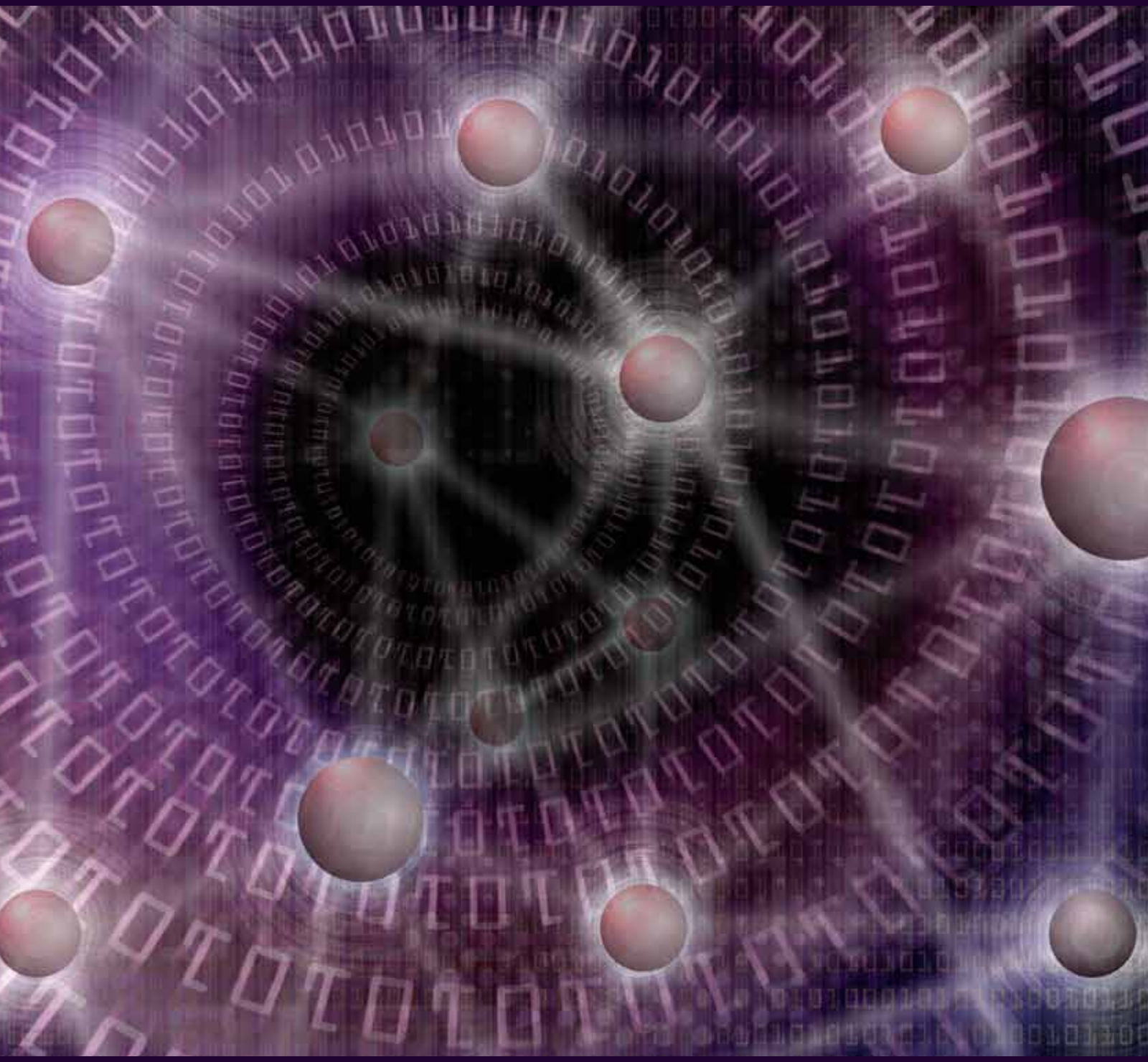
EURASIP Journal on Wireless Communications and Networking

Wireless Network Security

Guest Editors: Yang Xiao, Hui Chen, Shuhui Yang,
Yi-Bing Lin, and Ding-Zhu Du



Wireless Network Security

EURASIP Journal on
Wireless Communications and Networking

Wireless Network Security

Guest Editors: Yang Xiao, Hui Chen, Shuhui Yang,
Yi-Bing Lin, and Ding-Zhu Du

Contents

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 532434, 3 pages
doi:10.1155/2009/532434

Editorial
Wireless Network Security

Yang Xiao,[1] Hui Chen,[2] Shuhui Yang,[3] Yi-Bing Lin,[4] and Ding-Zhu Du[5]

[1] *Department of Computer Science, University of Alabama, P.O. Box 870290, Tuscaloosa, AL 35487-0290, USA*
[2] *Department of Mathematics and Computer Science, Virginia State University, Petersburg, VA 23806, USA*
[3] *Department of Math, Computer Science and Statistics, Purdue University, Calumet, 2200 169th Street, Hammond, IN 46323, USA*
[4] *Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 300, Taiwan*
[5] *Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA*

Correspondence should be addressed to Yang Xiao, yangxiao@ieee.org

Received 13 December 2009; Accepted 13 December 2009

Wireless networking has been enjoying fast development, evidenced by wide deployments of many wireless networks of various sizes, such as wireless personal area networks (WPANs), local area networks (WLANs), metropolitan area networks (WMANs), and wide area networks (WWANs). These wireless networks can be of different formations, such as cellular networks, ad hoc networks, and mesh networks, and can also be domain specific networks, such as vehicular communication networks and sensor networks. However, wireless networks are lack of physical security because the underlying communications are carried out by electromagnetic radiations in open space. Wireless networks pose a unique challenge in computer and network security community. The effort to improve wireless network security is linked with many technical challenges including compatibility with legacy wireless networks, complexity in implementation, and practical values in the real market. The need to address wireless network security and to provide timely solid technical contributions establishes the motivation behind this special issue.

This special issue received many submissions. Unfortunately, due to the limited space and volume, we can only choose twelve papers in this special issue, as a result of the peer-review process.

Wireless vehicular networks and sensor networks are two domain-specific networks that can have many important applications. This special issue includes a few papers investigating topics of locating and tracking malicious insiders and key management for sensor networks.

In vehicular communication networks that are hardened by public cryptographic systems, security modules including secret keys can be exposed to wrong hands due to weakness of physical security than those that can be enforced. With the security modules and secret keys, various security attacks can be launched via authenticated messages. Christine Laurendeau and Michel Barbeau designed a hyperbolic position bounding algorithm to localize the originator of an attack signal within a vehicular communication network. Their algorithm makes use of received signal strength reports for locating the source of attack signals without the knowledge of the power level of the station that is transmitting packets. Find the details of their work in the paper entitled "Probabilistic localization and tracking of malicious insiders using hyperbolic position bounding in vehicular networks."

Key management is always a challenging issue in wireless sensor networks due to resource limitation imposed by sensor nodes. Xiang et al. surveyed key establishment and distribution protocols in their paper entitled "In situ key establishment in large-scale sensor networks," where key establishment protocols are categorized as deterministic key predistribution, probabilistic key predistribution, and in situ key establishment protocols. Different from predistribution protocols, in situ protocol only requires a common shared key among all nodes to prevent node injection attack. Keys for securing pairwise communication among nodes are achieved by key establishment process after deployment. The paper provides an in-depth discussion and comparison of previously proposed three in situ key establishment protocols, namely, iPAK, SBK, and LKE. In addition, the study leads to an improvement where random keys can be easily computed from a secure pseudorandom function. This new approach requires no computation overhead at regular worker sensor nodes, and therefore has a high potential to conserve the network resource.

In the paper entitled "A flexible and efficient key distribution scheme for renewable wireless sensor networks," A-N. Shen et al. proposed a key distribute scheme for three-tier hierarchical wireless sensors networks that consist of base stations, cluster heads, and sensor nodes. By making use of secret keys generated by a bivariate symmetric polynomial function and well-designed message exchanges, the key distribution protocol can allow new sensor nodes to be added, deter node captures, and cope with the situations when base stations are either online or offline.

Routing protocols are integral components of multihop networks. Attacks on routing protocols can render such networks nonfunctional. Many wireless sensor networks can be viewed as multihop ad hoc networks. The following three papers discuss security issues of routing protocols.

Establishing trusts among sensor nodes can be an effective approach to counter attacks. In the paper entitled "Cautious rating for trust-enabled routing in wireless sensor networks," I. Maarouf et al. studied trust-aware routing for wireless sensor networks. Trust awareness of sensor nodes are commonly obtained by implementing a reputation system, where the measures of trustworthiness of sensor nodes are provided by a rating system. In the paper, the authors proposed and studied a new rating approach for reputation systems for wireless sensor networks called "*Cautious RAting for Trust Enabled Routing (CRATER)*."

In multihop wireless networks, designers of routing protocols concern not only network performance (such as bandwidth and latency) but also malicious attacks on routing protocols. Nevertheless, how to choose a path between two nodes in a network relies on both performance and security considerations. In their paper entitled "On multipath routing in multihop wireless networks: security, performance, and their tradeoff," L. Chen and J. Leneutre formulate the multipath routing problem as optimization problems with objectives as minimal security risks, maximal packet delivery ratio, or maximal packet delivery ratio under a given security risks. Polynomial time solutions to the optimization problems are proposed and studied.

Mobile Ad Hoc Networks (MANETs) are often subject to node capture attack. Once a node is captured by an adversary, all the security material stored in the node falls in the hands of the adversary. The captured node after reprogram or a newly deployed node operated by the adversary can make use of the stored security material to gain access to the networks and hence launch attacks on the network. Thus, it is beneficial to reduce the probability that nodes are detected and located, in particular, in hostile environments. X. Lu et al. proposed a routing protocol for wireless ad hoc networks where the antennas of nodes can act as both omnidirectional and directional antennas in the paper entitled "Minimizing detection probability routing in ad hoc networks using directional antennas." The routing protocol aims at reducing detection probability while finding a secure routing path in ad hoc networks where nodes employ directional antennas to transmit data to decrease the probability of being detected by adversaries.

Captured nodes pose security threats to many wireless networks. Capturing node is an important and yet very typical attack that is commonly launched to attack wireless ad hoc networks and sensor networks. Therefore, it should not come as a surprise that this issue includes another paper investigating this attack. M. Conti et al. in their paper entitled "Mobility and cooperation to thwart node capture attacks in MANETs" demonstrated that node mobility, together with local node cooperation, can be leveraged to design secure routing protocols that deters node capture attacks, among many other benefits.

This special issue also includes discussions on another type of an important attack, called "*coordinated attacks*," launched via Botnets. Advancements of wired and wireless networks have also enabled attackers to control applications running on many networked computers to coordinately attack while letting users to access remote computing resources much easily. Software applications in many hosts can form self-propagating, self-organizing, and autonomous overlay networks that are controlled by attackers to launch coordinated attacks. Those networks are often called Botnets. In their paper entitled "Botnet: classification, attacks, detection, tracing, and preventive measures," J. Liu et al. provide a survey on this subject. The paper discusses many fundamental issues regarding Botnets and sheds light on possible future research directions.

Ever-evolving mobile wireless networking technology leads to coexistence of many different wireless networks. Seamless and fast handover among different networks such as Wireless LANs (e.g., IEEE 802.11), WiMax (e.g., IEEE 802.16), and personal communication systems (e.g., GSM) becomes an important topic under investigation. The handover mechanisms need to not only maintain the security of the networks involved but also sustain the quality of the service (QoS) requirements of network applications. The following two papers study internetwork handover mechanisms.

In the paper entitled "Pre-authentication schemes for UMTS-WLAN interworking," A. Al Shidhani and V. Leung proposed and studied two secure pre-authentication protocols for the interworking Universal Mobile Telecommunication System (UMTS) and IEEE 802.11 Wireless Local Area Networks (WLANs). The authors also verified the proposed protocols by the Automated Validation of Internet Security Protocols and Applications (AVISPAs) security analyzer.

Growing interesting in multimedia access via mobile devices has led the IEEE 802.21 workgroup to standardize the Media Independent Handover (MIH) mechanisms that enable the optimization of handovers in heterogeneous networks for multimedia access. Based on the analysis on IPSec/IKEv2 and DTLS security solutions for secure MIH message transport, J.-J. Won et al. show that handover latency can be too large to be acceptable. They thus proposed and studied a secure MIH message transport solution that reduces authentication time. Find the detail of their work in the paper entitled "Secure media independent handover message transport in heterogeneous networks."

S. Song et al. study a related but different problem in mobile wireless networks in the paper entitled "A secure and lightweight approach for routing optimization in mobile IPv6." Mobile IPv6 (MIPv6) provides mobile terminals

uninterrupted access to networks while on the move via a mechanism called Router Optimization (RO). They found three weaknesses in RO that attribute to a session hijack attack where an adversary can join an ongoing sessions at a chosen location. They proposed an authentication mechanism that hardens RO. Via performance evaluation, they show that the improved protocol achieves strong security and at the same time requires minimal computational overhead.

Cooperative radio is an important wireless communications technology that can improve capacity of wireless channels. It has been a topic that attracts growing interests. This special issue nonetheless has included the paper entitled "Distributed cooperative transmission with unreliable and untrustworthy relay channels."

Cooperative radio is subject to malicious attacks and performance degradation caused by selfish behaviors. Z. Han and Y. (Lindsay) Sun demonstrated the security vulnerabilities of the traditional cooperative transmission schemes and proposed a trust-assisted cooperative scheme that can detect attacks and has self-healing capability.

In summary, this special issue reflects growing interests in wireless network security, without which the usability of wireless networks is questionable. We believe that this special issue is a good snapshot of current research and development of wireless network security and is an important reference for researchers, practitioners, and students.

In the end, we would like to extend our appreciation to every author who has submitted their work. We are very regretful that we could not include every decent paper in this special due to the page limitation. Without unselfish reviewers' countless efforts, it would be impossible for us to select these papers from the great number of submissions and to ensure the quality of the special issue. We are thus deeply indebt to our reviewers. Last, but not the least, we thank our editor Hend Abdullah and many other editorial staff members with the journal. Without their coordination and skillful management, we would not be able to finish our task as guest editors.

Yang Xiao
Hui Chen
Shuhui Yang
Yi-bing Lin
Ding-zhu Du

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 128679, 13 pages
doi:10.1155/2009/128679

Research Article

Probabilistic Localization and Tracking of Malicious Insiders Using Hyperbolic Position Bounding in Vehicular Networks

Christine Laurendeau and Michel Barbeau

School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6

Correspondence should be addressed to Christine Laurendeau, claurend@scs.carleton.ca

Received 12 December 2008; Accepted 1 April 2009

Recommended by Shuhui Yang

A malicious insider in a wireless network may carry out a number of devastating attacks without fear of retribution, since the messages it broadcasts are authenticated with valid credentials such as a digital signature. In attributing an attack message to its perpetrator by localizing the signal source, we can make no presumptions regarding the type of radio equipment used by a malicious transmitter, including the transmitting power utilized to carry out an exploit. Hyperbolic position bounding (HPB) provides a mechanism to probabilistically estimate the candidate location of an attack message's originator using received signal strength (RSS) reports, without assuming knowledge of the transmitting power. We specialize the applicability of HPB into the realm of vehicular networks and provide alternate HPB algorithms to improve localization precision and computational efficiency. We extend HPB for tracking the consecutive locations of a mobile attacker. We evaluate the localization and tracking performance of HPB in a vehicular scenario featuring a variable number of receivers and a known navigational layout. We find that HPB can position a transmitting device within stipulated guidelines for emergency services localization accuracy.

1. Introduction

Insider attacks pose an often neglected threat scenario when devising security mechanisms for emerging wireless technologies. For example, traffic safety applications in vehicular networks aim to prevent fatal collisions and preemptively warn drivers of hazards along their path, thus preserving numerous lives. Unmitigated attacks upon these networks stand to severely jeopardize their adoption and limit the scope of their deployment.

The advent of public key cryptography, where a node is authenticated through the possession of a public/private key pair certified by a trust anchor, has addressed the primary threat posed by an outsider without valid credentials. But a vehicular network safeguarded through a Public Key Infrastructure (PKI) is only as secure as the means implemented to protect its member nodes' private keys. An IEEE standard has been proposed for securing vehicular communications in the Dedicated Short Range Communications Wireless Access in Vehicular Environments (DSRC/WAVE) [1]. This standard advocates the use of digital signatures to secure vehicle safety broadcast messages, with tamper proof devices storing secret keys and cryptographic algorithms in each vehicle. Yet a convincing body of existing literature questions the resistance of such devices to a motivated attacker, especially in technologies that are relatively inexpensive and readily available [2, 3]. In the absence of strict distribution regulations, for example, if tamper proof devices for vehicular nodes are available off the shelf from a neighborhood mechanic, a supply chain exists for experimentation with these devices for the express purpose of extracting private keys. The National Institute of Standards and Technology (NIST) has established a certification process to evaluate the physical resistance of cryptographic processors to tampering, according to four security levels [4]. However, tamper resistance comes at a price. High end cryptographic processors certified at the highest level of tamper resistance are very expensive, for example, an IBM 4764 coprocessor costs in excess of 8000 USD [5]. Conversely, lower end tamper evident cryptographic modules, such as smartcards, feature limited mechanisms to prevent cryptographic material disclosure

or modification and only provide evidence of tampering after the fact [6]. The European consortium researching solutions in vehicular communications security, SeVeCom, has highlighted the existence of a gap in tamper resistant technology for use in vehicular networks [7]. While low end devices lack physical security measures and suffer from computational performance issues, the cost of high end modules is prohibitive. The gap between the two extremes implies that a custom hardware and software solution is required, otherwise low end devices may be adopted and prove to be a boon for malicious insiders.

Vehicle safety applications necessitate that each network device periodically broadcast position reports, or *beacons*. A malicious insider generating false beacons whose digital signature is verifiable can cause serious accidents and possibly loss of life. Given the need to locate the transmitter of false beacons, we have put forth a mechanism for attributing a wireless network insider attack to its perpetrator, assuming that a malicious insider is unlikely to use a digital certificate linked to its true identity. Any efforts to localize a malicious transmitter must assume that an attacker may willfully attempt to evade detection and retribution. As such, only information that is revealed outside a perpetrator's control can be utilized. A number of existing wireless node localization schemes translate the radio signal received signal strength (RSS) at a set of receivers into approximated transmitter-receiver (T-R) distances, in order to position a transmitter. However, these assume that the effective isotropic radiated power (EIRP) used by the signal's originator is known. While this presumption may be valid for the location estimation of reliable and cooperative nodes, a malicious insider may transmit at unexpected EIRP levels in order to mislead localization efforts and obfuscate its position. Our hyperbolic position bounding (HPB) algorithm addresses a novel threat scenario in probabilistically delimiting the candidate location of an attack message's originating device, assuming neither the cooperation of the attacker nor any knowledge of the EIRP [8]. The RSS of an attack message at a number of trusted receivers is employed to compute multiple hyperbolic areas whose intersection contains the source of the signal, with a degree of confidence.

We demonstrate herein that the HPB mechanism is resistant to varying power attacks, which are a known pitfall of RSS-based location estimation schemes. We present three variations of HPB, each with a different algorithm for computing hyperbolic areas, in order to improve computational efficiency and localization granularity. We extend HPB to include a mobile attacker tracking capability. We simulate a vehicular scenario with a variable number of receiving devices, and we evaluate the performance of HPB in both localizing and tracking a transmitting attacker, as a function of the number of receivers. We compare the HPB performance against existing location accuracy standards in related technologies, including the Federal Communications Commission (FCC) guidelines for localizing a wireless handset in an emergency situation.

Section 2 reviews existing work in vehicular node location determination and tracking. Section 3 outlines the HPB mechanism in its generic incarnation. Section 4 presents three flavours of the HPB algorithm for localizing and tracking a mobile attacker. Section 5 evaluates the performance of the extended HPB algorithms. Section 6 discusses the simulation results obtained. Section 7 concludes the paper.

2. Related Work

A majority of wireless device location estimation schemes presume a number of constraints that are not suitable for security scenarios. We outline these assumptions and compare them against those inherent in our HPB threat model in [9]. For example, a number of publications related to the location determination of vehicular devices focus on self-localization, where a node seeks to learn its own position [10, 11]. Although the measurements and information provided to these schemes are presumed to be trustworthy, this assumption does not hold for finding an attacker invested in avoiding detection and eviction from the network.

Some mechanisms for the localization of a vehicular device by other nodes are based on the principle of location verification, where a candidate position is proposed, and some measured radio signal characteristic, such as time of flight or RSS, is used to confirm the vehicle's location. For example, in [12, 13], Hubaux et al. adapt Brands and Chaum's distance bounding scheme [14] for this purpose. Yet a degree of cooperation is expected on the part of an attacker for supplying a position. Additionally, specialized hardware is necessary to measure time of flight, including nanosecond-precision synchronized clocks and accelerated processors to factor out relatively significant processing delays at the sender and receiver. Xiao et al. [15] employ RSS values for location verification but they assume that all devices, including malicious ones, use the same EIRP. An attacker with access to a variety of radio equipment is unlikely to be constrained in such a manner.

