IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Packet Hiding Methods for Preventing Selective Jamming Attacks

Vani. S[*1], Rachelin Sujae[2]

Department of Electronics and Communication Engineering, Bharath University, Selaiyur, Chennai, India

Vanidhivya@gmail.com

Abstract

Coverage and energy conservation are two major issues in wireless sensor networks (WSNs), especially when sensors are randomly deployed in large areas. In such WSNs, sensors are equipped with limited lifetime batteries and redundantly cover the target area. To face the short lifetime of the WSN, the objective is to optimize energy consumption while maintaining the full sensing coverage. A major technique to save the energy is to use a wake-up scheduling protocol through which some nodes stay active whereas the others enter sleep state so as to conserve their energy. This study presents an original algorithm for node

Self scheduling to decide which ones have to switch to the sleep state. The novelty is to take into account the remaining energy at every node in the decision of turning off redundant nodes. Hence, the node with a low remaining energy has priority over its neighbors to enter sleep state. The decision is based on a local neighborhood knowledge that minimizes the algorithm overhead. To verify and evaluate the proposed algorithm, simulations have been conducted and have shown that it can contribute to extend the network lifetime. A comparison with existing works is also presented and the performance gains are highlighted.

Introduction

Uninterrupted availability of the open wireless medium to interconnect participating nodes. However, the open nature of this medium leaves it vulnerable to multiple security threats. Anyone with a transceiver can eavesdrop on wireless transmissions, inject spurious messages, or jam legitimate ones. While eavesdropping and message injection can be prevented using cryptographic methods, jamming attacks are much harder to counter. They have been shown to actualize severe Denial-of-Service (DoS) attacks against wireless networks. In the simplest form of jamming, the adversary interferes with the reception of messages by transmitting a continuous jamming signal , or several short jamming pulses. Typically, jamming attacks have been considered under an external threat model, in which the jammer is not part of the network. Under this model, jamming strategies include the continuous or random transmission of high-power interference signals. However, adopting an “always- on” strategy has several disadvantages. First, the adversary has to expend a significant amount of energy to jam frequency bands of interest. Second, the continuous presence of unusually high interference levels makes this type of attacks easy to detect. Conventional antijamming techniques rely extensively on spread-spectrum (SS) communications , or some form of jamming evasion (e.g., slow frequency hopping, or spatial retreats.

SS techniques provide bit-level protection by spreading bits according to a secret pseudo noise (PN) code, known only to the communicating parties. These methods can only protect wireless transmissions under the external threat model. Potential disclosure of secrets due to node compromise neutralizes the gains of SS. Broadcast communications are particularly vulnerable under an internal threat model because all intended receivers must be aware of the secrets used to protect transmissions. Hence, the compromise of a single receiver is sufficient to reveal relevant cryptographic information.

In this paper, we address the problem of jamming under an internal threat model. We consider a sophisticated adversary who is aware of network secrets and the implementation details of network protocols at any layer in the network stack. The adversary exploits his internal knowledge for launching selective jamming attacks in which specific messages of “high importance” are targeted. For example, a jammer can target route-request/route-reply messages at the routing layer to prevent route discovery, or target TCP acknowledgments in a TCP session to severely degrade the throughput of an end-to-end flow. To launch selective jamming attacks, the adversary must be capable of implementing a “classify-then-jam” strategy before the completion of a wireless transmission. Such strategy can be actualized either

by classifying transmitted packets using protocol semantics, or by decoding packets on the fly. In the latter method, the jammer may decode the first few bits of a packet for recovering useful packet identifiers such as packet type, source and destination address. After classification, the adversary must induce a sufficient number of bit errors so that the packet cannot be recovered at the receiver. Selective jamming requires an intimate knowledge of the physical (PHY) layer, as well as of the specifics of upper layers.

Literature Survey

2.1 Jamming and Sensing of Encrypted Wireless Ad Hoc Networks

Problem Formulation

This paper considers the problem of an attacker disrupting an encrypted victim wireless ad hoc network through jamming. Jamming is broken down into layers and this paper focuses on jamming at the Transport/Network layer. Jamming at this layer exploits AODV and TCP protocols and is shown to be very effective in simulated and real networks when it can sense victim packet types, but the encryption is assumed to mask the entire header and contents of the packet so that only packet size, timing, and sequence is available to the attacker for sensing.

