Avoiding Selective Attacks with using Packet Hiding Approaches in Wireless Network

Patel Dhaval Dhirubhai[1], Singh Kashkumar Nirmalsingh[2]

[1]Computer Network and Engineering, EastWest Institute of Technology,
Visvesvaraya Technological University, Bangalore 560091, India
dhavalpokiya@yahoo.com

[2] Computer Science and Engineering, Babaria Institute of Technology,
Gujarat University Vadodara 391240, India
kashsingh99@yahoo.in

Abstract: ***We address the problem of selective attacks in wireless networks. In these attacks, the opponent selectively targets specific packets of "high" importance by exploiting his knowledge on the implementation details of network protocols at various layers of the protocol stack. We illustrate illustrating various selective attacks against the TCP protocol. We show that selective attacks can be launched by performing real-time packet classification at the physical layer. To mitigate these attacks, we develop three schemes that Avoid real-time packet classification by combining cryptographic primitives with physical-layer attributes. We analyze the security of our methods and evaluate their computational and communication overhead.***

Keywords: Wireless Network, Packet Classification, Selective Attacks, Time-Lock Puzzle, Cryptography.

1. Introduction

The open nature of the wireless medium leaves it vulnerable to intentional interference attacks, typically referred to as jamming. Wireless networks rely on the uninterrupted availability of the 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.

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.

Our findings indicate that selective attacks lead to DoS with very low effort on behalf of the jammer. To mitigate such attacks, we develop three schemes that prevent classification of transmitted packets in real time. Our schemes rely on the joint consideration of cryptographic mechanisms with PHY-layer attributes. We analyze the security of our schemes and show that they achieve strong security properties, with minimal impact on the network performance.

The remainder of the paper is organized as follows. In Section 2, we describe the problem addressed, and state the system model. In Section 3, we show the feasibility of system working of communication model. Section 4 illustrates the impact of selective jamming. In Sections 4 and 5, we develop methods for preventing selective jamming. Section 6, Result of work. After this, we will conclude.

2. Problem Statement and System Model

2.1 Problem Statement

Consider the scenario depicted in Figure. 1. Nodes A and B communicate via wireless link. Within the communication range of both A and B there is a jamming/attacker node K. When A transmits a packet m to B, node K classifies m by receiving only the first few bytes of m. K then corrupts m beyond recovery by interfering with its reception at B. We address the problem of preventing the attacking node from classifying m in real time, thus mitigating K's ability to perform selective jamming attacks. Our goal is to transform a selective Attacker to a random one. Note that in the present work, we do not address packet classification methods based on protocol semantics. So, all the models and communication are created in wireless network and transmission are made in wireless.

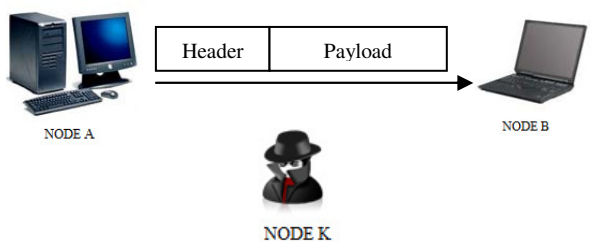

Figure 1: Realization of a selective jamming attack

2.2 System Model

Communication Model-Packets are transmitted at a rate of r bauds. Each PHY-layer symbol corresponds to n bits, where the value of n is defined by the underlying digital modulation scheme. Every symbol carries (α/β) n data bits, where α/β is the rate of the PHY-layer encoder. Here, the transmission bit rate is equal to nr bps and the information bit rate is (α/β) nR bps. Spread spectrum techniques such as frequency hopping spread spectrum (FHSS), or direct sequence spread spectrum (DSSS) may be used at the PHY layer to protect wireless transmissions from jamming. SS provides immunity to interference to some extent (typically 20 to 30 dB gain), but a powerful jammer is still capable of jamming data packets of his choosing.

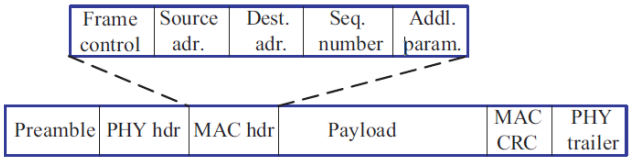

Figure 2: Generic frame format for a wireless network

Transmitted packets have the generic format depicted in Figure. 2. The preamble is used for synchronizing the sampling process at the receiver. The PHY layer header contains information regarding the length of the frame, and the transmission rate. The MAC header determines the MAC protocol version, the source and destination addresses, sequence numbers plus some additional fields. The MAC header is followed by the frame body that typically contains an ARP packet or an IP datagram. Finally, the MAC frame is protected by a cyclic redundancy check (CRC) code. At the PHY layer, a trailer may be appended for synchronizing the sender and receiver.

