IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 1 (Sep. - Oct. 2013), PP 42-45
www.iosrjournals.org

Prevention of Selective Jamming Attacks by Using Packet Hiding Methods

V.Jyothi[1], U. Vidya Sagar[2], S. Ramesh Kumar [3]
[1.] *PG Student Dept of CSE St. Johns college of engineering Yemmiganur A.P. India.*
[2.] *Assistant Professor Dept of CSE St. Johns college of engineering Yemminganur A.P. India.*
[3.] *Assistant Professor Head of the department, Dept of CSE, Chiranjeevi Reddy Institute of Engineering and Technology Anantapur A.P. India.*

Abstract: *The open nature of the wireless medium leaves it too weak to intentional interference attacks, typically defined as jamming. This intentional interference with wireless transmissions can be used as a launch pad for mounting Denial-of-Service attacks on wireless networks. Typically, jamming has been introduced under an external threat model. However, intruders with internal knowledge of protocol specifications and network secrets can launch low-effort jamming attacks that are difficult to detect and counter. In this work, we address the problem of selective jamming attacks in wireless networks. In these attacks, the hacker is active only for a short period of time, selectively targeting messages of high importance. We demonstrate the advantages of selective jamming in terms of network performance degradation and hacker effort by presenting two case studies; a selective attack on TCP and one on routing. We show that selective jamming attacks can be forwarded by performing real-time packet classification at the physical layer. To reduce these attacks, we develop three schemes that prevent real-time packet classification by combining cryptographic primitives with physical-layer attributes. We analyze the security of the proposed methods and evaluate their computational and communication overhead.*

I. Introduction

Wireless networks rely on the uninterrupted availability of the wireless medium to interconnect participating nodes. 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. 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. 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 anti-jamming techniques rely extensively on spread-spectrum (SS) communications , or some form of jamming evasion. SS techniques provide bit-level protection by spreading bits according to a secret pseudo-noise (PN) code, known only to the communicating parties.

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.

Investigations–We investigate the feasibility of real time packet classification for launching selective jamming attacks, under an internal threat model. We show that such attacks are relatively easy to actualize by exploiting knowledge of network protocols and cryptographic primitives extracted from compromised nodes. We investigate the impact of selective jamming on critical network functions. Our findings indicate that selective jamming attacks lead to a 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.

II. Problem Statement and Assumptions

We address the problem of preventing the jamming node from classifying in real time, thus mitigating ability to perform selective jamming. Our goal is to transform a selective jammer to a random one. Note that in the present work, we do not address packet classification methods based on protocol semantic**s.**

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 pre shared pair wise keys or asymmetric cryptography.

Attacker Model–We assume the adversary is in control of the communication medium and can jam messages at any part of the network of his choosing. The adversary can operate in full-duplex mode, thus being able to receive and transmit simultaneously. This can be achieved, for example, with the use of multi-radio transceivers. In addition, 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. For analysis purposes, we assume that the adversary can pro-actively jam a number of bits just below the ECC capability early in the transmission. He can then decide to irrecoverably corrupt a transmitted packet by jamming the *last symbol*. In reality, it has been demonstrated that selective jamming can be achieved with far less resources. 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.

III. Effect of Selective Jamming

In this section, we illustrate the impact of selective jamming attacks on the network performance. We used OPNET^{TM} Modeler 14.5 to implement selective jamming attacks in two multi-hop wireless network scenarios. In the first scenario, the attacker targeted a TCP connection established over a multi-hop wireless route. In the second scenario, the jammer targeted network-layer control messages transmitted during the route establishment process.

Selective Jamming at the Transport Layer–In the first set of experiments, we setup a file transfer of a 3 MB file between two users A and B connected via a multi-hop route. The TCP protocol was used to reliably transport the requested file. At the MAC layer, the RTS/CTS mechanism was enabled. The transmission rate was set to 11 Mbps at each link. The jammer was placed within the proximity of one of the intermediate hops of the TCP connection.

Four jamming strategies were considered: (a) selective jamming of cumulative TCP-ACKs.
(b) Selective jamming of RTS/CTS messages.
(c) Selective jamming of data packets,
(d) Random jamming of any packet. In each of the strategies, a fraction p of the targeted packets is jammed.

