers generate two similar bit strings that provide the basis for the cryptographic key. Due to the existence of noise and various sources of interference, there will generally be discrepancies between these two bit strings. However, symmetric key cryptography requires that both transceivers possess identical cryptographic keys. To this end, we propose an error reconciliation method, whose function is based on a two-layer neural network. The produced key is known only to the legitimate transceivers and is secure against eavesdropper activity.

The rest of the paper is organised as follows. In Section II, the system and channel model is described and an overview of the proposed key exchange protocol is given. A new channel thresholding method is proposed in Section III. In Section IV,

978-1-4577-1348-4/11/$26.00 ©2011 IEEE

we describe the training process and the operation of the neural network. Simulation results are presented in Section V, and some conclusions are drawn in Section VI.

II. System and Channel Model

A cryptographic key exchange scheme for a wireless communication setting has to adhere to some basic principles. First, it is assumed that the two transceiver nodes are communicating over a wiretapped channel. That being said, the key exchange scheme should (a) produce information-theoretically secure keys and (b) perform error reconciliation between the private keys generated by the transmitter and receiver, such that they become identical.

It is assumed that the transceivers communicate over an additive white Gaussian noise (AWGN) channel with slow Rayleigh fading, though the method presented in this paper is also applicable to other channel models such as Rician or Nakagami-*m* fading. Our approach utilizes the reciprocity principle in a wireless channel as in [2], where two transmitters operating in the same frequency band would experience the same channel characteristics at the same time. Also, we assume that the eavesdropper is at such a distance from both communicating nodes, that the envelope of the signal received by the eavesdropper is uncorrelated with the signal received by the legitimate receivers. This statement is true in most practical implementations. Specifically, an eavesdropper who is more than half a wavelength away from both legitimate transceivers will experience two independent fading channels to the two transceivers. According to [17], these channels are uncorrelated with the channel between the two legitimate nodes. We further assume that the number of deep fades in a specific time interval might be known to the eavesdropper but not the duration or time location of the deep fades.

A signal of a predetermined power is transmitted by both transceivers. However, transceivers cannot transmit and receive simultaneously, so they have to transmit and receive alternately, switching roles at a center frequency of f_s. Due to the principle of reciprocity, they will experience identical fade characteristics at the same time. Subsequently, they generate a sequence of predetermined length by sampling the received signal magnitude simultaneously over a predetermined time duration with sampling frequencies equal to f_s. The sampling frequency can be chosen based on the maximum Doppler shift of the channel, f_d, whereas the number of samples is determined by the desired computational complexity. Then, the transceivers apply a thresholding process to their respective sampled sequences, so that they each generate a bit string of equal length.

In practice, the principle of reciprocity is not true, due to the presence of noise, various sources of interference, and synchronization issues between the two transceivers in the sampling process. Thus, there are discrepancies between the bit strings generated by each transceiver, so they cannot be immediately used as a cryptographic key. In order to resolve this problem, a novel neural network based error reconciliation scheme is proposed and described in Section IV.

III. Least Square Thresholding

In [3], a thresholding method was proposed, where the threshold of the sampled sequences is determined by an automatic gain control (AGC) mechanism, so that it is independent of the transmit power and the link attenuation. Here, we present an alternative thresholding method which is more efficient even in environments where deep fades do not occur, e.g. in line-of-sight (LoS) situations.

Let g_A and g_B be the sampled sequences of length L that both transceivers - A and B - generate by sampling the channel magnitude envelope. Each sample is represented by $(i, g_K[i])$, where i is the position of the sample in the sequence and $g_K[i]$, $K \in \{A, B\}$, denotes its value. Transceivers A and B form the sets S_A and S_B, respectively, which contain all local maxima and minima of g_A and g_B.