Research Design

Within the framework defined so far this paper provides seven contributions.

- First it demonstrates the potential Transport/Network layer jamming gains within a simulated environment.
- Second a simulated jamming protocol is developed that allows testing on an ad hoc network of lap top computers.
- Third the potential jamming gains are demonstrated on a live network using the simulated jamming protocol.
- Fourth a sensor is developed that uses packet size, timing, and sequence. It uses off-line sensing to adapt an online sensor to the current network conditions and a probabilistic model of the sizes and inter-packet timing of different packet types. A historical method for detecting known protocol sequences is used to develop the probabilistic models.
- The fifth is an active jamming mechanism to force the victim network to produce known sequences for the historical analyzer.
- The sixth is the online classifier that makes packet type classification decisions. The method is tested on live data and found that for many packet types the classification is highly reliable.
- Finally the relative roles of size, timing, and sequence are discussed along with the implications for making networks more secure.

Findings

The simulation and experimental results show that jamming has the potential for large gains, if the packet types are identified. This section describes the approach to sensing packet types. There are two approaches to classifying packets into types. The first classifies packets as they arrive (so-called online classification). The second is allowed to collect more observations before making the decision on packet type (so-called offline classification). Online classification is the preferred approach, but as will be shown in the following subsections, both online and offline classification have a role.

Conclusions and Limitations

This paper presented initial results in designing such a layered attacker for the Transport/Network layer. Jamming can get significant jamming gains, well over 100, when it knows the packet type and timing. Interestingly most of these gains were produced by attacking packets above the ad hoc network layer. Protocols introduce highly predictable timing that can be exploited. The limited information of packet size, timing, and sequence is enough to accurately predict packet types. Future work will fully connect and test the jamming and sensing which were treated separately. The statistical sensing tools continue to be refined. A few representative attacks were presented and the test bed tools described here are being used to methodically evaluate other attacks. Scaling to larger ad hoc networks and networked attackers is the long term g0al.

2.2 Mitigating Control-Channel Jamming Attacks in Multi-channel Ad Hoc Networks

Problem Formulation

We address the problem of control-channel jamming attacks in multi-channel ad hoc networks. Deviating from the traditional view that sees jamming attacks as physical-layer vulnerability, we consider a sophisticated adversary who exploits knowledge of the protocol mechanics along with cryptographic quantities extracted from compromised nodes to maximize the impact of his attack on higher-layer functions

Research Design

New security metrics are defined that quantify the adversary's ability to localize and deny

legitimate nodes access to the control channel. We develop a randomized distributed channel establishment scheme that allows nodes to establish a new control channel using frequency hopping. Under our scheme, network nodes are able to temporarily construct a control channel until the jammer is removed from the network. Our scheme differs from classical frequency hopping in that the communicating nodes are not synchronized on the same hopping sequence, but each node follows a unique hopping sequence. This leads to unique identification of the set of compromised nodes by nearby nodes. Assuming perfect random sequence generators, we analytically evaluate the expected delay until a control channel is re-established and the expected fraction of time that the control channel is available. We verify our analytic results via extensive simulations.

Control Channel Maintenance

Consider a single cluster with each node being within one hop from the CH. Suppose the current control channel is jammed by an adversary. The main idea behind our scheme is to have each node of the cluster hop between channels in a pseudo-random fashion, following a unique hopping sequence not known to other nodes. This way if the jammer captures the hopping sequence of a compromised node, this node can be uniquely identified. Once the compromised node has been identified, the CH updates the hopping sequences of all nodes in the cluster except the compromised one. Hence, the effectiveness of a jammer that exploits knowledge from compromised nodes becomes equivalent to the effectiveness of a jammer that randomly hops between channels. Note that our method is not a permanent solution for the control channel allocation, nor can it permanently be used for data communications due to its high communication overhead and delay. Rather, our scheme temporarily restores a control channel until the jammer and any com-promised nodes are removed from the network.

