iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** VIII **Month of publication:** August 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Wireless Communication under Broadband Reactive Jammer Attacks

Archana.A[1], N.Kohila Assistant Professor[2]
[1,2](Department of Computer Applications, Vivekanandha College of Arts and Science for Women (Autonomous), Tiruchencode, Tamilnadu)

Abstract : A reactive jammer jams wireless channels only when target devices are transmitting. Compared to constant jamming, reactive jamming is harder to track and compensate against [2], [42]. Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) have been widely used as counter measures against jamming attacks. However, both will fail if the jammer jams all frequency channels or has high transmit power. In this paper, we propose an anti-jamming communication system that allows communication in the presence of a broadband and high power reactive jammer. The proposed system transmits messages by harnessing the reaction time of a reactive jammer. It does not assume a reactive jammer with limited spectrum coverage and transmit power, and thus can be used in scenarios where traditional approaches fail. We develop a prototype of the proposed system using GNURadio. Our experimental evaluation shows that when a powerful reactive jammer is presence, the prototype still keeps communication, whereas other schemes such as 802.11 DSSS fail completely.
KEYWORDS - Wireless Communication, Jamming Attacks, Reactive Jammer, Broadband

I. INTRODUCTION

Jamming attacks are well-known threats to wireless communication. A jammer uses a radio frequency device to transmit wireless signals. Due to the shared nature of wireless medium, signals of the jammer and the sender collide at the receiver, and the signal reception process is disrupted. Anti-jamming techniques have been extensively studied and proposed in the literature over the past decades Frequency Hopping Spread Spectrum (FHSS) (e.g., [13], [37]) and Direct Sequence Spread Spec-trum (DSSS) are dominantly used for the anti-jamming purpose.

Although FHSS and DSSS techniques were developed more than 30 years ago, until now these techniques and their variants are all limited by a common assumption that the jammer can only jam part of the communication channels or has limited transmit power. Unfortunately, if the jammer is broadband (i.e., it can jam all channels simultaneously) or has a high transmit power to overcome the spreading gain, these methods fail to maintain the anti-jamming communication. Hence, it seems that a broadband and high-power jammer is perfect and invincible. However, when such a jammer adopts reactive jamming strategy, a closer examination on the jammer's behavior reveals its "Achilles Heel".

Reactive jamming attacks are among the most effective jamming attacks [2]. Compared to constant jamming, reactive jamming is not only cost effective for the jammer, but also hard to track and remove due to its intermittent jamming behaviors [2]. To be reactive, a reactive jammer "stays quiet when the channel is idle, but starts transmitting a radio signal as soon as it senses activity on the channel" [45]. Channel sensing causes a short delay. For example, energy detection is the most popular channel sensing approach with very small sensing time [15]. When implemented in a fully parallel pipelined FPGA for fast speed [7], energy detection requires more than 1ms to detect the existence of target signals for a 0.6 detection probability and -110dBm signal strength. In addition, upon detecting the target signal, the jammer needs to switch its status from quiet to transmitting. The switching process further takes time. As another example, German SGS 2000 series military jammer has a switching time of about 50 s [3]. Therefore, before the jammer actually jams, the sender has already transmitted tR bits, where t is the reaction time of the jammer and R is the transmission rate of the sender.

A. Wireless communication under Broadband Reactive Jammer Attacks

This observation provides insights into designing counter-measures to deal with the broadband and high-power reactive jammers. It is easy for people to conceive that the receiver may collect information bits from the unjammed parts of received packets and try to assemble these bits together to obtain a meaningful message. However, significant technical challenges exist to prevent this intuition

©IJRASET: All Rights are Reserved

from being transformed into a real-world realization. For example, transmission errors like lost or duplicate bits may happen when there exist jamming attacks or a retransmission mechanism is employed. A small number of lost/duplicate bits can make many bits misaligned, which causes the failures in reconstructing the original message.