We define the sets S_K^{max} and S_K^{min}, which contain the local maxima and minima of g_K, respectively, multiplied by a scaling factor, $u \leq 1$. These sets are formally defined as

$$S_K^{\text{max}} = \{\, (i, u g_K[i]) \,|\, g_K[i-1] < g_K[i] \wedge g_K[i+1] < g_K[i], \\ i = 2, \cdots, L-1\} \tag{1}$$

$$S_K^{\text{min}} = \{\, (i, u g_K[i]) \,|\, g_K[i-1] > g_K[i] \wedge g_K[i+1] > g_K[i], \\ i = 2, \cdots, L-1\}. \tag{2}$$

Thus, the set S_K is

$$S_K = S_K^{\text{max}} \cup S_K^{\text{min}}. \tag{3}$$

Then, each transceiver calculates a least-square polynomial curve [20] by using the elements of S_K as data points. The degree of the polynomial can be selected depending on the length of the sampling time frame and the maximum Doppler shift. Afterwards a sequence of length L, s_K, is formed by both transceivers by sampling their respective least-square curves at the points $1, 2, \cdots, L$. Each transceiver generates a bit string ρ_K of length L by comparing each element of s_K with its respective element of g_K. For each $i \in [0, L]$, if the value of $s_K[i]$ is greater than that of $g_K[i]$, the corresponding element $\rho_K[i]$ of the bit string ρ_K is set equal to 0. Otherwise, it is set equal to 1. Thus, the bit strings ρ_A and ρ_B are

$$\rho_K[i] = \left\{ \begin{matrix} 1, & g_K[i] \leq s_K[i] \\ 0, & g_K[i] > s_K[i] \end{matrix} \right\}, i = 1, 2, \cdots, L. \tag{4}$$

Before the sampling process, a low-pass filter should be used in order to smoothen the sequence g_K and eliminate high-frequency components that can potentially harm the effectiveness of this method.

A flowchart representation of the proposed algorithm is given in Fig. 1, while an example of the proposed thresholding method on a simulated channel magnitude envelope is demonstrated in Fig. 2. Specifically, Fig. 2a depicts the perceived channel magnitude for both transceivers and their generated least-square thresholds, while Fig. 2b depicts the corresponding bit strings ρ_A and ρ_B.

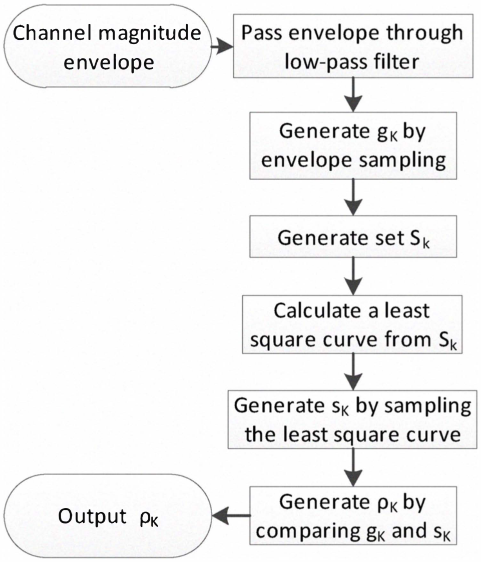
Channel magnitude envelope
Pass envelope through low-pass filter
Generate g_K by envelope sampling
Generate set S_k
Calculate a least square curve from S_k
Generate s_K by sampling the least square curve
Generate ρ_K by comparing g_K and s_K
Output ρ_K

Fig. 1. Flowchart of the proposed channel thresholding method

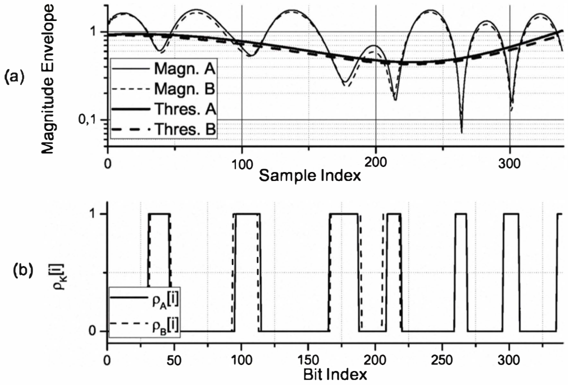

Fig. 2. (a) The channel magnitudes and thresholds for A and B, (b) The bit strings generated by A and B.