Conclusions and Limitations

From we conclude that, a randomized distributed channel establishment scheme that allows nodes to select a new control channel using frequency hopping. Our method differs from classical frequency hopping in that the communicating nodes are not synchronized to the same hopping sequence. Instead, each node follows a unique hopping sequence. We showed that our scheme can uniquely identify compromised nodes through their unique sequence and exclude them from the network. We evaluated the performance of our scheme based on the newly proposed metrics of evasion entropy, evasion delay, and evasion ratio. Our proposed scheme can by utilized as a temporary solution for the control channel re-establishment until the jammer and the compromised nodes are removed from the network**.**

2.3 The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks

Problem Formulation

In this paper, we examine radio interference attacks from both sides of the issue: first, we study the problem of conducting radio interference attacks on wireless networks, and second we examine the critical issue of diagnosing the presence of jamming attacks. Specifically, we propose four different jamming attack models that can be used by an adversary to disable the operation of a wireless network, and evaluate their effectiveness in terms of how each method affects the ability of a wireless node to send and receive packets. We then discuss different measurements that serve as the basis for detecting a jamming attack, and explore scenarios where each measurement by itself is not enough to reliably classify the presence of a jamming attack.

Research Design

we propose two enhanced detection protocols that employ consistency checking. The first scheme employs signal strength measurements as a reactive consistency check for poor packet delivery ratios, while the second scheme employs location information to serve as the consistency check. Throughout our discussions, we examine the feasibility and effectiveness of jamming attacks and detection schemes using the MICA2 Mote platform**.**

Findings

In order to understand the effect that a jammer would have on the received signal strength, we performed six experiments. In the first two experiments, we have two Motes, a sender A and a receiver B, which are 30 inches apart from each other. In the ¯first case, A transmits 20 packets per second, corresponding to a traffic rate of 5.28kbps, which we refer to as a CBR source. In the second case, A transmits at its maximum rate; as soon as the send function returns to the application level asynchronously, either because the packet is successfully sent or because the packet is dropped (the packet pumping rate is larger than the radio throughput), it posts the next send function. Such a sender is referred to as a MaxTraffic source, and corresponds to a raw traffic rate of 6.46kbps.

Conclusions and Limitations

We then studied the issue of detecting the presence of jamming attacks, and examined the ability of different measurement statistics to classify the presence of a jammer. We showed that by using signal strength, carrier sensing time, or the packet delivery ratio individually, one is not able to

definitively conclude the presence of a jammer. Therefore, to improve detection, we introduced the notion of consistency checking, where the packet delivery ratio is used to classify a radio link as having poor utility, and then a consistency check is performed to classify whether poor link quality is due to jamming. We introduced two enhanced detection algorithms:

- one employing signal strength as a consistency check
- one employing location information as a consistency check.

We evaluated the effectiveness of each scheme through empirical experiments and showed that each of the four jammer models we introduced can be reliably classified using our consistency checking schemes

2.4 Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service

Problem Formulation

In this paper we present two strategies that may be employed by wireless devices to evade a MAC/PHY-layer jamming-style wireless denial of service attack. The first strategy, channel surfing, is a form of spectral evasion that involves legitimate wireless devices changing the channel that they are operating on. The second strategy, spatial retreats, is a form of spatial evasion whereby legitimate mobile devices move away from the locality of the DoS emitter. We study both of these strategies for three broad wireless communication scenarios: two-party radio communication, an infrastructure wireless network, and an ad hoc wireless network. We evaluate several of our proposed strategies and protocols through ns-2 simulations and experiments on the Berkeley mote platform

Research Design

Although there are many different scenarios where a jamming-style DoS may take place, we will focus on three basic classes of wireless networks

1.Two-Party Radio Communication: The two-party scenario is the baseline case, in which A and B communicate with each other on a specific channel. As long as interferer X is close enough to either A or B, Its transmission will interfere with the transmission and reception of packets by A and B.

2.Infrastructured Wireless Networks: Infrastructure wireless networks, such as cellular networks or wireless local area networks (WLANs), consist of two main types of devices: access points and mobile de-vices. Access points are connected to each other via a separate, wired infrastructure. Mobile devices communicate via the access point in order to communicate with each other or the Internet. The presence of an interferer, such asX0 orX1, might make it impossible for nodes to communicate with their access point.

3.Mobile Ad Hoc Wireless Networks: Ad hoc networks involve wireless devices that establish opportunistic connections with each other in order to form a communication network.

