Physical Gate Based Preamble Obfuscation for Securing Wireless Communication

James Chacko, Kyle Juretus, Marko Jacovic, Cem Sahin, Nagarajan Kandasamy,
Ioannis Savidis and Kapil Dandekar
ECE Department, Drexel University, 3141 Chestnut St, Philadelphia, PA, 19104
{jjc652, kjj39, mj355, cs486, nk78, is338, krd26}@drexel.edu

***Abstract*—As wireless devices hold prominent roles as means of communication, developing strong security methods against sophisticated cyber-attacks has become paramount. A novel physical layer based technique for securing wireless communication between the transmitter and receiver is described in this paper. The technique involves obfuscating the preamble data of the baseband signal through unique keys that are independently generated at both the transmitter and the receiver based on channel characteristics known only to the pair. The obfuscation technique is developed on the Drexel Software Defined Communication testbed on a Xilinx Virtex 6 ML605 board.**

***Index Terms*—communication system security, field programmable gate arrays, physical layer, software radio**

I. Introduction

Wireless communication is governed by standards put forward by telecommunication regulatory bodies, which differ in implementation and are selected based on the type of coverage, throughput, and desired service goals. Commercial devices working within each standard share the same medium and communicate based on a defined set of policies that is described within the header structure of each packet. The strict packet structure, defined by the protocol, for reliable communication between two parties might be exploited by intruders, introducing security vulnerabilities. Challenges include man-in-the-middle attacks such as eavesdropping, spoofing, and denial-of-service related to reactive signal jamming. A novel method is described to secure a wireless communication channel through obfuscating the physical layer at the gate level logic by using unique keys that are generated individually at both communication nodes based on reciprocal channel characteristics [1].

Exploiting the strict structure of a packet by surgically attacking key policy markers to disrupt the correct reception of the packet has been discussed in the past [2]. A technique in which an adversary launches a low-power reactive jamming attack to distort data used in frequency offset estimation is demonstrated in [2]. An energized narrow-band based reactive jamming framework that generates an energy pulse that disrupts the reception of data upon detecting the packet preamble is described in [3]. Both attacks depend on exploiting the strict structure of a packet. In this paper, we describe a security technique that is capable of preventing attacks based on reactive intruder frameworks by concealing the preamble.

Information transmitted over the wireless medium is encrypted to secure against man-in-the-middle and spoofing attacks; however, ensuring the private knowledge of the encryption keys is an area of ongoing research. Security methods that rely on encryption and decryption techniques to secure a transmission reveal the sending and receiving nodes to the intruder. A unique method of securing packet transmission by combining a robust physical layer based symmetric key generation technique provided in [1] with gate level logic obfuscation within the baseband communication layer is described in this paper. The system is implemented on the Drexel Software Defined Communication (SDC) [4] testbed using a Xilinx Virtex 6 ML605 board.

The work described in this paper is novel both in the implementation and application of the technique. The technique provides flexibility in obfuscating the physical layer based on runtime data on the SDC testbed and the ability to apply an overlying security layer for point-to-point communication on demand. The remainder of the paper is structured as follows: a brief background on physical key and other preamble based obfuscation techniques are discussed in Section II. Our novel technique in securing wireless communication through physical key obfuscation of the preamble using a uniquely generated key is described in Section III. Implementation details and analysis of results are provided in Sections IV and V, respectively. Concluding remarks are discussed in Section VI.

II. Background

Securing wireless transmissions by manipulating the physical layer is an emerging area of research as attacks such as spoofing, de-authentication, and cracking encryption keys are easily executed due to the availability of software tools. Idiosyncratic design and spectral characteristics of the channel between a transmitter and receiver have been successfully exploited in the past to generate secret keys even with the presence of eavesdroppers [1], [5], [6]. Applications which use secret keys are numerous, ranging from intruder detection and authentication to countermeasures against attacks [7], [8]. An attacker is still capable of detecting packets that are secured within the medium due to the strict structure of the packets used in the communication protocol. The capability of detecting transmissions allows for reactive attacks which cause partial or full distortion of packets traversing the medium [2], [3]. The physical layer hardware encryption technique we developed defends against intruders attempting

