Packet-Thrashing Methods for Preventing Discriminatory Blocking Attacks

C.Mani,MCA.,M.Phil.,M.E
Associate Professor,
Department of Computer Applications,
Nandha Engineering College/Anna University,
Erode,
India.

P.Anandraj,
Final year,
Department of Computer Applications,
Nandha Engineering College/Anna University,
Erode,
India.

Abstract—*The open environment of the wireless middle leaves it susceptible to intentional interference attacks, normally referred to as blocking. This intended interference with wireless transmissions can be used as a mounting refutation of- Service attacks on wireless networks. Generally blocking has been addressed under an peripheral threat model. Though, adversaries with internal acquaintance of protocol specifications and network secrets can begin small-effort jamming attacks that are difficult to detect and counter. During this work, we address the problem of selective jamming attacks in wireless networks. In these attacks, the challenger is energetic simply for a small phase of time, selectively targeting messages of high importance. We demonstrate the advantages of selective jamming in terms of network performance degradation and challenger effort by presenting two case studies. a selective attack on TCP and one on routing. We show that selective jamming attacks can be launched by performing real-time packet classification at t he physical layer. En route for moderate these attacks, we extend three schemes to prevent existent-point packet organization by combining cryptographic primitives with physical-layer attributes. We analyze the security of our methods and evaluate their computational and communication transparency.*

Index Terms—*Selective Jamming, Denial-of-Service, Wireless Network s, Packet Classification.*

1 INTRODUCTION

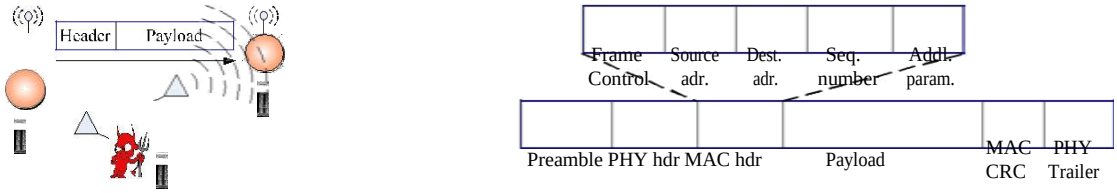
Wireless networks rely on the constant availability of the wireless medium to interconnect participating nodes. conversely, the open environment of this standard leaves it exposed to various security threats. everyone with a transceiver can overhear something on wireless transmissions, bring in spurious messages, otherwise jam legitimate ones. Though eaves dropping 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 [1],. into the simplest form of jamming, the adversary interferes with the reception of messages by transmitting a continuous jamming signal [2], or several short jamming pulses [3].

Opening,

A opening translation of this document was accessible at IEEE ICC 2010 convention. This research be supported in element by NSF (CNS-0844111, CNS-1016943). Every opinion, necessarily reflect the views of NSF .to node concession, neutralize the gains of SS.

Broad cast communications are particularly exposed under an internal threat model as all intended receiver must be aware of the secret used to protect transmissions.

During this document, we address the difficulty of jamming under an internal threat reproduction. a jammer can target route-request/route-counter 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 stream.



(a) (b)

Fig. 1. (a) Realization of a selective jamming attack, (b) a generic frame format for a wireless network.

Our scheme relies on the united thought of crypto-graphic mechanism with PHY-layer attributes. We analyze the protection of our scheme and show that they complete strong security property, with minimal blow on the network routine.

We illustrate the feasibility of selective jamming attacks. Section 4 illustrates the impact of selective jamming. In Sections 5, 6, and 7, we develop methods for preventing selective jamming. In Section 8, we evaluate the impact of our attack mitigation methods on the network performance. Section 9, presents related work. In Section 10, we conclude.

2 PROBLEM DECLARATION AND ASSUMPTIONS

2.1 Problem Statement

Consider the development depicted in Fig. 1(a). Nodes A and B correspond through a wireless connection. within the communication variety of both A and B there is a jamming join J. after A transmits a packet m to B, node J categorize by getting only the first little bytes of m. J subsequently corrupts m away from improvement by interfering with its reception at B. We address the problem of preventing the jamming node from classifying m in real time, thus mitigating J 's capability to perform selective jamming. Our goal is to transform a selective jammer to a random individual. Communication that in the present work, we execute not address packet methods based on protocol semantics, as described in [4]

2.2 System and Challenger Model

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. Communicate both in uncast 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.

Communication Model–Packets are transmitted at a rate of **R** bauds. Each PHY-layer symbol corresponds to q bits, where the value of q is defined by the underlying digital In conclusion, 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.

Adversary 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 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 [5], [6]. 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.