Location verification schemes for detecting false position reports may be beacon based or sensor based. Leinmüller et al. [16] filter beacon information through a number of plausibility rules. Because each beacon's claimed position is corroborated by multiple nodes, consistent information is assumed to be correct, based on the assumption of an honest majority of network devices. This presumption leaves the scheme vulnerable to Sybil attacks [17]. If a rogue insider can generate a number of Sybil identities greater than the honest majority, then the attacker can dictate the information corroborated by a *dishonest majority* of virtual nodes. In ensuring a unique geographical location for a signal source, our HPB-based algorithms can detect a disproportionate number of colocated nodes.

Tang et al. [18] put forth a sensor-based location verification mechanism, where video sensors, such as cameras and RFID readers, can identify license plates. However, cameras perform suboptimally when visibility is reduced, for example, at night or in poor weather conditions. This scheme is supported by PKI-based beacon verification and correlation by an honest majority, which is also vulnerable to insider and Sybil attacks. Another sensor-based mechanism

is suggested by Yan et al. [19], using radar technology for local security and the propagation of radar readings through beacons on a global scale. Again, an honest majority is assumed to be trustworthy for corroborating the beacons, both locally and globally.

Some existing literature deals explicitly with mobile device tracking, including the RSS-based mechanisms put forth by Mirmotahhary et al. [20] and by Zaidi and Mark [21]. These presume a known EIRP and require a large number of transmitted messages so that the signal strength variations can be filtered out.

3. Hyperbolic Position Bounding

The log-normal shadowing model predicts a radio signal's large-scale propagation attenuation, or *path loss*, as it travels over a known T-R distance [22]. The variations in signal strength experienced in a particular propagation environment, also known as the *signal shadowing*, behave as a Gaussian random variable with mean zero and a standard deviation obtained from experimental measurements. In this model, the path loss over T-R distance d is computed as

$$L(d) = \overline{L}(d_0) + 10\eta \log\left(\frac{d}{d_0}\right) + X_\sigma, \tag{1}$$

where d_0 is a predefined reference distance close to the transmitter, $\overline{L}(d_0)$ is the average path loss at the reference distance, and η is a path loss exponent dependent upon the propagation environment. The signal shadowing is represented by a random variable X_σ with zero mean and standard deviation σ.

In [8], we adapt the log-normal shadowing model to estimate a range of T-R *distance differences*, assuming that the EIRP is unknown. The minimum and maximum bounds of the distance difference range between a transmitter and a receiver pair R_i and R_j, with confidence level $\mathcal{C}$, are computed as

$$\begin{aligned}\Delta d_{ij}^{-} = {} & \left(d_0 \times 10^{(\mathcal{P}^- - \mathrm{RSS}_i - \overline{L}(d_0) - z\sigma)/10\eta}\right) \\ & - \left(d_0 \times 10^{(\mathcal{P}^- - \mathrm{RSS}_j - \overline{L}(d_0) + z\sigma)/10\eta}\right),\end{aligned} \tag{2}$$

$$\begin{aligned}\Delta d_{ij}^{+} = {} & \left(d_0 \times 10^{(\mathcal{P}^+ - \mathrm{RSS}_i - \overline{L}(d_0) + z\sigma)/10\eta}\right) \\ & - \left(d_0 \times 10^{(\mathcal{P}^+ - \mathrm{RSS}_j - \overline{L}(d_0) - z\sigma)/10\eta}\right),\end{aligned} \tag{3}$$

where RSS_k is the RSS measured at receiver R_k, $[\mathcal{P}^-, \mathcal{P}^+]$ represents a dynamically estimated EIRP interval, $z = \Phi^{-1}((1 + \mathcal{C})/2)$ represents the normal distribution constant associated with a selected confidence level $\mathcal{C}$, and $[-z\sigma, +z\sigma]$ is the signal shadowing interval associated with this confidence level. The amount of signal shadowing taken into account in the T-R distance difference range is commensurate with the degree of confidence $\mathcal{C}$. For example, a confidence level of $\mathcal{C} = 0.95$, where $z = 1.96$, encompasses a larger proportion of signal shadowing around the mean path loss than $\mathcal{C} = 0.90$, where $z = 1.65$. A higher confidence level, and thus a larger signal shadowing interval, translates into a wider range of T-R distance differences.

Hyperbolas are computed at the minimum and maximum bounds, Δd_{ij}^{-} and Δd_{ij}^{+}, respectively, of the distance difference range. The resulting candidate hyperbolic area for the location of a transmitter is situated between the minimum and maximum hyperbolas and contains the transmitter with probability $\mathcal{C}$. The intersection of hyperbolic areas computed for multiple receiver pairs bounds the position of a transmitting attacker with an aggregated degree of confidence, as demonstrated in [23].

4. Localization and Tracking of Mobile Attackers

We demonstrate that by dynamically computing an EIRP range, we render the HPB mechanism impervious to varying power attacks. We propose three variations of HPB for computing sets of hyperbolic areas and the resulting candidate areas for the location of a transmitting attacker. We also describe our HPB-based approach for estimating the mobility path of a transmitter in terms of location and direction of travel.

4.1. Mitigating Varying Power Attacks. The use of RSS reports has been criticized as a suboptimal tool for estimating T-R distances due to their vulnerability to varying power attacks [24]. An attacker that transmits at an EIRP other than the one expected by a receiver can appear to be closer or farther simply by transmitting a stronger or weaker signal. Our HPB-based algorithms are immune to such an exploit, since no fixed EIRP value is expected. Instead, measured RSS values are leveraged to compute a likely EIRP range, as demonstrated in Heuristic 1.

In order for HPB to compute a set of hyperbolic areas between pairs of receivers upon detection of an attack message, a candidate range $[\mathcal{P}^-, \mathcal{P}^+]$ for the EIRP employed by the transmitting device must be dynamically estimated. We use the RSS values registered at each receiver as well as the log-normal shadowing model captured in (1) for this purpose. The path loss $L(d)$ is replaced with its equivalent, the difference between the EIRP and the RSS_k measured at a given receiver R_k. Our strategy takes the receiver with the maximal RSS as an approximate location for the transmitter and computes the EIRP range a device at those coordinates would need to employ in order for a signal to reach the other receivers with the RSS values measured for the attack message.

We begin by identifying the receiver measuring the maximal RSS for an attack message. Given that this device is likely to be situated in nearest proximity to the transmitter, we deem it the *reference receiver*. For every other receiving device R_k, we use the log-normal shadowing model to calculate the range of EIRP $[\mathcal{P}_k^-, \mathcal{P}_k^+]$ that a transmitter would employ for a message to reach R_k with power RSS_k, assuming the transmitter is located at exactly the reference receiver coordinates. The global EIRP range $[\mathcal{P}^-, \mathcal{P}^+]$ for the attack message is calculated as the intersection of all receiver-computed ranges $[\mathcal{P}_k^-, \mathcal{P}_k^+]$.

```
1: i ⇐ n − 1
2: j ⇐ 1
3: while i > 0 and j < n do
4:     if P_i^− < P_j^+ then
5:         P^− ⇐ P_i^−
6:         P^+ ⇐ P_j^+
7:         exit
8:     end if
9:     if i > 1 then
10:        if P_{i−1}^− < P_j^+ then
11:            P^− ⇐ P_{i−1}^−
12:            P^+ ⇐ P_j^+
13:            exit
14:        end if
15:    end if
16:    i ⇐ i − 1
17:    j ⇐ j + 1
18: end while
```

Pseudocode 1

Heuristic 1 (EIRP range computation). Let $\mathbb{R}$ be the set of all receivers within range of an attack message. Let $\widetilde{R}_m$ be the maximal RSS receiver and thus be estimated as the closest receiver to the message transmitter, such that $\widetilde{R}_m \in \mathbb{R}$ and $\text{RSS}_m \geq \text{RSS}_j$ for all $R_j \in \mathbb{R}$. Given that EIRP $= \overline{L}(d_0) + 10\eta \log(d/d_0) + \text{RSS} + X_\sigma$ from the log-normal shadowing model, let the EIRP range $[\mathcal{P}_k^-, \mathcal{P}_k^+]$ at any receiver R_k be determined, with confidence $\mathcal{C}$, as

$$\mathcal{P}_k^- = \overline{L}(d_0) + 10\eta \log\left(\frac{d_{mk}}{d_0}\right) + \text{RSS}_k - z\sigma, \tag{4}$$

$$\mathcal{P}_k^+ = \overline{L}(d_0) + 10\eta \log\left(\frac{d_{mk}}{d_0}\right) + \text{RSS}_k + z\sigma \tag{5}$$

where d_{mk} is the Euclidian distance between R_k and $\widetilde{R}_m$, for any $R_k \in \mathbb{R} \setminus \{\widetilde{R}_m\}$.

The estimated EIRP range $[\mathcal{P}^-, \mathcal{P}^+]$ employed by a transmitter is the intersection of receiver-computed EIRP intervals $[\mathcal{P}_k^-, \mathcal{P}_k^+]$ within which every receiver $R_k \in \mathbb{R} \setminus \{\widetilde{R}_m\}$ can reach $\widetilde{R}_m$. Since $\mathcal{P}^-$ must be smaller than $\mathcal{P}^+$, we iterate through the ascending ordered sets $\{\mathcal{P}_k^-\}$ and $\{\mathcal{P}_k^+\}$, for all $R_k \in \mathbb{R} \setminus \{\widetilde{R}_m\}$, to find a supremum of EIRP values with minimal shadowing that is lower than an infimum of maximal shadowing EIRP values. Assuming the size of $\mathbb{R}$ is n, and thus the size of $\mathbb{R} \setminus \{\widetilde{R}_m\}$ is $n - 1$, we compute the estimated EIRP range $[\mathcal{P}^-, \mathcal{P}^+]$ as shown in Pseudocode 1.

The only case where the pseudocode above can fail is if every $\mathcal{P}_i^-$ is greater than every $\mathcal{P}_j^+$ for all $1 \leq i, j \leq n - 1$. This is impossible, since (4) and (5) taken together indicate that for any k, $\mathcal{P}_k^-$ must be smaller than $\mathcal{P}_k^+$.

The log-normal shadowing model indicates that, for a fixed T-R distance, the expected path loss is constant, albeit subject to signal shadowing, regardless of the EIRP used by a transmitter. Any EIRP variation induced by an attacker translates into a corresponding change in the RSS values measured by all receivers within radio range. As a result, an EIRP range computed with Heuristic 1 incorporates an attacker's power variation and is commensurate with the actual EIRP used, as are the measured RSS reports. The values cancel each other out when computing an HPB distance difference range, yielding constant values for the minimum and maximum bounds of this range, independently of EIRP variations.

Lemma 1 (varying power effect). *Let $\mathbb{R}$ be the set of all receivers within range of an attack message. Let a probable EIRP range $[\mathcal{P}^-, \mathcal{P}^+]$ for this message be computed as set forth in Heuristic 1. Let the distance difference range $[\Delta d_{ij}^-, \Delta d_{ij}^+]$ between a transmitter and receiver pair R_i, R_j be calculated according to* (2) *and* (3). *Then any increase (or decrease) in the EIRP of a subsequent message influences a corresponding proportional increase (or decrease) in RSS reports, effecting no measurable change in the range of distance differences $[\Delta d_{ij}^-, \Delta d_{ij}^+]$ estimated with a dynamically computed EIRP range.*

Proof. Let an original EIRP range $[\mathcal{P}_k^-, \mathcal{P}_k^+]$ computed for all receivers $R_k \in \mathbb{R}$ yield an estimated global EIRP range $[\mathcal{P}^-, \mathcal{P}^+]$. Let a new varying power attack message be transmitted such that the EIRP includes a power increase (or a decrease) of $\Delta\mathcal{P}$. Then for every $R_k \in \mathbb{R}$, the corresponding $\widehat{\text{RSS}}_k$ for the new attack message reflects the same change in value from the original RSS_k, for $\widehat{\text{RSS}}_k = \text{RSS}_k + \Delta\mathcal{P}$. Given new $\widehat{\text{RSS}}_k$ values for all $R_k \in \mathbb{R}$, the resulting EIRP range $[\widehat{\mathcal{P}}^-, \widehat{\mathcal{P}}^+]$ computed with Heuristic 1 includes the same change $\Delta\mathcal{P}$ over the original range of values $[\mathcal{P}^-, \mathcal{P}^+]$:

$$\begin{aligned}
\widehat{\mathcal{P}}^- &= \sup\left\{\widehat{\mathcal{P}}_k^-\right\} \\
&= \sup\left\{\overline{L}(d_0) + 10\eta \log\left(\frac{d_{mk}}{d_0}\right) + \widehat{\text{RSS}}_k - z\sigma\right\} \\
&= \sup\left\{\overline{L}(d_0) + 10\eta \log\left(\frac{d_{mk}}{d_0}\right) + \text{RSS}_k + \Delta\mathcal{P} - z\sigma\right\} \\
&= \sup\left\{\mathcal{P}_k^- + \Delta\mathcal{P}\right\} \\
&= \mathcal{P}^- + \Delta\mathcal{P}.
\end{aligned} \tag{6}$$

Conversely, we see that $\widehat{\mathcal{P}}^+ = \mathcal{P}^+ + \Delta\mathcal{P}$.

As a result, the distance difference range $[\Delta\hat{d}_{ij}^-, \Delta\hat{d}_{ij}^+]$ for the new message is equal to the original range $[\Delta d_{ij}^-, \Delta d_{ij}^+]$:

$$\begin{aligned}
\Delta\hat{d}_{ij}^- &= \left(d_0 \times 10^{(\widehat{\mathcal{P}}^- - \widehat{\text{RSS}}_i - \overline{L}(d_0) - z\sigma)/10\eta}\right) \\
&\quad - \left(d_0 \times 10^{(\widehat{\mathcal{P}}^- - \widehat{\text{RSS}}_j - \overline{L}(d_0) + z\sigma)/10\eta}\right) \\
&= \left(d_0 \times 10^{(\mathcal{P}^- + \Delta\mathcal{P} - \text{RSS}_i - \Delta\mathcal{P} - \overline{L}(d_0) - z\sigma)/10\eta}\right) \\
&\quad - \left(d_0 \times 10^{(\mathcal{P}^- + \Delta\mathcal{P} - \text{RSS}_j - \Delta\mathcal{P} - \overline{L}(d_0) + z\sigma)/10\eta}\right) \\
&= \left(d_0 \times 10^{(\mathcal{P}^- - \text{RSS}_i - \overline{L}(d_0) - z\sigma)/10\eta}\right) \\
&\quad - \left(d_0 \times 10^{(\mathcal{P}^- - \text{RSS}_j - \overline{L}(d_0) + z\sigma)/10\eta}\right) \\
&= \Delta d_{ij}^-.
\end{aligned} \tag{7}$$

The same logic can be used to demonstrate that $\Delta\hat{d}_{ij}^+ = \Delta d_{ij}^+$. □

A varying power attack is thus ineffective against HPB, as the placement of hyperbolic areas remains unchanged.

4.2. HPB Algorithm Variations. The HPB mechanism estimates the originating location of a single attack message from a static snapshot of a wireless network topology. Given sufficient computational efficiency, the algorithm executes in near real time to bound a malicious insider's position at the time of its transmission.

Hyperbolic areas constructed from (2) and (3) are used by HPB to compute a candidate area for the location of a malicious transmitter.

Definition 1 (hyperbolic area). Let $\mathbb{G}$ be the set of all (x, y) coordinates in the Euclidian space within radio range of a malicious transmitter. Let $\mathcal{H}_{ij}^{-}$ be the hyperbola computed from the minimum bound of the distance difference range between receivers R_i and R_j with confidence level $\mathcal{C}$, as defined by (2). Let $\mathcal{H}_{ij}^{+}$ be the hyperbola computed from the maximum bound of the distance difference range between R_i and R_j with the same confidence, as defined by (3). Then we define the hyperbolic area $\mathcal{A}_{ij}$ as situated between the hyperbolas $\mathcal{H}_{ij}^{-}$ and $\mathcal{H}_{ij}^{+}$ with confidence level $\mathcal{C}$. More formally, if $\delta(a, b)$ represents the Euclidian distance between any two points a and b, then

$$\mathcal{A}_{ij} = \left\{ p_k : \Delta d_{ij}^{-} \le \delta(p_k, R_i) - \delta\left(p_k, R_j\right) \le \Delta d_{ij}^{+} \ \forall p_k \in \mathbb{G} \right\} \tag{8}$$

where Δd_{ij}^{-} and Δd_{ij}^{+} are defined in (2) and (3).

A set of hyperbolic areas may be computed according to three different algorithms, depending on the set of receiver pairs considered.

Definition 2 (receiver pair set). Let Ω be any set of unique receivers R_k. Then $\mathcal{S}^{\Omega}$ is defined as the exhaustive set of unique ordered receiver pairs in Ω:

$$\mathcal{S}^{\Omega} = \left\{ \left\{ R_i, R_j \right\} : R_i, R_j \in \Omega,\ i < j \right\}, \tag{9}$$

where $s_h \neq s_k$ for all $s_h, s_k \in \mathcal{S}^{\Omega}$ where $h \neq k$, and $|\mathcal{S}^{\Omega}| = \binom{n}{2}$ where $n = |\Omega|$.

Our original HPB algorithm employs all possible combinations of receiver pairs to compute a set of hyperbolic areas. The intersecting space of the hyperbolic areas yields a probable candidate area for the location of a transmitter.

Algorithm 1 ($\mathbf{A}^{\alpha}$: all-pairs algorithm). The all-pairs algorithm $\mathbf{A}^{\alpha}$ computes hyperbolic areas between every possible pair of receivers. Let $\mathbb{R}$ be the set of all receivers within range of an attack message. Let $\mathcal{S}^{\mathbb{R}}$ represent the set of all unique ordered receiver pairs in $\mathbb{R}$, as put forth in Definition 2. Then the set of hyperbolic areas $\mathbb{H}^{\alpha}$ between all receiver pairs is stated as follows:

$$\mathbb{H}^{\alpha} = \left\{ \mathcal{A}_{ij}, \mathcal{A}_{ji} : \mathcal{A}_{ij}, \mathcal{A}_{ji} \text{ are computed as in Definition 1 for every } \left\{ R_i, R_j \right\} \in \mathcal{S}^{\mathbb{R}} \right\}. \tag{10}$$

The $\mathbf{A}^{\alpha}$ algorithm generates hyperbolic areas for every possible receiver pair, for a total of $\binom{n}{2}$ pairs given n receivers, as put forth in Algorithm 1. While this approach works adequately for four receivers, additional receiving devices have the effect of dramatically increasing computation time as well as reducing the success rate due to the accumulated amount of signal shadowing excluded. The HPB execution time is based on the number of hyperbolic areas computed, which in turn is contingent upon the number of receivers. For $\mathbf{A}^{\alpha}$, n receivers locate a transmitter with a complexity of $\binom{n}{2} = n \times (n-1)/2 \approx O(n^2)$.

An alternate algorithm $\mathbf{A}^{\beta}$ aims to scale down the computational complexity by reducing the number of hyperbolic areas. We separate the set of all receivers into subsets of size r. Each receiver subset computes an intermediate candidate area as the intersection of the hyperbolic areas constructed from all receiver pair combinations within that subset. The final candidate area for a transmitter consists of the intersection of the intermediate candidate areas computed over all receiver subsets.

Algorithm 2 ($\mathbf{A}^{\beta}$: r-pair set algorithm). The *r-pair* set algorithm $\mathbf{A}^{\beta}$ groups receivers in subsets of size r, computes intermediate candidate areas for each subset using the all-pairs approach within the subset, and yields an ultimate candidate area for a transmitter as the intersection of the receiver subset intermediate candidate areas. Let $\mathbb{R}$ be the set of all receivers within range of an attack message. Let Ψ represent the disjoint partition of $(m-1)$ sets of r receivers, with the mth element of Ψ containing the remaining receivers:

$$\Psi = \{\psi_k : \psi_k \subseteq \mathbb{R} \text{ for } 1 \le k \le m,\ |\psi_k| = r \text{ if } k < m,\ 2 \le |\psi_k| \le r \text{ if } k = m\}, \tag{11}$$

where $\psi_h \cap \psi_k = \varnothing$ for all $\psi_h, \psi_k \in \Psi$ with $h \neq k$. Let $\mathcal{S}^{\psi_k}$ represent the set of all unique, ordered receiver pairs in a given set of receivers $\psi_k \in \Psi$, as put forth in Definition 2. Then the set of hyperbolic areas $\mathbb{H}^{\beta}$ computed for sets of r receivers is stated as follows:

$$\mathbb{H}^{\beta} = \left\{ \mathcal{A}_{ij}, \mathcal{A}_{ji} : \mathcal{A}_{ij},\ \mathcal{A}_{ji} \text{ are computed as in Definition 1 for every } \left\{ R_i, R_j \right\} \in \mathcal{S}^{\psi_k}\ \forall \psi_k \in \Psi \right\}. \tag{12}$$

For the $\mathbf{A}^{\beta}$ algorithm, the number of hyperbolic areas depends on the set size r as well as the number of receivers n. Thus $\mathbf{A}^{\beta}$ locates a transmitter with a complexity of $(n/r + 1) \times \binom{r}{2} \approx O(n)$. For a small value of r, for example, $r = 4$, the execution time is proportional to at most $(3n/2 + 6)$.