Typically, ad hoc networks employ multi-hop routing protocols in order to deliver data from one network node to another. The presence of an interferer may bring down whole regions of the network.

Findings

In this functional retreat plan must satisfy the following two conditions:

(1) it must ensure that both parties leave the adversary's interference range;

(2) it must ensure that the two parties stay within each other's radio range. In order to accomplish these two requirements, we propose a three-stage protocol:

- Establish Local coordinates
- Exit the Interference Region
- Move Into Radio Range

Conclusions and Limitations

In this paper, we have presented two different strategies that may be employed to mitigate the effects of this type of DoS attacks. The rationale behind both strategies is that legitimate wireless users should avoid the interference as much as possible because there is no way to combat the adversary. The first strategy involves changing the transmission frequency to a range where there is no interference from the adversary. The second strategy involves wireless users moving to a new location where there is no interference.

2.5 Jamming-resistant Broadcast Communication without Shared Keys

Problem Formulation

In this work, we focus on a related but different problem for broadcast communication: How to enable robust anti-jamming broadcast without shared secret keys? Typical broadcast applications share the need for authenticity and availability of messages that are transmitted by base stations (senders) to a large, unknown number of potentially untrusted (malicious or selfish) receivers. In such settings, a sender communicates to a dynamic set of trusted receivers (i.e., the nodes are honest but may be unknown to the sender due to receiver dynamics) or to untrusted receivers (which might be interested in obtaining the information themselves but depriving others of it).

Research Design

As a solution to the described problem, we propose a scheme called Uncoordinated DSSS (UDSSS) that enables authentic spread-spectrum anti-jamming broadcast without the requirement of shared secrets. UDSSS follows a similar approach as

DSSS, it differs, however, in the following aspect: the spreading code is not pre-defined but chosen by the sender randomly out of a set of publicly available codes. Since no receiver can predict the choice of the sender, UDSSS prevents dishonest receivers from interfering with the communication (to other receivers) while it enables them to obtain the information themselves. After a certain time, every receiver will succeed in identifying the correct spreading code and its synchronization, thus despreading the signal. The required dispreading time depends on the coding strategy, the size of the spreading code set, and on the receivers' processing capabilities; we analyze this in detail.

Findings

We will also show that UDSSS can achieve the same performance as DSSS in the absence of jamming.

In summary, the main contributions of this work are:

- Identify anti-jamming broadcast without shared keys as a relevant problem and we show that it can be addressed using uncoordinated spread-spectrum techniques.
- Propose a scheme called Uncoordinated DSSS that supports broadcast anti-jamming communication without shared keys and enables communication in scenarios in which DSSS cannot be used.
- Analyze the performance of UDSSS. We show that a performance comparable to DSSS can be achieved in the absence of jamming and that the expected time for a message transmission to ten receivers takes less than 30 s on state-of-the-art systems under high jamming-probabilities.

We demonstrate the feasibility of UDSSS by a prototype implementation on a software-defined radio Platform, the reception of a typical message takes well below 20 s for 21 dB processing gain on this system. We note that this time can further be significantly reduced on a purpose-built platform (e.g., like the ones used for GPS receivers).

Conclusions and Limitations

We evaluated the performance and jamming resistance of our DSSS scheme analytically, through a prototype implementation, and by means of simulations for single and multiple receivers. For a state-of-the-art system (about 6000MIPS), the expected time for a message transfer to a group of 10 receivers takes less than 30 s for a high jamming probability of 80%. We accent that this time is reasonably short, given that with common (key-dependent) anti-jamming techniques the devices would not be able to broadcast jamming-resistant messages at all

Existing System

- The MANETs are infrastructure-less networks. The nodes are no need to follow any fixed infrastructure to utilize the network services.
- The absence of the dedicated link between the wireless nodes leaves the communication medium under threat.
- Any node can access the wireless link anonymous to the wireless node at any time.
- If a link is captured by the adversary node, then the communication through the link can be overheard.
- The adversary nodes can access and manipulate the data once they acquire the link and can perform Jamming attacks.
- When the node captures the link, the node captures the data, manipulate it and send the same packet frequently to the destination node.
- If the adversary node performs jamming against any particular data, then it is "Selective Jamming".
- The data selected by the adversary for selective jamming attack might be a priority data.
- So, there is a need of proper security mechanism to prevent the data from adversary and to ensure secured communication in the network.