Network Model-The network consists of a collection of nodes connected via wireless links. Nodes may communicate directly if they are within communication range, or indirectly via multiple hops. Nodes communicate both in unicast mode and broadcast mode. Communications can be either unencrypted or encrypted. For encrypted broadcast communications, symmetric keys are shared among all intended receivers. These keys are established using preshared pairwise keys or asymmetric cryptography.

Opponent Model-The adversary can operate in full-duplex mode, thus being able to receive and transmit simultaneously. We assume the adversary is in control of the communication medium and can jam messages at any part of the network of his choosing. In detail, the adversary is equipped with directional antennas that enable the reception of a signal from one node and jamming of the same signal at another. A jammer equipped with a single half-duplex transceiver is sufficient to classify and jam transmitted packets. However, our model captures a more potent adversary that can be effective even at high transmission speeds.

Hardware for performing cryptanalysis or any other required computation. Solving well-known hard cryptographic problems is assumed to be time-consuming. For the purposes of analysis, given a cipher text, the most efficient method for deriving the corresponding plaintext is assumed to be an exhaustive search on the key space.

3. Working of Communication System

In this section, we describe how the adversary can classify packets in real time, before the packet transmission is completed. Once a packet is classified, the adversary may choose to jam it depending on his strategy. Consider the generic communication system depicted in Figure.3.At the PHY layer, a packet m is encoded, interleaved, and modulated before it is transmitted over the wireless channel. At the receiver, the signal is demodulated, DE interleaved, and decoded, to recover the original packetm.

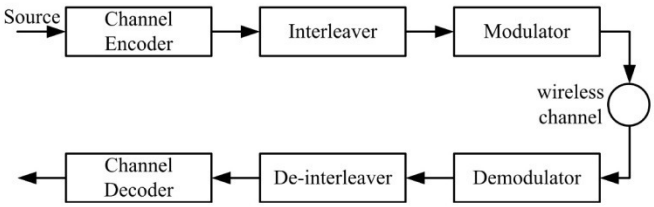

Figure 3: Example of an image with acceptable resolution

The opponent's ability in classifying a packet m depends on the implementation of the blocks in Figure. 3. The channel encoding block expands the original bit sequence m, adding necessary redundancy for protecting m against channel errors. For example, a α/β-block code may protect m from up to e errors per block. Alternatively, a α/β-rate convolutional encoder with a constraint length of L_{max}, and a free distance of e bits provides similar protection. For our purposes, we assume that the rate of the encoder is α/β. At the next block, interleaving is applied to protect m from burst errors. For simplicity, we consider a block interleaver that is defined by a matrix $A_{d\times\beta}$. The de-interleaver is simply the transpose of A. Finally, the digital modulator maps the received bit stream to symbols of length n, and modulates them into suitable waveforms for transmission over the wireless channel. Typical modulation techniques include OFDM, BPSK, 16(64)-QAM, and CCK.

In order to recover any bit of m, the receiver must collect d.β bits for de-interleaving. The d.β de-interleaved bits are then passed through the decoder. Ignoring any propagation and decoding delays, the delay until decoding the first block of data is ⌈dβ/n⌉ symbol durations. As an example, in the 802.11a standard, operating at the lowest rate of 6Mbps, data is passed via a 1/2-rate encoder before it is mapped to an OFDM symbol of q=48 bits. In this case, decoding of one symbol provides 24 bits of data. At the highest data rate of 54Mbps, 216 bits of data are recovered per symbol.