In this paper, we aim to create techniques that can solve the major challenges in utilizing the unjammed bits survived in the reaction time of a reactive jammer to establish the anti-jamming communication. Based on the proposed techniques, we implemented a real-world prototype anti-jamming system, which can collect unjammed bits and assemble them into an original message under the broadband and high-power reactive jamming attacks.

Note that our goal is to raise the wireless communication from non-existence in extremely hostile environments (e.g., battlefield) to being available, rather than support high-speed applications like video streaming in benign environments. When FHSS and DSSS cannot deliver a single bit, the pro-posed techniques can still maintain the wireless communication.

The contribution of this paper is three-fold: (1) we developed novel techniques to harness the reaction time of a reactive jammer for anti-jamming communication; (2) we designed a communication system that integrates the proposed techniques to enable information exchange between wireless devices under broadband and high-power reactive jamming attacks; (3) we implemented a prototype using the USRP platform [23], and evaluated the performance on top of the prototype implementation.

The rest of the paper is organized as follows. Section 2 discusses the technical challenges we face in this work. Sections 3, 4, and 5 present the proposed techniques for addressing these technical challenges. Section 6 describes the implementation and evaluation.

II. TECHNICAL CHALLENGES

To facilitate the readers' understanding of the technical chal-lenges, we first describe a basic design of the sender for achieving such an anti-jamming system. To transmit a mes-sage, the sender may take a random backoff before each transmission, as shown in Figure 1. This makes it hard for the reactive jammer to predict when the sender will start the next transmission. The jammer may attempt to jam the communication for longer time periods. However, this will increase the chance for the reactive jammer to be detected and removed. The sender transmits each bit of the message for multiple times (e.g., three times in Figure 1) to increase the chance that the receiver receives this bit.

Note that there may exist other ways to design the sender. For example, if the sender resides in the power range of the jammer, the need of random backoffs can be removed. Before each transmission, the sender may perform channel sensing to determine whether or not the jammer is transmitting. If not, the sender immediately sends bits without waiting for the backoff time to expire, and hence the system throughput can be greatly improved. Nevertheless, as our initial investigation, in this proposal we focus on the basic design that utilizes random backoffs and retransmissions.

Although the design of the sender can be simple and straightforward, the design of the corresponding receiver is difficult and complicated.

A receiver should have the following essential capabilities as shown in Figure 2. First, the receiver receives a series of bits from the wireless channel. Among these received bits, the receiver should be able to extract unjammed bits that carry useful information about original messages. Thus, the receiver needs to process each received bit with a jamming detector, which checks if this bit is jammed, and discard all jammed bits. Second, to assemble unjammed bits into an original message, the receiver should be able to achieve bit synchronization, i.e., to identify the correct positions of received unjammed bits in an original message. Accordingly, the receiver needs to feed the output of the jamming detector to a bit synchronization decoder to achieve this goal.

Finally, if a smart jammer knows that such an anti-jamming system that collects unjammed bits for communication is being used, in addition to reactive jamming, the jammer may try to defeat the system by transmitting fake bits to the receiver when the sender is not transmitting. These fake bits can mislead the receiver to incorrectly decode a message. In this paper, we refer to such attacks as pollution attacks. The receiver should be able to address pollution attacks to make the decoding of the original message feasible. Thus, the receiver needs to enforce defending techniques against pollution attacks throughout the communication.

Detecting Jammed Bits: Traditional jamming detection aims to find out if wireless communication is jammed (e.g., [36], [45]). To detect jamming, a receiver usually analyzes received signal samples to obtain statistical values like packet loss rate and bit error rate. These values enable the receiver to make a decision regarding whether or not the communication system is under jamming at-tacks. However, traditional techniques cannot be directly applied to achieve anti-jamming systems that reconstruct messages by assembling unjammed bits together, because in such systems the receiver needs to distinguish jammed bits from unjammed bits rather than merely detecting the existence of jamming. Therefore, reliable jamming detection techniques that can identify unjammed

©IJRASET: All Rights are Reserved

bits should be created to realize such anti-jamming systems.