Issues in existing system:

- No secret transmission
- Jamming attacks occurs
- No proper selective jamming

Proposed System

- Here, we propose a mechanism "Strong Hiding Commitment Scheme" (SHCS) to mitigate selective jamming.
- This method is based on cryptography which involves keying techniques.
- In this technique, the data are hided within the chain of several cryptographic keys.
- But the key used for hiding is refreshed and a new key is used at particular intervals for the same packet.
- The adversary node cannot read the data hided inside, even though he can access the link and data.
- The actual frequently changed keys are needed for the adversary to decrypt and read the message hided.
- The process involves the following methods:

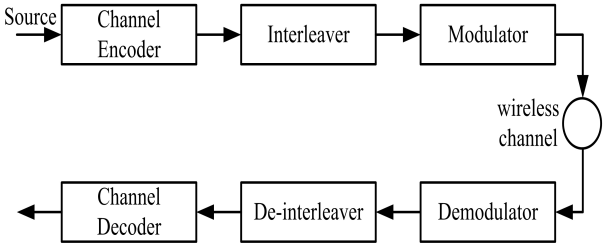
- Padding
- Permutation

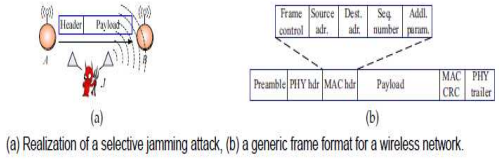
- First, the message is divided into several packets, and each packet is encrypted with random key values.
- This key value is changed frequently to keep the key values secret from the adversaries.
- The next step is padding. Here some bits are added to the encrypted data to modulate the data.
- Finally, the data is permuted and send to the destination.
- So, the adversary node can access the link, but he needs the refreshed key value to read the hidden data, which is annoying to the adversary.

Advantages

- Performance is high
- Selective jamming is efficient
- Security is highly efficient

System Architecture





(a) Realization of a selective jamming attack, (b) a generic frame format for a wireless network.

Fig 5.1 System flow diagram

SHCS:

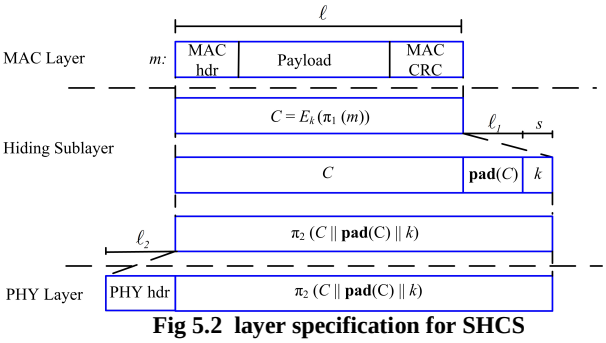


Fig 5.2 layer specification for SHCS

Algorithm

A strong hiding commitment scheme (shcs):

- PADDING

Symmetric encryption algorithm

Assume that the sender S has a packet m for R. First, S constructs $(C, d) = commit(m)$, where

$$C = E_k(\pi_1(m)), \quad d = k.$$

Here, the commitment function $E_k()$ is an off-the-shelf symmetric encryption algorithm (e.g., DES or AES [27]), π_1 is a publicly known permutation, and $k \in \{0,1\}^s$ is a randomly selected key of some desired key length s (the length of k is a security parameter). The sender broadcasts $(C \| d)$, where "$\|$" denotes the concatenation operation. Upon reception of d, any receiver R computes

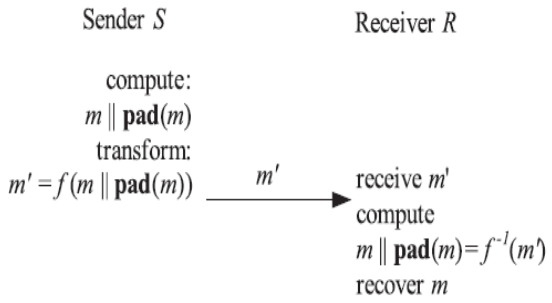
$$m = \pi_1^{-1}(D_k(C)),$$

where π_1^{-1} denotes the inverse permutation of π_1. To satisfy the strong hiding property, the packet carrying d is formatted so that *all bits of d are modulated in the last few PHY-layer symbols of the packet*. To recover d, any receiver must receive and decode the last symbols of the transmitted packet, thus preventing early disclosure of d. We now present the implementation details of SHCS.

