ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

An Enhanced Jamming Attack Detection Using Bittrickle Methodology

K.Gayathri[1], K.Renugadevi[1], L.Kiruthika[1], J.Rathika[1], D.Sreearthi[2]

Department of ECE, Angel College of Engineering and Technology, Tirupur, Tamilnadu, India[1]

Asst. Professor, Department of ECE, Angel College of Engineering and Technology, Tirupur, Tamilnadu, India[2]

ABSTRACT: When target devices are transmitting a reactive jammer jams the wireless channel; Reactive jamming is harder to track and compensate against the constant jamming methods. Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum(DSSS) have been widely used against Jamming attacks. If high power transmitter or frequency channel is jammed then both these method will fail. We use broad band and high power reactive jammer to avoid jamming attacks. In this proposed method we use bit trickle method to avoid jamming. Bit Trickle is a method which is used to find attackers when using broadband communication between transmitter and receiver. Users having secret key to access the data from anti jammer. Once the security key is correct then only the transmission occurs.

KEYWORDS: wireless network; jamming; anti-jamming; bit trickle method

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. . In FHSS, the sender and the receiver switch their communication channels periodically to avoid being jammed. In DSSS, the sender multiplies the original message with a pseudo random sequence to obtain the spreading gain. If the jammer's power is not strong enough to overwhelm the DSSS signals with the spreading gain, the receiver can use the same pseudorandom sequence to recover the message. Reactive jamming attacks are among the most effective jamming attacks. 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. 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" . Channel sensing causes a short delay. For example, energy detection is the most popular channel sensing approach with very small sensing time. When implemented in a fully parallel pipelined FPGA for fast speed, 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 . Therefore, before the jammer actually jams, the sender has already transmitted R bits, where t is the reaction time of the jammer and R is the transmission rate of the sender. It is easy for people to conceive that the receiver may collect information bits from the un jammed 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 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.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

II. RELATED WORK

This paper[1],The technique used false positive and false negative. We propose an anti-jamming communication system that allows communication in the presence of a broadband and high power reactive jammer. The advantage is it delivers information by harnessing the reaction time of the reactive jammer, but it does not assume a reactive jammer with limited spectrum coverage and transmit power.

This paper[2],we consider jamming detection using key exchange in wireless network which is based on (JADE) scheme. IT could defeat the jamming attack and maintain the considerable performance of the overall network. But it has low detection range.

This paper[3],uses the game theoretic strategies .Motivated by the high energy –consuming nature of jamming, We propose our defense strategy to defeat the jammer by draining its energy as fast as possible. Jammer replacement problem, scheduling, routing and resource allocation under jamming attacks are being identified

III. OVERVIEW OF BITTRICKLE

As discussed earlier, Bit Trickle exploits the sensing delay of reactive jamming to enable message transmission. In this section, we describe the high-level behaviors of the sender and the receiver, respectively, and then discuss the technical problems that need to be solved to build Bit Trickle.

A. Transmission at the Sender

The sender encodes a message using a *Bit Trickle encoder*, which enables the receiver to recover the message in the presence of partial message corruption. A message is encoded in a way such that the receiver can identify the boundary of are received encoded message, and thus no extra synchronization preamble is required for transmitting the encoded message. If a message is too long, the sender may first split it into short messages, and then encode those short messages.

To transmit encoded message, the sender needs to find times when the jammer is not jamming. The sender may take a random back off 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 .

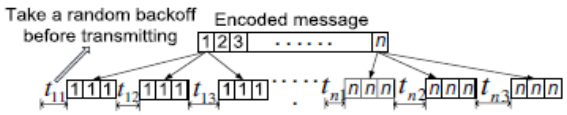


Fig. 1. Transmission at the sender

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. If the sender resides in the power range of the jammer, the need of random back offs 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 back off time to expire. The sender may transmit each bit of the encoded message for multiple times to increase the chance that the receiver receives this bit. Note that Bit Trickle can also address random jamming attacks. In both reactive and random jamming scenarios, a common feature is that multiple transmitted bits are lost. In Section IV, we will show how Bit Tickle deals with lost bits.

Figure 1 shows the most conservative situation, where the sender transmits one bit a time. The sender can improve the performance by transmitting multiple bits a time. To this end, the sender may learn how many bits can be delivered within the jammer's reaction time through, for example, detecting the point where jamming happens or using ACK packets from the receiver, which can be transmitted in a similar way. This paper focuses on the transmission of single bits. Extending the approach to transmitting multiple bits a time is straightforward.