Bit Synchronization: Bit synchronization errors will prevent the receiver from correctly assembling the un-jammed bits from the sender into an original message. Bit synchronization errors are mainly caused by lost, duplicated, and inserted bits. A bit of the sender's original message may get lost when it is jammed by the jammer. Also, if the sender transmits each bit of a message for multiple times, the receiver will receive extra bits. With the presence of lost and extra bits, the receiver cannot know the correct positions of received unjammed bits in the original message, and thus it fails to recover this message. Therefore, techniques should be created to establish the correct bit synchronization between the sender and the receiver in the presence of bit synchronization errors.

Dealing with Pollution Attacks: One possible way to distinguish the sender's bits from the jammer's fake bits is to use existing physical layer fingerprinting techniques such as radio-metrics (e.g., [6]) and radio frequency (RF) fingerprints (e.g., [25], [47]). However, recent research discoveries reveal that physical layer fingerprinting techniques are vulnerable to cer-tain security threats (e.g., mimicry attacks [21]), and it has been demonstrated that an attacker can easily forge physical layer fingerprints to impersonate a target wireless device (e.g., [11], [12], [21]). Thus, secure and reliable countermeasures against pollution attacks should be designed to make the use of unjammed bits for anti-jamming communication feasible.

In what follows, we present the proposed techniques to deal with these challenges. We made the following clarifications of the jammer. First, a general reactive jammer jams the channel when it detects the sender's transmission and stops jamming when the sender ends transmission. In this paper, however, we assume a more challenging reactive jammer model with unpredictable jamming behavior, i.e., the jammer jams the channel when it detects the sender's transmission, but after the sender stops transmission, instead of immediately stopping, the jammer chooses a random value (0), and lasts jamming for second(s).

Second, multiple jammers may collaborate to jam the communication channel. The impact of these jammers can be reduced to that of one jammer. Specifically, if the jammers take turns to jam the channel in a seamless way (i.e., all time slots are jammed), then the jamming impact is similar to that caused by one constant jammer. If the jammers take turns to jam the channel in an unseamless way (i.e., some time slots are not jammed), then the jamming impact is similar to that caused by one random jammer who jams the channel at a random time and lasts jamming for a random duration. For a constant jammer that jams all channels, overwhelms the spreading gain, and never stops, no existing methods can be used to beat such a jammer. Localization or social engineering approaches might be used to physically find the jammers so that they can be disabled. However, as long as unjammed time slots exist, the proposed techniques can decode bits received during the unjammed time slots into a meaningful message.

III. DETECTION OF (UN)JAMMED BITS

In this section, we develop a novel technique that utilizes physical layer modulation properties to identify (un)jammed bits.

A. Preliminaries on Modulation

I/Q modulation has been widely used in modern wireless systems, including WCDMA, WiMax, ZigBee, WiFi, and DVB (Digital Video Broadcasting). I/Q modulation encodes data bits into physical layer symbols, which are the trans-mission units in the physical layer. In the following, we use Quadrature Phase-Shift Keying (QPSK) modulation, a typical I/Q modulation, to illustrate how I/Q modulation works.

QPSK – An Example I/Q Modulation: QPSK encodes two bits into one symbol at a time. In Figure 3, bits 00, 01, 10, and 11 are represented by points whose coordinates are (0; 1), (1; 0), (0; 1), and (1; 0) in an I/Q plane, respectively. The I/Q plane is called a constellation diagram. A symbol is the coordinate of a point in the constellation diagram. For a bit sequence 0010, the modulation output are two symbols: (0; 1) and (0; 1). A received symbol is not exactly the same as the original symbol sent by the sender, since wireless channels usually introduce noise to signals that pass through them [13]. To demodulate, the receiver finds the point that is closest to the received symbol in the constellation diagram. For example, in Figure 3, the point closest to the received symbol is (0, 1). Thus, the demodulation output is 00.

