# Secure and Robust Symmetric Key Generation Using Physical Layer Techniques Under Various Wireless Environments

Cem Sahin, Brandon Katz, and Kapil R. Dandekar

Drexel Wireless Systems Lab, Department of Electrical and Computer Engineering,
Drexel University, Philadelphia, PA, 19104, USA
Emails: {cs486, bzk23}@drexel.edu, dandekar@coe.drexel.edu

***Abstract*— Due to the unavoidable shared nature of the wireless radio spectrum, providing security for wireless communications presents unique challenges. Cryptographic keys have been used as a primary technique for securing wireless networks but are not completely secure. The symmetric keys used for cryptographic purposes are vulnerable to man-in-the-middle and brute force attacks. More recently, PHY-layer based techniques have gained significant attention, where the principle of channel reciprocity and the natural randomness of the measured wireless channel to generate secret keys is leveraged. Although, PHY-layer secret key generation has been investigated and presented in the literature before, there has been a lack of discussion in terms of obtaining symmetric keys on both ends of the link. In this work, we present a novel PHY-layer secret key generation technique that aims for symmetric encryption keys while limiting the information leaked to an eavesdropper to ensure complete secrecy and to prevent potential intelligent attacks. Our algorithm utilizes forward and backward channel state information in a given wireless link and combats the minute differences observed in these pairs by generating channel trend information. We demonstrate our technique using experimental results from a realtime SDR testbed connected to a wireless channel emulator. Our results indicate that the algorithm is successful in generating symmetric keys under various wireless environments without revealing any information pertaining to the key via the open wireless channel. Our secret key extraction rate can reach values in excess of 17.3 Kbps.**

***Index Terms*— Data security, cryptography, encryption, channel estimation, wireless networks, SISO.**

## I. Introduction

The pervasiveness of wireless communications introduces new security challenges. It is imperative that the security of wireless connections is intact so that no information is leaked to anyone but the intended receiver. The shared radio frequency propagation medium in wireless communications makes it harder to provide this type of security. Current cryptographic techniques utilize symmetric keys applied at higher layers of the network stack in order to encrypt the data that is about to be transferred. Although, it is possible to have stronger encryption with longer and more random keys, these cryptographic techniques still have their vulnerabilities. Man-in-the-middle and eavesdropping attacks are possible, where two legitimate users are exchanging salts and nonces during the key establishment phase. Any sensitive information leaked at this stage can give the attacker additional knowledge or hints towards the key being used. From an information theoretic point of view, any information release regarding the encryption of the communication jeopardizes the secrecy of the link itself. When this leaked information is combined with the low level of randomness in the keys that are being used today [1] along with an intelligent attack, we quickly establish a motive to minimize information leakage and to maximize randomness of encryption keys in order to enhance the secrecy of wireless communications.

In recent years, a new method of key generation scheme that aims to extract randomness from the Physical (PHY) layer properties of the wireless channel has gained popularity [2], [3]. The wireless channel between two nodes is highly random and changes significantly due to multipath fading. There have been many different methods proposed regarding how to generate secret keys from the PHY layer for different uses. In [4], the authors propose an algorithm that is based on interpolation, ranking, decorrelation, and quantization, where they were able to extract secret keys using wireless sensors. Another proposed approach uses averaging the channel fingerprints and applying thresholds to extract binary bits from channel information [5]. These methods leverage the highly random wireless channel for the purposes of secret key extraction.

It has been observed that, although PHY layer secret key generation schemes are useful in providing the necessary randomness for extracting secret keys, they come with their own challenges, especially agreeing on a symmetric key without any leaked information to an eavesdropper [6]. Standard key establishment steps that are currently used in wired systems, which require nonces sent in clear text in initial agreement steps, would not be applicable in PHY layer secret key generation algorithms. Previous work has attempted ways of key reconciliation [5]; however, there was still information such as index numbers being sent over unsecured wireless channels. With this work, we propose a novel algorithm that uses PHY layer secret key generation schemes, where no sensitive information is sent in clear text during the key establishment phase. We present a systematic testing of PHY layer key generation with different channel environments, which has been omit-

978-1-4673-9806-0/16/$31.00 © 2016 IEEE

ted by other works. We measure the effectiveness of our algorithm using several emulated channels and report our findings in the following sections.