978-1-5090-4588-4/17/$31.00

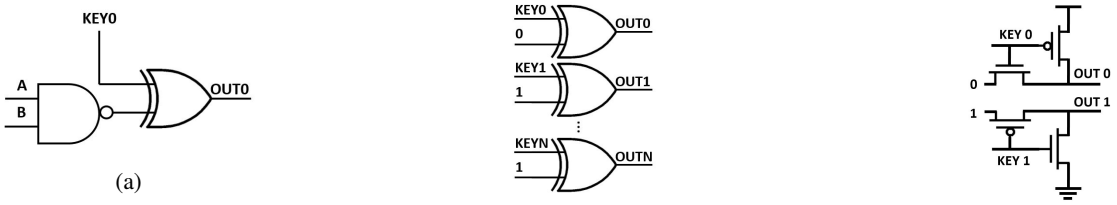

Fig. 1: Encryption implementations using (a) XOR-based logic encryption of a NAND gate, (b) XOR preamble encryption, and (c) ASIC preamble encryption.

Truth Table

KEY0	KEY1	A	B	OUT
0	0	X	X	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	X	X	1

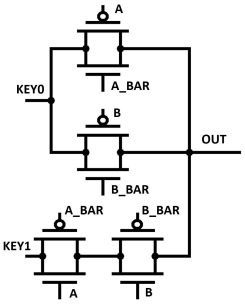

Fig. 2: Transmission gate encryption topology.

to reactively attack transmitted packets by masking the channel from detection.

The technique described in this paper prevents intruders using standard compliant radio transceivers from performing narrow band reactive jamming and eavesdropping based attacks. An attacker with unlimited power, memory, and time may capture every transmission over the air and then post process the data to sync unto the transmitted packet, but the described obfuscation technique significantly compromises the consecutive steps in decoding that involve training on known data. The intended receiver is designed to retrieve the original preamble data from the received signal to sync and decode the packet. It is important to note that this paper makes a realistic assumption that the attacker is bounded to power and resource utilization margins.

The use of a physical layer based technique provides additional security that complements traditional software approaches. Design changes for the implementation of the techniques require additional overhead in comparison to standard approaches; however, the technique described in this paper uses minimal additional hardware resources. Unique signatures, or secret keys, are derived through the analysis of channel or other communication link measurements and are engineered into the transmitted signal. In this paper, a method is developed a method which combines channel measurement based secret key generation with hardware based encryption to secure wireless transmission.

A. Physical Key Obfuscation

Physical keys have been used in integrated circuits (ICs) to withhold detailed information regarding the design and functionality of a circuit from untrusted third-parties [9]–[11]. Concealing circuit information is a vital protection mechanism against the threats of intellectual property (IP) theft, IC counterfeiting and overproduction, and the insertion of malicious hardware (such as hardware Trojans) into an IC. There are various methods for implementing physical key obfuscation, in this section the XOR [9], [10] and gate level [11] logic encryption methodologies are described.

1) XOR-based Logic Encryption: XOR based logic encryption [9], [10] utilizes an XOR at the output of a gate already present in the circuit to corrupt the output of the original gate if an incorrect key is used. An example is shown in Fig. 1a, where $KEY0$ controls the corruption on the net labeled $OUT0$. When $KEY0$ is 0, the XOR behaves as a buffer, and the circuit functions as a NAND gate. If $KEY0$ is set to 1, then the XOR acts as an inverter, resulting in an AND gate behavior.

Inverters are added before or after the original gate to further obfuscate the functionality of the key gate to prevent an adversary from knowing the key value solely on the use of the XOR/XNOR [9]. Requiring the original gate and the XOR gate for encryption introduces a large per-gate overhead of approximately 140% in propagation delay, 85% increase in power, and 125% area overhead when compared to a standard cell with no encryption [11]. Note that the analysis of overhead does not consider the addition of the inverters, which further increases the cost of implementing XOR based encryption.

2) Gate Level Logic Encryption: Gate level logic encryption introduces key based security into the gate design itself in order to reduce the per-gate overhead of implementing XOR based encryption [11]. Implementing an obfuscated AND/NAND with gate level logic encryption is shown in Fig. 2.