Selective Jamming at the Network Layer–In this scenario, we simulated a multi-hop wireless network of 35 nodes, randomly placed within a square area. The AODV routing protocol was used to discover and establish routing paths [19]. Connection requests were initiated between random source/destination pairs. Three jammers were strategically placed to selectively jam non-overlapping areas of the network. Three types of jamming strategies were considered: (a) a continuous jammer, (b) a random jammer blocking only a fraction p of the transmitted packets, and (c) a selective jammer targeting route request (RREQ) packets.

IV. Hiding Techniques

4.1 Hiding Based on 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.

***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 = {(ti, fi)} denotes the events occurring at protocol stages ti (i = 1, 2), as per functions fi (i = 1, 2). During commitment stage t1, A uses a

commitment function f1 = commit() to generate a pair (C, d) = commit(m), where (C, d) is called the commitment/decommitment pair. At the end of stage t1, A releases the commitment C to V . In the open stage t2, A releases the opening value d. Upon reception of d, V opens the commitment C, by applying function f2 = open(), thus obtaining a value of m′ = open(C, d). This stage culminates in either acceptance (m′ = m) or rejection (m′ 6= m) of the commitment by V . Commitment schemes satisfy the following two fundamental properties:

- ***Hiding:*** For every polynomial-time party V interacting with A, there is no (probabilistic) polynomially-efficient algorithm that would allow V to associate C with m and C′ with m′, without access to the decommitment values d or d′ respectively, and with non-negligible probability.
- ***Binding***: For every polynomial-time party A interacting with V , there is no (probabilistic) polynomially 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 Hiding Based on Cryptographic Puzzles

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.

Let a sender S have a packet m for transmission. The sender selects a random key k ∈ {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 = Ek(π1(m)). At the receiver side, any receiver R solves the received puzzle P′ to recover key k′ and then computes m′ = π−1(Dk′ (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′.

The per-packet communication overhead of CPHS is equal to the length of P, in addition to the padding added by the encryption function. If the puzzle is realized using time-locks, the length of P is equal to the lengths of Kh, a, and t. The value Kh is computed modulo g and has the same length as g. Similarly, a has a length equal to the length of g. The size of t is potentially smaller than a, g, and Kh, and depends on the computational capability of the adversary. The security of time locks depends on the difficulty in factoring g or finding ϕ (g), where ϕ () denotes the Euler ϕ −function. Typical values of g are in the order of 1,024 bits. Since messages need to remain hidden for only a short period of time, the modulo can be chosen to be of much smaller size and be periodically refreshed. In the case of hash-based puzzles, the communication overhead is equal to the transmission of the key k1 which is of length s1 and the hash value h(k). The typical length of hash function is 160 bits.

4.3 Hiding Based on All-Or-Nothing Transformations

In this section, we propose a solution based on All-Or-Nothing Transformations (AONT) that introduces a modest communication and computation overhead. Such transformations were originally proposed by Rivest to slow down brute force attacks against block encryption algorithms. An AONT serves as a publicly known and completely invertible pre-processing step to a plaintext before it is passed to an ordinary block encryption algorithm. A transformation f, mapping message m = {m1, ,mx} to a sequence of pseudo-messages m′ = {m′ 1, · · · ,m′ x′}, is an AONT if: (a) f is a bisection, (b) it is computationally infeasible to obtain any part of the original plaintext, if one of the pseudo-messages is unknown, and (c) f and its inverse f−1 are efficiently computable.

When a plaintext is pre-processed by an AONT before encryption, all cipher text blocks must be received to obtain any part of the plaintext. Therefore, brute force attacks are slowed down by a factor equal to the number of cipher text blocks, without any change on the size of the secret key. Note that the original AONT proposed in is computationally secure. Several AONT schemes have been proposed that extend the definition of AONT to undeniable security. Under this model, all plaintexts are equiprobable in the absence of at least one pseudo-message.

The linear AONT requires only elementary arithmetic operations such as string addition and multiplication, making it particularly fast due to its linear nature. The package transform requires x′ symmetric encryptions at the sender and an equal amount of symmetric decryptions at the receiver. Note that the length of the plaintext for the x′ encryptions is relatively small compared to the length of message m (indexes 1 . . . x are

encrypted). Therefore, only one cipher text block is produced per pseudo-message. Assuming a pseudo-message block size equal to the cipher text block size ℓb, the computational overhead of the x′ encryptions required by the package transform is equivalent to the overhead of one encryption of a message of length ℓ + ℓb.