©IJRASET: All Rights are Reserved

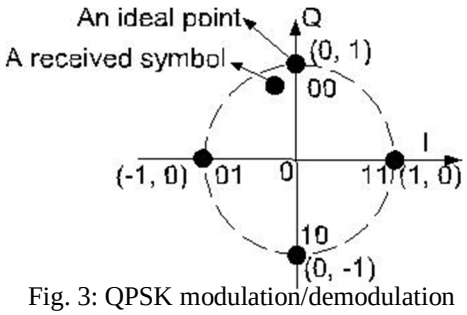


Fig. 3: QPSK modulation/demodulation

B. Observation

Intuitively, jamming signals can introduce a large distortion to signals transmitted by the sender, since the goal of the jammer is to corrupt the signals. If a received symbol is jammed, it may greatly deviate from its ideal point in the constellation diagram and can hardly be recovered. To get more insights in this process, we perform experiments to examine the impacts of jamming on symbol locations.

We collect the received symbols using USRPs [23], which are radio frequency (RF) front ends equipped with analog to digital (AD) and digital to analog (DA) converters. In our experiments, three USRPs are used as the sender, the receiver, and the jammer, respectively, each of which is connected to a computer. Automatic gain control (AGC) is employed by USRPs. We set the bit rate as 1Mbps, carrier frequency as 5GHz, and modulator as QPSK.

We consider a normal scenario and a jamming scenario. In the first one, only the sender transmits randomly generated packets to the receiver, while in the second one, both the sender and the jammer transmit random packets to the receiver concurrently. The receiver record the coordinates of the received symbols in the constellation diagram.

In the normal scenario, as shown in Figure 4, the received symbols form four clusters, each of which centers around an ideal point of QPSK. However, in the jamming scenario, the received symbols randomly spread over the constellation diagram. Thus, it is hard to identify the ideal points for the received symbols, and demodulation errors may happen frequently.

C. Detection Method

Let dunjam (or djam) be the distance between an unjammed (or a jammed) symbol and the origin in the constellation dia-gram. As shown in the above experiment, unjammed symbols are close to their ideal constellation points, and thus dunjam approximately equals to the distance between an ideal point and the origin. In contrast, jammed symbols deviate from their ideal points. Due to AGC, such deviation is actually a convergence from ideal points toward the origin rather than an expansion out of the constellation diagram range. Hence, unlike unjammed symbols, jammed symbols are randomly distributed within the constellation diagram, and the expected value of djam is smaller than that of dunjam. For example, in Figures 4 and 5, the average distance between a received symbol and the origin is 2.2524 and 1.2628, respectively.

We propose to use the distance d between a received symbol and the origin of the constellation diagram as a metric to detect the existence of jammed symbols. For each received symbol, we compute the corresponding distance d, and compare d with a threshold t. If d > t, the received symbol is marked as unjammed. Otherwise, it is jammed and we discard it.

Note that different metrics can be explored to accommodate different variants of I/Q modulation. For example, rectangular based I/Q modulation (e.g., 64 QAM) may use the distance between a received symbol and the closest constellation point as the detection metric.

IV. FALSE POSITIVES AND FALSE NEGATIVES

False positives (FP) and false negatives (FN) are two types of errors that may happen in the detection. In a false positive, In a false negative, the temporal sequences are larger than t, and thus a jammed symbol is incorrectly classified as an unjammed symbol. In Theorems 1 and 2 we derive both probabilities of false negative and positive.