The transmission gate design physically replaces the original gate, removing the need for both the original gate and XOR. Removing the original gate leaves less information of the original design, and reduces the overhead needed to encrypt a design. For example, the NAND gate encrypted in Fig. 1a is now implemented by setting $KEY0$ to 1 and $KEY1$ to 0. Utilizing gate level logic encryption results in an approximately 23% reduction in propagation delay through the circuit, 29% reduction in power consumption, and 19.8% less area usage as compared to an implementation utilizing XOR based logic encryption [11].

B. Channel Spectrum based Key Generation

Various encryption methods are available both at higher layers of the network stack and at the Physical (PHY) layer.

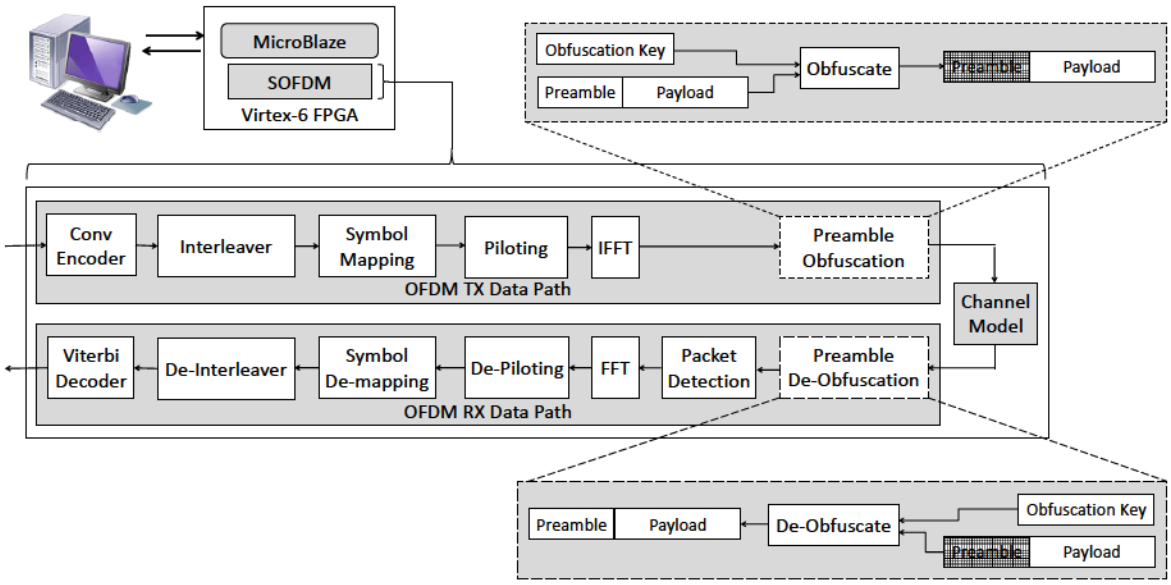

Fig. 3: System layout of the SOFDM PHY layer within the SDC testbed showing the physical key obfuscation and de-obfuscation modules in the transmit and receive chain.

One of the most widely-used encryption techniques rely on generating a pair of public and private keys that leverage computationally hard problems to solve [12]. Although the algorithm offers a solution for not sharing the decryption key with the public, the asymmetric nature of RSA cryptography disqualifies the key generation algorithm from the application defined in this paper. The logic gates, as described above, require symmetric keys established on both ends of the communication channel. Symmetric key encryption techniques, such as the Advanced Encryption Standard (AES), have been widely adopted and require additional steps to agree on a secret key. Possible steps include; 1) a key management server, 2) pre-shared keys, and/or 3) nonces, salts, or initialization vectors transmitted between the two parties during the key agreement phase. The information transmitted over an unsecured channel during the key agreement phase leaves the communication link vulnerable to eavesdropping attacks. Physical layer techniques have been shown to leverage the randomness extracted from the wireless channel to generate a symmetric secret key minimizing any information leaked in plain-text [1], [6], [13].

For the purposes of this paper, any symmetric cryptography technique is acceptable. However, to leverage the additional benefits offered by wireless PHY layer-based techniques, the algorithm defined in [1] is chosen, where the authors lay the foundation for real-time, standards-compliant PHY layer encryption.