A third HPB algorithm, the perimeter-pairs variation $\mathbf{A}^{\gamma}$, is proposed to bound the geographic extent of a candidate area within an approximated transmission range, based on the coordinates of the receivers situated farthest from a signal source. We establish a rudimentary perimeter around a transmitter's estimated radio range, with the logical center of this range calculated as the centroid of all receiver coordinates. The range is partitioned into four

quadrants from the center, along two perpendicular axes. Four perimeter receivers are identified as the farthest in each quadrant from the center. Hyperbolic areas are computed between all combinations of perimeter receiver pairs as well as between every remaining nonperimeter receiver and the perimeter receivers in the other three quadrants.

Algorithm 3 ($\mathbf{A}^{\gamma}$: perimeter-pairs algorithm). The perimeter-pairs algorithm $\mathbf{A}^{\gamma}$ partitions a transmitter's radio range into four quadrants. Four perimeter receivers are determined. Hyperbolic areas are computed between all pairs of perimeter receivers, as well as between every perimeter receiver and the nonperimeter receivers of other quadrants. Let $\mathbb{R}$ be the set of all receivers within range of an attack message. Let $R\chi = (x_c, y_c)$ be the centroid of all $R_i \in \mathbb{R}$. Let $\mathbb{Q}$ be the disjoint set of all receivers $R_i \in \mathbb{R}$ partitioned into four quadrants from the centroid $R\chi$:

$$\begin{aligned}\mathbb{Q} = \{Q_k\colon Q_k = \{R_i\colon R_i \in \mathbb{R},\ R_i = (x_i, y_i),\\ x_i \geq x_c,\ y_i \geq y_c \text{ for } k = 1,\\ x_i < x_c,\ y_i \geq y_c \text{ for } k = 2,\\ x_i < x_c,\ y_i < y_c \text{ for } k = 3,\\ x_i \geq x_c,\ y_i < y_c \text{ for } k = 4\}\}.\end{aligned} \tag{13}$$

Let the set $\mathcal{N}$ of perimeter receivers contain one receiver ρ_k for each of the four quadrants, such that ρ_k is the farthest receiver from the centroid $R\chi$ in quadrant k:

$$\begin{aligned}\mathcal{N} = \{\rho_k\colon \rho_k = q_i \text{ such that } q_i \in Q_k,\\ \delta\left(q_i, R\chi\right) \geq \delta\left(q_j, R\chi\right)\ \forall q_j \in Q_k\\ \forall Q_k \in \mathbb{Q}\},\end{aligned} \tag{14}$$

where $\delta(a, b)$ represents the Euclidian distance between any two points a and b. Also let the set of nonperimeter receivers in a given quadrant be determined as all receivers in that quadrant other than the perimeter receiver:

$$\overline{\mathcal{N}} = \left\{\overline{\rho}_k\colon \overline{\rho}_k = \{Q_k \setminus \{\rho_k\}\} \text{ for every } Q_k \in \mathbb{Q}\right\}. \tag{15}$$

Let $\mathcal{S}^{\mathcal{N}}$ represent the set of all unique, ordered perimeter receiver pairs, as put forth in Definition 2. Then the set of hyperbolic areas $\mathbb{H}^{\gamma}$ is stated as follows:

$$\begin{aligned}\mathbb{H}^{\gamma} = \Big\{&\mathcal{A}_{ij}, \mathcal{A}_{ji}\colon \mathcal{A}_{ij},\ \mathcal{A}_{ji} \text{ are computed as in Definition 1}\\ &\text{for every } \left\{R_i, R_j\right\}\\ &\in \left\{\mathcal{S}^{\mathcal{N}} \cup \left\{\left\{R_i, R_j\right\}\colon R_i = \rho_k \text{ for every } \rho_k \in \mathcal{N},\right.\right.\\ &\left.\left.R_j \in \overline{\rho}_m \text{ for every } \overline{\rho}_m \in \overline{\mathcal{N}} \text{ where } m \neq k\right\}\right\}\Big\}.\end{aligned} \tag{16}$$

For example, Figure 1 illustrates a transmitter T and a set of receivers. The grid is partitioned into four quadrants from the computed receiver centroid. The set of perimeter receivers, as the farthest receivers from the centroid in each quadrant (I to IV), form a rudimentary bounding area for the location of the transmitter. The $\mathbf{A}^{\gamma}$ algorithm computes hyperbolic areas between all pairs of perimeter receivers, in this case between all possible pairs in $\mathcal{N} = \{R_3, R_4, R_7, R_5\}$. Additional receiver pairs are formed between the remaining nonperimeter receivers $\{R_1, R_2, R_6, R_8\}$ and the perimeter receivers of other quadrants. Receiver R_6, for instance, is situated in quadrant II, so it is included in a receiver pair with each perimeter receiver in $\{R_3, R_7, R_5\}$.

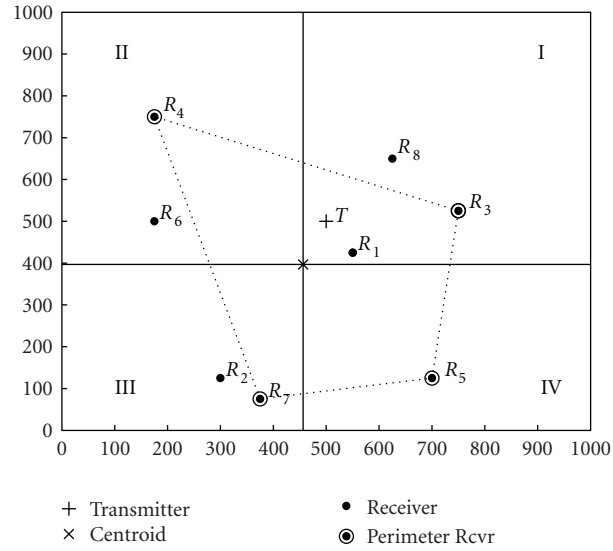


Figure 1: Example of perimeter receivers.

In terms of complexity, the $\mathbf{A}^{\gamma}$ algorithm is equivalent to $\mathbf{A}^{\beta}$. Given n receivers and four perimeter receivers such that $|\mathcal{N}| = 4$, $\mathbf{A}^{\gamma}$ executes in time $\binom{4}{2} + 3(n-4) = 3n - 6 \approx O(n)$.

The candidate area for the location of a malicious transmitter is computed as the intersection of a set of hyperbolic areas, $\mathbb{H}^{\alpha}$, $\mathbb{H}^{\beta}$, or $\mathbb{H}^{\gamma}$, determined according to Algorithms 1, 2, or 3.

Definition 3 (candidate area). Let $\mathbb{G}$ be the set of all (x, y) coordinates in our sample Euclidian space. Let $\mathbb{V} \subseteq \mathbb{G}$ be the subset of all coordinates situated on the road layout of a vehicular scenario. Then the *grid candidate area* GA^{ℓ}, where $\ell \in \{\alpha, \beta, \gamma\}$, is defined as the subset of grid points in $\mathbb{G}$ situated in the intersection of every hyperbolic area computed according to Algorithms $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$, or $\mathbf{A}^{\gamma}$:

$$\begin{aligned}\mathrm{GA}^{\ell} = \Bigg\{&p_k\colon p_k \in \mathbb{G},\ p_k \in \bigcap_{h=1}^{h \leq m} \mathcal{A}_h \in \mathbb{H}^{\ell}\\ &\text{where } \ell \in \{\alpha, \beta, \gamma\},\ m = \left|\mathbb{H}^{\ell}\right|\Bigg\}.\end{aligned} \tag{17}$$

Similarly, the *vehicular candidate area* VA^{ℓ}, where $\ell \in \{\alpha, \beta, \gamma\}$, is defined as the subset of vehicular layout points in $\mathbb{V}$ situated in the intersection of every hyperbolic area computed according to Algorithms $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$, or $\mathbf{A}^{\gamma}$:

$$\begin{aligned}\mathrm{VA}^{\ell} = \Bigg\{&p_k\colon p_k \in \mathbb{V},\ p_k \in \bigcap_{h=1}^{h \leq m} \mathcal{A}_h \in \mathbb{H}^{\ell}\\ &\text{where } \ell \in \{\alpha, \beta, \gamma\},\ m = \left|\mathbb{H}^{\ell}\right|\Bigg\}.\end{aligned} \tag{18}$$

While a candidate area contains a malicious transmitter with probability $\mathcal{C}$, the tracking of a mobile device requires a unique point in Euclidian space to be deemed the likeliest position for the attacker. In free space, we can use the centroid of a candidate area, which is calculated as the average of all the (x, y) coordinates in this area. In a vehicular scenario, we use the road location closest to the candidate area centroid.

Definition 4 (centroids). The grid centroid of a given GA, denoted as $G\chi$, consists of the average (x, y) coordinates of all points within the GA:

$$G\chi = (x_G, y_G), \quad \text{such that } x_G = \frac{\sum_{i=1}^{|\text{GA}|} x_i}{|\text{GA}|}, \quad y_G = \frac{\sum_{i=1}^{|\text{GA}|} y_i}{|\text{GA}|}, \quad \forall\, p_i = (x_i, y_i) \in \text{GA}. \tag{19}$$

The *vehicular centroid* of a given VA, represented as $V\chi$, is the closest vehicular point to the average coordinates of all points within the VA:

$$V\chi = v_k, \quad \text{such that } v_k \in \mathbb{V}, \quad p_h = (x_V, y_V), \quad \text{where } x_V = \frac{\sum_{i=1}^{|\text{VA}|} x_i}{|\text{VA}|}, \quad y_V = \frac{\sum_{i=1}^{|\text{VA}|} y_i}{|\text{VA}|}, \quad \forall\, p_i = (x_i, y_i) \in \text{VA}, \quad \delta(p_h, v_k) \le \delta\left(p_h, v_j\right),\ \forall\, v_j \in \mathbb{V}. \tag{20}$$

4.3. Tracking a Mobile Attacker. We extend HPB to approximate the path followed by a mobile attacker, as it continues transmitting. By computing a new candidate area for each attack message received, a malicious node can be tracked using a set of consecutive candidate positions and the direction of travel inferred between these points. We establish a mobility path in our vehicular scenario as a sequence of vehicular layout (x, y) coordinates over time, along with a mobile transmitter's direction of travel at every point.

Definition 5. A mobility path $\mathbb{P}$ is defined as a set of consecutive coordinates $p_i = (x_i, y_i)$ and angles of travel θ_i over a time interval T:

$$\mathbb{P} = \{\{p_i, \theta_i\} :\ p_i = (x_i, y_i) \text{ is the transmitter location at } t_i \in T,\ \theta_i = \operatorname{atan2}(y_i - y_{i-1}, x_i - x_{i-1})\}, \tag{21}$$

where atan 2 is an inverse tangent function returning values over the range $[-\pi, +\pi]$ to take direction into account (as first defined for the Fortran 77 programming language [25]).

In order to approximate the dynamically changing position of an attacker, we discretize the time domain T into a series of time intervals t_i. At each discrete t_i, we sample a snapshot of the vehicular network topology consisting of a set of receiving devices and their locations. Our approach is analogous to the discretization phase in digital signal processing, where a continuous analog radio signal is sampled periodically for conversion to digital form. We thus estimate the mobility path $\mathbb{P}$ taken by an attacker by executing an HPB algorithm for an attack message received at every interval t_i over a time period T. The vehicular centroids of the resulting candidate areas constitute the estimated attacker positions, and the angle from one estimated point to the next determines the approximated direction of travel.

Algorithm 4 (mobile attacker tracking). Let $\mathcal{M}$ be the set of consecutive attack messages received over a time interval. Then the estimated mobility path $\hat{\mathbb{P}}$ of a transmitter over the message base $\mathcal{M}$ is computed as follows:

$$\hat{\mathbb{P}} = \left\{\left(\hat{p}_i, \hat{\theta}_i\right) \colon \hat{p}_i = (\hat{x}_i, \hat{y}_i) = V\chi_i \text{ for } m_i \in \mathcal{M},\ \hat{\theta}_i = \operatorname{atan2}(\hat{y}_i - \hat{y}_{i-1}, \hat{x}_i - \hat{x}_{i-1})\right\}. \tag{22}$$

For every attack message $m_i \in \mathcal{M}$, an estimated transmitter location $\hat{p}_i$ must be determined. An execution of HPB using the RSS values corresponding to m_i yields a vehicular candidate area VA_i, as put forth in Definition 3. The road centroid of VA_i is computed as $V\chi_i$, according to Definition 4. It is by definition the closest point in the vehicular layout to the averaged center of the VA_i, and thus the natural choice for an estimated value $\hat{p}_i$ of the true transmitter location p_i. The direction of travel of a transmitter is stated in Definition 5 as the angle between consecutive positions in Euclidian space. We follow the same logic to compute the estimated direction of travel $\hat{\theta}_i$ between transmitted messages m_{i-1} and m_i as the angle between the corresponding estimated positions $\hat{p}_{i-1}$ and $\hat{p}_i$.

Example 1. Figure 2 depicts an example mobility path of a malicious insider, with consecutive traveled points labeled from 1 to 20. The transmitter broadcasts an attack message at every fourth location, labeled as points 4, 8, 12, 16 and 20.

For each attack message, we execute the $\mathbf{A}^{\gamma}$ HPB variation, for confidence level $\mathcal{C} = 0.95$, using eight randomly positioned receivers, and a vehicular candidate area VA^{γ} is computed. The estimated locations and directions of travel are depicted in Figure 3. The initial point's direction of travel cannot be estimated, as there is no previous point from which to ascertain a traveled path. In this example, point 4 is localized at 100 meters from its true position, points 8, 16 and 20 at 25 meters, while point 12 is found in its exact location.

5. Performance Evaluation

We describe a simulated vehicular scenario to evaluate the localization and tracking performance of the extended HPB mechanisms described in Section 4.2. In order to model a mobile attacker transmitting at 2.4 GHz, we employ Rappaport's log-normal shadowing model [22] to generate simulated RSS values at a set of receivers, taking into account an independently random amount of signal shadowing experienced at each receiving device. According to Rappaport, the log-normal shadowing model has been used extensively in experimental settings to capture radio signal

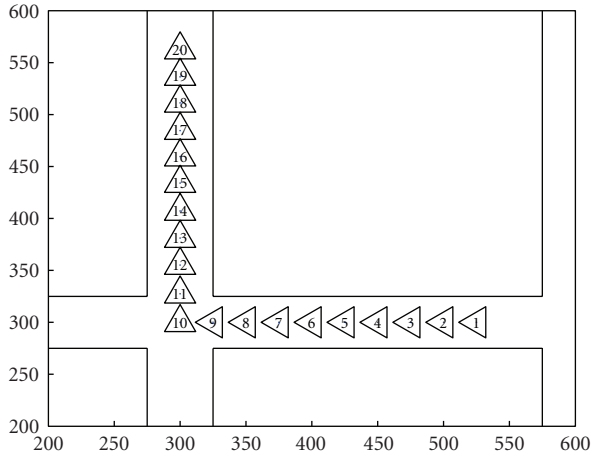


Figure 2: Example of attacker mobility path.

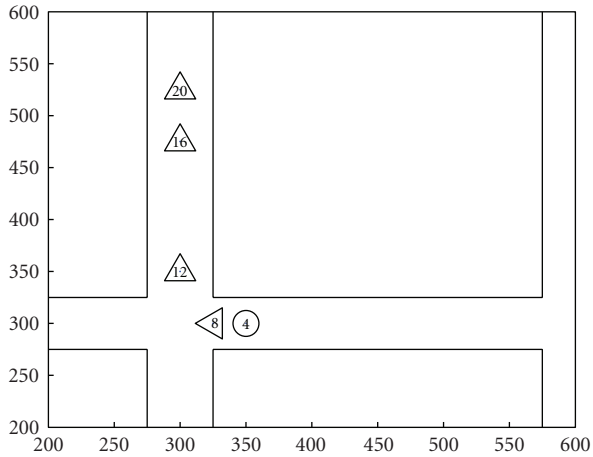


Figure 3: Example of mobile attacker localization.

propagation characteristics, in both indoor and outdoor channels, including in mobility scenarios. In our previous work, we have evaluated HPB results with both log-normal shadowing simulated RSS values and RSS reports harvested from an outdoor field experiment at 2.4 GHz [9]. We found that the simulated and experimental location estimation results are nearly identical, indicating that at this frequency, the log-normal shadowing model is an appropriate tool for generating realistic RSS values.

We compare the success rates of the $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\gamma}$ algorithms at estimating a malicious transmitter's location within a candidate area, as well as the relative sizes of the grid and vehicular candidate areas. We model a mobile transmitter's path through a vehicular scenario and assess the success in tracking it by measuring the distance between the actual and estimated positions, in addition to the difference between the approximated direction of travel and the real one.

5.1. Hyperbolic Position Bounding of Vehicular Devices. Our simulation uses a one square kilometer urban grid, as depicted in Figure 4. We evaluate the all-pairs $\mathbf{A}^{\alpha}$, 4-pair set $\mathbf{A}^{\beta}$ and perimeter-pairs $\mathbf{A}^{\gamma}$ HPB algorithms with four, eight, 16 and 32 receivers. In each HPB execution, four of the receivers are fixed road-side units (RSUs) stationed at intersections. The remaining receivers are randomly positioned on-board units (OBUs), distributed uniformly on the grid streets. Every HPB execution also sees a transmitter placed at a random road position within the inner square of the simulation grid. We assume that in a sufficiently dense urban setting, RSUs are positioned at most intersections. As a result, any transmitter location is geographically surrounded by four RSUs within radio range. For each defined number of receivers and two separate confidence levels $\mathcal{C} \in \{0.95, 0.90\}$, the HPB algorithms, $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\gamma}$, are executed 1000 times. For every execution, RSS values are generated for each receiver from the log-normal shadowing model. We adopt existing experimental path loss parameter values from large-scale measurements gathered at 2.4 GHz by Liechty et al. [26, 27]. From $\eta = 2.76$ and a signal shadowing standard deviation $\sigma = 5.62$, we augment the simulated RSS values with an independently generated amount of random shadowing to every receiver in a given HPB execution. Since the EIRP used by a malicious transmitter is unknown, a probable range is computed according to Heuristic 1.

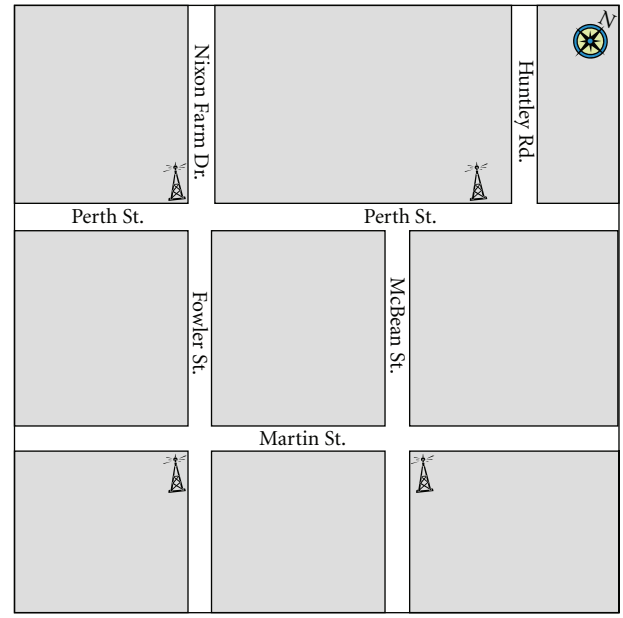


Figure 4: Urban scenario—Richmond, Ontario.

For every HPB execution, whether the $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$ or $\mathbf{A}^{\gamma}$ algorithm is used, we gather three metrics: the success rate in localizing the transmitter within a computed candidate area GA; the size of the unconstrained candidate area GA as a percentage of the one square kilometer grid; the size of the candidate area restricted to the vehicular layout VA as a percentage of the grid. The success rate and candidate area size results we obtain are deemed 90% accurate within a 2% and 0.8% confidence interval, respectively. The average HPB execution times for each algorithm on an HP Pavilion laptop with an AMD Turion 64×2 dual-core processor are shown in Table 1. As expected from our complexity analysis, the $\mathbf{A}^{\alpha}$

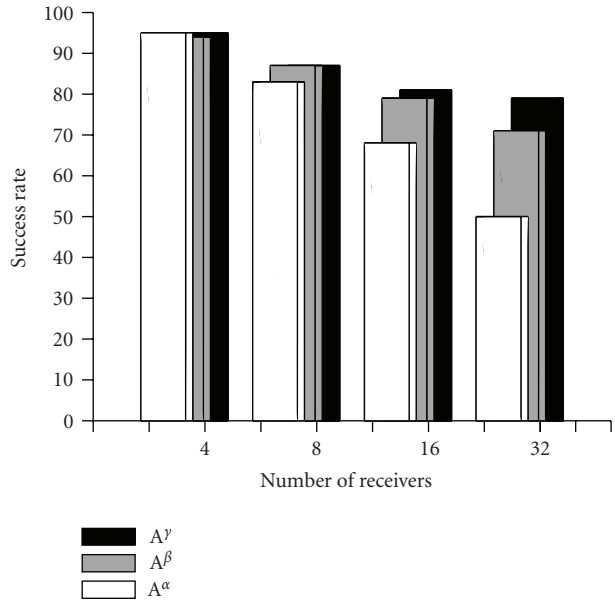


Figure 5: Success rate for $\mathcal{C} = 0.95$.

Table 1: Average HPB execution time (seconds).