3 REAL-TIME PACKET CATEGORIZATION

In this section, we describe how the rival can classify packets in real time, previous to the packet transmission is completed. Once a packet is classify, the opponent may choose to jam it depending on his strategy. Consider the basic communication system depicted in Fig. 2. At the PHY layer, a packet m is determined, 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 packet m.

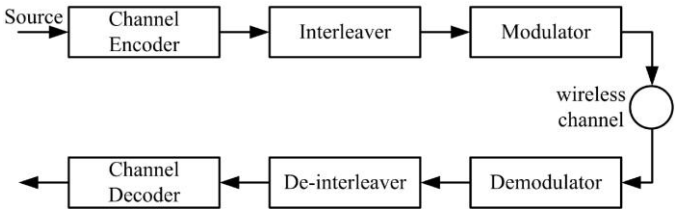


Fig. 2. A generic communication system diagram.

Data is accepted using a 1/2-rate encoder after it is mapped to an symbol of q = 48 bits. into this case, decoding of one symbol provide 24 bits of data. At the highest data rate of 54 Mbps, 216 bits of data are improved per symbol.

At the subsequently stage, the 1/2-rate convolution encoder

maps the packet to a sequence of 1,180 bits. into revolve, the output of the encoder is split into 25 blocks of 48 bits each and interleaved on a per-sign basis. in conclusion, both of the blocks is modulated as an OFDM symbol for transmission. The in sequence controlled in each of the 25 OFDM symbols is as follows:

- secret code 1-2 contain the PHY-layer heading and the first byte of the MAC description. The PHY header reveals the length of the packet, the transmission rate, and synchronization information. The earliest byte of the MAC header reveals the protocol version and the type and subtype of the MAC frame (e.g., DATA, ACK).
- secret code 3-10 contain the source and destination MAC addresses, and the length of the IP packet header.
- secret code 11-17 contain the source and destination IP addresses the size of the TCP datagram accepted by the IP packet, and other IP layer information. The original two bytes of the TCP datagram reveal the source port.
- secret code 18-23 contain the TCP destination port, sequence number, acknowledgment number, TCP flags, window size, and the heading checksum.
- Symbols 24-25 contain the MAC CRC code.

An perceptive clarification to selective jamming would be the encryption of transmitted packets (including headers) with a still key. Conversely, for broadcast communications, this static decryption key must be identified to all proposed receivers and consequently, is susceptible to compromise. For example, consider the cipher-block chaining (CBC) mode of encryption [7]. To encrypt a message m with a key k and an initialization vector IV, message m is split into x blocks $m_1, m_2, \ldots m_x$, and each cipher text block c_i, is generated as:

$$c_1 = IV,\ c_{i+1} = E_k (c_i \oplus m_i),\ i = 1, 2, \ldots, x, \qquad (1)$$

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

$$m_i = c_i \oplus D_k (c_{i+1}),\ i = 1, 2, \ldots, x. \qquad (2)$$

4 IMPACT OF SELECTIVE JAMMING

Transmission aperture this leads to a dispensable dawdling after of the purpose. Communication that, for average of $p > 0.4$, the TCP connection is abort for the case of random and TCPACK jamming, allocated to the recurring timeouts at the sender.

The jammers remain dynamic. at this time, intended for selective jamming attacks, we implicit that 13% of the package have to be degraded in order to be drop [8]. during the casing of random jamming, the adversary is not aware of the type of packets transmitted (by means of processing the title of these packets). Therefore, he is unspecified to pack the complete package in order to drop it. We view to discerning jamming require the jamming of roughly one order of importance less packets than random jamming.

Selective Jamming at the Transport Layer -This is because, as the packet transmission rate of the sender drops fewer packets needed to be selectively under attack. 3(d), we study that targeting control packets such as RTS/CTS messages and TCP-ACKs yields the lowest jamming activity, because control packets are significantly smaller compared to data packets. Such low-effort jamming attacks are not only efficient in terms of energy payments, but also challenging in localizing and physically removing the jamming devices.

Discerning blocking at the Network Layer–In this scenario, we simulated a multichip wireless network of 35 nodes, randomly placed within a square area. The AODV routing protocol was used to discover and establish routing paths [9]. 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.

In the above definition, it is easily seen that the release of d_{part} must be limited to a fraction of d, in order for m to remain hidden. If a significant part of d becomes known to the verifier, trivial attacks, such as brute forcing the unknown bits of d, become possible.