III. Physical Key Based Wireless Preamble Encryption

A. XOR Implementation

An XOR gate is added between each bit and the final preamble bit output to translate the standard based preamble to an encrypted preamble, as shown in Fig. 1b. If a bit requires flipping, the key with the corresponding bit is set to 1 to invert the input, allowing for the generation of any encrypted output sequence. With XOR based encryption, the original standards based preamble is preserved when the key is set to 0, which ensures communication is established if the pairing of the receiver and transmitter is unsuccessful with the encrypted preamble. The XOR methodology is also beneficial when the standard preamble is not known, or the design is meant to support multiple standards, as the preamble input need not be known for correct operation.

B. Challenges in Preamble Encryption

The inherent repetitive structure of the preamble is maintained for synchronization, since auto-correlation based techniques are required for both coarse timing and frequency estimates. Even if an attacker is able to determine the coarse timing point of the signal through standard auto-correlation methods, cross-correlation based fine timing synchronization fails as the receiver is using the incorrect reference signal. If timing synchronization is not resolved by the attacker, the packet is rendered useless. Encrypting the reference symbol used for channel estimation causes an incorrect channel estimate at each sub-carrier, which results in the corruption of the packet. Although an attacker with infinite processing time is capable of determining the correct estimates through iterative minimization approaches of the recovered Error Vector Magnitude, such methods are impractical under real-time constraints as modification of the key is possible at any time instant.

IV. Experimental Setup and Implementation

The preamble obfuscation technique described in this paper was implemented and validated on a hardware based software

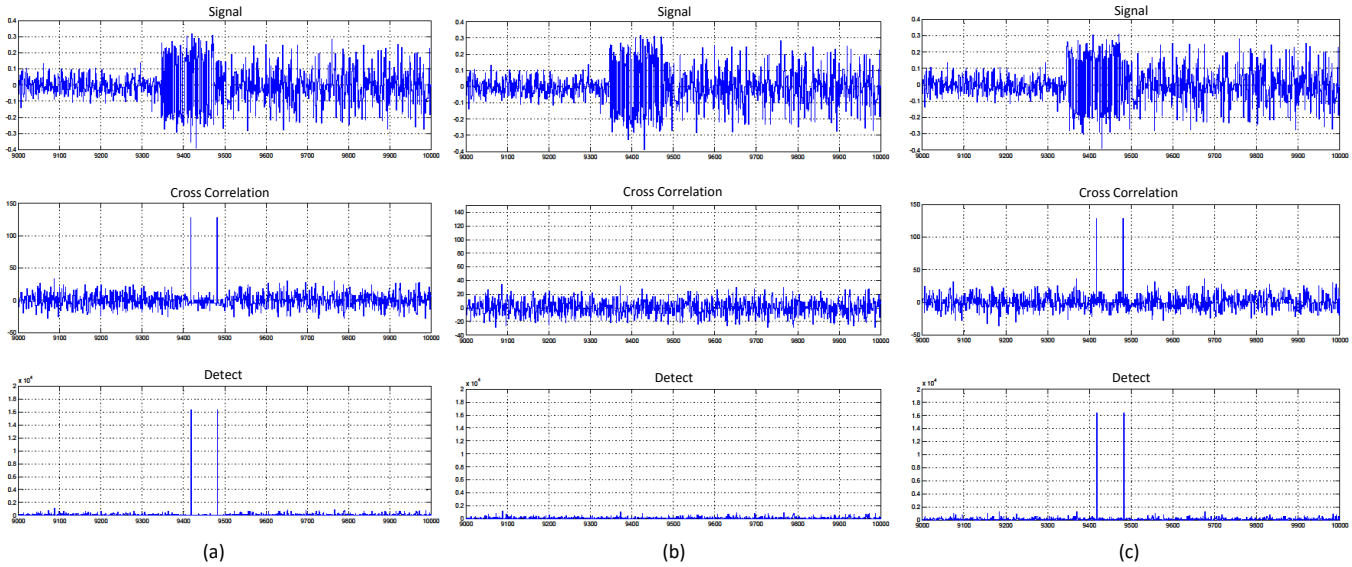

Fig. 4: Received Signal, Cross-Correlation and Fine Detect outputs captured at the receiver for three experimental setups. (a) implementation without encryption or decryption. (b) implementation with encryption at the transmitter side (c) implementation with encryption at the transmitter and decryption at the receiver side.

defined radio to permit gate level logic encryption in the design of the transmitter and receiver. The Software Defined Communication (SDC) testbed developed by the Drexel Wireless System Laboratory (DWSL) was used. SDC is a highly flexible hardware based physical layer implementation for Scalable Orthogonal Frequency Division Multiplexed (S-OFDM) signals, allowing for rapid prototyping for wireless research.