# Rcvrs	$\mathbf{A}^{\gamma}$		$\mathbf{A}^{\beta}$		$\mathbf{A}^{\alpha}$	
	Mean	Std dev.	Mean	Std dev.	Mean	Std dev.
4	0.005	0.000	0.023	0.001	0.023	0.001
8	0.023	0.001	0.045	0.001	0.104	0.003
16	0.075	0.001	0.090	0.002	0.486	0.142
32	0.215	0.059	0.195	0.053	2.230	0.766

variation is markedly slower, and the computational costs increase as additional receivers participate in the location estimation effort. For example in the case of eight receivers, a single execution of $\mathbf{A}^{\gamma}$ takes 23 milliseconds, while $\mathbf{A}^{\alpha}$ requires over 100 milliseconds.

The comparative success rates of the $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\gamma}$ approaches are illustrated in Figure 5, for confidence level $\mathcal{C} = 0.95$. While $\mathbf{A}^{\gamma}$ exhibits the best localization success rate, every algorithm sees its performance degrade as more receivers are included. With four receivers for example, all three variations successfully localize a transmitter 94-95% of the time. However with 32 receivers, $\mathbf{A}^{\gamma}$ succeeds in 79% of the cases, while $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\alpha}$ do so in 71% and 50% of executions. Given that each receiver pair takes into account an amount of signal shadowing based on the confidence level $\mathcal{C}$, it also probabilistically ignores a portion $(1 - \mathcal{C})$ of the shadowing. As more receivers and thus more receiver pairs are added, the error due to excluded shadowing accumulates. The results obtained for confidence level $\mathcal{C} = 0.90$ follow the same trend, although the success rates are slightly lower.

Figures 6 and 7 show the grid and vehicular candidate area sizes associated with our simulation scenario, as computed with algorithms $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\gamma}$, for confidence level $\mathcal{C} = 0.95$. The size of the grid candidate area GA corresponds to 21% of the simulation grid, with four receivers, for both $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\alpha}$, while $\mathbf{A}^{\gamma}$ narrows the area to only 7%. In fact, the $\mathbf{A}^{\gamma}$ approach exhibits a GA size that is independent of the number of receivers. Yet for $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\alpha}$, the GA size is noticeably lower with additional receivers. This finding reflects the use of perimeter receivers with $\mathbf{A}^{\gamma}$. These specialized receivers serve to restrict the GA to a particular portion of the simulation grid, even with few receivers. However, this variation does not fully exploit the presence of additional receiving devices, as these only support the GA determined by the perimeter receivers. The size of the vehicular candidate area VA follows the same trend, with a near constant size of 0.64% to 1% of the grid for $\mathbf{A}^{\gamma}$, corresponding to a localization granularity within an area less than 100 m $\times$ 100 m, assuming the transmitter is aboard a vehicle traveling on a road. The $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\alpha}$ algorithms compute vehicular candidate area sizes that decrease as more receivers are taken into account, with $\mathbf{A}^{\alpha}$ yielding the best localization granularity. But even with four receivers, $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\alpha}$ localize a transmitter within a vehicular layout area of 1.6% of the grid, or 125 m $\times$ 125 m.

Generally, both the GA and VA sizes decrease as the number of receivers increases, since additional hyperbolic areas pose a higher number of constraints on a candidate area, thus decreasing its extent. We see in Figures 6 and 7 that $\mathbf{A}^{\beta}$ consistently yields larger candidate areas than $\mathbf{A}^{\alpha}$ for the same reason, as $\mathbf{A}^{\alpha}$ generates a significantly greater number of hyperbolic areas. For example, while $\mathbf{A}^{\alpha}$ computes an average GA^{α} of 10% and 3% of the simulation grid with eight and 16 receivers, $\mathbf{A}^{\beta}$ yields areas of 15% and 9%, respectively. By contrast, $\mathbf{A}^{\gamma}$ yields a GA size of 5-6% but its reliability is greater, as demonstrated by the higher success rates achieved. The nearly constant 5% GA size computed with $\mathbf{A}^{\gamma}$ has an average success rate of 81% for 16 receivers, while the 9% GA generated by $\mathbf{A}^{\beta}$ is 79% reliable and the 3% GA obtained with $\mathbf{A}^{\alpha}$ features a dismal 68% success rate. Indeed, Figures 5 and 6 taken together indicate that smaller candidate areas provide increased granularity at the cost of lower success rates, and thus decreased reliability. This phenomenon is consistent with the intuitive expectation that a smaller area is less likely to contain the transmitter.

5.2. Tracking a Vehicular Device. We generate 1000 attacker mobility paths $\mathbb{P}$, as stipulated in Definition 5, of 20 consecutive points evenly spaced at every 25 meters. Each path begins at a random start location along the central square of the simulation grid depicted in Figure 4. We keep the simulated transmitter location within the area covered by four fixed RSUs, presuming that an infinite grid features at least four RSUs within radio range of a transmitter. The direction of travel for the start location is determined randomly. Each subsequent point in the mobile path is contiguous to the previous point, along the direction of travel. Upon reaching an intersection in the simulation grid, a direction of travel is chosen randomly among the ones available from the current position, excluding the reverse direction.

The $\mathbf{A}^{\alpha}$, $\mathbf{A}^{\beta}$ and $\mathbf{A}^{\gamma}$ algorithms are executed at every fourth point p_i of each mobility path $\mathbb{P}$, corresponding to a transmitted attack signal at every 100 meters. The algorithms

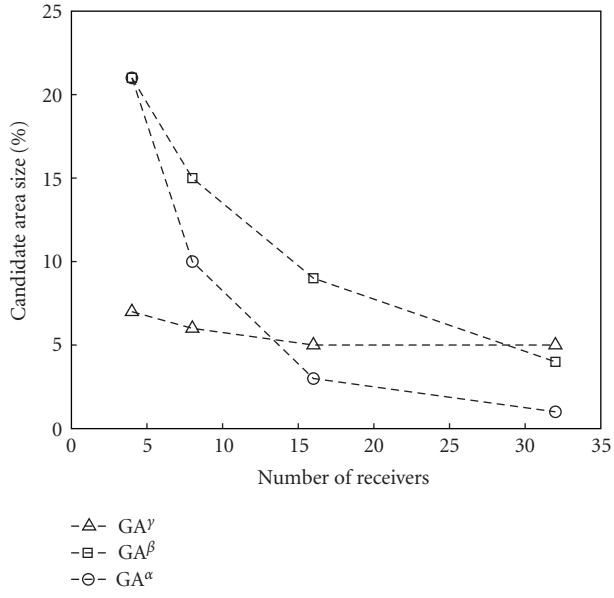


Figure 6: Grid candidate area size for $\mathcal{C} = 0.95$.

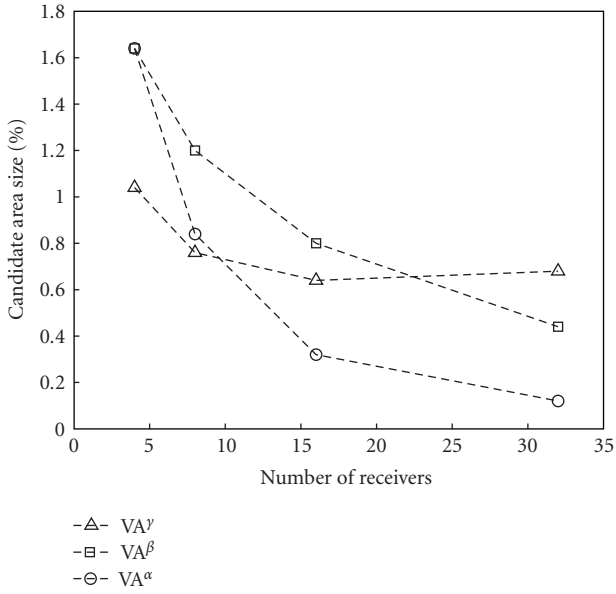


Figure 7: Vehicular candidate area size for $\mathcal{C} = 0.95$.

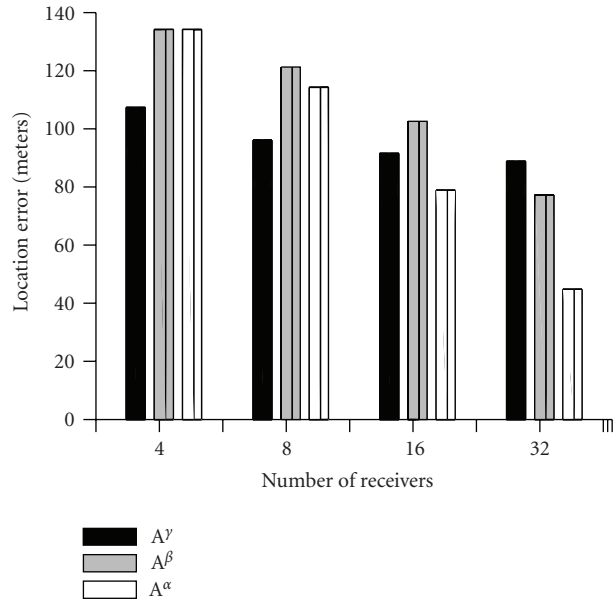


Figure 8: Location error for $\mathcal{C} = 0.95$.

are executed for confidence levels $\mathcal{C} \in \{0.95, 0.90\}$, with each of four, eight, 16 and 32 receivers. In every case, the receivers consist of four static RSUs, and the remaining are OBUs randomly placed at any point on the simulated roads.

For each execution of $\mathbf{A}^\alpha$, $\mathbf{A}^\beta$ and $\mathbf{A}^\gamma$, a vehicular candidate area VA is computed, and its centroid $V\chi$ is taken as the probable location of the transmitter, as described in Algorithm 4. Two metrics are aggregated over the executions: the root mean square *location error*, as the distance in meters between the actual transmitter location p_i and its estimated position $\hat{p}_i = V\chi_i$; and the root mean square *angle error* between the angle of travel θ_i for each consecutive actual transmitter location and the angle $\hat{\theta}_i$ computed for the approximated locations.

The location error for the $\mathbf{A}^\alpha$, $\mathbf{A}^\beta$ and $\mathbf{A}^\gamma$ algorithms, given confidence level $\mathcal{C} = 0.95$, is illustrated in Figure 8. As expected, the smaller VA sizes achieved with a greater number of receivers for $\mathbf{A}^\alpha$ and $\mathbf{A}^\beta$ correspond to a more precise transmitter localization. The location error associated with the $\mathbf{A}^\alpha$ algorithm is smaller, compared to $\mathbf{A}^\beta$, for the same reason. Correspondingly, the nearly constant VA size obtained with $\mathbf{A}^\gamma$ yields a similar result for the location error. For instance with confidence level $\mathcal{C} = 0.95$, eight and 16 receivers produce a location error of 114 and 79 meters, respectively, with $\mathbf{A}^\alpha$ but of 121 and 102 meters with $\mathbf{A}^\beta$. The location error with $\mathbf{A}^\gamma$ is once more nearly constant, at 96 and 91 meters. The use of all receiver pairs to compute a VA with $\mathbf{A}^\alpha$ allows for localization that is up to 40–50% more precise than grouping the receivers in sets of four or relying on perimeter receivers when 16 or 32 receiving devices are present. Despite its granular localization performance, the $\mathbf{A}^\alpha$ approach works best with large numbers of receivers, which may not consistently be realistic in a practical setting. Another important disadvantage of the $\mathbf{A}^\alpha$ approach lies in its large complexity of $O(n^2)$ for n receivers, when compared to $\mathbf{A}^\beta$ and $\mathbf{A}^\gamma$ with a complexity of $O(n)$, as discussed in Section 4.2.

Figure 9 plots the root mean square location error in terms of VA size for the three algorithms. While $\mathbf{A}^\alpha$ and $\mathbf{A}^\beta$ yield smaller VAs for a large number of receivers, the VAs computed with $\mathbf{A}^\gamma$ offer more precise localization with respect to their size. For example, a 0.7% VA size obtained with $\mathbf{A}^\gamma$ features a 96 meter location error, while a similar size VA computed with $\mathbf{A}^\beta$ and $\mathbf{A}^\alpha$ generates a 102 and 114 meter location error, respectively.

The error in estimating the direction of travel exhibits little variation in terms of number of receivers and choice

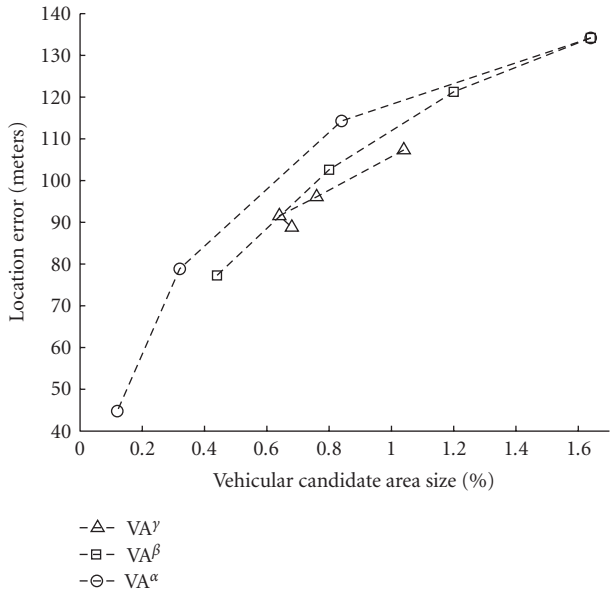


Figure 9: Location error for vehicular candidate area size.

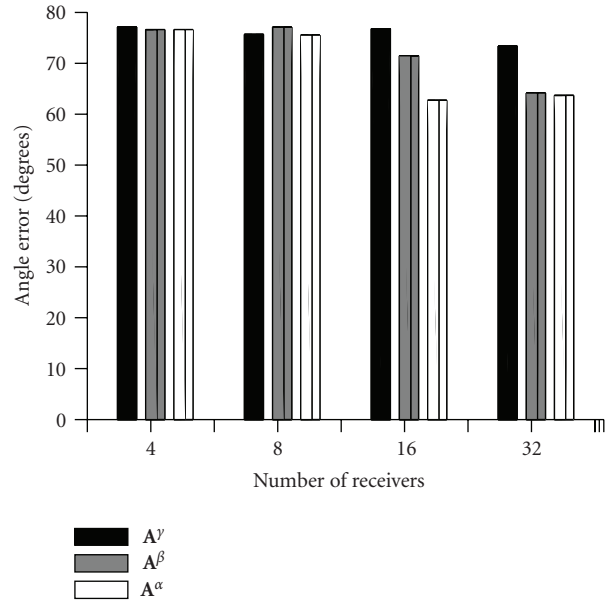


Figure 10: Direction of travel angle error for $\mathcal{C} = 0.95$.

of HPB algorithm, as shown in Figure 10. With eight and 16 receivers, for confidence level $\mathcal{C} = 0.95$, $\mathbf{A}^{\beta}$ approximates the angle of travel between two consecutive points within 77° and 71°, respectively, whereas $\mathbf{A}^{\alpha}$ estimates it within 76° and 63°. $\mathbf{A}^{\gamma}$ exhibits a slightly higher direction error at 76° and 77°. It should be noted that for all three algorithms, for all numbers of receivers, the range of angle errors only spans 14°. So while the granularity of localization is contingent upon the HPB methodology used and the number of receivers, the three variations perform similarly in estimating the general direction of travel.

6. Discussion

The location error results of Figure 8 shed an interesting light on the HPB success rates discussed in Section 5.1. For example in the presence of 32 receivers, for confidence level $\mathcal{C} = 0.95$, only 50% of $\mathbf{A}^{\alpha}$ executions yield a candidate area containing a malicious transmitter, as shown in Figure 5. Yet the same scenario localizes a transmitter with a root mean square location error of 45 meters of its true location, whether it lies within the corresponding candidate area or not. This indicates that while a candidate area may be computed in the wrong position, it is in fact rarely far from the correct transmitter location. This may be a result of our strict definition of a successful execution, where only a candidate area in the intersection of all hyperbolic areas is considered. We have observed in our simulations that a candidate area may be erroneous solely because of a single misplaced hyperbolic area, which results in either a wrong location or an empty candidate area. In our simulations tracking a mobile attacker, we notice that while $\mathbf{A}^{\gamma}$ and $\mathbf{A}^{\beta}$ generate an empty VA for 10% and 14% of executions, $\mathbf{A}^{\alpha}$ does so in 31% of the cases. This phenomenon is likely due to the greater number of hyperbolic areas generated with the $\mathbf{A}^{\alpha}$ approach and the subsequent greater likelihood of erroneously situated hyperbolic areas. While the success rates depicted in Figure 5 omit the executions yielding empty candidate areas as inconclusive, future work includes devising a heuristic to recompute a set of hyperbolic areas in the case where their common intersection is empty.

In comparing the location accuracy of HPB with related technologies, we find that, for example, differential GPS devices can achieve less than 10 meter accuracy. However, this technology is better suited to self-localization efforts relying on a device's assistance and cannot be depended upon for the position estimation of a noncooperative adversary. The FCC has set forth regulations for the network-based localization of wireless handsets in emergency 911 call situations. Service providers are expected to locate a calling device within 100 meters 67% of the time and within 300 meters in 95% of cases [28]. In the minimalist case involving four receivers, the HPB perimeter-pairs variation $\mathbf{A}^{\gamma}$ localizes a transmitting device with a root mean square location error of 107 meters. This translates into a location accuracy of 210 meters in 95% of cases and of 104 meters in 67% of executions. While the former case is fully within FCC guidelines, the latter is very close. With a larger number of receivers, for example, eight receiving devices, $\mathbf{A}^{\gamma}$ yields an accuracy of 188 meters 95% of the time and of 93 meters in 67% of cases. Although HPB is designed for the location estimation of a malicious insider, its use may be extended to additional applications such as 911 call origin localization, given that its performance closely matches the FCC requirements for emergency services.

7. Conclusion

We extend a hyperbolic position bounding (HPB) mechanism to localize the originator of an attack signal within a vehicular network. Because of our novel assumption that

the message EIRP is unknown, the HPB location estimation approach is suitable to security scenarios involving malicious or uncooperative devices, including insider attacks. Any countermeasure to this type of exploit must feature minimalist assumptions regarding the type of radio equipment used by an attacker and expect no cooperation with localization efforts on the part of a perpetrator.

We devise two additional HPB-based approaches to compute hyperbolic areas between pairs of trusted receivers by grouping them in sets and establishing perimeter receivers. We demonstrate that due to the dynamic computation of a probable EIRP range utilized by an attacker, our HPB algorithms are impervious to varying power attacks. We extend the HPB algorithms to track the location of a mobile attacker transmitting along a traveled path.

The performance of all three HPB variations is evaluated in a vehicular scenario. We find that the grouped receivers method yields a localization success rate up to 11% higher for a 6% increase in candidate area size over the all-pairs approach. We also observe that the perimeter-pairs algorithm provides a more constant candidate area size, independently of the number of receivers, for a success rate up to 13% higher for a 2% increase in candidate area size over the all-pairs variation. We conclude that the original HPB mechanism using all pairs of receivers produces a smaller localization error than the other two approaches, when a large number of receiving devices are available. We observe that for a confidence level of 95%, the former approach localizes a mobile transmitter with a granularity as low as 45 meters, up to 40–50% more precisely than the grouped receivers and perimeter-pairs methods. However, the computational complexity of the all-pairs variation is significantly greater, and its performance with fewer receivers is less granular than the perimeter-pairs method. Of the two approaches with complexity $O(n)$, the perimeter-pairs method yields a success rate up to 8% higher for consistently smaller candidate area sizes, location, and direction errors.

In a vehicular scenario, we achieve a root mean square location error of 107 meters with four receivers and of 96 meters with eight receiving devices. This granularity is sufficient to satisfy the FCC-mandated location accuracy regulations for emergency 911 services. Our HPB mechanism may therefore be adaptable to a wide range of applications involving network-based device localization assuming neither target node cooperation nor knowledge of the EIRP.

We have demonstrated the suitability of the hyperbolic position bounding mechanism for estimating the candidate location of a vehicular network malicious insider and for tracking such a device as it moves throughout the network. Future research is required to assess the applicability of the HPB localization and tracking mechanisms in additional types of wireless and mobile technologies, including wireless access networks such as WiMAX/802.16.

Acknowledgments

The authors gratefully acknowledge the financial support received for this research from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Automobile of the 21st Century (AUTO21) Network of Centers of Excellence (NCE).

References

[1] IEEE Intelligent Transportation Systems Committee, "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages," IEEE Std 1609.2-2006, July 2006.

[2] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov, "Cryptographic processors—a survey," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 357–369, 2006.

[3] R. Anderson and M. Kuhn, "Tamper resistance: a cautionary note," in *Proceedings of the 2nd USENIX Workshop on Electronic Commerce*, pp. 1–11, Oakland, Calif, USA, November 1996.

[4] National Institute of Standards and Technology, "Security Requirements for Cryptographic Modules," Federal Information Processing Standards 140-2, NIST, May 2001.

[5] IBM, "IBM 4764 PCI-X Cryptographic Coprocessor," http://www.ibm.com.

[6] D. E. Williams, "A Concept for Universal Identification," White paper, SANS Institute, December 2001.

[7] SeVeCom, "Security architecture and mechanisms for V2V/V2I, deliverable 2.1," Tech. Rep. D2.1, Secure Vehicle Communication, Paris, France, August 2007, edited by Antonio Kung.

[8] C. Laurendeau and M. Barbeau, "Insider attack attribution using signal strength-based hyperbolic location estimation," *Security and Communication Networks*, vol. 1, no. 4, pp. 337–349, 2008.