We organize our paper as follows. In Section II, the threat model and the motivation behind this technique is introduced. Section III goes into the details of the algorithm developed. In Section IV, we present our experimental evaluation details including the setup, discussion on results, and comparison to existing encryption schemes, which is followed by a discussion of our future work. We finally conclude our paper in Section VI.

## II. Threat Model

In this work, we assume two legitimate nodes are trying to establish a secure wireless link between each other using traditional cryptography techniques such as AES or DES. We further assume that there is an intelligent eavesdropper that is listening to these two nodes since the start of their communication, where the eavesdropper is successful in capturing sensitive information regarding their secret key. Using this knowledge, the eavesdropper manages to crack the key and can now listen to the legitimate users' conversation without them noticing. Our algorithm aims to stop this threat. We assume the eavesdropper will have knowledge of our technique.

## III. Key Generation Algorithm

Our algorithm, which is designed for orthogonal frequency-division multiplexing (OFDM) systems, collects channel state information (CSI) data to extract randomness from the wireless channel. We start by sending packets that contain dummy or non-confidential data back and forth between two legitimate users. For each received packet, the nodes extract CSI, apply an averaging filter to smooth out the extreme changes and store them inside a matrix. Within the matrix, each column corresponds to the subcarrier index and the rows indicate the packet number. We call this collection of individual CSI measurements the channel trend information (CTI). CTI is used to determine the overall fading trend of each data subcarrier. The confidence constant, N, is set by the user (or by a smart controller) and indicates the number of agreeing ones or zeros required before a secret bit can be *locked*. These secret bits are then concatenated to form a secret key. The value of N also determines the number of dummy packets that needs to be transmitted before the key generation takes place. Algorithm 1 shows the pseudocode implementation of our key generation algorithm.

It should be noted that apart from transmitting packets with dummy data, our algorithm does not leak any sensitive information to the open wireless medium. The main advantage of our algorithm comes from this secrecy. For a seamless integration, our algorithm piggybacks on regular WiFi (or any other OFDM-based protocol) packets.

**Algorithm 1** Extracting Symmetric Keys from Wireless PHY Layer

**procedure** GENERATESYMMETRICKEY(N)
  **initialize** $C \leftarrow$ length of *data subcarriers*
  **initialize** *key[0 to C-1]* $\leftarrow 0$
  **initialize** *CTI[0 to 2N-1][0 to C-1]* $\leftarrow 0$
*start*:
  **for** $i \leftarrow 0, 2N-1$ **do** ▷ /** Obtain CSI and save it to CTI array **/
    send packet with dummy data
    receive packet with dummy data
    apply averaging filter to the CSI measurement
    $CTI[i][0 \rightarrow C-1] \leftarrow$ **abs**(CSI measurement)
  **for** $i \leftarrow 0, C-1$ **do** ▷ /** Play the game **/
    $temp \leftarrow 0$
    **for** $j \leftarrow 1, 2N-1$ **do**
      **if** $CTI[j][i] > CTI[j-1][i]$ **then** $temp \leftarrow temp + 1$
    **if** $temp >= N$ **then** $key[i] \leftarrow 1$
    **else**$key[i] \leftarrow 0$
  **checkKeyStrength**(key) ▷ /** Ensure key is strong **/
  **if** key is strong **then**
    **return** $key$
  **else**
    **go to** *start*

Keeping in mind the fact that both sides of the communication link are running the algorithm at the same time but independently, we rely on the channel coherence time to be slower than our packet-to-packet transmission time to ensure both ends observe the same fading effects.

## IV. Experimental Evaluation

### A. Setup

In order to be able to evaluate the performance of our algorithm, we needed to test it under different wireless channel environments, such as an empty room or an office setting. To accomplish this task, a Spirent SR5500 wireless channel emulator was integrated into our experiment. The SR5500 needed real RF signals to be sent in for processing. We decided to use two Wireless Open-Access Research Platform (WARP) [7] nodes operated via their WARPLab implementation, which allows all signal processing to be completed on a host PC via MATLAB. A laptop was designated to orchestrate the experiment using WARPLab. This laptop was connected to the two WARP boards via Ethernet cables. Rather than using antennas, RF cables were used to connect the WARP board ports to the SR5500. In order to ensure the same RF chain is being used, MiniCircuits ZFRSC-42-S+ splitters were introduced to our setup and they allowed for the each board to be able