V. Evaluation of Packet-Hiding Techniques

In this section, we evaluate the impact of our packet-hiding techniques on the network performance via extensive simulations. We used the OPNET^{TM} Modeler 14.5 to implement the hiding sub layer and measure its impact on the effective throughput of end-to-end connections and on the route discovery process in wireless ad-hoc networks. We chose a set of nodes running 802.11b at the PHY and MAC layers, AODV for route discovery, and TCP at the transport layer. Aside from our methods, we also implemented a simple MAC layer encryption with a static key. Our packet-hiding methods require the processing of each individual packet by the hiding sub layer. We emphasize that the incurred processing delay is acceptable, even for real-time applications. The SCHS requires the application of two permutations and one symmetric encryption at the sender, while the inverse operations have to be performed at the receiver. Such operations can be implemented in hardware very efficiently. Symmetric encryption such as AES can be implemented at speeds of tens of Gbps/s when realized with Application Specific Integrated Circuits (ASICs) or Field Programmable Gate Arrays (FPGAs). These processing speeds are orders of magnitude higher than the transmission speeds of most current wireless technologies, and hence, do not impose a significant delay.

Similarly, the AONT-HS performs linear operations on the packet that can be efficiently implemented in hardware. We note that a non-negligible processing delay is incurred by the CPHS. This is due to the cryptographic puzzle that must be solved at the receiver. CPHS should only be employed when the symbol size at the PHY layer is too small to support the SHCS and AONTHS solutions. The processing delays of the various schemes are taken into account in our experimental evaluations.

In the third set of experiments, we evaluated the performance of TCP in a congested ad-hoc network. We considered the same network topology used in the second set of experiments. Twenty source/destination pairs simultaneously exchanged 2MB files using TCP. we show the effective throughput averaged over all 20 TCP connections. We observe that efficient packet-hiding techniques such as SHCS, and AONT-HS have a relatively small impact on the overall throughput. This is because in a congested network, the performance is primarily dependent on the queuing delays at the relay nodes. The communication overhead introduced by the transmission of the packet hiding parameters is small and hence, does not significantly impact the throughput. On the other hand, for CPHS, we observe a performance reduction of 25% − 30% compared to the case of no packet-hiding. This reduction is attributed to the delay introduced by CPHS for the reception of each packet. Note that in the congested network scenario, the throughput reduction of CPHS is smaller compared to the non-congested one because nodes can take advantage of the queuing delays to solve puzzles.

VI. Conclusion

We addressed the problem of selective jamming attacks in wireless networks. We considered an internal adversary model in which the jammer is part of the network under attack, thus being aware of the protocol specifications and shared network secrets. 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. Our schemes combine cryptographic Primitives such as commitment schemes, cryptographic puzzles, and all-or-nothing transformations (AONTs) with physical layer characteristics. We analyzed the security of our schemes and quantified their computational and communication overhead.

Bibliography

[1] T. X. Brown, J. E. James, and A. Seth. Jamming and sensing of encrypted wireless ad hoc networks. In Proceedings of MobiHoc, pages 120–130, 2006. [2] M. Cagalj, S. Capkun, and J.-P. Hubaux. Wormhole-based antijamming techniques in sensor networks. *IEEE Transactions on Mobile Computing*, 6(1):100–114, 2007.
[3] A. Chan, X. Liu, G. Noubir, and B. Thapa. Control channel jamming: Resilience and identification of traitors. In *Proceedings of ISIT*, 2007.
[4] T. Dempsey, G. Sahin, Y. Morton, and C. Hopper. Intelligent sensing and classification in ad hoc networks: a case study. *Aerospace and Electronic Systems Magazine, IEEE*, 24(8):23–30, August 2009.
[5] Y. Desmids. Broadcast anti-jamming systems. *Computer Networks*, 35(2-3):223–236, February 2001.
[6] K. Gaj and P. Chodowiec. FPGA and ASIC implementations of AES. *Cryptographic Engineering*, pages 235–294, 2009.
[7] O. Goldreich. *Foundations of cryptography: Basic applications*. Cambridge University Press, 2004.
[8] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceedings of MobiSys*, 2008.
[9] IEEE. IEEE 802.11 standard. http://standards.ieee.org/getieee802/ download/802.11-2007.pdf, 2007.
[10] A. Juels and J. Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *Proceedings of NDSS*, pages 151–165, 1999.