[9] C. Laurendeau and M. Barbeau, "Hyperbolic location estimation of malicious nodes in mobile WiFi/802.11 networks," in *Proceedings of the 2nd IEEE LCN Workshop on User MObility and VEhicular Networks (ON-MOVE '08)*, pp. 600–607, Montreal, Canada, October 2008.

[10] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular ad hoc networks: a new challenge for localization-based systems," *Computer Communications*, vol. 31, no. 12, pp. 2838–2849, 2008.

[11] R. Parker and S. Valaee, "Vehicular node localization using received-signal-strength indicator," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, part 1, pp. 3371–3380, 2007.

[12] J.-P. Hubaux, S. Čapkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security & Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[13] S. Čapkun and J.-P. Hubaux, "Secure positioning in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 221–232, 2006.

[14] S. Brands and D. Chaum, "Distance-bounding protocols," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT '94)*, vol. 765 of *Lecture Notes in Computer Science*, pp. 344–359, Springer, Perugia, Italy, May 1994.

[15] B. Xiao, B. Yu, and C. Gao, "Detection and localization of sybil nodes in VANETs," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '06)*, pp. 1–8, Los Angeles, Calif, USA, September 2006.

[16] T. Leinmüller, E. Schoch, and F. Kargl, "Position verification approaches for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 16–21, 2006.

[17] J. R. Douceur, "The Sybil attack," in *Peer-to-Peer Systems*, vol. 2429 of *Lecture Notes in Computer Science*, pp. 251–260, Springer, Berlin, Germany, 2002.

[18] L. Tang, X. Hong, and P. G. Bradford, "Privacy-preserving secure relative localization in vehicular networks," *Security and Communication Networks*, vol. 1, no. 3, pp. 195–204, 2008.

[19] G. Yan, S. Olariu, and M. C. Weigle, "Providing VANET security through active position detection," *Computer Communications*, vol. 31, no. 12, pp. 2883–2897, 2008.

[20] N. Mirmotahhary, A. Kohansal, H. Zamiri-Jafarian, and M. Mirsalehi, "Discrete mobile user tracking algorithm via velocity estimation for microcellular urban environment," in *Proceedings of the 67th IEEE Vehicular Technology Conference (VTC '08)*, pp. 2631–2635, Singapore, May 2008.

[21] Z. R. Zaidi and B. L. Mark, "Real-time mobility tracking algorithms for cellular networks based on Kalman filtering," *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 195–208, 2005.

[22] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[23] C. Laurendeau and M. Barbeau, "Probabilistic evidence aggregation for malicious node position bounding in wireless networks," *Journal of Networks*, vol. 4, no. 1, pp. 9–18, 2009.

[24] Y. Chen, K. Kleisouris, X. Li, W. Trappe, and R. P. Martin, "The robustness of localization algorithms to signal strength attacks: a comparative study," in *Proceedings of the 2nd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '06)*, vol. 4026 of *Lecture Notes in Computer Science*, pp. 546–563, Springer, San Francisco, Calif, USA, June 2006.

[25] American National Standards Institute, "Programming Language FORTRAN," ANSI Standard X3.9-1978, 1978.

[26] L. C. Liechty, *Path loss measurements and model analysis of a 2.4 GHz wireless network in an outdoor environment*, M.S. thesis, Georgia Institute of Technology, Atlanta, Ga, USA, August 2007.

[27] L. C. Liechty, E. Reifsnider, and G. Durgin, "Developing the best 2.4 GHz propagation model from active network measurements," in *Proceedings of the 66th IEEE Vehicular Technology Conference (VTC '07)*, pp. 894–896, Baltimore, Md, USA, September-October 2007.

[28] Federal Communications Commission, 911 Service, FCC Code of Federal Regulations, Title 47, Part 20, Section 20.18, October 2007.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 427492, 12 pages
doi:10.1155/2009/427492

Research Article

In Situ Key Establishment in Large-Scale Sensor Networks

Yingchang Xiang,[1] Fang Liu,[2] Xiuzhen Cheng,[3] Dechang Chen,[4] and David H. C. Du[5]

[1] *Department of Basic Courses, Rizhao Polytechnic College, Rizhao, Shandong 276826, China*
[2] *Department of Computer Science, University of Texas - Pan American, Edinburg, Texas 78539, USA*
[3] *Department of Computer Science, The George Washington University, Washington, DC, 20052, USA*
[4] *Department of Preventive Medicine and Biometrics, Uniformed Services University of the Health Sciences, Bethesda, MD 20817, USA*
[5] *Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota, USA*

Correspondence should be addressed to Xiuzhen Cheng, cheng@gwu.edu

Received 1 January 2009; Accepted 11 April 2009

Recommended by Yang Xiao

Due to its efficiency, symmetric key cryptography is very attractive in sensor networks. A number of key predistribution schemes have been proposed, but the scalability is often constrained by the unavailability of topology information before deployment and the limited storage budget within sensors. To overcome this problem, three in-situ key establishment schemes, SBK, LKE, and iPAK, have been proposed. These schemes require no preloaded keying information but let sensors compute pairwise keys after deployment. In this paper, we propose an in-situ key establishment framework of which iPAK, SBK, and LKE represent different instantiations. We further compare the performance of these schemes in terms of scalability, connectivity, storage, and resilience. Our simulation results indicate that all the three schemes scale well to large sensor networks. We also notice that SBK outperforms LKE and LKE outperforms iPAK with respect to topology adaptability. Finally, observing that iPAK, SBK, and LKE all rely on the key space models that involve computationally intensive modular operations, we propose an improvement that rely on random keys that can be easily computed from a secure pseudorandom function. This new approach requires no computation overhead at regular worker sensors, therefore has a high potential to conserve the network resource.

1. Introduction

Secure communication is a critical requirement for many sensor network applications. Nevertheless, the constrained capabilities of smart sensors (battery supply, CPU, memory, etc.) and the harsh deployment environment of a sensor network (infrastructureless, wireless, ad hoc, etc.) make this problem very challenging. A secure sensor network requires a "sound" key establishment scheme that should be easily realized by individual sensors, should be localized to scale well to large sensor networks, should require small amount of space for keying information storage, and should be resilient against node capture attacks.

Symmetric key cryptography is attractive and applicable in sensor networks because it is computationally efficient. As reported by Carman et. al [1], a middle-ranged processor such as the Motorola MC68328 "DragonBall" consumes 42 mJ (840 mJ) for RSA encryption (digital signature) and 0.104 mJ for AES when the key size for both cases is 1024 bits. Therefore establishing a shared key for pairwise communication becomes a central problem for sensor network security research. Ever since the pioneer work on key predistribution by Eschenauer and Gligor [2] in the year 2002, a variety of key establishment schemes have been reported, as illustrated in Figure 1.

Key predistribution is motivated by the observation that no topology information is available before deployment. The two extreme cases are the *single master key scheme*, which preloads a master key to all sensors, and the *all pairwise keys scheme*, which preloads a unique key for each pair of sensors. The first one is weak in resilience while the second one has a high storage overhead. Other than these two extreme cases there exist a number of probabilistic-based key predistribution schemes [2–11], which attract

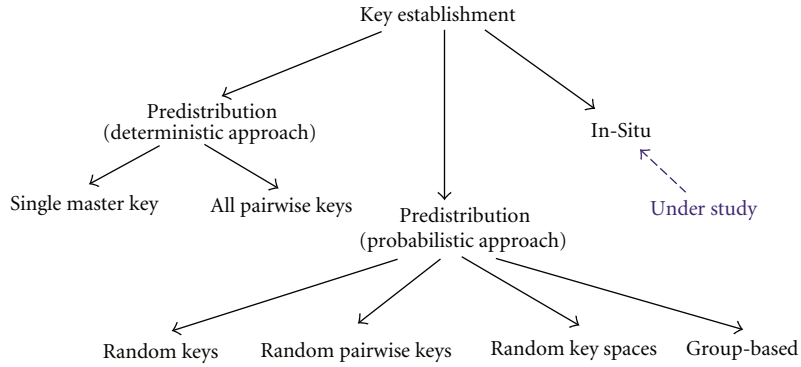


Figure 1: Existing Key Establishment Schemes - A Taxonomy.

most of the research interests in securing sensor networks. The probabilistic-based schemes require each sensor to preload keying information such that two neighboring sensors compute a shared key after exchanging part of the stored information after deployment. Generally speaking, the larger the amount of keying information stored within each sensor, the better the connectivity of the key-sharing graph, the higher the computation and communication overheads. A major drawback of the schemes in this category is the storage space wastage since a large amount of keying information may never be utilized during the lifetime of a sensor. Consequently, the scalability of key predistribution is poor, since the amount of required security information to be preloaded increases with the network size. Furthermore, many of the probabilistic-based approaches bear poor resilience as the compromise of any sensors could release the pairwise key used to protect the communications between two uncompromised sensors. In summary, probabilistic-based key predistribution could not achieve good performance in terms of scalability, storage overhead, key-sharing probability, and resilience simultaneously.

Recently, three in-situ key establishment schemes, iPAK [12], SBK [13] and LKE [14], have been proposed for the purpose of overcoming all the problems faced by key predistribution. Schemes in this category require no keying information to be predistributed, while sensors compute shared keys with their neighbors after deployment. The basic idea is to utilize a small number of service sensors as sacrifices for disseminating keying information to worker sensors in the vicinity. Worker sensors are in charge of normal network operations such as sensing and reporting. Two worker sensors can derive a common key once they obtain keying information from the same service sensor. In this paper, we first propose the in-situ key establishment framework, of which iPAK, SBK, and LKE represent different instantiations. Then we report our comparison study on the performance of these three schemes in terms of *scalability, connectivity, storage overhead* and *resilience*. Our results indicate that all the three in-situ schemes scale well to large sensor networks as they require only local information. Furthermore, we also notice that SBK outperforms LKE and LKE outperforms iPAK with respect to topology adaptability. Finally, observing that iPAK, SBK, and LKE all rely on the key space models that involve intensive computation overhead, we propose an improvement that rely on random keys that could be easily generated by a secure pseudorandom function.

This paper is organized as follows. Major key predistribution schemes are summarized in Section 2. Preliminaries, models, and assumptions are sketched in Section 3. The in-situ key establishment framework is introduced in Section 4, and the three instantiations are outlined in Section 5. Performance evaluation and comparison study are reported in Section 6. Finally, we summarize our work and discuss the future research in Section 7.

2. Related Work: Key Predistribution

In this section, major related works on key predistribution are summarized and compared. We refer the readers to [10, 15] for a more comprehensive literature survey.

The basic *random keys scheme* is proposed by Eschenauer and Gligor in [2], in which a large key pool $\mathcal{K}$ is computed offline and each sensor picks m keys randomly from $\mathcal{K}$ without replacement before deployment. Two sensors can establish a shared key as long as they have at least one key in common. To enhance the security of the basic scheme in against small-scale attacks, Chan et al. [3] propose the q-composite keys scheme in which $q > 1$ number of common keys are required for two nodes to establish a shared key. This scheme performs worse in resilience when the number of compromised sensors is large.

In these two schemes [2, 3], increasing the number of compromised sensors increases the percentage of compromised links shared by uncompromised sensors. To overcome this problem, in the same work Chan et al. [3] propose to boost up a unique key for each link through multi-path enhancement. For the same purpose, Zhu et al. [16] propose to utilize multiple logic paths. To improve the efficiency of key discovery in [2, 3], which is realized by exchanging the identifiers of the stored keys, or by a challenge-response procedure, Zhu et al. [16] propose an approach based on the pseudo-random key generator seeded by the node id. Each sensor computes the key identifiers and preloads the corresponding keys based on its unique id. Two sensors can determine whether they have a common key based on their ids only. Note that this procedure does not improve the

security of the key discovery procedure since an attacker can still Figure out the key identifiers as long as the algorithm is available. Further, a smart attacker can easily beat the pseudo-random key generator to compromise the network faster [17]. Actually for smart attackers, challenge-response is an effective way for key discovery but it is too computationally intensive. Di Pietro et al. [17] propose a pseudo-random key predeployment scheme that supports a key discovery procedure that is as efficient as the pseudo-random key generator [16] and as secure as challenge-response.

To improve the resilience of the random keys scheme in against node capture attacks, *random pairwise keys schemes* have been proposed [3, 4], in which a key is shared by two sensors only. These schemes have good resilience against node capture attacks since the compromise of a sensor only affects the links incident to that sensor. The difference between [3] and [4] is that sensors in [3] are paired based on ids while in [4] are on virtual grid locations. Similar to the random keys schemes, random pairwise keys schemes do not scale well to large sensor networks. Neither do they have good key-sharing probability due to the conflict between the high keying storage redundancy requirement and the memory constraint.

To improve the scalability of the random keys schemes, two *random key spaces schemes* [5, 7] have been proposed independently at ACM CCS 2003. These two works are similar in nature, except that they apply different key space models, which will be summarized in Subsection 3.1. Each sensor preloads several keying shares, with each belonging to one key space. Two sensors can establish a shared key if they have keying information from the same key space. References [7] also proposes to assign one key space to each row or each column of a virtual grid. A sensor residing at a grid point receives keying information from exactly two key spaces. This realization involves more number of key spaces. Note that these random key spaces schemes also improve resilience and key-sharing probability because more key spaces are available, and because two sensors compute a unique key within one key space for their shared links.

Compared to the works mentioned above, *group-based schemes* [6, 8, 9, 11] have the best performance in scalability, key-sharing probability, storage, and resilience due to the relatively less randomness involved in these key predistribution schemes. Du et al. scheme [6] is the first to apply the group concept, in which sensors are grouped before deployment and each group is dropped at one deployment point. Correspondingly, a large key pool $\mathcal{K}$ is divided into subkey spaces, with each associated with one group of sensors. Subkey spaces overlap if the corresponding deployment points are adjacent. Such a scheme ensures that close-by sensors have a higher chance to establish a pairwise key directly. But the strong assumption on the deployment knowledge (static deployment point) renders it impractical for many applications. Also relying on deployment knowledge, the scheme proposed by Yu and Guan in [9] significantly reduces the number of potential groups from which neighbors of each node may come, yet still achieves almost perfect key-sharing probability with low storage overhead. Two similar works [8, 11] have been proposed at ACM Wise 2005 independently. In [8], sensors are equally partitioned based on ids into disjoint *deployment groups* and disjoint *cross groups*. Each sensor resides in exactly one deployment group and one cross group. Sensors within the same group can establish shared keys based on any of the key establishment schemes mentioned above [2–4, 18, 19]. In [11], the deployment groups and cross groups are defined differently and each sensor may reside in more than two groups. Note that these two schemes inherit many nice features of [6], but release the strong topology assumption adopted by [6]. A major drawback of these schemes is the high communication overhead when path keys are sought to establish shared keys between neighboring sensors.

Even with these efforts, the shared key establishment problem still has not been completely solved yet. As claimed by [20, 21], the performance is still constrained by the conflict between the desired probability to construct shared keys for communicating parties and the resilience against node capture attacks, under a given capacity for keying information storage in each sensor. Researchers have been actively working toward this to minimize the randomness [22, 23] in the key predistribution schemes. Due to space limitations, we could not cover all of them thoroughly. Interested readers are referred to a recent survey [15] and the references therein.

Architectures consisting of base stations for key management have been considered in [24] and [25], which rely on base stations to establish and update different types of keys. In [1], Carman et al. apply various key management schemes on different hardware platforms and evaluate their performance in terms of energy consumption, for and so forth. Authentication in sensor networks has been considered in [24–26], and so forth.

The three in-situ key establishment schemes [12–14] are radically different from all those mentioned above. They rely on service sensors to facilitate pairwise key establishment between worker sensors after deployment. The service sensors could be predetermined [12], or self-elected based on some probability [13] or location information [14]. Each service sensor carries or computes a key space and distributes a unique piece of keying information to each associated worker sensor in its neighborhood via a computationally asymmetric secure channel. Two worker sensors are able to compute a pairwise key if they obtain keying information from the same key space. As verified by our simulation study in Section 6, in-situ schemes can simultaneously achieve good performance in terms of scalability, storage overhead, key-sharing probability, and resilience.

3. Preliminaries, Models, and Assumptions

3.1. Key Space Models. The two key space models for establishing pairwise keys, one is polynomial-based [19] and the other is matrix-based [18], have been tailored for sensor networks at [7] and [5], respectively. These two models are similar in nature.

The polynomial-based key space utilizes a bivariate λ-degree polynomial $f(x,y) = f(y,x) = \sum_{i,j=0}^{\lambda} a_{ij}x^j y^j$ over a finite field F_q, where q is a large prime number (q must be large enough to accommodate a cryptographic key) . By pluging in the id of a sensor, we obtain the keying information (called a *polynomial share*) allocated to that sensor. For example, sensor i receives $f(i,y)$ as its keying information. Therefore two sensors knowing each other's id can compute a shared key from their keying information as $f(x,y) = f(y,x)$. For the generation of a polynomial-based key space $f(x,y)$, we refer the readers to [19].

The matrix-based key space utilizes a $(\lambda+1)\times(\lambda+1)$ public matrix (Note that G can contain more than $(\lambda+1)$ columns.) G and a $(\lambda+1)\times(\lambda+1)$ private matrix D over a finite field $GF(q)$, where q is a prime that is large enough to accommodate a cryptographic key. We require D to be symmetric. Let $A = (D\cdot G)^T$. Since D is symmetric, $A\cdot G$ is symmetric too. If we let $K = A\cdot G$, we have $k_{ij} = k_{ji}$, where k_{ij} is the element at the ith row and the jth column of K, $i,j = 1,2,\ldots,\lambda+1$. Therefore if a sensor knows a row of A, say row i, and a column of G, say column j, then the sensor can compute k_{ij}. Based on this observation, we can allocate to sensor i a keying share containing the ith row of A and the ith column of G, such that two sensors i and j can compute their shared key k_{ij} by exchanging the columns of G in their keying information. We call (D,G) a matrix-based key space, whose generation has been well-documented by [18] and further by [5].

Both key spaces are λ-collusion-resistent [18, 19]. In other words, as long as no more than λ sensors receiving keying information from the same key space release their stored keying shares to an attacker, the key space remains perfectly secure. Note that it is interesting to observe that the storage space required by a keying share from either key space at a sensor can be very close $((\lambda+1)\cdot\log q$ for the polynomial-based key space [19] and $(\lambda+2)\cdot\log q$ for the matrix-based key space [18]) for the same λ, as long as the public matrix G is carefully designed. For example, [5] proposes to employ a Vandermonde matrix over $GF(q)$ for G, such that a keying share contains one row of A and the seed element of the corresponding column in G. However, the column of G in a keying share must be restored when needed, resulting in $(\lambda-1)$ modular multiplications.

Note that iPAK, SBK and LKE work with both key space models. In these schemes, service sensors need to generate or to be preloaded with a key space and then distribute to each worker sensor a keying share. Two worker sensors can establish a shared key as long as they have keying information from the same key space. Note that for a polynomial-based key space, two sensors need to exchange their ids while for a matrix-based key space, they need to exchange the columns (or the seeds of the corresponding columns) of G in their keying shares.

3.2. Rabin's Public Cryptosystem. Rabin's scheme [27] is a public cryptosystem, which is adopted by the in-situ key establishment schemes to set up a computationally asymmetric secure channel through which keying information can be delivered from a service sensor to a worker sensor.

3.2.1. Key Generation. Choose two large distinct primes p and q such that $p \equiv q \equiv 3 \bmod 4$. (p,q) is the private key while $n = p\cdot q$ is the public key.

3.2.2. Encryption. For the encryption, only the public key n is needed. Let P_l be the plain text that is represented as an integer in Z_n. Then the cipher text $c = P_l^2 \bmod n$.

3.2.3. Decryption. Since $p \equiv q \equiv 3 \bmod 4$, we have

$$\begin{aligned} m_p &= c^{p+1/4} \bmod p, \\ m_q &= c^{q+1/4} \bmod q. \end{aligned} \tag{1}$$

By applying the extended Euclidean algorithm, y_p and y_q can be computed such that $y_p\cdot p + y_q\cdot q = 1$.

From the Chinese remainder theorem, four square roots $+r, -r, +s, -s$ can be obtained:

$$\begin{aligned} r &= \left(y_p\cdot p\cdot m_q + y_q\cdot q\cdot m_p\right) \bmod n \\ -r &= n - r \\ s &= \left(y_p\cdot p\cdot m_q - y_q\cdot q\cdot m_p\right) \bmod n \\ -s &= n - s. \end{aligned} \tag{2}$$

Note that Rabin's encryption [27] requires only one squaring, which is several hundreds of times faster than RSA. However, its decryption time is comparable to RSA. The security of Rabin's scheme is based on the factorization of large numbers; thus, it is comparable to that of RSA too. Since Rabin's decryption produces three false results in addition to the correct plain text, a prespecified redundancy, a bit string R, is appended to the plain text before encryption for ambiguity resolution.

3.3. Network Model and Security Assumptions. We consider a large-scale sensor network with nodes dropped over the deployment region through vehicles such as aircrafts. Therefore no topology information is available before deployment. Sensors are classified as either *worker nodes* or *service nodes*. Worker sensors are in charge of sensing and reporting data, and thus are expected to operate for a long time. Service sensors take care of key space generation and keying information dissemination to assist in bootstrapping pairwise keys among worker sensors. They may die early due to depleted energy resulted from high workload in the bootstrapping procedure. In this sense, they are sacrifices. Nevertheless, we assume service sensors are able to survive the bootstrapping procedure.