An intuitive solution to selective jamming would be the encryption of transmitted packets (including headers) with a

static key. However, for broadcast communications, this static decryption key must be known to all intended receivers and hence, is susceptible to compromise. An adversary in possession of the decryption key can start decrypting as early as the reception of the first cipher text block. For example, consider the cipher-block chaining (CBC) mode of encryption. To encrypt a message m with a key k and an initialization vector IV, message m is split into x blocks m_1, m2. . . mx, and each cipher text block c_i, is generated as:

$$c_1=IV,\ ci_{+1}=E_k\ (c_i \oplus m_i),\ i = 1, 2. . . x. \quad (1)$$

Where E_k (m) denotes the encryption of m with key k. The plaintext m_i is recovered by:

$$m_i = c_i \oplus D_k\ (ci+1),\ i=1, 2. . . x. \quad (2)$$

Note from (2) that reception of c_{i+1} is sufficient to recover m_i if k is known (c_1=IV is also known). Therefore, realtime packet classification is still possible.

4. Hiding Approaches with Commitments

In this section, we show that the problem of real-time packet classification can be mapped to the hiding property of commitment schemes, and propose a packet-hiding scheme based on commitments.

4.1 Commitment Schemes

Commitment schemes are cryptographic primitives that allow an entity A, to commit to a value m, to an entity V while keeping m hidden. Commitment schemes are formally defined as follows.

Commitment Scheme: A commitment scheme is a two-phase interactive protocol defined as a triple {X, M, E}. Set X={A,V} denotes two probabilistic polynomial-time interactive parties, where A is known as the committer and V as the verifier; set M denotes the message space, and set E = {(t_i, f_i)} denotes the events occurring at protocol stages t_i(i=1, 2), as per functions f_i (i=1,2). During commitment stage t_1, A uses a commitment function f_1=commit () to generate a pair (C, d) = commit (m), where (C, d) is called the commitment/DE commitment pair. At the end of stage t_1, A releases the commitment C to V. In the open stage t_2, A releases the opening value d. Upon reception of d, V opens the commitment C, by applying function f_2=open (), thus obtaining a value of m′=open(C, d). This stage culminates in either acceptance (m′=m) or rejection (m′≠m) of the commitment by V. Commitment schemes satisfies the following two fundamental properties:

Hiding: For every polynomial-time party V interacting with A, there is no (probabilistic) polynomials-efficient algorithm that would allow V to associate C with m and C′ with m′, without access to the DE commitment values d or d′ respectively, and with non-negligible probability.

Binding: For every polynomial-time party A interacting with V, there is no (probabilistic) polynomial efficient algorithm that would allow A to generate a triple (C, d, d′), such that V accepts the commitments (C, d) and (C, d′), with non-negligible probability.

4.2 Dual Hiding Scheme on Commitment

We propose a dual hiding scheme on commitment (DHSC), which is based on symmetric cryptography. Our main motivation is to satisfy the strong hiding property while keeping the computation and communication overhead to a minimum. Assume that the sender S has a packet m for R. First, S constructs (C, d) = commit (m), where,

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

Here, the commitment function E_k () is an off-the-shelf symmetric encryption algorithm (e.g., DES or AES), 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 $\pi 1^{-1}$ 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 DHSC.

4.3 Implementation Detail of DHSC

The proposed SHCS requires the joint consideration of the MAC and PHY layers. To reduce the overhead of SHCS, the DE commitment value d (i.e., the decryption key k) is carried in the same packet as the committed value C. This saves the extra packet header needed for transmitting d individually. To achieve the strong hiding property, a sub layer called the "hiding sub layer" is inserted between the MAC and the PHY layer. This sub layer is responsible for formatting m before it is processed by the PHY layer. The functions of the hiding sub layer are outlined in Figure. 4.