The main advantage of the above proposed thresholding technique is its ability to detect fades of smaller depth compared to a constant threshold. This is useful because, assuming that the eavesdropper knows the number of deep fades in this time interval, the time required to perform a brute force check on the value of ρ_K increases for a larger number of fades, thus strengthening the system against such attacks. Also, the ability of the legitimate transceivers to detect fades of small depth might cause the eavesdropper to miscalculate the number of fades by ignoring or not being aware of fades with a smaller depth.

IV. An Efficient Error Reconciliation Algorithm based on Neural Networks

In this section, an efficient neural network based error reconciliation method is presented. Let the transceivers A and B have generated bit strings ρ_A and ρ_B, respectively, of length L. The eavesdropper is oblivious to the fading characteristics of the communication channel between the two legitimate nodes, and thus cannot deduce the values of ρ_A and ρ_B. The correlation between the channels perceived by A and B will always be less than 1. This means that there will be discrepancies between the bit strings generated by the thresholding process. This is clearly demonstrated in Fig. 2, where the two bit strings are not identical, though the channel envelopes are highly correlated. However, if ρ_A and ρ_B are not identical, they cannot be used as cryptographic keys. The method presented in this section uses these two similar bit strings in order to generate a cryptographic key of arbitrary length, which will be known to both transceivers.

A. Neural Network Description and Operation

Let L be the length of ρ_A and ρ_B, ρ the cryptographic key, and L_t the length of ρ, which can be selected arbitrarily based on the desired key length. The neural network that will be used handles binary inputs and outputs and consists of an input layer of L nodes, a hidden layer of N nodes, and an output layer of L_t nodes. The value of N is selected taking into account that higher values increase the complexity, but also the error reconciliation capability of the key exchange scheme. A graphical representation of this neural network can be seen in Fig. 3, where the hidden layer neurons are denoted by H_j and the output layer neurons by Z_j.

An overview of this scheme is described as follows.

1) Transceiver A creates a binary neural network with the parameters as noted above and randomly initializes its synaptic weights.
2) Transceiver A randomly generates the cryptographic key ρ of length L_t.
3) The neural network is trained by using a training set that consists of inputs similar to ρ_A. This is described in Section IV-B.
4) The synaptic weights of the neural network are transmitted to transceiver B.
5) Transceiver B applies ρ_B as an input to the neural network with the received synaptic weights. The neural network should output ρ.

In Step 2, the cryptographic key is randomly generated. Here, it should be noted that in order to maximize the security of the key, each bit of ρ, which is denoted by $\rho[i]$, $i = 1, 2, \cdots, L_t$, should be generated in a way that ensures $P(\rho[i] = 0) = P(\rho[i] = 1)$. This is done so that for security purposes the generated key is uniformly distributed. Also, in Step 4, to ensure the correct transmission of the required information, the use of an error correction scheme is recommended.

B. Training of the Neural Network

Initially, we define the training set that is used for the training process of the neural network. In [21], an efficient bit string representation was proposed, where each bit string is represented by a set of pairs whose first element denotes the position of the beginning of a fade, and its second element denotes the end of the fade. For the reader's convenience,