B. Reception at the Receiver

The receiver's task is to extract the un jammed bits and reconstruct the original message from these un jammed bits,

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

which are possibly collected from multiple transmission

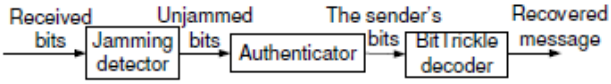


Fig. 2. Reception at the Receiver

Extracting Un jammed Bits: Figure 2 shows the high-level view of the receiver's operations. The receiver processes each received bit with the *jamming detector*, which checks if this bit is jammed, and discard all jammed bits. The output of the jamming detector is thus a collection of un jammed bits.

Dealing with Pollution Attacks: An intelligent jammer may attempt to pollute the un jammed bits to defeat our scheme. Specifically, the jammer transmits fake bits to the receiver when the sender is not transmitting. Those fake bits can cause a high decoding complexity (e.g., exponential complexity) at the receiver. Therefore, the receiver should have the ability to remove the jammer's bits. Accordingly, the receiver feeds the output of the jamming detector to an *authenticator*, which distinguishes the sender's bits from the jammer's bits by using physical layer authentication approaches such as radio metrics (e.g., [4]) and radio frequency (RF) fingerprints (e.g., [14]). As cryptographic authentication faces cryptanalysis based attacks (e.g., birthday attacks against digital signatures), physical layer authentication may also face similar threats as revealed by [7]. A hybrid of multiple physical layer approaches may be explored to defense against sophisticated attacks. How to improve the authentication capability, including cryptographic and physical layer authentication, is complementary to this work.

We assume that the jammer cannot break physical layer authentication approaches. However, In practice, false negatives and false positives may happen with small probabilities when those approaches are employed. With a false negative, the jammer's bits are identified as the sender's bits. With a false positive, the sender's bits are identified as the jammer's bits. Although false negatives and false positives are events of small probabilities, they may introduce a small number of inserted bits and lost bits. In Section IV, we will show how the receiver handles inserted and lost bits.

Reconstructing Original Message: After obtaining un jammed bits, the receiver still faces several challenges in reconstructing the original message: First, a bit may get lost, if itself and all its copies are jammed by the jammer or mistaken as a false positive. Second, the sender transmits each bit for multiple times, and thus the receiver may receive duplicate bits. In addition, false negatives insert a small number of jammer's bits into the input of the decoder. Therefore, to deal with transmission errors such as inserted, lost, and duplicate bits, the receiver utilizes a *Bit Trickle decoder*, which corresponds to the *Bit Trickle encoder* used by the sender.

C. Technical Challenges

Detecting (Un)Jammed Bits: Traditional jamming detection aims to find out if wireless communication is jammed (e.g., [16], [19]). However, jamming detection in Bit Trickle needs to distinguish jammed bits from un jammed bits. One may suggest the use of received signal strength (RSS) of each bit to distinguish jammed and un jammed bits. Unfortunately, this method will fail when the distribution of RSS values are inherently time-varying due to reasons like the movement of communicators or the use of power control techniques. In Section III, we propose a novel jamming detector that uses modulation properties to distinguish jammed and un jammed bits. The proposed detector does not rely on RSS values, and thus can be used in general wireless applications that have either dynamic or static RSS.

Bit Trickle Decoder: Transmission errors such as lost or duplicate bits may happen when there exist jamming attacks or a retransmission mechanism is employed. A small number of lost or duplicate bits can make many bits misaligned, which greatly reduce the efficiency of ECC. To deal with lost and duplicate bits, we develop Bit Trickle decoder, which can find the original position for each received bit in the encoded message, and enable the use of ECC with high efficiency

. IV. 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

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

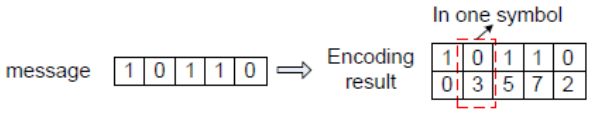
Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