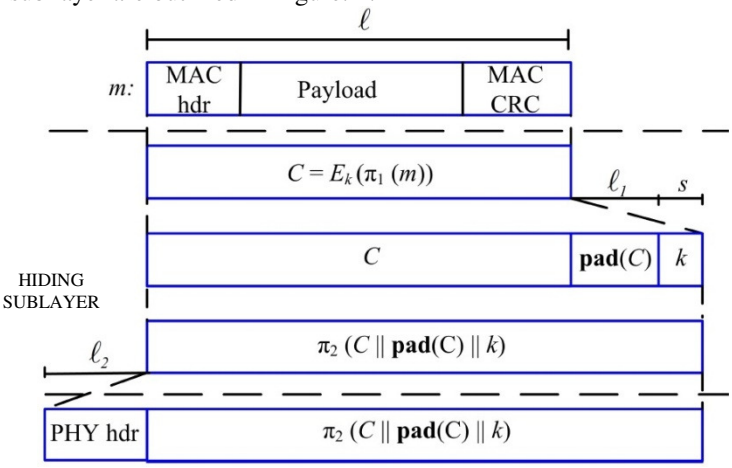

Figure 4: Hiding sub layer processing

Consider a frame m at the MAC layer delivered to the hiding sub layer. Frame m consists of a MAC header and the payload, followed by the trailer containing the CRC code. Initially, m is permuted by applying a publicly known permutation π_1. The purpose of π_1 is to randomize the input to the encryption algorithm and delay the reception of critical packet identifiers such as headers. After the permutation, π_1 (m) is encrypted using a *random* key k to produce the commitment value C=Ek (π_1 (m)). Although the random permutation of m and its encryption with a random key k seemingly achieve the same goal (i.e., the randomization of the cipher text).

In the next step, a padding function **pad** () appends **pad**(C) bits to C, making it a multiple of the symbol size. Finally, C||**pad**(C) ||k is permuted by applying a publicly known permutation π_2. The purpose of π_2 is to ensure that the interleaving function applied at the PHY layer does not disperse the bits of k to other symbols. We now present the padding and permutation functions in detail.

Padding–The purpose of padding is to ensure that k is modulated in the minimum number of symbols needed for its transmission. This is necessary for minimizing the time for which parts of k become exposed. Let ℓ_1 denote the number of bits padded to C. For simplicity, assume that the length of C is a multiple of the block length of the symmetric encryption algorithm and hence, has the same length ℓ as the original message m. Let also ℓ_2 denote the length of the header added at the PHY layer the frame carrying (C, d) before the encoder has a length of (ℓ + ℓ1 + ℓ2+s) bits. Assuming that the rate of the encoder is α/β the output of the encoder will be of length, α/β (ℓ + ℓ1 + ℓ2 + s). For the last symbol of transmission to include (α/β) q bits of the key k, it must hold that,

$$\ell 1 = \alpha/\beta\ (q-((\ell + \ell 2)\ \alpha/\beta) \bmod q). \qquad (3)$$

Permutation–The hiding layer applies two publicly known permutations π_1 and π_2 at different processing stages. Permutation π_1 is applied to m before it is encrypted. The purpose of π_1 is twofold. First, it distributes critical frame fields which can be used for packet classification across multiple plaintext blocks. Hence, to reconstruct these fields, all corresponding cipher text blocks must be received and decrypted. Moreover, header information is pushed at the end of π_1 (m). This prevents early reception of the corresponding cipher text blocks.

For example, consider the transmission of a MAC frame of length 2,336 bytes which carries a TCP data packet. The MAC header is 28 bytes long and has a total of 18 distinct fields. TCP header is 20 bytes long (assuming no optional fields) and has 17 distinct fields. Assume the encryption of a fixed block of 128 bits. Packet π_1(m) is partitioned to 146 plaintext blocks {p_1, p_2, . . . , p_{146}}, and is encrypted to produce 146 cipher text blocks C = $c_1||c_2||$ . . . $||c_{146}$. Each field of the TCP and MAC headers is distributed bit-by-bit from the most significant bit (MSB) to the least significant bit (LSB) to each of the plaintext blocks in the reverse block order. This process is depicted in Figure. 5.

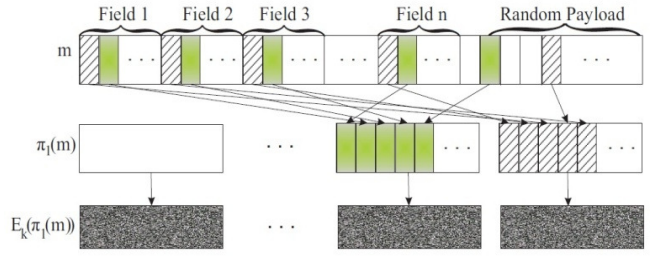

Figure 5: Application on permutation π_1 on packet m

For fields longer than one bit, bits are numbered from the LSB to the MSB and are placed in reverse order to each plaintext block. To recover any field $_i$ that is ℓ_i bits long, the last ℓ_i cipher text blocks must be received and decrypted. If $\ell_i > \ell_b$, where ℓ_b denotes the cipher text block length, the bit placement process continues in a round robin fashion. The second goal of the permutation π1 is to randomize the plaintext blocks. Assuming a random payload, the permutation distributes the payload bits to all plaintext blocks processed by the encryption function, thus randomizing each cipher text block.