V. BIT SYNCHRONIZATION

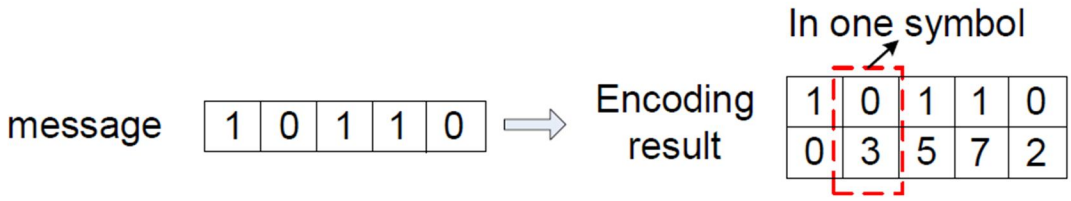
The original message is first encoded with a traditional ECC (e.g., Reed-Solomon codes). ECC corrects substitution errors (i.e., bit "'1" is replaced by "0" and vice-versa). The proposed bit synchronization encoding scheme further encodes the ECC-coded message to allow a receiver to decode the correct positions of received bits and recover from synchronization errors.

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

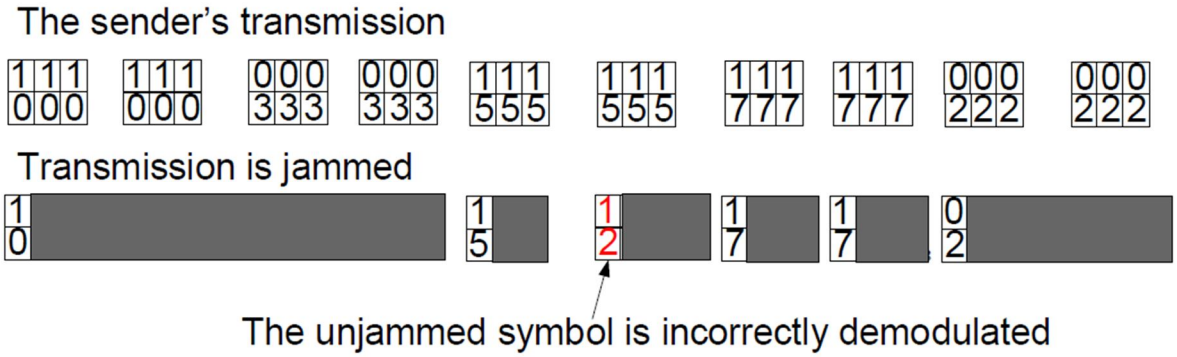
A. Basic Idea

1) *Bit Synchronization Encoding:* The sender and the receiver agree on a sequence that is formed by n integers, where n is the length of the message (a long message may be spited into several short messages of n bits). We call such an integer sequence a positioning code and each integer in the sequence a label. As shown in Figure 8, the message is 10110 and the positioning code is 03572. For 1 i 5, the sender labels the i-th bit of the message using the i-th label in the positioning code (e.g., the second bit is 0 and its label is 3). In the labeling, the sender uses one symbol to represent both a bit and its label. (Details of labeling will be presented in Section 4.2.) Note that a symbol is the transmission unit of physical layer. Once a receiver receives a symbol, the receiver knows both the bit and its label.



Bit Synchronization Encoding at the Sender

2) *Transmission Errors:* After encoding, the sender transmits each symbol for multiple times. The sender transmits the first symbol for 3 times in the first transmission, and transmits this symbol again for 3 times in the second transmission.
3) *Bit Synchronization Decoding:* The receiver demodulates each received symbol to extract the bit and corresponding label carried by this symbol. Figure 10 shows an example following Figure 9. The extracted bits and labels are 111110 and 052772, respectively. The receiver then takes two steps to correct synchronization errors.



Bit synchronization decoding at the receiver

The first step is merging, in which bits are merged into a single bit if they are identical and have the same label. The merging result is 11110 and the corresponding labels are 05272. An incorrect merging may happen if multiple bits in the BTmessage are identical and use the same label. The second step is alignment, which consists of two substeps:

a) *Dealing with False Negatives:* Although most un-jammed symbols can be correctly demodulated by the existing demodulation techniques, the demodulation of a small amount of them may be incorrect due to the channel noise. These wrong symbols are actually random incoherent pieces and the correlation between their labels and the positioning code is weak. Therefore, to filter out inserted bits, we perform alignment on the most correlated part between the positioning code and the merged received labels. We find the largest common subsequence (LCS) between the positioning code and received labels, and align the LCS with the positioning code. For example, in Figure 10, the received labels are 05272, where the underlined 2 is the inserted label from the jammer. The LCS between 05272 and the positioning code 03572 is 0572. Thus, the inserted label is filtered.