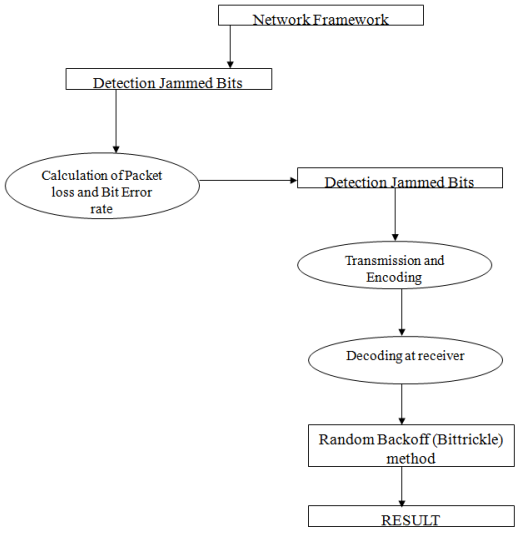
message for multiple times, the receiver will receive extra bits. With the presence of lost and extra bits, the receiver cannot know the correct positions of received un jammed bits in the original message, and thus it fails to recover this message. Therefore, techniques should be created to establish 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 use physical layer finger print such as radio-metrics and radio frequency (RF) fingerprinting techniques are vulnerable to certain security threats has been demonstrated that an attacker can easily forge physical layer fingerprints to impersonate a target wireless. Thus, secure and reliable countermeasures against pollution attacks should be designed to make the use of un jammed 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 and lasts jamming for second.

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. 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. In the labeling, the sender uses one symbol to represent both a bit and its label. Once a receiver receives a symbol, the receiver knows both the bit and its label. The sender repeats its transmission until the last symbol is transmitted. Due to jamming and retransmissions, a symbol may get lost or duplicated. Also, channel noise may introduce a small number of incorrectly demodulated symbols, and thus extra inserted symbols can be resulted.



V.FLOW CHART:



ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

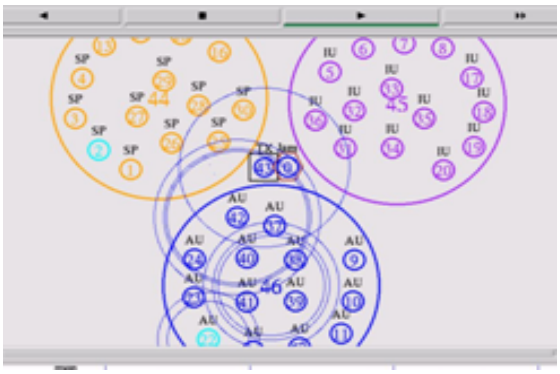
International Journal of Innovative Research in Computer and Communication Engineering

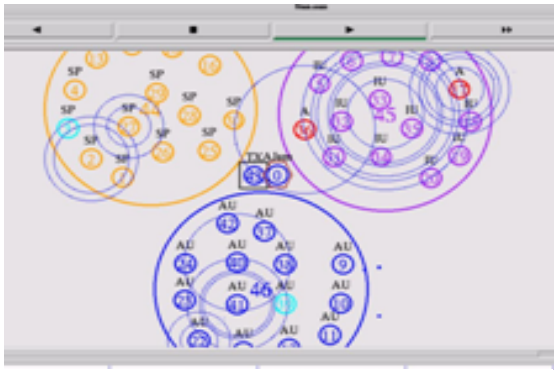
(A High Impact Factor, Monthly, Peer Reviewed Journal)

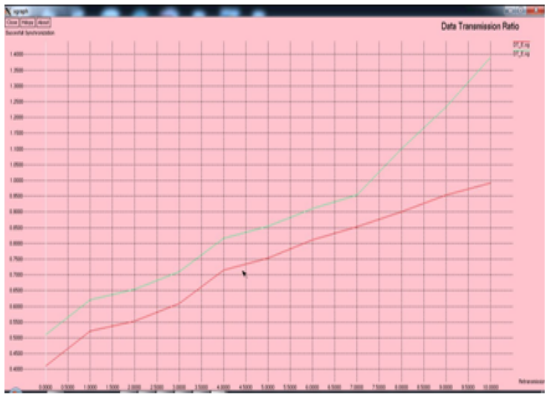
Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

VI.PROCESS







VII.BENEFITS

Low cost:

There is no need to spend time and money to obtain licenses since Linux and much of its software comes with the GNU General Public License. It is able to start working immediately without worrying that your software may stop working anytime because the free trial version expires. Additionally, there are large repositories from which you can freely download high quality software for almost any task you can think of.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