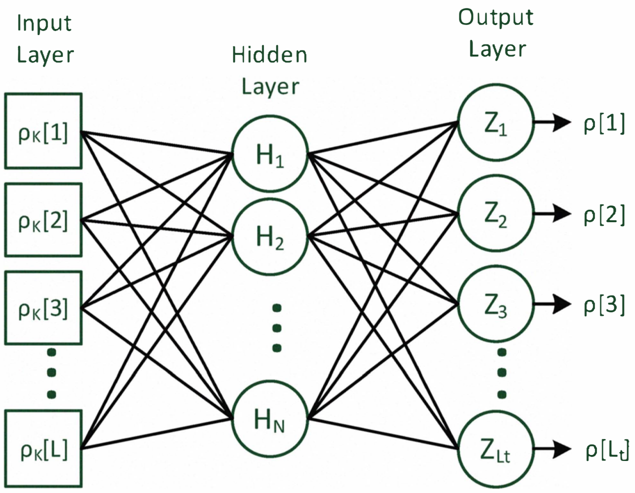

Fig. 3. The neural network used in the proposed method

ρ_K - which has previously been defined in (4) - can thus be redefined as

$$\rho_K \equiv \left\{ \left(i_l^K, j_l^K \right) \mid l = 1, \cdots, t \right\}, \tag{5}$$

where t denotes the number of fades detected by the thresholding process. Let

$$h_l(s,k) = \begin{Bmatrix} s, & k = l \\ 0, & k \neq l \end{Bmatrix}, l, s, k \in \mathbb{Z}. \tag{6}$$

Given the bit string ρ_A and using the notation described above, we define the strings $T_1^{s,k}$ and $T_2^{s,k}$ as

$$T_1^{s,k} = \left\{ \left(i_l^A + h_l(s,k), j_l^A \right) \mid \left(i_l^A, j_l^A \right) \in \rho_A \right\}, \tag{7}$$

$$T_2^{s,k} = \left\{ \left(i_l^A, j_l^A + h_l(s,k) \right) \mid \left(i_l^A, j_l^A \right) \in \rho_A \right\}. \tag{8}$$

Thus, we define the set of bit strings T, which constitute the neural network's training set, as

$$T = \left\{ T_j^{s,k} \mid s \in [-M, M], k \in [1, t], j \in \{1, 2\} \right\}, \tag{9}$$

where M is a parameter that denotes the maximum shift that is applied to a fade's beginning or end. Practically, this means that the training set consists of strings that are formed by $\rho_A[i]$, modified so that the beginning or the end of one fade is shifted by $|s|$ bits to the left or to the right. The motivation behind this formation of the training set is that, as demonstrated in Fig. 2, discrepancies between ρ_A and ρ_B usually occur at the edges of a fade, so the training set should be formed by inputs that simulate such errors. Thus, the training set is formed so that the neural network can detect and correct this kind of discrepancies.

When applying an input bit string to the neural network, the set $\{0, 1\}$ is mapped to the set $\{-1, 1\}$ in order to perform the necessary calculations. We have two layers of neurons, one for the hidden layer, whose inputs are the nodes of the input layer, and one for the output layer, whose inputs are the nodes of the hidden layer. In order to describe the neural network's training algorithm, we consider a layer with N_1 inputs and N_2 nodes - and subsequently N_2 outputs. This can refer to either the hidden or the outer layer. The synaptic weight for the j-th input of the i-th neuron on this layer is denoted by w_{ij}. All weights w_{ij} are initialized randomly to be either -1 or 1, so that $P(w_{ij} = -1) = P(w_{ij} = 1)$. If x_{ij} denotes the j-th input of the i-th neuron, then its output is

$$\sigma_i = \text{sgn} \left(\sum_{j=1}^{N_1} w_{ij} x_{ij} \right), \tag{10}$$

where sgn is a function which extracts the sign of a real number. It follows from (10) that the possible outputs of a neuron are -1 and 1. Now if, for example, $\rho_{\text{out}}[i], i = 1, \cdots, N_2$, is the target output of this layer, then the output error of the i-th neuron is defined as

$$e_i = \rho_{\text{out}}[i] - \sigma_i. \tag{11}$$

For the neural network's training, a Hebbian learning rule [22] is used, so that the alteration in the value of w_{ij} is defined as

$$\Delta w_{ij} = c e_i x_{ij}, \tag{12}$$

where $c \in \mathbb{R}^+$ denotes a training parameter that can be selected arbitrarily. The altered value of w_{ij} is simply calculated as

$$w'_{ij} = w_{ij} + \Delta w_{ij}. \tag{13}$$

Given the bit string $\rho_A[i]$, the neural network's training process as performed by transceiver A consists of the following steps:

1) The cryptographic key ρ is generated and the neural network's synaptic weights are initialized.
2) The output layer of the neural network is trained by using ρ_A as an input m times, as described by equations (10)-(13). The synaptic weights of the output layer are updated accordingly, while the synaptic weights of the hidden layer remain unchanged.
3) The output of the hidden layer, denoted by ρ_h, is calculated with ρ_A as input.
4) The training set T is formed as in (9) and the neurons of both hidden and output layers are trained by using ρ_h and ρ as target outputs, respectively. Each element of T is used m times.

In the process described above, m is a parameter that can be selected based on the time available for the neural network's training. Simulations revealed that values of $m > 1$ generally improve the performance of the neural network's error reconciliation capabilities.

V. Simulations and Discussion

In this section, we present simulation results for the performance of the proposed key exchange scheme. In the simulations, the parameters were chosen as follows: the sampling frequency was $f_s = 100$ Hz, the bit strings ρ_A and ρ_B had a length of 340 bits, the training parameter was set to $c = 0.1$, the repetition parameter was $m = 3$, and the maximum shift parameter was $M = 4$. These parameters were chosen so that the training process will be feasible on a wireless node. We also assumed that the synaptic weights were equal at transceivers A and B, meaning that there were no transmission errors.

Fig. 4 depicts the average level-crossing rate (LCR) for the proposed least-square thresholding method, plotted against the degree of the polynomial, calculated with a scaling factor of $u = 1$. The simulations were performed for two maximum Doppler shift frequencies: 1.54 Hz and 2.22 Hz, which are appropriate for indoor WLAN networks. Rayleigh and Rician channels were simulated for each of these frequencies. The K-factor for the Rician channels was $K = 10$ dB, a reasonable value for indoor channels. It should be noted that a zero degree polynomial represents a constant threshold which was used in [3]. We observe that in all cases, a polynomial of a relatively small degree greater than zero can lead to an increase of the LCR. This applies even to the line-of-sight scenarios, represented here by the Rician channel model. We can thus conclude that the proposed thresholding technique has better fade detection capabilities than the method which uses a constant threshold. As previously mentioned, better fade crossing detection leads to an improved security level against eavesdropper activity.

Fig. 5 depicts the key agreement percentage after the error reconciliation process, plotted against the number of bit discrepancies between ρ_A and ρ_B, rounded up to the nearest even number. The simulations were performed for three key lengths: 50, 100, and 200 bits, the maximum Doppler shift was $f_d = 1.54$ Hz, and the scaling factor was $u = 0.7$. We observe that in all cases, the success rate drops as the number of errors increase. Also, the key agreement rate remains above 90% for up to 8 bit discrepancies, a reasonable assumption for most practical implementations. We also observe that the success rate decreases as the key length increases, which is visible especially for larger numbers of errors. This indicates that smaller key lengths lead to better error reconciliation capabilities. Another error reconciliation method proposed in [21], named "SFIR Construction #2", was also tested. The corresponding key agreement percentages are depicted in Fig. 5. We observe that the proposed neural network-based method offers much higher agreement rates in all cases. It should be noted that in SFIR Construction #2, the key length does not affect the scheme's error reconciliation capability, so only one graph is presented for this method.

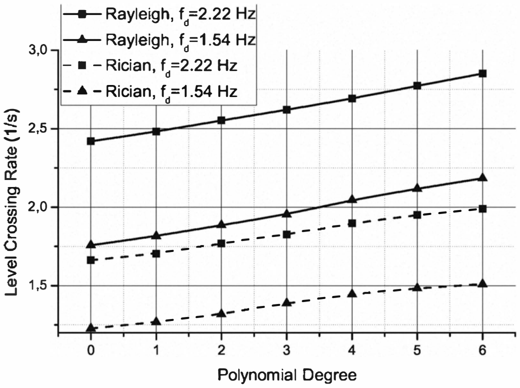

Fig. 4. The level crossing rate for Rayleigh and Rician channels plotted against the polynomial degree of the proposed thresholding method