4.1 A Strong Hiding Commitment Scheme (SHCS)

We propose a strong hiding commitment scheme (SHCS), 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,

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

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

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

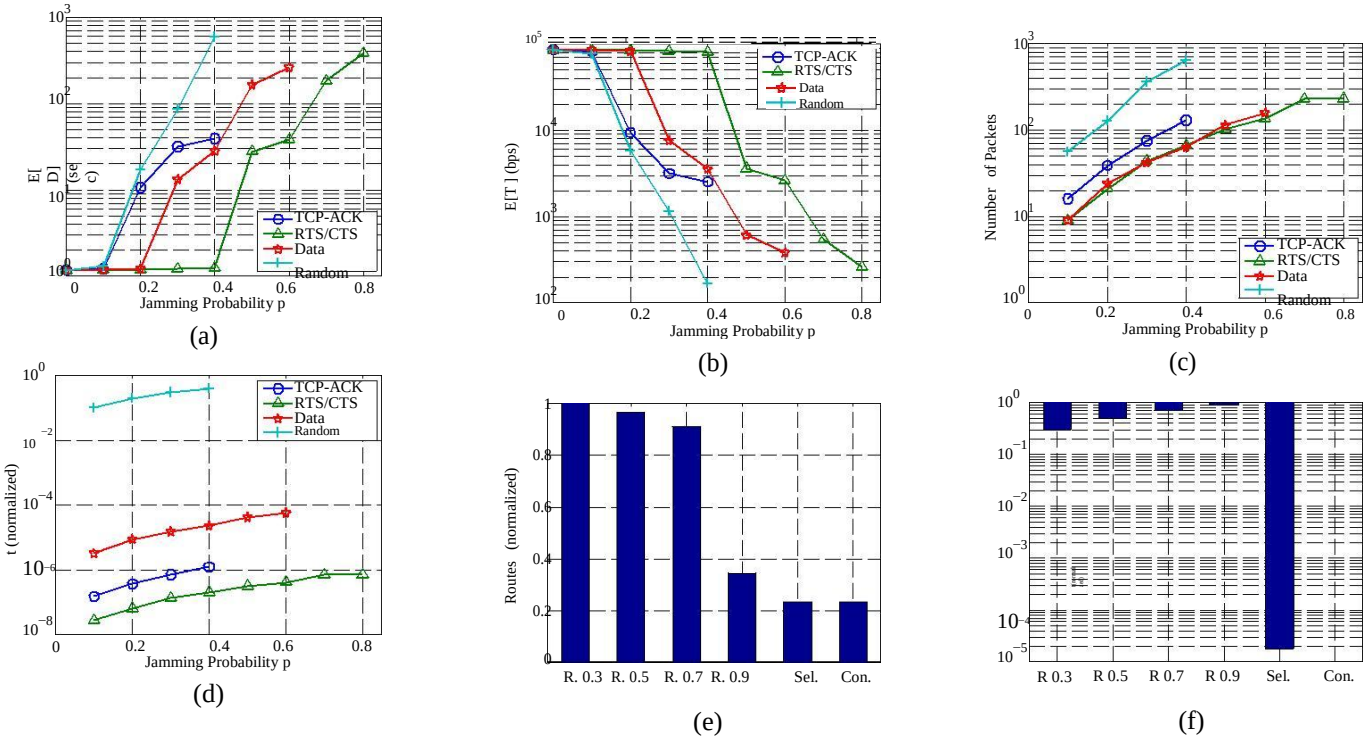


Fig. 3. (a) Average application delay E[D], (b) average effective throughput E[T], (c) number of packets jammed, (d) fraction of time the jammer is active, (e) number of connections established in the network, (f) fraction of time the jammer is active. R p: random jammer with probability p; Con.: constant jammer; Sel.: selective jammer.

4.2 Execution Information of SHCS

The proposed SHCS requires the joint thought of the MAC and PHY layers. This save the extra packet description needed for transmit d individually.

Consider a frame m at the MAC layer delivered to the hiding sub layer. Frame m consists of a MAC header and the payloads, followed by the trailer contain 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 response of serious packet identifiers such as headers. After the permutation, $\pi_1(m)$ is encrypted using a *random* key k to produce the commitment value $C = E_k(\pi_1(m))$.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 Section 5.4 we show that both are necessary to achieve packet hiding.

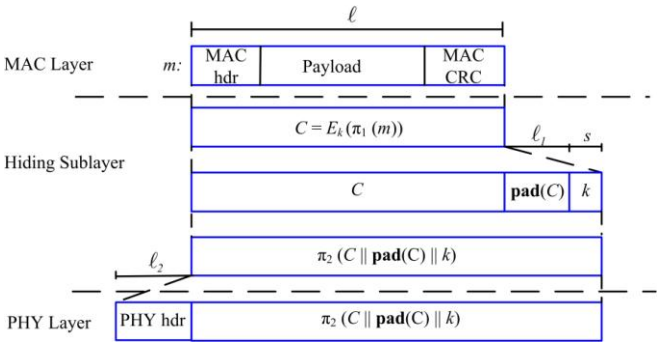


Fig. 4. Processing at the hiding sub layer.