In our consideration, sensors are not tamper resistant. The compromise or capture of a sensor releases all its security information to the attacker. Nevertheless, a sensor deployed in a hostile environment must be designed to survive at least a short interval longer than the key bootstrapping procedure when captured by an adversary; otherwise, the whole network can be easily taken over by the opponent [28].

We further assume that a cryptographically secure key k_0 is preloaded to all sensors such that all communications in the key establishment procedure can be protected by a

popular symmetric cryptosystem such as AES or Triple-DES. Therefore k_0 is adopted mainly to protect against false sensor injection attacks, and any node deployed by an adversary can be excluded from key establishment. Note that k_0 is strong enough such that it is almost impossible for an adversary to recover it before the key establishment procedure is complete, and the release of k_0 after the key establishment procedure does not negatively affect the security of the in-situ key establishment schemes since all sensitive information involved in the key establishment procedure is protected via a different technique. All sensors should remove their stored keying information (k_0 and/or the key space/pool) at the end of the key bootstrapping procedure.

4. The In-Situ Key Establishment Framework

Compared to the predistribution schemes, in-situ key establishment schemes distribute keying information for shared key computation after deployment.

All the in-situ key establishment contains three phases: *service node determination and key space construction*, *service node association and keying information acquisition*, and *shared key derivation*. iPAK, SBK, and LKE mainly differ from each other in the first phase, which will be detailed afterwards. Now we sketch the framework for in-situ key establishment in sensor networks.

4.1. Service Node Determination and Key Space Construction. In the first phase, service nodes are either preselected (in iPAK[12]), or self-elected with some probability (in SBK[13]) or based on sensors' physical location (in LKE[14]). A λ-collusion resistent key space (either polynomial-based [19] or matrix-based [18]) is allocated to [12] or generated by [13, 14] each service sensor.

Before deployment, each sensor randomly picks up two primes p and q from a pool of large primes without replacement. The prime pool is precomputed by high-performance facilities, which is out of the scope of this paper. Primes p and q will be used to form the private key such that Rabin's public cryptosystem [27] can be applied to establish a secure channel for disseminating keying information in the second phase.

4.2. Service Node Association and Keying Information Acquisition. Once a service sensor finishes the key space construction, it broadcasts a beacon message notifying others of its existence after a random delay. A worker node receiving the beacon will acquire keying information from the service sensor through a secure channel established based on Rabin's cryptosystem between the two nodes. As illustrated in Figure 2, the service node association and keying information acquisition is composed of the following three steps.

4.2.1. Key Space Advertisement. A service node S announces its existence through beacon broadcasting when its key space is ready. The beacon message should include: (i) a unique key space id, (ii) the public key n, where $n = p \times q$ and (p, q) is the corresponding private key preloaded before deployment, and (iii) the coverage area of the service sensor, which is determined in LKE by a grid size L, and specified in iPAK and SBK by a forwarding bound H, the maximum distance in hop count over which the existence of a key space can be announced. The message will be forwarded to all sensors within S's coverage area.

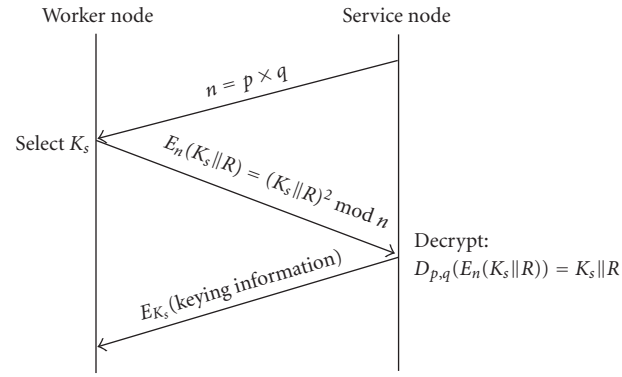


Figure 2: Service sensor association. A worker node associates itself to a service sensor to obtain the keying information through a secure channel established based on Rabin's public cryptosystem.

4.2.2. Secure Channel Establishment. Any worker node requesting the keying information from a service node needs to establish a secure channel to the associated service node. Recall that we leverage Rabin's public key cryptosystem [27] for this purpose. After obtaining the public key n, a worker sensor picks up a random key K_s and computes $E_n(K_s\|R) = (K_s\|R)^2 \bmod n$, where R is a predefined bit pattern for ambiguity resolution in Rabin's decryption. $E_n(K_s\|R)$, along with the location information, is transmitted to the corresponding service sensor. After Rabin's decryption, the service sensor obtains $D_{p,q}(E_n(K_s\|R)) = K_s\|R$, where K_s will be utilized to protect the keying share transmission from the service sensor to the work sensor.

Note that in this protocol, each worker sensor executes one Rabin's encryption for each service sensor from which an existence announcement is received, whereas the computationally intensive decryption of Rabin's system is performed only at service sensors. This can conserve the energy of worker sensors to extend the operation time of the network. In this aspect, service nodes work as sacrifices to extend the network lifetime.

4.2.3. Keying Information Acquisition. After a shared key K_s is established between a worker node and a service node, the service sensor allocates to the node a keying share from its key space. The keying information, encrypted with K_s based on any popular symmetric encryption algorithm (AES, DES, etc.), is transmitted to the requesting worker node securely. Any two worker nodes receiving keying information from the

same service node can derive a shared key for secure data exchange in the future.

After disseminating the keying information to all worker sensors in the coverage area, *the service sensor should erase all stored key space information for security enhancement*.

4.3. Shared Key Derivation. Two neighboring nodes sharing at least one key space (having obtained keying information from at least one common service sensor) can establish a shared key accordingly. The actual computation procedure is dependent on the underlying key space model. We refer the readers for the details to Subsection 3.1. Note that this procedure involves the exchange of either node ids, if polynomial-based key space model is utilized [19], or columns (seeds) of the public matrix, if matrix-based key space model is utilized [18]. To further improve security, nonces can be introduced to protect against replay attacks.

5. Service Sensor Election for the In-Situ Key Establishment Schemes

All the in-situ key establishment schemes rely on service sensors for keying information dissemination after deployment. As stated before, the major difference among the three schemes lies in how service sensors are selected, which is sketched in this section.

5.1. iPAK. Service node election in iPAK is trivial. They are predetermined by the network owner. iPAK considers a heterogeneous sensor network consisting of two different types of sensors, namely, worker nodes and service nodes. Since the number of service sensors is expected to be much smaller than that of the worker sensors, service sensors are assumed to have much higher capability (computational power, energy, and so forth) in order to complete the key bootstrapping procedure before they run out of energy.

Each service node is preloaded with all the necessary information, including one key space and two large primes. Worker sensors and service sensors are deployed together, with the proportion predetermined by ρ, where $\rho = \lambda \cdot N_s/N_w$, and $N_s(N_w)$ is the number of service nodes (worker nodes). The serving area of a service node is predetermined by the forwarding bound T_0, the utmost hop distance from the service node that the keying information can be disseminated.

5.2. SBK. Compared to iPAK, SBK does not differentiate the roles of worker sensors and service sensors before deployment. Instead, sensors determine their roles after deployment by probing the local topology of the network. In SBK, service sensors are elected based on a probability P_s, which is initialized as $P_s = 1/\lambda$. Once elected, a service sensor constructs a λ-collusion-resistent key space and serves worker sensors within its coverage area that is determined by the forwarding bound T_0. T_0 is defined according to the expected network density, which should satisfy $N_{T_0} \leq \lambda$ where N_{T_0} is the average number of neighbors within T_0 hops in the network.

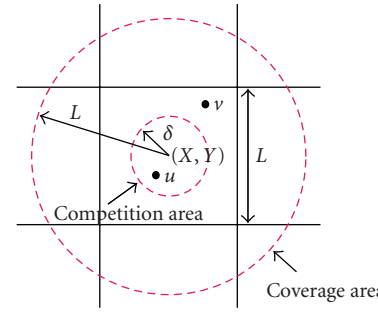


Figure 3: LKE: A virtual grid, with each grid size of L, is computed based on location information. Sensor u is selected from the competition area and will take care of key establishment for nodes residing in the coverage area.

In SBK, the service node election is conducted for several rounds. At the beginning of each round, a non-service sensor that does not have any service node within $T_0 - 1$ hops decides to become a service node with the probability P_s. If a sensor succeeds in the self-election, it sets up a key space, announces its status to T_0-hop neighbors after a random delay, and then enters the next phase for keying information dissemination. Otherwise, it listens to key space advertisements. Upon receiving any new key space announcements from a service node that is at most $T_0 - 1$ hops away, the sensor becomes a worker node, erases its primes, and enters the next phase for service sensor association and keying information acquisition. Note that the reception of a service node announcement also suppresses sensors who have self-elected as service nodes but have not broadcasted their decisions to broadcast their status. If no service node within $T_0 - 1$ hops is detected in the current round, the sensor participates in the next round.

To speed up the key bootstrapping procedure, an enhanced scheme, iSBK, is also proposed in [13], which achieves high connectivity in less time by generating more service sensors. In iSBK, the service sensor election probability P_s is initialized as $P_s = 1/N_{T_0-1}$, and is doubled in each new round until it reaches 1.

5.3. LKE. Similar to SBK, LKE [14] is a self-configuring key establishment scheme. However, the role differentiation is based on location information instead of a probability P_s. Right after deployment, each sensor positions itself and computes a virtual grid with the grid size of L. As illustrated in Figure 3, each grid contains a *competition area*, the disk region within a radius of δ from the grid center. At most one service sensor will be selected from the competition area.

An eligible sensor first waits a random delay. If it receives no competition message from others, it announces its decision to be a service sensor. Otherwise, the sensor self-configures as a worker sensor. Note that all the eligible sensors are within δ-distance from the grid center with $\delta = R/\sqrt{5}$, where R is the nominal transmission range. The setting of δ ensures that all eligible sensors within a grid can communicate with each other directly.

Each service sensor will establish a λ-collusion-resistent key space and serve those worker sensors residing in the *coverage area*, the disk region centered at the grid center with a radius of L. The setting of L satisfies $\pi L^2 = \lambda \times A/N$, where A is the deployment area, and N is the total number of nodes to be deployed. Thus, each service node is expected to serve λ nodes in a uniformly distributed network. To improve performance, iLKE is proposed, which adaptively generates service nodes based on a hierarchical virtual grid structure such that each service sensor will serve at most λ worker sensors.

6. Performance Evaluation

In this section, we study the performance of iPAK, SBK, and LKE via simulation. Note that we focus on worker sensors only, as service sensors are sacrifices that will not participate in the long-lasting networking operations. We will evaluate the in-situ key establishment schemes in terms of the following metrics via simulation: *Scalability, Resilience, Connectivity, Storage*, and *Cost*. These performance metrics will be defined at which our corresponding simulation results are reported.

6.1. Simulation Settings. We consider a sensor network of 300 or 500 nodes deployed over a field of 100 by 100. The sensors are uniformly distributed in the network, with each node capable of a fixed transmission range of 10. All the results are averaged over 100 runs.

In SBK and LKE, the two system parameters that affect the performance are the node density and λ, the security parameter of the λ-collusion-resistant key spaces. In iPAK, two more system parameters to be specified are ρ and T_0, where ρ determines the fraction of service nodes to be deployed, and T_0 determines the serving area of a service node. In our simulation study, we measure the performance of the three schemes under the same node density and security parameter λ, and conFigure the other parameters (T_0 and ρ) accordingly for a fair comparison.

In iPAK, the serving area of a service sensor is specified by the preconfigured parameter T_0. While in SBK and LKE, a service sensor determines its coverage area according to λ and the node density. Specifically, a service sensor serves worker sensors within T_0-hop (in SBK) or L-distance (in LKE), respectively, where $N_{T_0} \leq \lambda$ and $\pi L^2 = \lambda \times A/N$, T_0 is the maximum number satisfying $N_T \leq \lambda$ and N_T is the average number of neighbors within T hops in the network, N is the number of sensors in the network, and A is the deployment area. In the simulation, we select T_0 (for SBK and iPAK) and L (for LKE) that satisfy

$$N_{T_0} \leq \lambda = \frac{N}{A} \times \pi L^2. \tag{3}$$

Specifically, we consider $N = 300$ or 500 sensors in the network, estimate N_T, the average number of neighbors within T-hop using the ER model [12] (see Table 1), decide the forwarding bound T_0 for a given security parameter λ (see Table 2), and measure the performance accordingly.

Table 1: N_T, the number of neighbors within T hops, computed from ER model, used in Tests 1, 2, and 5.

T	1	2	3	4	5
$N_T(N = 300)$	9	26	55	101	164
$N_T(N = 500)$	16	48	106	194	310

Table 2: T_0, the forwarding bound, used in Tests 1 and 2.

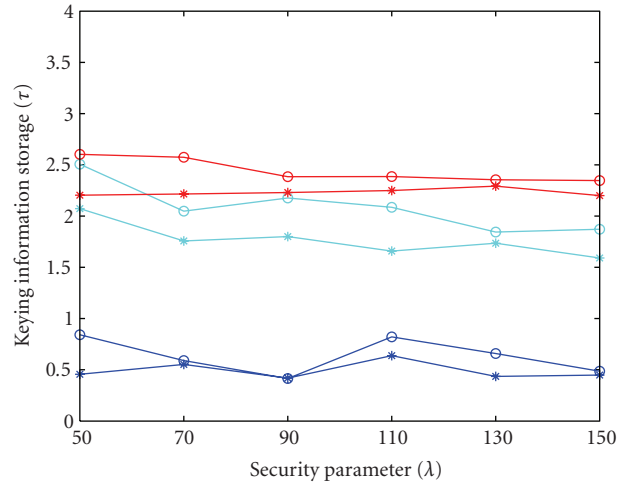
λ	50	70	90	110	130	150
$T_0(N = 300)$	2	3	3	4	4	4
$T_0(N = 500)$	2	2	2	3	3	3

Another parameter to be considered in iPAK is ρ, where $\rho = \lambda \times N_s/N_w$ and $N_s(N_w)$ is the number of service sensors (worker sensors). iPAK specifies the proportion of the two different sensors before deployment. While in SBK and LKE, service sensors are elected based on probability or location after deployment. In SBK, a service sensor is elected with the probability $P_s = 1/\lambda$, with the expectation that each service sensor serves only λ worker sensors. Thus, N_s/N_w is expected to be $1/\lambda$ in SBK. While in LKE, the network is divided into grids, and one service sensor is elected from each grid. Hence, $N_s/N_w \approx (\lceil \sqrt{A}/L \rceil)^2/N \approx A/NL^2 = \pi/\lambda$, where L is the grid size which satisfies $\pi L^2 = \lambda \times A/N$. Therefore, we consider two settings in the simulation: one is to compare iPAK and SBK with $\rho = 1$, the other is to compare iPAK and LKE with $\rho = \pi$.

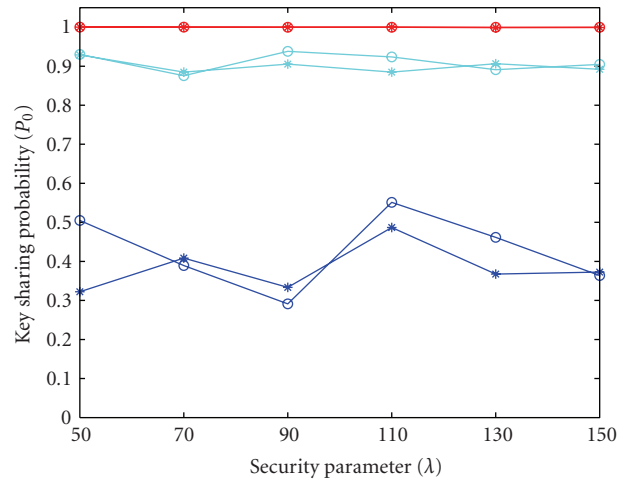
6.2. Comparison on Scalability, Storage, Connectivity and Cost. Given a series of λ values, we first measure the performance of iPAK, SBK and LKE in terms of *storage*, measured by τ, the number of keying information units (polynomial shares [19] or crypto shares [18]) obtained by a worker sensor; *connectivity*, measured by the key sharing probability P_0, the fraction of communication links that are secured by shared keys; and *cost*, measured by the percentage of service nodes generated [13, 14] or allocated [12] by the in-situ schemes.

We consider a network of 300 or 500 nodes, and employ the ER model to estimate N_T, the number of nodes within T hops in the network. The derived N_T values are given in Table 1. Then for each given λ, we set T_0 which is the maximal number satisfying $N_T \leq \lambda$. The T_0 values used in iPAK and SBK are reported in Table 2. According to the analysis in Section 6.1, we conduct three experiments: one is to compare SBK and iPAK, with $\rho = 1$ in iPAK; one is to compare LKE and iPAK, with $\rho = \pi$ in iPAK; one is to compare SBK and LKE under the same λ and node density. The results are presented in Figures 4, 5, and 6, respectively.

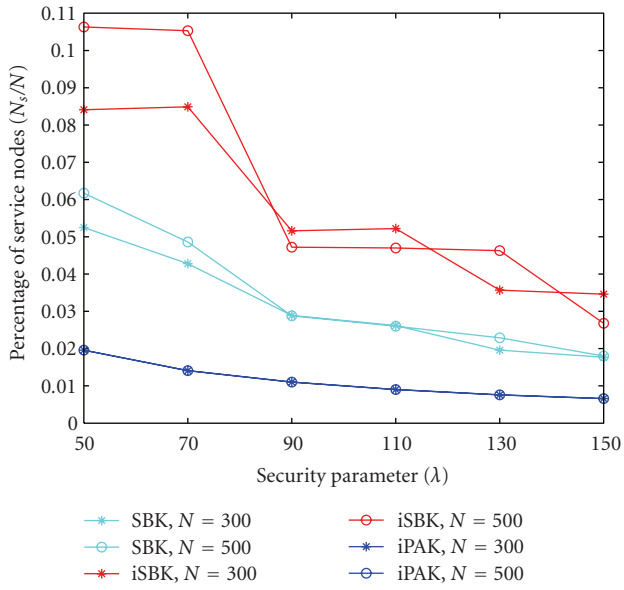
As illustrated in Figures 4, and 5, SBK and LKE can reach better connectivity than iPAK. By adjusting the number of service nodes to be generated, SBK and LKE respond actively to different network conditions with a high key sharing probability. However, iPAK has no such self-adjustability due to the predetermined ρ and T_0 values. Hence, iPAK requires that the system parameters should be carefully planned beforehand for specific network conditions. Nevertheless,



(a) Storage

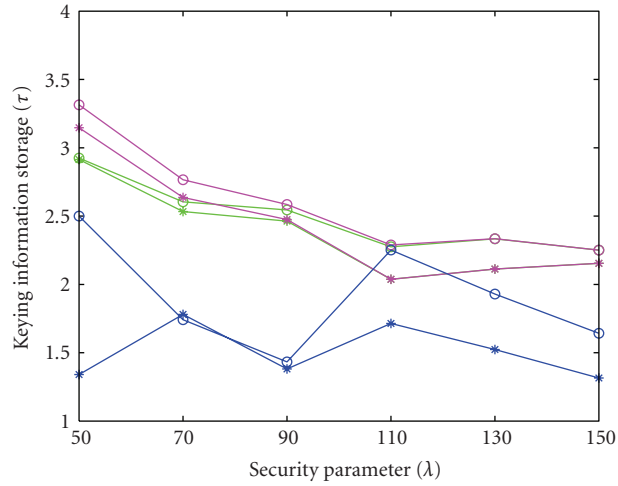


(b) Connectivity

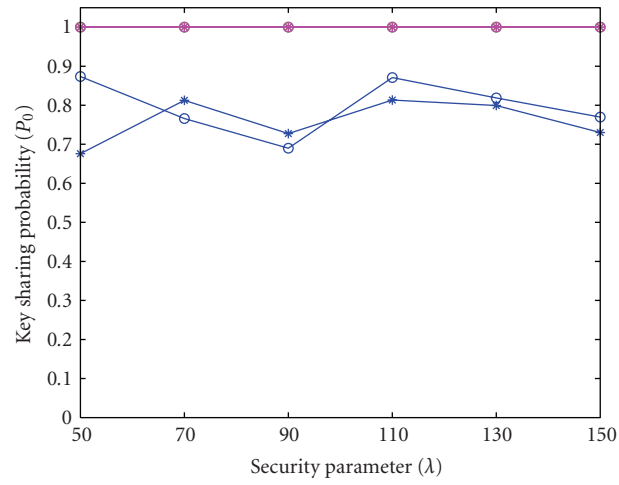


(c) Cost

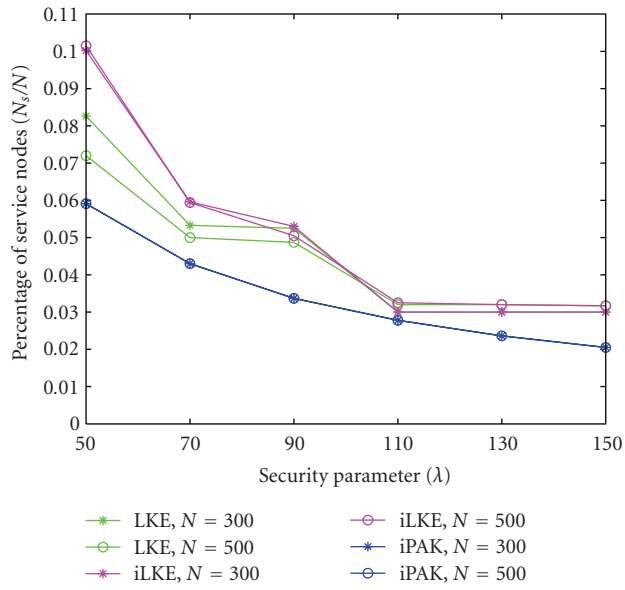
Figure 4: Test 1. iPAK versus SBK (iPAK: $\rho = 1$, $N_{T_0} \leq \lambda$): Comparison on storage, connectivity, and cost.