b) *Generating Alignment Output:* In the LCS 0572, the labels 0, 5, 7, and 2 match the 1st, 3rd, 4th, and the last label in the positioning code, respectively. Thus, the receiver knows that the second bit is lost, and corrects synchronization errors by filling a bit that can be either 1 or 0 in the position shown in Figure 10. The alignment output is further processed by traditional ECC to recover the original message. There may exist multiple alignment outputs. In Section 4.3, we develop a fast alignment

©IJRASET: All Rights are Reserved

approach that not only achieves desired alignment accuracy, but also reduces the overhead by only trying a subset of all combinations.

B. Encoding at Sender

The sender adds special data content (e.g., 11111) to both the beginning and the end of a message, so that a receiver can recognize the boundary of a message. We refer to the special data content as a message delimitation code (MDC).

Afterwards, the sender labels the i-th bit of the message by packing the i-th bit and the i-th label of the positioning code into one physical layer symbol. For example, assume that i-th bit is 1 and its label is 2. The sender appends 10 (i.e, binary form of 2) to the data bit 1, and the result is 110, which are modulated into one symbol (e.g., a 8PSK symbol). To improve efficiency, bits in the MDC are not labeled. For an M-ary modulator that encodes $\log_2 M$ bits by one symbol, the maximum value of a label of the positioning code should be $2^{\log_2 M} 1 = {}^{M}_{2} 1$. For example, an 8PSK symbol uses one bit to carry data information and two bits to carry the label. Hence, a label is less than or equal to 3 (i.e., 11). Packing a data bit and its label in one symbol achieves atomicity: data bits are always associated with their labels. Upon receiving a symbol, the receiver knows both the data bit and its label.

C. Decoding at Receiver

Before decoding, the receiver searches for boundaries of a message. The boundary of the message is identified if the receiver can observe an MDC or a certain data pattern that is a part of MDC. For example, assume that the MDC equals to 1111111, message if the receiver receives 1111111, multiple consecutive 1's (e.g., 1111), or multiple consecutive 1's interleaved with quite a few 0's (e.g., 1110111). The third condition deals with bits inserted by false negatives.

To reduces the chances that the entire MDC is jammed, the sender and the receiver can increase the length of the MDC according to the severity of jamming attacks, so that the receiver can observe at least a part of the MDC.

The receiver then demodulates the symbols of the received message, extracts a data bit and a label from each symbol, and takes two steps to correct synchronization errors.

1) *Basic Alignment Method:* If the length of the position-ing code is small, we can do alignment in a brute force way. Specifically, assume that the length of L is q. The receiver can find all length-q subsequences of S, and compare each of them with L. For each subsequence that equals to L, the receiver generates an alignment output by padding 1's or 0's into the positions of lost bits. For example, assume that padding bits are 1's and the received message after merging is 00. For L = 17 and S = 1317, the alignment outputs are 0110 and 1100. Each alignment output is further processed by traditional ECC decoding, where replacement errors (i.e., 1 ! 0 or 0 ! 1) are corrected. Since there may exist multiple alignment outputs, the receiver may obtain multiple decoding results, among which the one that can pass cyclic redundancy check (CRC) or authentication is the recovered message. If the length of S is large, this method is time consuming. We develop a fast alignment approach below to reduce the overhead.
2) *A Fast Alignment Method:* To achieve fast alignment, we propose to only find one alignment. We further show that given proper configurations, this single alignment leads to a very small error probability. We use a simple greedy strategy to obtain a single alignment. Specifically, the receiver compares labels of L with those of the positioning code S, trying to find S's leftmost or rightmost subsequence that equals to L. For example, if L = 17 and S = 1177, the S's leftmost and rightmost subsequence that equals to L is underlined in 1177 and 1177, respectively. The positions of the leftmost/rightmost subsequence is 13/24, and thus the corresponding decision is that the first and the third bits of the message are received (or the second and the last bits are received).