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 principle of padding is to ensure that k is modulated in the minimum number of secret code desirable for its transmission. This is necessary for minimize the time for which part of k develop into open to the basics. Let ℓ_1 denote the number of bits padded to C. For simplicity, assume that the length of C is a several of the block extent 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 $(\ell + \ell_1 + \ell_2 + s)$ bits. Assuming that the rate of the encoder is α/β the output of the encoder will be of length, $\frac{\beta}{\alpha}(\ell + \ell_1 + \ell_2 + s)$. meant for the last symbol of transmission to include $\frac{\alpha}{\beta}$ q bits of the key k, it must hold that,

$$\ell_1 = \overline{B}\,\text{-}q - \text{-}(\ell + \ell_2) \quad \overline{\alpha}\,\text{-} \bmod q)\text{-}.$$

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. Moreover, header information is pushed at the end of π_1 (m). This prevents early reception of the corresponding cipher text blocks.

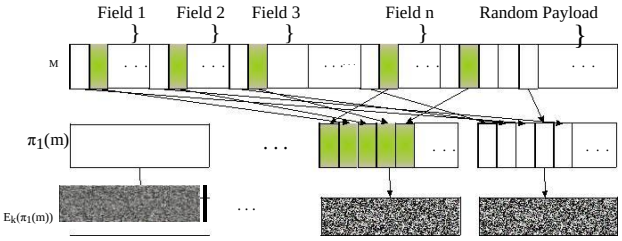


Fig. 5. Application of permutation π_1 on packet m.

4.3 Protection Analysis

In this section, we analyze the security of SHCS by evaluating the ability of J in classifying a transmitted packet at different stages of the packet transmission.

Release of C–We first examine if J can classify m by observing the commitment value C. Though C and k are part of the same packet, symbols corresponding to C .To minimize the communication over-head, k must be selected to be of the smallest length adequate for the protection of C, for the time required to transmit one packet. Assuming the encryption of a plaintext block m_i with a key k_i randomly maps to a cipher text $c_i = E_{ki}(m_i)$, every cipher text $c_i \in \mathbf{C}$ occurs with probability $p_C = \frac{1}{|C|}$. The problem of finding the probability that all $|\mathbf{M}||\mathbf{K}|$ cipher texts produced by the encryption of all plaintexts with all keys are unique, can be formulated as a "birthday problem" [10]:

$$\Pr[\text{cipher texts unique}] \approx e^{\frac{-|\mathbf{M}|\cdot|\mathbf{K}|(|\mathbf{M}|\cdot|\mathbf{K}|-1)}{2|C|}}.$$

As an example, consider the encryption of a message $m = \{m_1, m_2, \ldots m_x\}$ with a key k of length 56 bits, using blocks of 128 bits. For a fairly small plaintext space (e.g., $|\mathbf{M}| = 16$), the probability of cipher text uniqueness is equal to 99.8%. Thus, the adversary can recover k, by launching a codebook attack on m_1. Randomization of the plaintext ensures that all plaintexts are possible, thus equating the plaintext space with the cipher text space.

Partial release of d–Depending on the PHY layer implementations, d = k requires $n \geq 1$ symbols for its transmission Assuming that the adversary waits until the maximum number of bits of k are released, the key search space before the transmission of the last two symbols is equal to $2^{2\frac{\alpha}{\beta}q}$ keys. The adversary must be capable of performing on average $\mathbf{N} = 2^{(2\frac{\alpha}{\beta}q - 1)}\,\mathbf{R}$ decryptions per second in order to find k before the last symbol is transmitted[2]. Here, we have assumed that, on average, half the key space must be searched.

Binding property–The binding property is not a security requirement of SHCS under our adversary model. the jammer may launch denial of service attacks by making the receiver R to accept a $k' \mathbf{6}= k$ such that $m' = D_{k'}(C)$ is a meaningful message. Even though SHCS is not designed to ensure the binding property of commitment schemes, generating a $k' \mathbf{6}= k$ that opens a valid value of $m' \mathbf{6}= m$ is a computationally hard task. In order to find such a k', the jammer has to launch a brute force attack on C. Here, not only the attack must be performed in a timely manner, but m' has to be in the right

Given that k is transmitted right after C, the jammer has no time to find an appropriate k' that would lead to the decryption of an acceptable m', assuming that such m' exists. If m' is not meaningful, substituting k with k' is equivalent to a jamming attack on m without classification (no selectivity).