An 802.11-2012 based experimental setup using the Wireless Open-Access Research Platform (WARP) [14] was designed to collect wireless channel fingerprints. The measurements used interrupt based sampling and piggy-backed on standard WiFi packets to enable standards-compliant transmissions [1]. The symmetric keys generated by the use of the algorithm in [1] were then utilized for preamble obfuscation as described in the next section, which is done in two phases. In Phase 1, the preamble obfuscation technique, as a proof of concept, is realized in a higher level language. MATLAB was used to create an OFDM based packet with an obfuscated preamble and known data within the payload. The packet, with its obfuscated preamble, was then transmitted over WARPLab to introduce channel affects and received back into the MATLAB workspace. The inability of software scripts to decode the transmitted payload with an obfuscated preamble validated the efficacy of the technique. Implementing gate level obfuscation using one of the techniques described in Section II-A with MATLAB was avoided due to increased complexity. In Phase 2, gate level obfuscation was implemented within the physical layer of the SDC testbed.

Every module comprising the SOFDM core within the SDC testbed is built to be insensitive to functional latencies occurring across other baseband modules. Therefore, the extra processing latency introduced by the obfuscation module implemented between the IFFT module and the transmit buffers did not interfere with the physical layer implementation [15]. In addition, the global configuration control of the SDC through the on board microblaze processor enables changing keys on demand if required.

In order to develop and study the implementation of the gate level logic techniques, it is essential to include debug probe locations into the hardware modules targeted for the obfuscation. The simulation framework of SDC built in MATLAB sysGen has embedded probe locations used to transmit and receive packets through an emulated channel. For the work developed in this paper, the output from the fine packet detection modules were probed. As a first step, a packet was transmitted and received without the added obfuscation to provide a reference of captured data from the fine packet detection.

The experimental setup with the added preamble obfuscation and de-obfuscation modules at, respectively, the transmitter and receiver are shown in Fig. 3. The obfuscation module in the transmitter was added between the IFFT and the front-end buffer, while the de-obfuscation module was added between the front-end buffer and packet detection module in the receiver. The obfuscation module consists of control circuits that encrypt the preamble data section of a transmitted packet while leaving the payload in the original form. The control is capable of differentiating the preamble from the payload using the packet control handles of the SDC provided within the modules. The obfuscation keys are stored in a shared memory space accessible to the on board microprocessor, which enables setting and changing the key in real time (if needed). The framework was specifically chosen to ensure the start of communication exchanges between the transmitter and any receiver using the same protocol. Once communication is

established, channel spectrum realizations are processed to generate a key only if obfuscation hardware is available, which secures the communication link.

In phase 2, the unique key that was independently generated at both the transmitter and the receiver using the technique described in Section II-B was loaded into the shared registers. On the transmitter side, the obfuscation register is used to encrypt the preamble, while decryption occurs at the receiver. The detailed layout of the obfuscation module is shown in Fig. 3. The captured data from the fine packet detection module was used for verification of the implementation and is described in the next section.

V. Results and Analysis

The results observed at the output of the cross correlation module of the receiver are discussed in this section. The cross correlation module is used for timing synchronization of the received packet. The ability to decrypt the preamble back to the original form plays an important role in enabling corrective modules to act on the received signal, which otherwise renders the received packet useless. The affects of the encrypted preamble for communication blocks after the cross correlation are not discussed in this paper.

The top subplot in Fig. 4(a) is the signal captured at the receiver without the added physical key obfuscation of the preamble at the transmitter or the de-obfuscation at the receiver. The received signal is then cross correlated with the known preamble to detect the fine start of the packet. The output of the cross correlation of the captured signal is shown in the middle subplot of Fig. 4(a) and the two peaks in the bottom subplot of Fig. 4(a) represent positive detection as the threshold of 10^4 is passed, which was the set value for the given experiment. The use of preamble obfuscation at the transmitter renders the fine packet detection incapable of detecting the fine packet start, which corresponds to the absence of peaks in the output of the cross correlation (see Fig. 4(b)). The application of the physical key at the receiver enables the fine packet detection, as is shown by the cross correlation results of Fig. 4(c). The results indicate that encryption and decryption of the communication link is feasible through obfuscation of the preamble.