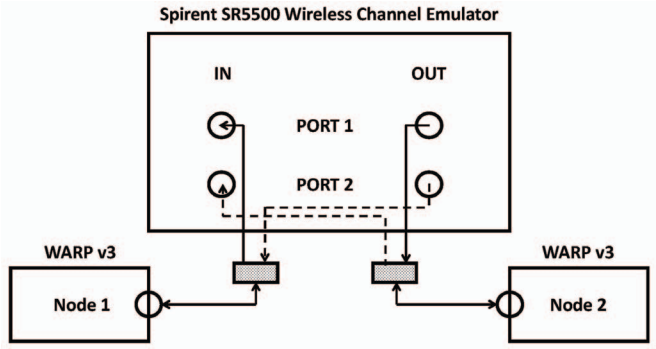


Fig. 1. Experimental setup.

to connect to an input and an output port on the channel emulator. Fig. 1 displays a sketch of this experimental setup.

### B. Results and Discussion

We collected data using 500 dummy packets transmitted and received between the two WARP nodes, which we post processed using different confidence constants (N). For each different N value, a sliding window was used to start generating keys from the first data point until the end of the packets were reached. We measured the performance of our algorithm by observing the percentage of successful symmetric keys, which were averaged over the sliding window. Because the value of N affects the number of packets required to build the CTI, as it increased, the number of final readings available for averaging decreased lowering the statistical confidence slightly.

Fig. 2 displays the various realistic indoor WiFi channel scenarios studied and summarizes the performance of our algorithm. The algorithm was successful in extracting symmetric keys in a range between 20% and 100% depending on the type of wireless channel environment. This means that there may be times where several trials might be necessary before a symmetric key is established. The static channel with single number of path and no delay, shows that as N increased, the decision of the algorithm got stronger and it was able to agree on the key in an increasing trend eventually reaching full agreement on a single try as N=14. This is the result of the increasing confidence of the algorithm. In a residential setting, following the same principle, our results show that as N increased, the success rate on extracting symmetric keys also did. For the Office and Commercial settings, we continue to observe the same trend. In order to agree on a symmetric key in a short amount of time, it might be a better choice to retry the algorithm rather than continuing to increase the value of N for heavily multipath environments.

Since the keys generated in this work are based on data subcarriers, current length in our algorithm is limited to 48 bits for an IEEE 802.11 WiFi link. However, adding more

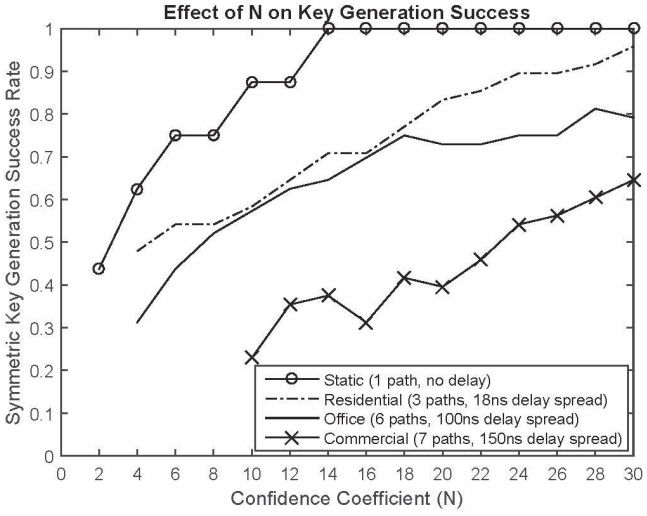


Fig. 2. Average success rate on generating symmetric keys using our PHY layer technique in terms of the confidence coefficient, N, value.

channel realizations with the use of pattern reconfigurable antennas [8] would enhance the key length greatly. The number of modes available on the antenna would multiply our current length. For an antenna with 5 modes, by concatenating the keys, we would reach a length of 240 bits. Our future work will include details of this investigation.