The binding property can be theoretically achieved if a random string r is appended to m [11]. In this case, the commitment/recommitments pair (C, d) is,

$$C = (\gamma, \delta) = (E_k(m||r), r), \quad d = k.$$

Provided that r is sufficiently long, a computationally bounded jammer cannot find a k' such that $D_{k'}(C) = m'||r$. In this case, r preserves the integrity of message m. Since the addition of r is not necessary for preventing real-time classification of m, we leave the implementation of the binding property to the discretion of the system designer.

4.4 *Source Overhead of SHCS*

Communication Overhead–For every packet m, a random key k of length s is appended. Also, $(\ell_b - (\ell \bmod \ell_b))$ bits of overhead are added by the encryption algorithm, to convert a plaintext of length ℓ to a multiple of the encryption block. Thus, the communication overhead of SHCS is equal to $s + (\ell_b - (\ell \bmod \ell_b))$, per packet. Here, we do not account for the padding string **pad**(C), because the addition of **pad**(C) does not increase the number of transmitted symbols.

Computation Overhead–The computation overhead of SHCS is one symmetric encryption at the sender and one symmetric decryption at the receiver. Because the header information is permuted as a trailer and encrypted, all receivers in the vicinity of a sender must receive the entire packet and decrypt it, before the packet type and destination can be determined. However, in wireless protocols such as 802.11, the complete packet is received at the MAC layer before it is decided if the packet must be discarded or be further processed [12]. If some parts of the MAC header are deemed not to be useful information to the jammer, they can remain unencrypted in the header of the packet, thus avoiding the decryption operation at the receiver.

5 THRASHING 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 [10]. The advantage of the puzzle-based scheme is that its security does not 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.

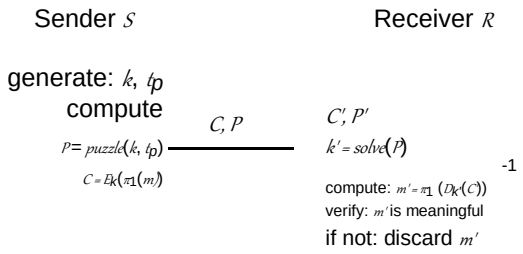


Fig. 6. The cryptographic puzzle-based hiding scheme.

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.

5.1 *Cryptographic Puzzle Hiding Scheme (CPHS)*

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, t_p), where puzzle() denotes the puzzle generator function, and t_p denotes the time required for the solution of the puzzle. Parameter t_p 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′. Fig. 6 shows the details of CPHS.

5.2 *Implementation Details of CPHS*

In this section, we consider several puzzle schemes as the basis for CPHS. For each scheme, we analyze the implementation details which impact security and performance. Cryptographic puzzles are primitives originally suggested by Markel as a method for establishing a secret over an insecure channel [3]. They find a wide range of applications from preventing DoS attacks to providing broadcast authentication and key escrow schemes.

Time-lock Puzzles–Rives et al. proposed a construction called *time-lock puzzles*, which is based on the iterative application of a precisely controlled number of modulo operations [1]. 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 to puzzle 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 =

mod 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 φ(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, K_h, 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, a are unknown, any value of k is possible [15].

Puzzles based on hashing–Computationally limited receivers can incur significant delay and energy consumption when dealing with modulo arithmetic. In this case, CPHS can be implemented from cryptographic puzzles which employ computationally efficient cryptographic primitives.

Client puzzles proposed in [6], 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.

5.3 *Security Analysis of CPHS*

With the completion of the transmission of P , any receiver can recover m. Therefore, a selective jammer must attempt to classify m before the transmission of P has been completed. We analyze the security of CPHS at different stages of its execution.

Reception of C–The jammer can attempt to classify m by crypt analyzing cipher text $C = E_k (\pi_1 (m))$. This attack is identical to the effort of classifying m with the transmission of C at the SHCS. The same analysis presented in Section 5.4 holds for the case of CPHS. The selection of a key of adequate length (e.g., 56-bit DES key) is sufficient to prevent both cipher text-only and codebook attacks.

Solving P –The transmission of k in the form of a puzzle P prevents any receiver from recovering k for at least time t_p, after the puzzle has been received. transmission of the last symbol of P. to last symbol are $2^{2\ \alpha\beta\ q}$. Assuming a brute force attack on the missing bits of the puzzle, the computational load of the adversary increases on average to $2^{2\ \alpha\beta\ q-1}\ t_p$.

The value of t_p has already been selected to prevent the puzzle solution until its transmission is completed. Hence, early solution of P before all its bits are received cannot be achieved. the selection of t_p. Therefore, this method is applicable even to wireless systems where q obtains relatively small values.

5.4 *Resource Overhead of CPHS*

Communication Overhead–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 K_h, a, and t. The value K_h is computed modulo.