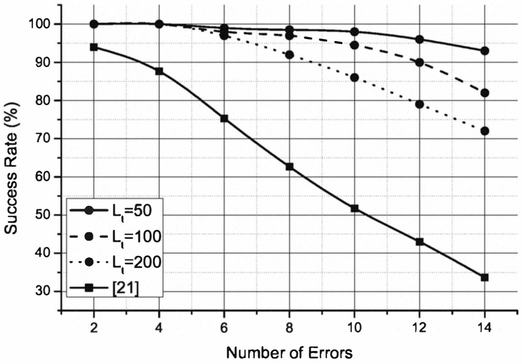

Fig. 5. The key agreement success rate for the proposed method and a method from the literature plotted against the number of bit errors for 3 key lengths

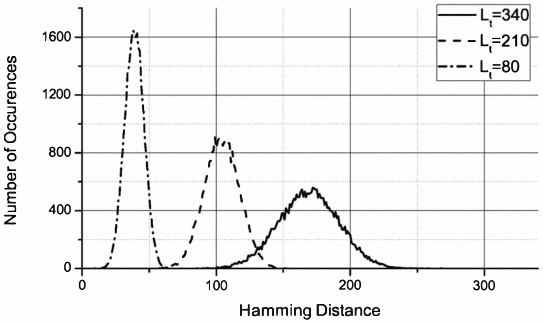

Fig. 6. The Hamming distance of the neural network's output from the actual cryptographic key during a brute force check.

In order to test the method's security level, a simulation was conducted where transceivers A and B applied the proposed key exchange scheme, while an eavesdropper who has intercepted the neural network's synaptic weights performs a brute force check by testing various inputs to the neural network. Three key lengths were tested in order to examine the security of the scheme in relation to the key length. The lengths examined were 80, 210, and 340 bits. The maximum Doppler shift was $f_d = 1.54$ Hz. In each of these cases the eavesdropper performed 30000 checks. Fig. 6 depicts the number of outputs with a specific Hamming distance from the actual cryptographic key (unknown to the eavesdropper) plotted against the Hamming distance for each of the three key lengths. We observe that the Hamming distance from the actual key follows a near-Gaussian distribution with a mean value equal to half of the key's length. This demonstrates that the eavesdropper cannot extract any useful information from the neural network's outputs. It should also be noted that the eavesdropper has no way to determine the Hamming distance of the neural network's outputs from the actual key, or whether an output is equal to the actual key, indicating that the proposed key exchange method is information-theoretically secure. If, however, we have to minimize the probability that the neural network intercepted by the eavesdropper will generate the actual key as an output, it is best to choose a large key length. This is because the quotient of the standard deviation of the Gaussian curve, σ, and the key length, L_t, σ/L_t, decreases as the key length increases. This means that the Hamming distance of 0 is relatively farther from the center of the distribution for larger key lengths.

One of the most important advantages of the proposed key exchange method is its customizability. For example, in a noisy channel where there might be a large number of discrepancies between ρ_A and ρ_B, it is possible to increase the error reconciliation capabilities of the neural network by increasing the size of the training set or the number of neurons in the hidden layer. However, in this case, the neural network's training process will require more time. Thus, there is a trade-off between these two aspects of the key exchange method's operation. The simulations performed also revealed some interesting observations about the key length. Specifically, an increase in the key length produces more secure keys, but might reduce, to some extent, the key agreement probability. A trade-off can also be made between these two factors.

VI. Conclusions

We have introduced a novel PHY-layer key exchange algorithm for wireless communications and demonstrated its performance and security level. We presented a least-square based channel thresholding method in order to extract a bit string from the channel's fading characteristics. Next, the concept behind neural network based error reconciliation was presented, and its specific characteristics, as well as the proposed training process, were thoroughly described.