- PERMUTATION
- CRYPTOGRAPHIC PUZZLE HIDING SCHEME

The AONT-based hiding scheme contains

Brute force attacks against block encryption algorithms



Module Specification

Modules:

NETWORK CREATION AND ROUTING:

- JAMMING IMPLEMENTATION
- RESULT ANALYSIS
- SHCS IMPLEMENTATION
- RESULT ANALYSIS

- RESULT COMPARISON ANALYSIS

Modules Description:

Module 1: Network Creation And Routing

In this module, a network is to be created. First nodes are to be created and configured. Wireless properties, storage are to be applied to the nodes. Nodes are deployed randomly across the network. A routing protocol is to be implemented across the network. A sample routing is to be performed across the network to check the connectivity.

Module 2: Jamming Implementation

In this module, a jamming is to be created across the network. First sender and receiver nodes are selected. A communication is configured between the source and destination. A jammer node is selected among the nodes manually. This node is configured to perform jamming across the network. this jammer node selects the particular data packet that flowing through it and perform jamming and send the packets with abnormal rate to the target node. The target node cannot receive the data packet with abnormal rate so start to drop the data packets.

Module 3: Result Analysis

In this module, the impact of the jamming in the network is to be analyzed. The results are plotted as X-graph. The parameters like throughput, packet drop are to taken from trace file of the ns2 output and graphs are plotted.

Module 4: Shcs Implementation

In this module, SHCS module is to be implemented. First the cryptographic keys are to be generated using any cryptographic algorithm like RSA. Then the data is divided into packets and these packets are encrypted using the newly created key. Then some bits are added with the encrypted data as padding process to hide the identity of the data. Now the data is permutated and transferred to the destination node. The cryptographic key is refreshed periodically to hide the key from the jammer node.

Module 5: Result Analysis

In this module, the impact of the jamming in the network is to be analyzed. The results are plotted as X-graph. The parameters like throughput, packet drop are to taken from trace file of the ns2 output and graphs are plotted.

Module 6: Result Comparison Analysis

In this module, the graphs plotted in module 3 and 5 are to be configured. Based on the analysis result decision is to be taken and the future work is determined.

Software Specification

Network Simulation

In communication and computer network research, network simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities(host/routers, data links, packets, etc) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab,

Various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions. When a simulation program is used in conjunction with live applications and services in order to observe end-to-end performance to the user desktop, this technique is also referred to as network emulation.

Network Simulator

A network simulator is a piece of software or hardware that predicts the behavior of a network, without an actual network being present. The network simulator is the program in charge of calculating how the network would behave. Such software may be distributed in source form (software) or packaged in the form of a dedicated hardware appliance. Users can then customize the simulator to fulfill their specific analysis needs. Simulators typically come with support for the most popular protocols in use today, such as IPv4, IPv6, UDP, and TCP.

Uses of Network Simulators

Network Simulators serve a variety of needs. Compared to the cost and time involved in setting up an entire test bed containing multiple networked computers, routers and data links, network simulators are relatively fats and inexpensive. They allow engineers to test scenarios that might be particularly difficult or expensive to emulate using real hardware – for instance, simulating the effects of a sudden burst in traffic or a DoS attack on a network service. Networking simulators are particularly useful in allowing designers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment.

Network simulators, as the name suggests are used by researchers, developers and QA to design various kinds of networks, simulate and then analyze the effect of various kinds of networks, simulate and then analyze the effect of various parameters on the network performance. A typical network simulator encompasses a wide range of networking technologies and help the users to build complex networks from basic building blocks like variety of nodes and links With the help of simulators one can design hierarchical networks using various types of

nodes like computers, hubs, bridges, routers, optical cross-connects, multicast router, mobile units, MSAUs etc.