VI. CONCLUSION

We developed an anti-jamming system that can enable wireless communication when a broadband and high power reactive jammer is present. The designed system delivers information by harnessing the reaction time of a reactive jammer. It does not assume a reactive jammer with limited spectrum coverage and transmit power, and thus can be used in scenarios where traditional approaches fail. We implemented a prototype of such system based on GNU Radio. Our results showed that the prototype achieved a reasonable throughput when 802.11 DSSS and GNURadio benchmark were completely disabled by the jammer.

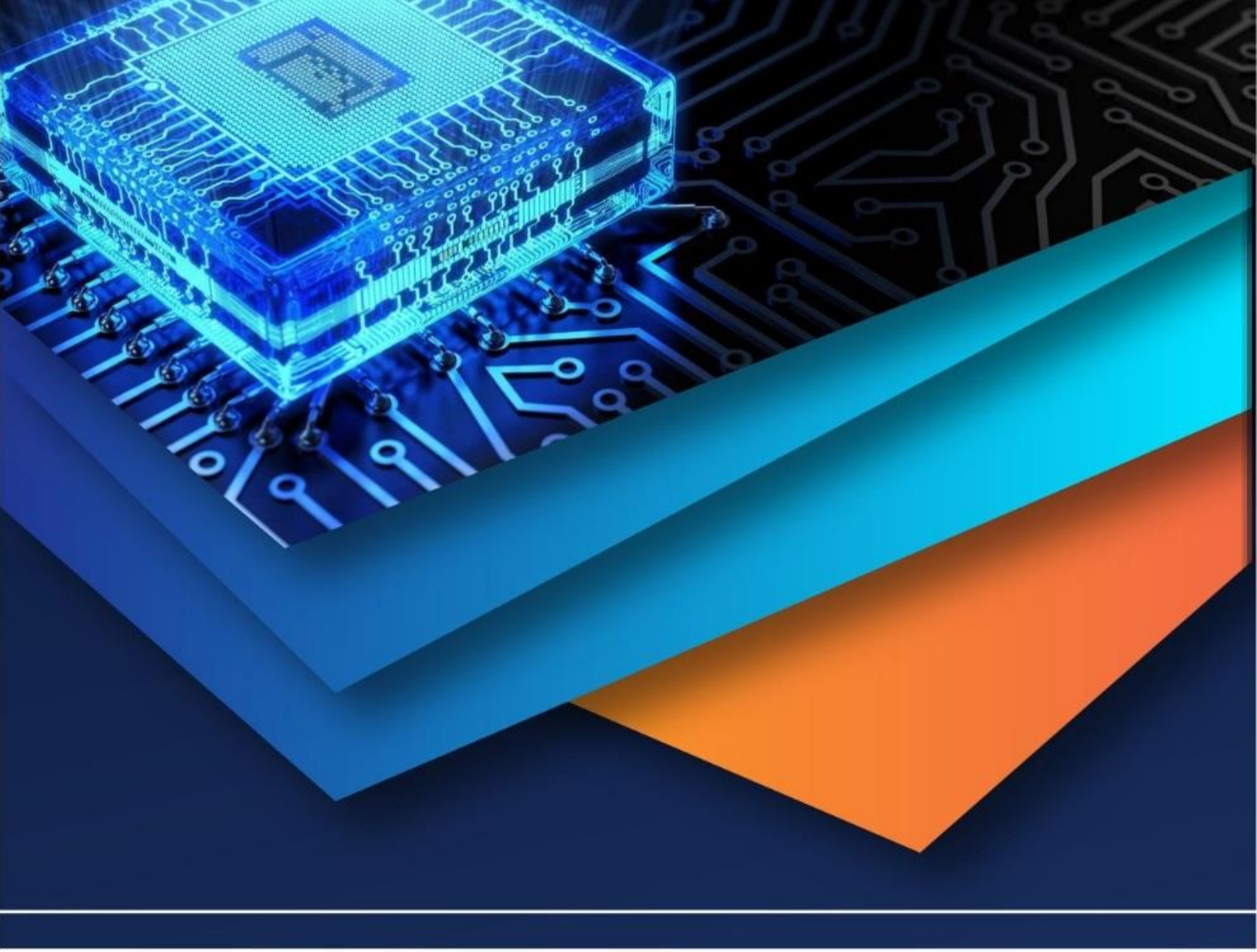
REFERENCES

[1] GNU Radio - The GNU Software Radio. http://www.gnu.org/software/ gnuradio/.
[2] Reactive jamming technologies. http://www.ece.gatech.edu/academic/ courses/ece4007/08fall/ece4007l02/lm5/jammer.doc.

©IJRASET: All Rights are Reserved

[3] Jane's Military Communications, Edition 22, 2001–2002. Jane's Infor-mation Group INC, Virginia, USA, 2002.
[4] L. Baird, W. Bahn, and M. Collins. Jam-resistant communication without shared secrets through the use of concurrent codes. Technical report, US Air Force Academy, 2007.
[5] A. J. Berni and W. D. Greeg. On the utility of chirp modulation for digital signaling. IEEE Trans. on Communications, 21:748–751, 1973.
[6] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In Proceedings of the 14th ACM international conference on Mobile computing and networking (MobiCom '08), pages 116–127, 2008.
[7] D. Cabric, A. Tkachenko, and R. W. Brodersen. Experimental study of spectrum sensing based on energy detection and network cooperation. In ACM TAPAS '06: Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum, 2006.
[8] J. Chiang and Y. Hu. Extended abstract: Cross-layer jamming detection and mitigation in wireless broadcast networks. In Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07), 2007.
[9] J. Chiang and Y. Hu. Dynamic jamming mitigation for wireless boradcast networks. In Proceedings of IEEE International Conference on Computer Communications (INFOCOM '08), 2008.