Permutation π_2 is applied to reverse the effects of interleaving on the bits of k, so that k is contained at the packet trailer. Interleaving can be applied across multiple frequencies on the same symbol (e.g., in OFDM), or it may span multiple symbol. For example, consider a d×β block interleaver. Without loss of generality, assume that β = q, and let the last n rows of the last block passed via the interleaver correspond to the encoded version of the random key k. Permutation π_2 rearranges the bits of k at the interleaver matrix $A_{d\times\beta}$ in such a way that all bits of k appear in the last n columns. Therefore, the bits of k will be modulated as the last n symbols of the transmitted packet. Note that this operation affects only the interleaver block(s) that carries k. For the rest of the packet, the interleaving function is performed normally, thus preserving the benefits of interleaving. For PHY layer implementations in which interleaving is applied on a per symbol basis (e.g., 802.11a and 802.11g), the application of permutation π_2 is not necessary.

5. Hiding Approaches on Cryptography Puzzle

In this section, we present a packet hiding scheme based on cryptographic puzzles. The main idea behind such puzzles is to force the recipient of a puzzle execute a pre-defined set of computations before he is able to extract a secret of interest. The time required for obtaining the solution of a puzzle depends on its hardness and the computational ability of the solver. The advantage of the puzzle based scheme is that its security does not rely on the PHY layer parameters. However, it has higher computation and communication overhead.

In our context, we use cryptographic puzzles to temporary hide transmitted packets. A packet m is encrypted with a randomly selected symmetric key k of a desirable length s. The key k is blinded using a cryptographic puzzle and sent to the receiver. For a computationally bounded adversary, the puzzle carrying k cannot be solved before the transmission of the encrypted version of m is completed and the puzzle is received. Hence, the adversary cannot classify m for the purpose of selective jamming.so, We will address the puzzle that how we will solute it. And this hiding scheme is totally based on puzzle technique with cryptographic.

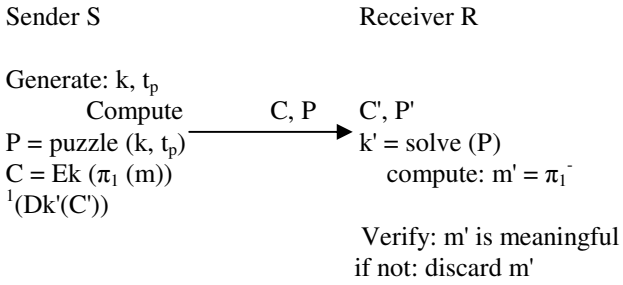

Figure 6: The cryptographic puzzle based hiding technique

5.1 Playing Cryptography Puzzle Scheme (PCPS)

Let a sender S have a packet m for transmission. The sender selects a random key $k \in \{0, 1\}^s$, of a desired length. S generates a puzzle P = puzzle (k, tp), where puzzle () denotes the puzzle generator function, and tp denotes the time required for the solution of the puzzle. Parameter tp is measured in units of time, and it is directly dependent on the assumed computational capability of the adversary, denoted by N and measured in computational operations per second. After generating the puzzle P, the sender broadcasts (C, P), where $C=E_k (\pi_1 (m))$. At the receiver side, any receiver R solves the received puzzle P′ to recover key k′ and then computes $m'=\pi^{-1}(D_{k'} (C'))$. If the decrypted packet m′ is meaningful (i.e., is in the proper format, has a valid CRC code, and is within the context of the receiver's communication), the receiver accepts that m′=m. Else, the receiver discards m′. Figure-6 shows the details of PCPS.

5.2 Implementation Detail of PCPS

In this section, we consider several puzzle schemes as the basis for PCPS. For each scheme, we analyze the implementation details which impact security and performance. Cryptographic puzzles are primitives originally suggested by Merkle as a method for establishing a secret over an insecure channel. They find a wide range of applications from preventing DoS attacks to providing broadcast authentication and key escrow schemes.

Time-lock Puzzles–Which is based on the iterative application of a precisely controlled number of modulo operations. Time-lock puzzles have several attractive features such as the fine granularity in controlling t_p and the sequential nature of the computation. Moreover, the puzzle generation requires significantly less computation compared puzzling solving.