Stability:
Linux doesn't need to be rebooted periodically to maintain performance levels. It doesn't freeze up or slow down over time due to memory leaks and such. Continuous up-times of hundreds of days (up to a year or more) are not uncommon.
Performance:
Linux provides persistent high performance on workstations and on networks. It can handle unusually large numbers of users simultaneously, and can make old computers sufficiently responsive to be useful again.
Network friendliness:
Linux was developed by a group of programmers over the Internet and has therefore strong support for network functionality; client and server systems can be easily set up on any computer running Linux. It can perform tasks such as network backups faster and more reliably than alternative systems.
Flexibility:
Linux can be used for high performance server applications, desktop applications, and embedded systems. You can save disk space by only installing the components needed for a particular use. You can restrict the use of specific computers by installing for example only selected office applications instead of the whole suite.
Compatibility:
It runs all common UNIX software packages and can process all common file formats.
Choice:
The large number of Linux distributions gives you a choice. Each distribution is developed and supported by a different organization. You can pick the one you like best; the core functionalities are the same; most software runs on most distributions.
Fast and easy installation:
Most Linux distributions come with user-friendly installation and setup programs. Popular Linux distributions come with tools that make installation of additional software very user friendly as well.
Full use of hard disk:
Linux continues work well even when the hard disk is almost full.
Multitasking:
Linux is designed to do many things at the same time; e.g., a large printing job in the background won't slow down your other work.
Security:
Linux is one of the most secure operating systems. "Walls" and flexible file access permission systems prevent access by unwanted visitors or viruses. Linux users have to option to select and safely download software, free of charge, from online repositories containing thousands of high quality packages. No purchase transactions requiring credit card numbers or other sensitive personal information are necessary.
Open Source:
If software is developed which requires knowledge or modification of the operating system code, Linux's source code is at your fingertips. Most Linux applications are Open Source as well.

3.5 NETWORK SIMULATOR-2

After setting up the platform, software named ns2 was set up on it which was used for all the analysis and simulation work apart from other tools used. Ns2 is the de facto standard for network simulation. Its behavior is highly trusted within the networking community. It is developed at ISI, California, and is supported by the DARPA and NSF. Ns2 is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. This means that most of the simulation scripts are created in Tcl. If the components have to be developed for ns2, then both Tcl and C++ have to be used. Ns2 uses two languages because any network simulator, in general, has two different kinds of things it needs to do. On the one hand, detailed simulations of protocols require a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

VIII.CONCLUSION

We developed Bit Trickle Method in order to find out the jammed nodes and find an alternate node randomly. We aim to create techniques that can solve major challenges in utilizing un Jammed bits in the reaction time of a reactive jammer to establish anti jamming communication. We implemented a real world prototype anti jamming system which can collect un Jammed bits and assemble them into a original message under broad band and high power reactive jamming attacks. There is no data loss and it is very speed and secure.

REFERENCES

[1] Reactive jamming technologies. http://www.ece.gatech.edu/academic/courses/ece4007/08fall/ece4007l02/lm5/jammer.doc.
[2] Y. Liu and P. Ning. Bittrickle: Defending against broadband and highpower reactive jamming attacks. Technical Report TR-2011-17, NC State University, Computer Science Department, July 2011.
[3] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential dsss: Jamming-resistant wireless broadcast communication. In *Proceedings of the 2010 IEEE INFOCOM*, 2010.
[4] Robert A. Scholtz. *Spread Spectrum Communications Handbook*. McGraw-Hill, 2001.
[5] W. Xu, W. Trappe, and Y. Zhang. Anti-jamming timing channels for wireless networks. In *WiSec '08: Proceedings of the first ACM conference on Wireless network security*, pages 203–213, New York,NY, USA, 2008. ACM.
[6] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "Short paper: Reactive jamming in wireless networks: How realistic is the threat?" in Proc. 4th ACM Conf. Wireless Netw. Secur., 2011, pp. 47–52.
[7] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: Attack and defense strategies," IEEE Netw., vol. 20, no. 3, pp. 41–47, May/Jun. 2006
[8] K. Grover, A. Lim, and Q. Yang, "Jamming and anti–jamming techniques in wireless networks: A survey," Int. J. Ad Hoc Ubiquitous Comput., vol. 17, no. 4, pp. 197–215, 2014.
[9] D. Giustiniano, V. Lenders, J. B. Schmitt, M. Spuhler, and M. Wilhelm, "Detection of reactive jamming DSSS-based wireless networks," in Proc. 6th ACM Conf. Secur. Privacy Wireless Mobile Netw., 2013, pp. 43–48.
[10] E.-K. Lee, S. Y. Oh, and M. Gerla, "Randomized channel hopping scheme for anti-jamming communication," in Proc. IFIP Wireless Days (WD), Oct. 2010, pp. 1–5.
[11] Y. Liu and P. Ning, "BitTrickle: Defending against broadband and highpower reactive jamming attacks," in Proc. IEEE INFOCOM, Mar. 2012, pp. 909
[12] W. Xu, W. Trappe, and Y. Zhang, "Anti-jamming timing channels for wireless networks," in Proc. 1st ACM Conf. Wireless Netw. Secur., New York, NY, USA, 2008, pp. 203–213.