Our WARPLab implementation is limited to about 4 packets per second, which yields a very low secret bit extraction rate. However, a more realistic standards-compliant IEEE 802.11n implementation at 65 Mbps is capable of transmitting up to 6250 packets per second assuming an average packet size of 1300 bytes. Following our algorithm steps, assuming an N value of 10, we require a total of 40 packets. At the current speed, sending these packets back to back will only take about 7.7 ms. We are able to generate 48 secret bits in 6.4 ms, which gives us a bit generation rate of 7.5 Kbps. Noticing the maximum speed of 150 Mbps using one stream of data, our generation rate can go up to 17.3 Kbps and even higher with smaller packet sizes. These secret key extraction rates are significantly higher than similar PHY layer encryption schemes [5], [4], which reach rates of 1 to 44 bps. Friendly jamming discussed in [9] is able to reach speeds up to 18 Kbps, which we can match and surpass using varying packet sizes. For the cases where multiple tries are needed to agree on the symmetric key, our high secret key extraction rate allows for seamless operations with minimal delays (for 150 Mbps connection, 33.4 ms is needed for 10 tries) experienced by the user.

### C. Comparison to Existing Encryption Algorithms

Although it is possible, the symmetric key generation algorithm we propose in this paper is not intended for

replacing current wireless security schemes but rather best suited for augmenting them. Any phase that requires a nonce, an initialization vector or a pre-shared key can be complemented by the key generated by the above algorithm. As an example, the four-way handshake in the 802.11i standard can leverage the additional source of randomness our scheme provides. During the initial setup, the access point (AP) sends a nonce to the client station (STA). STA then uses this nonce to achieve Pairwise Transient Key (PTK). The nonce sent by the AP can be bypassed to use our key, which is generated without sending the actual data for the nonce. Such applications, where the extracted Physical layer key replaces or augments one of the security steps, can be a valid technique for preventing eavesdropping attacks.

It should be noted that our algorithm does fall behind in symmetric key success rate and can not be fully compared against the same feature of the existing encryption algorithms. This is because current algorithms utilize nonces, initialization vectors transmitted over the link or pre-shared keys to agree on the symmetric key, which provide highly reliable key symmetricity. Our algorithm, however, does not follow the same procedure to establish the key. This, however, is acceptable as we provide additional privacy by not exchanging and nonces or initialization vectors during the key establishment phase.

## V. Future Work

Having access to multiple sources of test channels is important to understanding and optimizing the presented key generation algorithm. Future testing will include simulated channels to allow more rapid testing as well as initial testing of environments encountered outside of the lab. The WINNER (Wireless World Initiative New Radio) Phase II channel model presented in [10] has been selected as the simulation starting point. The WINNER II model provides a geometrical simulation with low-level controls which will allow testing mobile as well as multi-node situations. The model is based on measurements of standardized scenarios seen in the real world of communications which will be instrumental in proving the algorithm in standard use cases such as in a cellular network with motion. A more thorough analysis of our algorithm in terms of its success rate will be possible with the comparison between emulated, simulated, and over-the-air wireless channels.

## VI. Conclusion

In this work, we presented our wireless PHY layer secret key generation algorithm that provides symmetric keys without any sensitive information being leaked to the unsecured wireless channel. We ran a verification experiment using WARP software-defined radio nodes and a Spirent SR5500 channel emulator programmed with various channels. At the end of the experiment, we were able to show the success rate associated with our algorithm and identify the strength of it in terms of both key extraction rate and the enhanced secrecy.

## Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Numbers CNS-1228847 and CNS-1422964.

## References

[1] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter, "Ron was wrong, whit is right." *IACR Cryptology ePrint Archive*, vol. 2012, p. 64, 2012.

[2] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, March 2008, pp. 3013–3016.

[3] Y. Liu, S. Draper, and A. Sayeed, "Secret key generation through ofdm multipath channel," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, March 2011, pp. 1–6.

[4] J. Croft, N. Patwari, and S. K. Kasera, "Robust uncorrelated bit extraction methodologies for wireless sensors," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 70–81. [Online]. Available: http://doi.acm.org/10.1145/1791212.1791222

[5] 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 *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 321–332. [Online]. Available: http://doi.acm.org/10.1145/1614320.1614356

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

[7] "Warp project." [Online]. Available: http://warpproject.org

[8] D. Patron, H. Paaso, A. Mammela, D. Piazza, and K. Dandekar, "Improved design of a crlh leaky-wave antenna and its application for doa estimation," in *Antennas and Propagation in Wireless Communications (APWC), 2013 IEEE-APS Topical Conference on*, Sept 2013, pp. 1343–1346.

[9] S. Gollakota and D. Katabi, "Physical layer wireless security made fast and channel independent," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 1125–1133.

[10] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, "Winner ii channel models," *Radio Technologies and Concepts for IMT-Advanced*, pp. 39–92, 2009.