[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms, 2nd. MIT Press, 2001.
[11] B. Danev, H. Luecken, S. Capkun, and K. E. Defrawy. Attacks on physical-layer identification. In Procceedings of the 3nd ACM Conference on Wireless Networking Security (WiSec '10), pages 89–98, March 2010.
[12] M. Edman and B. Yener. Active attacks against modulation-based radiometric identification. Technical Report TR 09-02, Rensselaer Polytechnic Institute, 2009.
[13] A. Goldsmith. Wireless Communications. Cambridge University Press, August 2005.
[14] S. Hengstler, D. P. Kasilingam, and A. H. Costa. A novel chirp mod-ulation spread spectrum technique for multiple access. In Proceedings of the IEEE International Symposium on Spread Spectrum Techniques and Applications, pages 73–77, September 2002.
[15] H. Kim and K. G. Shin. In-band spectrum sensing in cognitive radio networks: energy detection or feature detection? In MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking, pages 14–25, 2008.
[16] L. Lazos, S. Liu, and M. Krunz. Mitigating control-channel jamming attacks in multi-channel ad hoc networks. In Proceedings of 2nd ACM Conference on Wireless Networking Security (WiSec '09), March 2009.
[17] M. Li, I. Koutsopoulos, and R. Poovendran. Optimal jamming attacks and network defense policies in wireless sensor networks. In Proceed-ings of IEEE International Conference on Computer Communications (INFOCOM '07), 2007.
[18] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang. Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure. In Proceedings of the 26th Annual Computer Security Applications Conference ACSAC '10, December 2010.
[19] Y. Liu and P. Ning. Poster: Mimicry attacks against wireless link signature. In Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'11), 2011.
[20] Y. Liu and P. Ning. BitTrickle: Defending against broadband and high-power reactive jamming attacks. In Technical Report TR-2011-17, NC State University, Computer Science Department, July 2011.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)