In a time-lock puzzle, the puzzle constructor generates a composite modulus g=u*v, where u and v are two large random prime numbers. Then, he picks a random a, $1 < a < g$ and hides the encryption key in $K_h = k + a^{2t} \bmod g$, where $t=t_{p*}N$, is the amount of time required to solve for k. Here, it is assumed that the solver can perform N squaring modulo g per second. Note that K_h can be computed efficiently if $\varphi(g) = (u-1)(v-1)$ or the factorization of g are known, otherwise a solver would have to perform all t squaring to recover k. The puzzle consists of the values P = (g, Kh, t, a).

In our setup, the value of the modulus g is known a priori and need not be communicated (may change periodically). The sender reveals the rest of the puzzle information in the order (K_h, t, a). Note that if any of t, an unknown, any value of k is possible.

Puzzles based on hashing–Computationally limited receivers can incur significant delay and energy consumption When dealing with modulo arithmetic. In this case, PCPS can be implemented from cryptographic puzzles which employ computationally efficient cryptographic primitives. Client puzzles proposed in, use one-way hash functions with partially disclosed inputs to force puzzle solvers search through a space of a precisely controlled size. In our context, the sender picks a random key k with $k = k_1 \| k_2$. The lengths of k_1 and k_2 are s_1, and s_2, respectively. He then computes $C=E_k (\pi_1 (m))$ and transmits (C, k_1, h (k)) in this particular order. To obtain k, any receiver has to perform on average 2^{s2-1} hash operations (assuming perfect hash functions). Because the puzzle cannot be solved before h(k) has been received, the adversary cannot classify m before the completion of m's transmission.

6. Result of Work

In first Approach i.e. dual or strong hiding method which is apply for more security on packets m than it is very difficult to getting data by attackers so we have to apply permutation on packets in this our project are applying 2 times permutation for hiding packets. And at destination, receiver receives packets with using invert procedure.

In second Approach, playing cryptography puzzle, in these method sender send packets with puzzle system so that is puzzling system so at destination has to solve the puzzle that how to solve the puzzle and getting key with solution of puzzle. So that is also difficult to solve this puzzle on attackers.

Conclusion

We addressed the problem of selective attacks in wireless networks. We showed that the jammer can classify transmitted packets in real time by decoding the first few symbols of an ongoing transmission. 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. We analyzed the security of our schemes and quantified their computational and communication overhead.

Future Scope

The shuffling process will be done 2 or more time, so security of data and encryption process will be very strong. For transmitting data padding must be use for size compression that will be used for increase the transfer rate. In Last puzzling process, more puzzles will be invented and it can be saluted in easy ways.

References

[1] O. Goldreich. Foundations of cryptography: Basic applications. Cambridge University Press, 2004.

[2] M. Simon, J. Omura, R. Scholtz, and B. Levitt. Spread spectrum communications handbook. McGraw-Hill Companies, 1994.

[3] T. X. Brown, J. E. James, and A. Sethi. Jamming and sensing of encrypted wireless ad hoc networks. In Proceedings of MobiHoc, pages 120–130, 2006.

[4] Y. Desmedt. Broadcast anti-jamming systems. Computer Networks, 35(2-3):223–236, February 2001.

[5] W. Xu, W. Trappe, Y. Zhang, and T.Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In Proceedings of MobiHoc, pages 46–57, 2005.

[6] SciEngines. Break DES in less than a single day. http://www. sciengines.com, 2010.

[7] IEEE. IEEE 802.11 standard.2007 http:// standards.ieee.org/getieee802/download/802.11-

2007.pdf.W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In the 6th ACM international symposium on Mobile ad hoc networking and computing, pages 46–57, 2005.

[8] G. Lin and G. Noubir. On link layer denial of service in data wireless LANs. Wireless Communications and Mobile Computing, 5(3):273–284, May 2004.

[9] C. P¨opper, M. Strasser, and S. ˇCapkun. Jamming-resistant broadcast communication without shared keys. In Proceedings of the USENIX Security Symposium, 2009.

[10] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders. Reactive jamming in wireless networks: How realistic is the threat? In Proceedings of WiSec, 2011.

Author Profile

Patel Dhaval Dhirubhai received the M.Tech. Degree in Computer Network & Engineering from Visvesvaraya Technological University Bangalore in 2013 and B.E degree in Computer Science & Engineering from Babaria institute of technology, Gujarat University, Vadodara in 2011

Singh Kashkumar Nirmalsingh received the B.E degree in Computer Science & Engineering from Babaria institute of technology, Gujarat University, Vadodara in 2011; He is working in Industrial Training Institute in Bardoli, Dist. Surat