(a) Storage

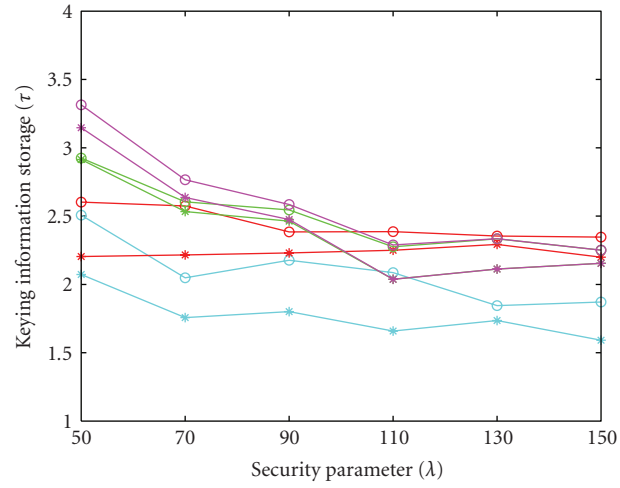


(b) Connectivity

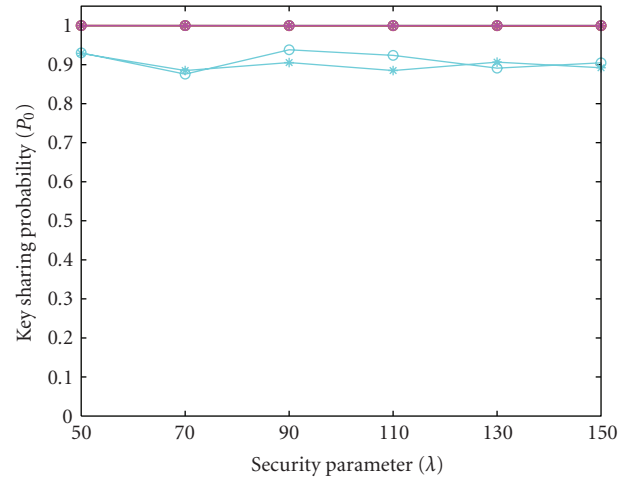


(c) Cost

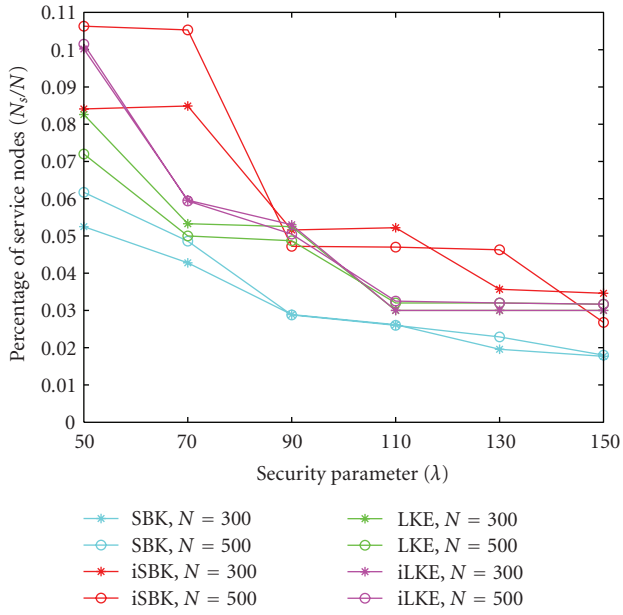
Figure 5: Test 2. iPAK versus LKE (iPAK: $\rho = \pi$, $N_{T_0} \leq \lambda$): Comparison on storage, connectivity, and cost.



(a) Storage



(b) Connectivity



(c) Cost

Figure 6: Test 3. SBK versus LKE: Comparison on storage, connectivity, and cost.

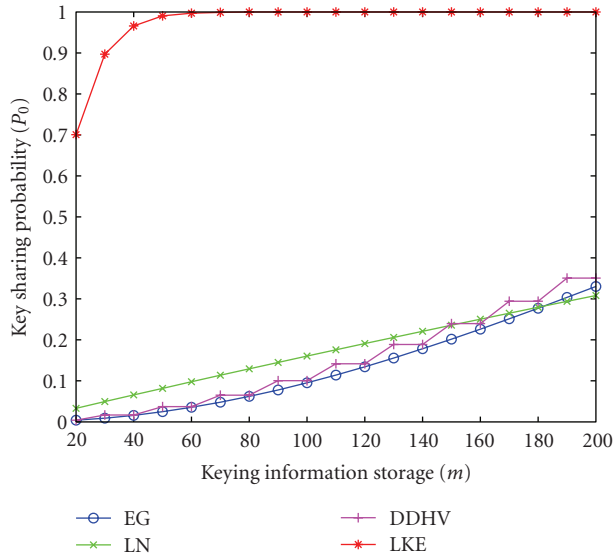


Figure 7: Test 4. Comparison of In-Situ schemes and Probabilistic-based Key Predistribution Schemes: Key Sharing Probability vs. Keying Information Storage.

iPAK has the least on-site operating complexity, since node role differentiation and key space construction are already finished before deployment.

Note that the performance of iPAK can be improved by choosing the appropriate system parameters. For example, we set $\rho = 1$ in Test 1 for a fair comparison between iPAK and SBK. $\rho = 1$ indicates $N_s/N_w = 1/\lambda$, which is just the lower bound for the fraction of service sensors to ensure the desired key-sharing probability under the limitation of $N_{T_0} \leq \lambda$. Thus, the key-sharing probability of iPAK is low in Figure 4. However, by selecting $\rho = \pi$ in Test 2, iPAK can achieve a much better connectivity with a small increase in the storage overhead. Hence, we can safely claim that iPAK, as well as SBK and LKE, can be configured to reach a high connectivity with a small amount of keying information storage in worker sensors. By using service nodes as sacrifices, all of the three in-situ schemes can avoid the storage space wastage that is existent in all the probabilistic-based key predistribution schemes, since the keying information is only disseminated within the close neighborhood.

As illustrated in Figure 6, we also observe that SBK and LKE behave similarly, while SBK can always burden worker sensors with similar storage overhead while achieving high connectivity, which is attributed to SBK's excellent topology adaptability. In SBK, sensors differentiate their roles as either service nodes or worker nodes after deployment by probing the local connectivity of the network, and then service nodes disseminate the keying information according to the specific network connectivity. But in LKE, a deterministic procedure based on location information is conducted for role differentiation and keying information distribution. Thereafter, we can expect SBK to perform better than LKE in adapting to different network conditions.

To further study the scalability of the in-situ schemes, we select LKE to compare with several probabilistic-based

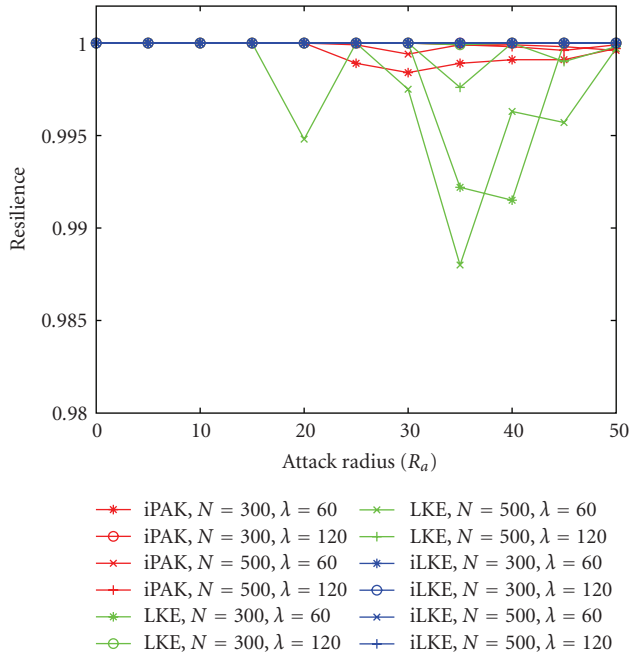


Figure 8: Test 5. iPAK vs. LKE (iPAK: $\rho = \pi, N_{T_0} \leq \lambda$). Comparison on Resilience Against Node Capture Attack.

Table 3: T_0, the forwarding bound, used in Test 5.

λ	60	120
$T_0(N = 300)$	3	4
$T_0(N = 500)$	2	3

key predistribution schemes. Figure 7 plots the relationship between P_0 and m, the number of memory units for keying information storage in a worker node (for a λ-collusion-resistent key space, m is determined by τ, the number of keying information units a sensor can obtain in the form of $m = (\lambda+1) \times \tau$ for the polynomial-based key space [19], and $m = (\lambda + 2) \times \tau$ for the matrix-based key space [18]) . We measure LKE's key sharing probability and compare it with that of the basic random key predistribution scheme (EG) [2], the random polynomial-based key space predistribution scheme (LN) [7] and the random matrix-based key space predistribution scheme (DDHV) [5]. The settings in EG and DDHV are the same as those in [6]. In EG, the key pool is of size 100, 000. In DDHV, we set the security parameter $\lambda = 19$ and the key pool size of 241 key spaces. For LN and LKE, both are considered in a network with 600 nodes, with each node storing 3 polynomial shares (we select 3 since it is a typical value for LKE in uniform network distribution as proved in [14]). The results show that the in-situ scheme can reach a much higher connectivity than the probabilistic-based predistribution schemes given the same amount of storage budget. Since the in-situ key establishment schemes are purely localized, they can completely remove the randomness inherent to the key predistribution schemes and hence achieve a much better scalability.

In summary, all of the three in-situ schemes obtain high scalability in network size. They can reach high connectivity with small amount of storage overhead, while SBK outperforms LKE, LKE outperforms iPAK in terms of topology adaptability.

6.3. Comparison on Resilience. To evaluate the resilience of the in-situ schemes, we consider a smart attack where an adversary compromises all nodes within a disk of radius R_a, and measure the resilience with the following metric.

6.3.1. Resilience. Given an attack radius R_a, the resilience against node capture attacks is defined to be the fraction of the compromised links incident to at least one compromised sensor among all the compromised links. Note that the metric resilience is in the range $(0, 1]$, where a value closer to 1 represents a better resilience.

We consider only iPAK and LKE in our simulation study, since in SBK there are at most λ worker nodes within a λ-collusion-resistent key space. Thus, the resilience of SBK remains to be 1 no matter how many nodes are captured and no matter what the network topology will be.

In the simulation, we set $\rho = \pi$ in iPAK to compare with LKE. T_0 (see Table 3) is the maximal number that satisfies $N_T \leq \lambda$, where N_T (see Table 1) is evaluated with the ER model.

As illustrated in Figure 8, both iPAK and LKE can effectively prevent the leakage of security information about uncaptured nodes, while iPAK outperforms LKE under the constraint that $N_{T_0} \leq \lambda$. We also observe that iLKE achieves the "perfect" security, which allows an adversary to learn nothing about the uncaptured sensors from those being directly attacked.

In terms of resilience, iPAK, SBK and LKE perform differently since they follow different regulations on n_s, the number of keying information to be released in a λ-secure key space. SBK requires strictly that n_s be at most λ, while iPAK has no such provision at all. In Test 4, the regulation $N_{T_0} \leq \lambda$ indicates that each λ-collusion-resistent key space is expected to cover no more than λ worker sensors, which brings about the strong resilience as illustrated in Figure 8. As for LKE, the improved scheme (iLKE) follows the same requirement as in SBK, while the basic scheme has no requirement on n_s but defines for each key space a coverage region that is expected to contain λ nodes in a uniformly distributed network. Hence, we observe that LKE and iLKE behave similarly in a uniform network distribution, while iLKE remains "perfectly" secure and LKE shows a small fluctuation in resilience. Such a fluctuation is attributed to the topology that is not perfectly uniform in our simulation.

In summary, SBK and iLKE perform the best in maintaining the security of the system. LKE can achieve a strong resilience under uniform network distribution, while iPAK must set T_0 as $N_{T_0} \leq \lambda$ to work against node capture attack.

6.4. Discussion on Computation Overhead. From the in-situ key establishment framework, we know that the computation overhead of a worker sensor comes from three sources:

encrypting a shared key k_s between a service sensor and itself in secure channel establishment, decoding the keying information obtained from the associated service node in keying information acquisition, and calculating the pairwise keys shared with its neighbors in shared key derivation. The first involves one modular squaring, while the second requires a symmetric decryption operation. These operations are repeated for each service sensor with which the worker sensor associated with.

For each neighbor, a work sensor needs to compute a pairwise key if they share a common key space. In general, given the keying information, computing a shared key with one neighbor takes $(\lambda + 1)$ modular multiplications for both key space models. Furthermore, if the matrix-based key spaces are used and only a seed, instead of the whole column of the public matrix G, is included as the keying information, each worker sensor needs $(\lambda + 1)$ more modular operations in order to recover the complete matrix share for each key space.

Modular operations are expensive in terms of energy consumption and computation time, which could make our in-situ schemes unapplicable to many practical sensor network settings. Therefore, we propose to utilize the secure pseudorandom functions (PRF) defined by the 802.11i working group and the Wi-Fi Alliance. These PRFs exploit the computationally light-weight HMAC-SHA-1, with each incorporating a different text string as input [29] to generate nonoverlapping key spaces. In our case, the text string can be the ID or the location information of the service node. Therefore in iPAK, each service node is preloaded with a PRF while in LKE and SBK, the elected service nodes run their stored PRFs to generate key spaces containing random keys. Then the service sensor securely deliver a set of pairwise keys to each associated worker sensor, as long as the worker sensor conveys the list of neighbors to the service sensor in the association phase.

Note that we can treat the PRF as another key space model, based on which each service sensor generates a random key pool that will supply pairwise keys to the associated worker sensors. It is obvious that no computation is needed at the worker sensor side. However, this zero computation overhead does not come for free: each worker sensor needs to collect the list of neighbors and send this information to all the associated service sensors. Therefore worker sensors tradeoff computation overhead with communication overhead. Furthermore, the λ-collusion resistent advantage is also lost as the PRF key space does not hold this property.

7. Conclusion

In this paper, we have studied iPAK, SBK and LKE, the three in-situ key establishment schemes proposed recently for large-scale sensor networks. We also introduce a simple improvement by exploiting a secure pseudorandom function to replace the matrix-based or the polynomial key space such that no computation is needed at the worker sensor to further conserve the resources. Our simulation results indicate that all the three in-situ key establishment schemes achieve high scalability in network size since they are purely localized. In addition, SBK and LKE outperform iPAK in terms of topology adaptability, SBK and iLKE have the best resilience against node capture attack, and iPAK has a better operating complexity. Our future research includes a more extensive performance study under different topology conditions and a comparison study with the probabilistic key predistribution schemes.

Acknowledgment

This research is supported in part by the US National Science Foundation under the CAREER Award CNS-0347674 and the Grant CCF-0627322.

References

[1] D. W. Carman, P. S. Kruss, and B. J. Matt, "Constraints and approaches for distributed sensor network security," Tech. Rep. 00-010, NAI Labs, Glenwood, Md, USA, September 2000.

[2] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, Washington, DC, USA, November 2002.

[3] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy (S&P '03)*, pp. 197–213, Berkeley, Calif, USA, May 2003.

[4] H. Chan and A. Perrig, "PIKE: peer intermediaries for key establishment in sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 524–535, Miami, Fla, USA, March 2005.

[5] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 42–51, Washington, DC, USA, October 2003.

[6] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 586–597, Hong Kong, March 2004.

[7] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 52–61, Washington, DC, USA, October 2003.

[8] D. Liu, P. Ning, and W. Du, "Group-based key predistribution for wireless sensor networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '05)*, Cologne, Germany, September 2005.

[9] Z. Yu and Y. Guan, "A key pre-distribution scheme using deployment knowledge for wireless sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 261–268, Los Angeles, Calif, USA, April 2005.

[10] Z. Yu, Y. Wei, and Y. Guan, "Key management for wireless sensor networks," in *Handbook of Wireless Mesh & Sensor Networking*, G. Aggelou, Ed., McGraw-Hill, New York, NY, USA, 2007.

[11] L. Zhou, J. Ni, and C. V. Ravishankar, "Efficient key establishment for group-based wireless sensor deployments," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '05)*, pp. 1–10, Cologne, Germany, September 2005.

[12] L. Ma, X. Cheng, F. Liu, F. An, and J. Rivera, "iPAK: an in-situ pairwise key bootstrapping scheme for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1174–1184, 2007.

[13] F. Liu, X. Cheng, L. Ma, and K. Xing, "SBK: a self-configuring framework for bootstrapping keys in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 858–868, 2008.

[14] F. Liu and X. Cheng, "LKE: a self-configuring scheme for location-aware key establishment in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 224–232, 2008.

[15] S. A. Camtepe and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey," RPI Technical Report TR-05-07, Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA, March 2005.

[16] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach," in *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP '03)*, p. 326, Atlanta, Ga, USA, November 2003.

[17] R. Di Pietro, L. V. Mancini, and A. Mei, "Efficient and resilient key discovery based on pseudo-random key pre-deployment," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 217–224, Santa Fe, NM, USA, April 2004.

[18] R. Blom, "An optimal class of symmetric key generation systems," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT '84)*, pp. 335–338, Paris, France, April 1984.

[19] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaroe, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '92)*, vol. 740 of *Lecture Notes in Computer Science*, pp. 471–486, Santa Barbara, Calif, USA, August 1992.

[20] W. Du, R. Wang, and P. Ning, "An efficient scheme for authenticating public keys in sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 58–67, ACM Press, Urbana-Champaign, Ill, USA, May 2005.

[21] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[22] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Security of Ad Hoc and Sensor Networks in Association with 10th ACM Conference on Computer and Communications Security*, pp. 72–82, Fairfax, Va, USA, October 2003.

[23] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 29–42, ACM Press, Washington, DC, USA, October 2004.

[24] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, (MOBICOM '01)*, pp. 189–199, Rome, Italy, July 2001.

[25] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 62–72, Washington, DC, USA, October 2003.

[26] R. Watro, D. Kong, S.-F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 59–64, Washington, DC, USA, October 2004.

[27] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Tech. Rep. MIT/LCS/TR-212, MIT Laboratory for Computer Science, Cambridge, Mass, USA, 1979.

[28] R. Anderson, H. Chan, and A. Perrig, "Key infection: smart trust for smart dust," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP '04)*, pp. 206–215, Berlin, Germany, October 2004.

[29] J. Edney and W. A. Arbaugh, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Addison-Wesley, Reading, Mass, USA, 2004.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 240610, 9 pages
doi:10.1155/2009/240610

Research Article

A Flexible and Efficient Key Distribution Scheme for Renewable Wireless Sensor Networks

An-Ni Shen,[1] Song Guo,[1] and Victor Leung[2]

[1] *School of Computer Science and Engineering, University of Aizu, Fukushima-Ken 965-8580, Japan*
[2] *Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z4*

Correspondence should be addressed to Song Guo, sguo@u-aizu.ac.jp

Received 1 February 2009; Accepted 11 April 2009

Recommended by Yang Xiao

Many applications of wireless sensor network require secure data communications, especially in a hostile environment. In order to protect the sensitive data and the sensor readings, secret keys should be used to encrypt the exchanged messages between communicating nodes. Traditional asymmetric key cryptosystems are infeasible in WSN due to its low capacity at each senor node. In this paper, we propose a new key distribution scheme for hierarchical WSNs with renewable network devices. Compared to some of the existing schemes, our key establishment methods possess the following features that are particularly beneficial to the resource-constrained large-scale WSNs: (1) robustness to the node capture attack, (2) flexibility for adding new network devices, (3) scalability in terms of storage cost, and (4) low communication overhead.

1. Introduction

Wireless sensor networks (WSNs) have been envisioned to be very useful for a broad spectrum of emerging civil and military applications [1]. However, sensor networks are also confronted with many security threats such as node compromise, routing disruption, and false data injection, because they normally operate in unattended, harsh, or hostile environment. Among all these threats, the WSNs are particularly vulnerable to the node compromise because sensor nodes are not tamper-proof devices. An adversary might easily capture the sensor devices to acquire their sensitive data and keys and then abuse them to further compromise the communication between other noncaptured nodes. This typical threat is known as the *node capture attack*. In order to conquer such problem, it is desirable to design key distribution protocols to support secure and robust pairwise communication among any pair of sensors.