The size of t is potentially smaller than a, g, and K_h, 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 [7]. In the case of hash-based puzzles, the communication overhead is equal to the transmission of the key k_1 which is of length s_1 and the hash value h(k). The typical length of hash function is 160 bits [8].

Computation Overhead–In time-lock puzzles, the sender has to apply one permutation on m, perform one symmetric encryption, and one modulo squaring operation to hide k. On the receiver side, the receiver has to perform t modulo squaring operations to recover k, one symmetric decryption to recover π_1 (m), and apply the inverse permutation. In the case of hash-based puzzles, the modulo squaring operation is substituted by, on average, $2^{s2\ -1}$ hashing operations.

6 HIDING BASED ON ALL-OR-NOTHING TRANS-FORMATIONS

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 Rivets to slow down brute force attacks against block encryption algorithms [9].

Blocks, without any change on the size of the secret key. the original AONT proposed in [2] is computationally secure. Several AONT schemes have been proposed that extend the definition of AONT to undeniable security [1]. Under this model, all plaintexts are equiprobable in the absence of at least one pseudo-message.

6.1 *An AONT-based Hiding Scheme (AONT-HS)*

In our context, packets are pre-processed by an AONT be-fore transmission but remain unencrypted. The jammer cannot perform packet classification until all pseudo-messages corresponding to the original packet have been received and the inverse transformation has been applied. Packet m is partitioned to a set of x input blocks $m = \{m_1, \ldots, m_x\}$, which serve as an input to an AONT $f : \{F_u\}^x \rightarrow \{F_u\}^{x'}$. Here, F_u denotes the alphabet of blocks m_i and x' denotes the number of output pseudo-messages with $x' \geq x$. The set of pseudo-messages $m' = \{m'_1, \ldots, m'_{x'}\}$ is transmitted over the wireless medium. At the receiver, the inverse transformation f^{-1} is applied after all x' pseudo-messages are received, in order to recover m.

6.2 *Implementation details of the AONT-HS*

In this section, we describe two AONTs which can be employed in AONT-HS; a linear transformation and the original package transformation.

Linear AONT–In Stinson show how to construct a linear AONT when the alphabet of the input blocks is a finite field F_u, with the order u being a prime power. He showed that if an invertible matrix $M = \{m_{ij} | m_{ij} \in F_u, m_{ij} \mathbf{6}= 0\}_{x \times x}$ exists, then the transformation $f(m) = mM^{-1}$ is a linear AONT. He also provided a method for constructing such M which is as follows.

Let $u = v^i$, where v is prime and i is a positive integer. Choose $\lambda \in F_u$ such that $\lambda \in / \{n - 1 \pmod v, n - 2 \pmod v\}$ and define the linear AONT LT to be,

$$LT = \begin{matrix} 1. & 0. & \bullet\bullet\bullet. & 0. & 1. \\ 0.. & 0.. & \bullet\bullet\bullet.. & 1.. & 1.. \\ 1 & 1 & \bullet\bullet\bullet & 1 & \lambda \end{matrix} \quad (4)$$

Given $m = \{m1, \ldots, mx\}$,

$$m'x = \lambda mx + \sum_{j=1}^{x-1} mj, \quad m'i = mi + m'x, \quad 1 \le i \le (x-1). \quad (5)$$

Conversely, given $m' = \{m'1, \ldots, m'x\}$, the original input $m = \{m1, \ldots, mx\}$ is recovered as follows:

$$mi = mi' - mx', \quad 1 \le i \le (x-1), \quad (6)$$

$$mx = \gamma(m1' + \ldots mx'-1 - mx'), \quad \gamma = \frac{1}{n - \lambda - 1}. \quad (7)$$

(6), (7) that if any of the {m′i} is missing, all values of mi are possible, for every i. Thus, the linear AONT provides undeniable security.

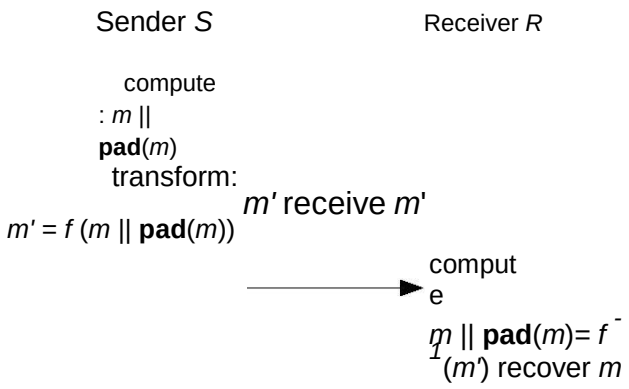


Fig. 7. The AONT-based Hiding Scheme (AONT-HS).