Generating the key (key_{obj}), to obfuscate the preamble is a function of the spectrum reciprocity key (key_{chn}). The function that derives key_{obj} from key_{chn}, in its current implementation, randomly selects a subset of length determined by the preamble length. Additional work involves developing the function to only choose the key_{obj} subsets that preserve certain preamble characteristics essential for decoding with respect to the noisy/fading communication channels. The current function-channel pair worked for all key subsets iterated through.

VI. Conclusion

A physical layer based technique for securing wireless communications by means of preamble obfuscation was described. The results indicate that encryption schemes integrated into the wireless physical layer permits logic gate based packet detection obfuscation. Our technique uses wireless channel fingerprints to extract symmetric secret keys on both ends of the communication channel. The keys were then used to successfully encrypt the packet preamble. By leveraging the Drexel SDC testbed, it was shown that an intruder without the correct decryption key is unable to detect the wireless packet using standard wireless packet detectors.

Acknowledgments

The authors thank Brandon Katz and Danh Nguyen for their insight to this project. The research was supported by funding from the National Science Foundation Grant No. CNS-1228847 and DUE-1241631.

References

[1] B. Z. Katz, C. Sahin, and K. R. Dandekar, "Real-time wireless physical layer encryption," in *2016 IEEE 17th Annual Wireless and Microwave Technology Conference (WAMICON)*, Apr. 2016, pp. 1–4.

[2] H. Rahbari, M. Krunz, and L. Lazos, "Security vulnerability and countermeasures of frequency offset correction in 802.11a systems," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, Apr. 2014, pp. 1015–1023.

[3] D. Nguyen, C. Sahin, B. Shishkin, N. Kandasamy, and K. R. Dandekar, "A real-time and protocol-aware reactive jamming framework built on software-defined radios," in *Proceedings of the 2014 ACM Workshop on Software Radio Implementation Forum*. New York, NY, USA: ACM, Jan. 2014, pp. 15–22.

[4] B. Shishkin, D. Pfeil, D. Nguyen, K. Wanuga, J. Chacko, J. Johnson, N. Kandasamy, T. P. Kurzweg, and K. R. Dandekar, "Sdc testbed: Software defined communications testbed for wireless radio and optical networking," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011 International Symposium on*, May 2011, pp. 300–306.

[5] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "Secret key extraction from wireless signal strength in real environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 917–930, May 2013.

[6] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, Sept. 2008, pp. 128–139.

[7] B. W. Ramsey, B. E. Mullins, M. A. Temple, and M. R. Grimaila, "Wireless intrusion detection and device fingerprinting through preamble manipulation," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 585–596, Sept. 2015.

[8] J. Xiong and K. Jamieson, "Securearray: Improving wifi security with fine-grained physical-layer information," in *Proceedings of the 19th Annual International Conference on Mobile Computing; Networking*. New York, NY, USA: ACM, Sept 2013, pp. 441–452.

[9] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending piracy of integrated circuits," in *Proceedings of the IEEE/ACM Design, Automation and Test in Europe*, Oct. 2008, pp. 1069 – 1074.

[10] J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu and R. Karri, "Fault analysis-based logic encryption," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 410 – 424, Feb. 2015.

[11] K. Juretus and I. Savidis, "Reduced overhead gate level logic encryption," in *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*, May 2016, pp. 15 – 20.

[12] R. Rivest, A. Shamir, and L. Adleman, "Cryptographic communications system and method," Sep. 20 1983, US Patent 4,405,829. [Online]. Available: https://www.google.com/patents/US4405829

[13] C. Sahin, B. Katz, and K. R. Dandekar, "Secure and robust symmetric key generation using physical layer techniques under various wireless environments," in *2016 IEEE Radio and Wireless Symposium (RWS)*. IEEE, 2016, pp. 211–214.

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

[15] J. Chacko, C. Sahin, D. Nguyen, D. Pfeil, N. Kandasamy, and K. Dandekar, "FPGA-based latency-insensitive OFDM pipeline for wireless research," in *High Performance Extreme Computing Conference (HPEC), 2014 IEEE*, Sept 2014, pp. 1–6.