Simulations have shown that the proposed method presents a multitude of benefits, such as efficiency, security and customizability. It is also able to function well in noisy channels where discrepancies appear between the bit strings generated by the two transceivers. Future work will address authentication issues and explore the capability of neural networks to perform user verification. Also, the key exchange algorithm's performance will be tested in various practical settings, such as indoor networks and communication between handheld devices. Due to limited space in this paper, in the journal version of this work security issues if the synaptic weights are known by the adversary will be taken under consideration, and the proposed method will be compared with other key generation methods for wireless channels.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, November 1976.

[2] J. Hershey, A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Trans. Commun.*, vol. 43, no. 1, pp. 3 –6, Jan. 1995.

[3] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks." in *ACM Conf. on Comput. and Commun. Security'07*, 2007, pp. 401–410.

[4] S. Mathur, N. M, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *MobiCom 08*, 2008, pp. 128–139.

[5] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2571–2579, 2008.

[6] P. Yu, J. Baras, and B. Sadler, "Physical-layer authentication," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 38–51, 2008.

[7] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Trans. Mob. Comput.*, vol. 9, no. 1, pp. 17–30, 2010.

[8] J. Croft, N. Patwari, and S. K. Kasera, "Robust uncorrelated bit extraction methodologies for wireless sensors," in *Proc. of the 9th ACM/IEEE International Conf. on Inf. Process. in Sensor Netw.*, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 70–81.

[9] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *Proc. of the 29th Conf. on Inf. Commun.*, ser. INFOCOM'10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 1837–1845.

[10] G. R. Tsouri and D. Wulich, "Securing OFDM over wireless time-varying channels using subcarrier overloading with joint signal constellations," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, pp. 6:1–6:18, March 2009.

[11] M. Wilhelm, I. Martinovic, and J. B. Schmitt, "Key generation in wireless sensor networks based on frequency-selective channels - design, implementation, and analysis," *CoRR*, vol. abs/1005.0712, 2010.

[12] S. T.-B. Hamida, J.-B. Pierrot, and C. Castelluccia, "An adaptive quantization algorithm for secret key generation using radio channel measurements," in *Proc. of the 3rd International Conf. on New Technologies, Mobility and Security*, ser. NTMS'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 59–63.

[13] M. Wilhelm, I. Martinovic, and J. B. Schmitt, "On key agreement in wireless sensor networks based on radio transmission properties," in *5th IEEE Workshop on Secure Netw. Protocols, 2009. NPSec 2009.*, oct. 2009, pp. 37–42.

[14] M. Forman and D. Young, "The generation of shared cryptographic keys through half duplex channel impulse response estimation at 60GHz," in *2010 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, 2010, pp. 627–630.

[15] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proc. of the 15th Annual International Conf. on Mobile Comput. and Netw.*, ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 321–332.

[16] M. Di Renzo and M. Debbah, "Wireless physical-layer security: The challenges ahead," in *International Conf. on Advanced Technologies for Commun., 2009. ATC '09*, 2009, pp. 313 –316.

[17] S. Mathur, A. Reznik, C. Ye, R. Mukherjee, A. Rahman, Y. Shah, W. Trappe, and N. Mandayam, "Exploiting the physical layer for enhanced security [security and privacy in emerging wireless networks]," *IEEE Wireless Commun.*, vol. 17, no. 5, pp. 63 –70, 2010.

[18] M. Zafer, D. Agrawal, and M. Srivatsa, "A note on information-theoretic secret key exchange over wireless channels," in *47th Annual Allerton Conf. on Commun., Contr., and Comput., 2009. Allerton 2009*, 30 2009.

[19] I. Martinovic, P. Pichota, and J. B. Schmitt, "Jamming for good: a fresh approach to authentic communication in WSNs," New York, NY, USA, pp. 161–168, 2009.

[20] F. B. Hildebrand, *Introduction to numerical analysis*, 2nd ed. McGraw-Hill New York,, 1973.

[21] R. Liu and W. Trappe, *Securing Wireless Communications at the Physical Layer*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[22] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.