The Package Transform–In the package transform [21], given a message m, and a random key k′, the output pseudo-messages are computed as follows:

$$mi' = mi \oplus Ek'(i), \text{ for } i = 1, 2, \ldots, x \quad (8)$$

$$mx'+1 = k' \oplus e1 \oplus e2 \oplus \bullet\bullet\bullet \oplus ex, \quad (9)$$

where $ei = Ek0(m'i \oplus i)$, for $i = 1, 2, \ldots, x$, and k0 is a fixed publicly-known encryption key. With the reception of all pseudo-messages message m is recovered as follows:

$$k' = mx' + 1 \oplus e1 \oplus e2 \oplus \bullet\bullet\bullet \oplus ex, \quad (10)$$

$$mi = mi' \oplus Ek'(i), \text{ for } i = 1, 2, \ldots, x. \quad (11)$$

6.3 *Security Analysis of the AONT-HS*

Partial reception of m'_i, $i < x'$–In the AONT-HS, the jammer may attempt to classify m without receiving all m'_i $(1 \le i \le x')$. By definition, AONTs prevent the computation of *any part of* m without the reception of all the pseudo-messages. In fact, for the linear AONT, undeniable security is achieved. The jammer can launch a brute force attack on m as early as the reception of m'_1. However, the system of equations formed by m'_i's when at least one is missing, has a number of solutions equal to the message space. All these solutions are equiprobable.

Partial release of m'_x –With the partial release of the last pseudo-message m'_x, the space of the possible original messages m is reduced.

The search space for m'_x is reduced to its smallest value before the transmission of the last two symbols, in which case the probable values of m are equal to $2^{2\alpha\beta q}$. The adversary must be capable of solving on average $2^{2\alpha\beta q - 1}$ systems of linear equations in time equal to the length of one symbol ($_R^1$ sec), in the case of the linear AONT, or perform the same number of decryptions for the case of the package transform. For instance when q = 48 and α/β = 1/2 (802.11a), the search space is equal to 1.4×10^{14}. As in the case of SHCS, when the value of q becomes small ($q \le 2^{\beta - \alpha} \log_2 \mathbf{N} + 1$), a brute force attack on m is possible. Therefore, AONT-HS is suitable for PHY layer implementations where q is sufficiently large.

6.4 *Resource Overhead of the AONT-HS*

Communication Overhead–In AONT-HS, the original set of x messages is transformed to a set of x' pseudo-messages, with $x' \ge x$. Additionally, the function **pad**() appends $(\ell_b - (\ell \bmod \ell_b))$ bits in order to make the length of m a multiple of the length ℓ_b of the pseudo-messages m'. Hence, the communication overhead introduced is $(\ell_b(x' - x) + \ell_b - (\ell \bmod \ell_b))$ bits. For the linear AONT, $x = x'$, and therefore, only the padding communication overhead is introduced. For the package transform, the overhead

is equal to the length of one pseudo-message ($x' = x + 1$).

Computation Overhead–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 sym-metric decryptions at the receiver. Note that the length of the plaintext for the x' encryptions is relatively small com-pared to the length of message m Assuming a pseudo-message block size equal to the cipher text block size ℓ_b, 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 $\ell + \ell_b$.

7 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.

Impact on real-time systems– 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 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) [26]. 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. As suggested in Section 6, CPHS should only be employed when the symbol size at the PHY layer is too small to support the SHCS and AONT-HS solutions. The processing delays of the various schemes are taken into account in our experimental evaluations.

Experimental evaluation–In the first set of experiments we setup a single file transfer between a client and server, connected via a multi-hop route. The client requested a 1 MB file from the server. We evaluated the effects of packet hiding by measuring the effective throughput of the TCP connection in the following scenarios: (a) No packet hiding (N.H.), (b) MAC-layer encryption with a static key (M.E.), (c) SHCS (C.S.), (d)

Time-lock CPHS (T.P.), (e) Hash-based CPHS (H.P.), (f) Linear AONT-HS (L.T.), and (g) AONT-HS based on the package transform (P.T.).

In Fig. 8(a), we show the effective throughput aver-aged over 100 different traces. We observe that MAC-layer encryption, SHCS and the linear AONT-HS achieve an effective throughput close to the throughput in the absence of packet hiding. This is justified by the the relatively small communication overhead of each hiding method and the small queuing delay at intermediate routers due to the absence of any cross traffic. The AONT-HS based on the package .

Techniques based on cryptographic puzzles decrease the effective throughput of the TCP connection to half, compared to the no hiding case. This performance is anticipated since the time required to solve a puzzle after a packet has been received at the MAC layer is equal to the transmission time of each packet. While this constitutes a significant performance reduction, we emphasize that cryptographic puzzles were suggested as a candidate solution only when the symbol size is so small that more efficient hiding methods do not provide adequate levels of security.