NS (SIMULATOR)

Ns or the Network simulator(also popularly called ns-2) is a discrete event network simulator. It is popular in academia for its extensibility(due to its open source model) and plentiful online documentation. Ns is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc networking research. Ns supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. It can be also used as limited –functionality network emulator. Ns is licensed for use under version 2 of the GNU General Public License.

About TCL

Tool Command Language (TCL) is an interpreted script language developed by Dr.John Ousterhout at the University of California, Berkely, and now developed and maintained by Scriptics. Tcl is comparable to : Netscape JavaScript Microsoft's Visual Basic .The UNIX-derived Practical Extraction and Reporting Language IBM's Restructured Extended Executor In general, script languages are easier and faster to code in than the more structured, complied languages such as C and C++. Script languages are sometimes considered good "glue" languages for tying several compiled programs together. Or, as stand-alone programs, they can allow you to create simple but powerful effects on their own. TclBlend is a version of Tcl that can access certain Java languages facilities. Tcl has a companion program, Tool kit (Tk), to help create a Graphical User Interface with Tcl.

About OTCL

OTCL is an object oriented extension of Tcl and created by David Wetherall. It is used in network simulator(NS-2) and usually run under Unix environment.

About GEDIT

GEDIT is a UTF-8 compatible text editor for the GNOME computer desktop environment. Designed as a general purpose text editor, gedit emphasizes simplicity and ease of use. It includes tools for editing source code and structured text such as markup languages. It is designed to have a clean, simple graphical user interface according to the philosophy of the GNOME project ,and it is the default text editor for GNOME.

Gedit includes syntax highlighting for various program code and text markup formats. Gedit also has GUI tabs for editing multiple files. Tabs can be moved between various windows by the user. It can edit remote files using GVFS(Gnome VFS is now deprecated) libraries. It supports a full undo and redo system aS well as search and replace. Other typical code orinted features include line numbering, bracket matching, text wrapping, current line highlighting, automatic indentation and automatic file backup. Some advanced features of gedit include Multilanguage spellchecking and a flexible plugin system allowing to dynamically add new features.

Conclusion

We evaluated the impact of selective jamming attacks on network protocols such as TCP and routing. Our findings show that a selective jammer can significantly impact performance with very low effort. We developed three schemes that transform a selective jammer to a random one by preventing real-time packet classification. Our schemes combine cryptographic primitives such as commitment schemes, cryptographic puzzles, and all-or-nothing transformations with physical-layer characteristics. We analyzed the security of our schemes and quantified their computational and communication overhead

References

[1] T.X. Brown, J.E. James, and A. Sethi, "Jamming and Sensing of Encrypted Wireless Ad Hoc Networks,"Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc),pp. 120-130,2006.

[2] M. Cagalj, S. Capkun, and J.-P. Hubaux, "Wormhole-Based Anti-Jamming Techniques in Sensor Networks," IEEE Trans. Mobile Computing,vol. 6, no. 1, pp. 100-114, Jan. 2007.

[3] A. Chan, X. Liu, G. Noubir, and B. Thapa, "Control Channel Jamming: Resilience and Identification of Traitors," Proc. IEEE Int'l Symp. Information Theory (ISIT),2007. PROA NO AND LAZOS: PACKET-HIDING METHODS FOR PREVENTING SELECTIVE JAMMING ATTACKS 113

[4] T. Dempsey, G. Sahin, Y. Morton, and C. Hopper, "Intelligent Sensing and Classification in Ad Hoc Networks: A Case Study,"IEEE Aerospace and Electronic Systems Magazine, vol. 24, no. 8, pp. 23-30, Aug. 2009.

[5] Y. Desmedt, "Broadcast Anti-Jamming Systems,"Computer Net-works,vol. 35, nos. 2/3, pp. 223-236, Feb. 2001.

[6] O. Goldreich,Foundations of Cryptography: Basic Applications. Cambridge Univ. Press, 2004.

[7] B. Greenstein, D. Mccoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall, “Improving Wireless Privacy with an Identifier-Free Link Layer Protocol,”Proc. Int’l Conf. Mobile Systems, Applications,and Services (MobiSys),2008.
[8] IEEE, IEEE 802.11 Standards, http://standards.ieee.org/getieee802/download/802.11-2007.pdf, 2007.