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 2 MB files using TCP. 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.

8 ASSOCIATED WORK

Jamming attacks on voice communications have been launched since the 1940s. In the context of digital communications, the jamming problem has been addressed under various threat models. We present a classification based on the selective nature of the adversary.

8.1 Previous work on Selective Blocking

In [8], Thence studied the impact of an external selective jammer who targets various control packets at the MAC layer. To perform packet classification, the adversary exploits inter-

packet timing information to infer eminent packet transmissions. In [2], Law et al. proposed the estimation of the probability distribution of inter-packet transmission times for different packet types based on network traffic analysis. Future transmissions at various layers were predicted using estimated timing information. Using their model, the authors proposed selective jamming strategies for well known sensor network MAC protocols.

Selective jamming attacks have been experimentally implemented using software-defined radio engines. Wilhelm et al. implemented a USRP2-based jamming plat-form called Refracts that enables selective and reactive jamming [5]. Refracts was shown to be agnostic to technology standards and readily adaptable to any desired jamming strategy. The success rate of a selective jamming attack against a 802.15.4 network was measured to be 99.96%. Blips et al. studied selective jamming attacks against the rate-adaptation mechanism of 802.11 [6].

Selective jammer targeting specific packets in a point-to-point 802.11 communication was able to reduce the rate of the communication to the minimum value of 1 Mbps, with relatively little effort (jamming of 5-8 packets per second). The results were experimentally verified using the USRP2/GNU radio platform.

Several researchers have suggested channel-selective jamming attacks, in which the jammer targets the broadcast control channel. It was shown that such attacks reduce the required power for performing a Do's attack by several orders of magnitude [3]. To protect control-channel traffic, the replication of control transmission in multiple channels was suggested in The "locations" of the control channels where cryptographically protected. In, Lazes et al. proposed a randomized frequency hopping algorithm to protect the control channel from inside jammers. Stressed et al. proposed a frequency hopping anti-jamming technique that does not require the existence of a secret hopping sequence, shared between the communicating parties [8].

8.2 Non-Selective Jamming Attacks

Conventional methods for mitigating jamming employ some form of SS communications. The transmitted signal is spread to a better bandwidth following a PN sequence. Without the knowledge of this sequence, a large quantity of energy (typically 20-30 dB gain) is requisite to interfere with an constant transmission. Lin et al. showed that jamming 13% of a packet is sufficient to overcome the ECC capabilities of the receiver [40]. Us et al. categorized jammers into four models: (a) a constant jammer, (b) a deceptive jammer that broadcasts fabricated messages, (c) a random jammer, and (d) a reactive jammer that jams only if activity is sensed .They further studied the problem of detecting the presence of jammers by measuring performance metrics such as packet delivery ratio Cabal et al. proposed wormhole-based anti-jamming techniques for wireless sensor networks (WSNs) [4]. Using a wormhole link, sensors within the jammed region establish communications with outside nodes, and notify those regarding ongoing jamming attacks.

9 CONCLUSION

We address the problem of selective jamming attacks in wireless networks. We considered an internal opponent model in which the jammer is part of the network under attack, consequently person aware of the protocol specifications and shared network secrets. We show that the jammer can classify transmit packets in real time by decoding the original few symbols of an continuing transmission. We evaluated the contact 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.

REFERENCES

[1] 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.

[2] M. Cagalj, S. Capkun, and J.-P. Hubaux. Wormhole-based anti-jamming techniques in sensor networks. IEEE Transactions on Mobile Computing, 6(1):100–114, 2007.

[3] 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.

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

[5] K. Gaj and P. Chodowiec. FPGA and ASIC implementations of AES. Cryptographic Engineering, pages 235–294, 2009.

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

[7] B. Greenstein, D. Mccoy, J. Pang, T. Kohno, S. Seshan, and D. Wether-all. Improving wireless privacy with an identifier-free link layer protocol. In Proceedings of MobiSys, 2008.

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

[9] L. Lazos, S. Liu, and M. Krunz. Mitigating control-channel jamming attacks in multi-channel ad hoc networks. In Proceedings of the 2nd ACM conference on wireless network security, pages 169–180, 2009.

[10] 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.

[11] X. Liu, G. Noubir, and R. Sundaram. Spread: Foiling smart jammers using multi-layer agility. In Proceedings of INFOCOM, pages 2536– 2540, 2007.

[12] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential DSSS: Jamming-resistant wireless broadcast communication. In Proceedings of INFOCOM, San Diego, 2010.