To prevent from the node capture attack is a challenging task in sensor networks that have scarce resources in energy, computation, and communication. Therefore, only lightweight energy efficient key distribution mechanisms are affordable. For example, the conventional asymmetric key cryptosystem, such as RSA [2] and Diffie-Hellman [3], cannot be implemented in sensor nodes due to their very limited capacities. As the first naive solution, all sensor devices are preloaded the same master key and thus any two nodes can use this master key for secure communication after deployment. However, if one sensor node is physically captured by an adversary, it would compromise the entire network secrecy. Another possible approach is to assign a distinct pairwise key for each pair of sensor nodes before they are deployed. Each sensor node needs to store $(n-1)$ keys, where n is the size of the network. The solution provided secure against the node captured attack but not scalable. Moreover, addition of new sensors to a deployed network is extremely difficult.

WSNs can be broadly classified into flat WSNs and hierarchical WSNs. In a flat WSN, all senor nodes have the same computational and communication capacities. In a hierarchical WSN, however, some special sensor devices, called Cluster Head (CH), have much higher capacities than other sensor nodes. By applying some clustering algorithms like [4], the whole set of sensor devices could be partitioned into several distinct clusters such that each cluster has at least one CH. Under this arrangement, each sensor node forwards the generated packets to its local CH by short-range

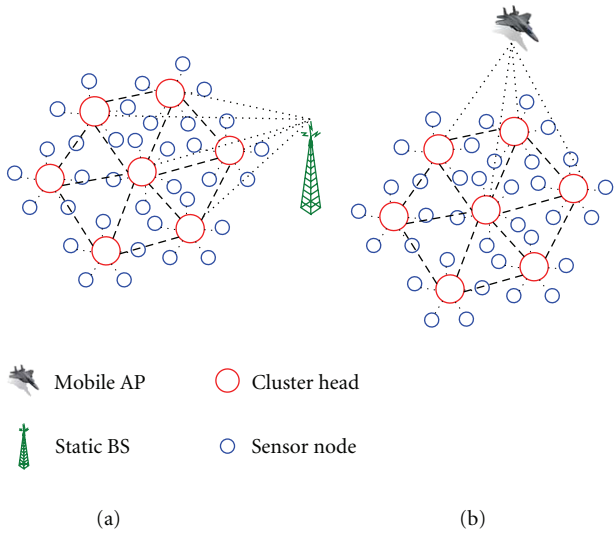


Figure 1: A three-tier hierarchical WSN.

transmissions, and the CH then performs a preprocessing for the raw data received from all other senor nodes in the cluster and finally forwards the aggregated data to the sink node, or Base Station (BS), by long-range transmissions. Key distribution protocols have already been studied comprehensively in flat WSNs, for example, in [5–8]. Recent research has more focused on the hierarchical architecture for large-scale resource-constrained WSNs, because it has been shown in [9] that a hierarchical architecture can provide better performance, in terms of communication overhead, than a flat architecture in such networks.

To solve the key agreement problem in hierarchical WSNs, Jolly et al. proposed a key predistribution scheme LEKM [10]. Before deployment, each CH stores a set of keys in its memory and each sensor node randomly selects a key from a CH and stores it with the CH's Id in its memory. After deployment, each sensor node establishes a securely link with the CH that has been selected. This is done at each sensor node by exchanging key information over the whole network. Such scheme has no computational cost at both sensor node and CH in key establishment phase and is robust against node capture attack after the key establishment phase. However, it has high storage and communication overhead at CHs.

Another proposal IKDM [9] is a polynomial-based protocol for hierarchical WSN. In the IKDM scheme, each sensor node or CH has fixed storage cost in predistribution phase. In order to improve the resilience against the node captured attack, the preloaded key of each sensor node is the exclusive-or result of ℓ ($\ell \geq 1$) number of bivariate polynomial keys which can be fetched by its CH from ℓ number of distinctive CHs all over the network. The parameter ℓ defines the tradeoff between the communication overhead and the robustness to the node capture attacks at the cluster heads. While the large ℓ can improve the security level of the network, it will also result in significant message exchanges for establishing secure links.

In real applications, new network devices need to be added into an already deployed network from time to time in order to replace the power-exhausted or compromised devices such that the performance of the whole network would not significantly degrade. However, most of schemes, for example, [9, 10], cannot provide a full solution to the key management for adding new cluster heads and sensor nodes in hierarchal renewable WSNs. In summary, the security and efficiency requirements in a WSN may include secrecy and authentication, robustness against node capture attack, dynamic membership management (including new network device addition), strong network connectivity, scalability to large-scale networks, and low complexities on memory, computation, and communication overhead. These challenges motivate us to propose scalable and robust pairwise key distribution mechanism between sensor devices in large-scale WSNs. In particular, our methods possess the following features that are particularly beneficial to the resource-constrained WSNs: (1) robustness to the node capture attack, (2) flexibility on key establishment for adding new network devices, (3) scalability in terms of storage cost, and (4) low communication overhead.

The rest of this paper is organized as follows. Section 2 presents our network model. Section 3 gives an overview of our proposal. Section 4 describes a group of protocols for our key distribution mechanism. Section 5 analyzes the security and evaluates the performance of our proposal. Section 6 summarizes our findings.

2. Network Model

As in other hierarchical models of sensor network [9–11], our system also assumes that a sensor network is divided into clusters, which are the minimum unit for detecting events. A cluster head coordinates all the actions inside a cluster and each pair of cluster heads in their transmission range can communicate directly with each other. Moreover, we assume a single base station (BS) or an access point (AP) in the network and works as the network controller to collect event data. As illustrated in Figure 1(a), the BS is a fixed infrastructure located in the network with virtually unlimited computational and communication power, unlimited memory storage capacity, and very large radio transmission range to ensure the full coverage of the whole network area. Another application scenario given in Figure 1(b) shows that the information collected by cluster heads from all its sensor nodes is retrieved by a mobile AP periodically. During the information retrieval operation, the AP broadcasts a beacon to activate cluster heads in its coverage area. Activated cluster heads then transmit their data to the AP through a common wireless channel. In the rest of paper, we use the general term BS for such network controller for describing our key distribution mechanism without discriminating the above two scenarios.

Our model has three different types of network devices: base station, cluster head, and normal sensor node. Each low-cost sensor node has low data processing capability, limited memory storage and battery power supplies, and

Table 1: Notations.

Symbol	Explanation
S_i	The Id of the sensor node $i\,(1 \le i \le n)$
CH_i	The Id of cluster head in cluster $i\,(1 \le i \le m)$
BS	The Id of the base station
$N_S(\mathrm{CH}_a)$	The set of all sensor nodes in cluster a, that is, there is a pairwise key between CH_a and any sensor node $S_i \in N_S(\mathrm{CH}_a)$
λ_S	The average number of sensor nodes in a cluster
$N_{CH}(\mathrm{CH}_a)$	The set of all neighboring cluster heads of cluster a, that is, there is a pairwise key between CH_a and any cluster head $\mathrm{CH}_b \in N_{CH}(\mathrm{CH}_a)$
λ_{CH}	The average number of neighboring cluster heads for a cluster head

short radio transmission range. Sensor nodes are restricted to direct communications with its CH only. The CHs are equipped with high power batteries, large memory storages, powerful antenna and data processing capacities, and thus can execute relatively complicated numerical operations. As the most powerful node in a WSN, the BS works as the central controller for data collect and key management. For the latter function, the BS maintains the topology of the whole network (the Ids of network devices and their connectivity information) and the method to generate keys for any secure link just based on Ids. In particular, we introduce two working modes for the BS: (1) on-line mode and (2) off-line mode.

In an on-line working mode, the key generation method at the BS can be requested from any cluster head and the BS should response in a timely manner. However, such on-line service is not always available at the BS. For example, the BS cannot response the request in certain period of time, in which it is already dedicated to some important and uninterruptable tasks as illustrated in Figure 1(a), or the requesting cluster head is not in its service area as illustrated in Figure 1(b). Under both cases, the BS is configured to work in the off-line mode, and the alternative methods for key generation relying on other network devices should be provided by the key distribution protocol.

A three-tier hierarchical wireless sensor network can thus be modeled as a simple graph G with a finite node set, including a base station, m cluster heads, and n sensor nodes. A secure wireless link corresponding to the wireless communication channel belongs to the arc set of G only if there exists a pairwise key between the transmission nodes of the link. In Table 1, we summarize the notations used in the rest of the paper.

3. Overview of Our Key Distribution Scheme

In this section, we present the foundations and basic idea of our key distribution scheme based on a three-tier hierarchal network model.

3.1. Key Distribution in Renewable WSNs. Specifics of wireless sensor networks, such as strict resource constraints and large network scalability, require a proposed security protocol to be not only secure but also efficient. Recent research shows that preloading symmetric keys into sensors before they are deployed is a practical method to deal with the key distribution and management problem in wireless sensor networking environments. After the deployment, if two neighboring nodes have some common keys, they can setup a secure link by the shared keys. As surveyed in [9], the existing schemes can be classified into the following three categories: random key predistribution schemes, polynomial-key predistribution schemes, and location-based key predistribution schemes.

In our key distribution scheme, a key distribution server (KDS) is available for both of the following cases. (1) KDS is installed in the base station, by which the keys can be delivered instantaneously when the BS is on-line to the requester. (2) It is available to the network deployer when the keys are required to be preloaded into network devices.

In many applications, new network devices need to be replenished into an already deployed network to replace the power-exhausted or compromised devices. The corresponding key management should be provided in order to setup the secure link between a new added network device and an existing one. To our best knowledge, there are no full solutions to the dynamic membership management for key distribution in hierarchal WSNs with renewable cluster head and sensor node. For example, some of them can only support the sensor node addition in the case when BS is on-line. The objective of our key distribution protocols is to provide a complete and flexible solution for such renewable WSNs. In particular, we will provide the key distribution protocols for both sensor node and cluster head when the BS is on-line or off-line.

3.2. Symmetric Polynomial Function. In our key distribution scheme, a bivariate symmetric polynomial function (s.p.f.) is used to generate the key for each link of the network. The t-degree bivariate symmetric polynomial function $f(x, y)$, introduced in [12], is defined as

$$f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j. \tag{1}$$

The coefficients $a_{ij}\,(0 \le i, j \le t)$ are randomly chosen from a finite field $GF(Q)$, in which Q is a prime number that is large enough to accommodate a cryptographic key. As implied by its name, the symmetric property of a bivariate polynomial function satisfies $f(x, y) = f(y, x)$. In our key distribution scheme, the KDS maintains two bivariate polynomial functions:

(i) the s.p.f. $f_{\text{CH-NS}}(x, y)$ is used to establish the key between existing cluster head and new sensor node,

(ii) the s.p.f. $f_{\text{CH-NCH}}(x, y)$ is used to establish the key between existing cluster head and new cluster head.

After the pairwise key $K_{a,b}$ between network devices a and b is generated from the above polynomial functions by

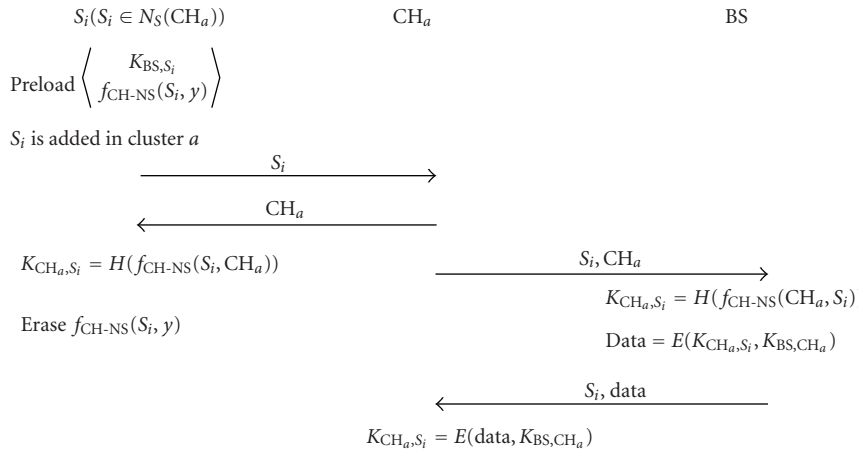


Figure 2: Protocol illustration of adding a new sensor node when BS is on-line.

substituting the variables with Ids of the two communicating parties, the *data* over the link can therefore be securely transmitted as $E(data, K_{a,b})$, which is a symmetric encryption function using $K_{a,b}$ as the key.

By applying the symmetric property, a secure link can be easily built up by just exchanging the Ids of transmission nodes. However, such scheme suffers the t-security problem, which means a t-degree bivariate polynomial key scheme can only keep secure against coalitions of up to t compromised sensors. When the number of compromised nodes is less than t, the coefficients of the polynomial cannot be derived even all the compromised nodes put their stored information together. But once more than t nodes are compromised, the adversary can crack the coefficients of the polynomial such that all the pairwise keys in the entire group would be cracked. Although increasing the value of t can improve the security property of bivariate polynomial key scheme, it is not suitable for wireless sensor networks due to the limited memory size of sensors. In order to conquer this limitation, the pairwise key x calculated from the polynomials will be further scrambled by a one-to-one hash function $H(x)$.

4. Key Distribution Protocols

Our scheme supports new network device (sensor node and cluster head) addition for both BS on-line and off-line scenarios with the minimum assumption that the deployed network has completed its key establishment, that is, the key $K_{a,b}$ for any secure link (a, b) is already shared by both network devices a and b. Furthermore, our proposed scheme can provide forward secrecy as well as full prevention from the node capture attack for large-scale sensor networks.

4.1. BS is On-Line. Let S_i be the new sensor node to be added in the network. In order to calculate the key between S_i and its cluster head, the calculation can be done at the BS if it is working at the on-line mode. Suppose new sensor node S_i is randomly added into the network and eventually belongs to cluster CH_a. The following Protocol 1, as illustrated in Figure 2, is to establish a secure link between S_i and CH_a.

Protocol 1 (sensor addition when BS is on-line).

(1) The new sensor node S_i is randomly deployed to the existing network with preloaded information: the s.p.f. $f_{\text{CH-NS}}(S_i, y)$ and a key K_{BS,S_i}.

(2) After S_i is deployed, it exchanges Ids with its cluster head CH_a.

(3) S_i evaluates its stored s.p.f. $f_{\text{CH-NS}}(S_i, y)$ at $y = \mathrm{CH}_a$ to establish the key between itself and its cluster head as $K_{\mathrm{CH}_a,S_i} = H(f_{\text{CH-NS}}(S_i, \mathrm{CH}_a))$. After calculating the pairwise key, S_i erases the preloaded s.p.f. $f_{\text{CH-NS}}(S_i, y)$ immediately to avoid potential attacks.

(4) CH_a requests the new key between CH_a and S_i from BS by forwarding the Id of S_i and its own Id.

(5) BS then calculates the corresponding key using the s.p.f. $f_{\text{CH-NS}}$ as and returns the encrypted key $E(K_{\mathrm{CH}_a,S_i}, K_{\mathrm{BS},\mathrm{CH}_a})$ back to CH_a.

(6) CH_a decrypts the received date to recover K_{CH_a,S_i} using the key $K_{\mathrm{BS},\mathrm{CH}_a}$, which was already loaded at CH_a since its very initial deployment, that is, $K_{\mathrm{CH}_a,S_i} = E(E(K_{\mathrm{CH}_a,S_i}, K_{\mathrm{BS},\mathrm{CH}_a}), K_{\mathrm{BS},\mathrm{CH}_a})$.

Now we consider the addition of a new cluster head and the corresponding key distribution procedures when the BS is on-line. We assume the CH_a is to be replaced by a new cluster head $\mathrm{CH}_{a'}$, due to its low power level. Note that in the replacement phase of cluster head, the communication keys with existing network devices (i.e., cluster head and sensor node) are also renewed, not simply making use of the copies of the previous keys. This process avoids potential attack activities and achieves the forward secrecy. In other words, even the attacker could intercept packets and analysis data to compromise the key of old cluster head, it still cannot decrypt the secret data using the old keys.

The following Protocol 2, as illustrated in Figure 3, is to build up the keys between the new cluster head $\mathrm{CH}_{a'}$, and all existing sensor nodes S_i $(S_i \in N_S(\mathrm{CH}_a))$ in the same cluster as well as the keys between the new cluster head $\mathrm{CH}_{a'}$, and all its neighboring cluster heads CH_b $(\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a))$.

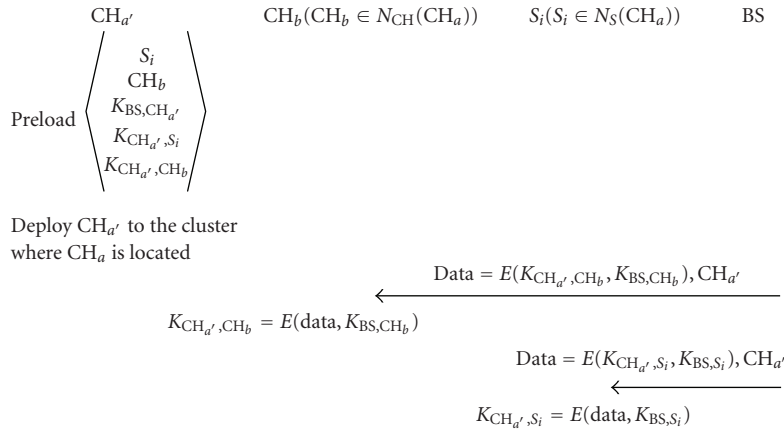


Figure 3: Protocol illustration of adding a new cluster head when BS is on-line.

Protocol 2 (CH addition when BS is on-line).

(1) The following secret information is created and preloaded into $CH_{a'}$:

   (i) the pairwise key with base station $K_{BS,CH_{a'}}$,
   (ii) for each sensor node $S_i \in N_S(CH_a)$, its Id and the key $K_{CH_{a'},S_i} = H(f_{CH\text{-}NS}(CH_{a'}, S_i))$,
   (iii) for each cluster heads $CH_b \in N_{CH}(CH_a)$, its Id and key $K_{CH_{a'},CH_b} = H(f_{CH\text{-}NCH}(CH_{a'}, CH_b))$,

(2) The new cluster head $CH_{a'}$ is then deployed physically to the cluster area where the old cluster CH_a is located.

(3) The base station transmits the encrypted key $E(K_{CH_{a'},CH_b}, K_{BS,CH_b})$ to each neighboring cluster head CH_b of CH_a such that it can be decrypted as $K_{CH_{a'},CH_b}$ at the side of CH_b using the key K_{BS,CH_b}, that is, $K_{CH_{a'},CH_b} = E(E(K_{CH_{a'},CH_b}, K_{BS,CH_b}), K_{BS,CH_b})$.

(4) Similarly, BS transmits the encrypted key $E(K_{CH_{a'},S_i}, K_{BS,S_i})$ to each senor node S_i of CH_a such that it can be decrypted as $K_{CH_{a'},S_i}$ at the side of S_i, that is, $K_{CH_{a'},S_i} = E(E(K_{CH_{a'},S_i}, K_{BS,S_i}), K_{BS,S_i})$, using the key K_{BS,S_i}.

4.2. BS is Off-Line

Protocol 3 (sensor addition when BS is off-line).

(1) The new sensor node S_i is randomly deployed to the existing network with the following preloaded information:

   (i) the pairwise key K_{BS,S_i} shared with BS,
   (ii) the Id of a cluster head CH_b, which is an arbitrary CH already in the network,
   (iii) the key $K_{CH_b,S_i} = H(f_{CH\text{-}NS}(S_i, CH_b))$ shared with CH_b,
   (iv) the encrypted key $E(K_{CH_b,S_i}, K_{BS,CH_b})$ of K_{CH_b,S_i} using K_{BS,CH_b},

(2) The added sensor node S_i sends the join-request message to the cluster head CH_a with the preloaded secret information CH_b and $E(K_{CH_b,S_i}, K_{BS,CH_b})$ and erases $E(K_{CH_b,S_i}, K_{BS,CH_b})$ afterwards.

(3) Based on CH_b, CH_a then knows to request the secret key from CH_b by providing information $E(K_{CH_b,S_i}, K_{BS,CH_b})$ and Id of S_i.

(4) After receiving the request message, CH_b uses K_{BS,CH_b} to decrypt $E(K_{CH_b,S_i}, K_{BS,CH_b})$ and obtain the pairwise key K_{CH_b,S_i}. CH_b then re-encrypts it using K_{CH_a,CH_b} as the key and sends $E(K_{CH_b,S_i}, K_{CH_a,CH_b})$ back to CH_a. Finally, CH_b deletes $E(K_{CH_b,S_i}, K_{BS,CH_b})$, $E(K_{CH_b,S_i}, K_{CH_a,CH_b})$, and K_{CH_b,S_i} immediately.

(5) CH_a decrypts $E(K_{CH_b,S_i}, K_{CH_a,CH_b})$ by K_{CH_a,CH_b} to obtain the key K_{CH_b,S_i} with S_i.

Similar to the on-line case, we assume that the new sensor node S_i is randomly added into the network and eventually belongs to cluster CH_a. In order to create the key between S_i and CH_a, a cluster head CH_b is randomly assigned as the proxy of BS as illustrated in Figure 3. All required information to generate the key should be first forwarded to CH_b. The detailed process is described in Protocol 3.

We notice that the cluster head CH_b may be physically located far from CH_a due to the random deployment process of the sensor nodes, resulting in a relatively high communication overhead between CH_a due CH_b. In order to reduce such overhead, up to ℓ number of CHs are randomly chosen as potential proxies of BS and the corresponding keys are all generated and stored in S_i. CH_a will choose the closest one, for example, with minimum hops, as the selected proxy by looking up its routing table based on their Ids. Comparing to the on-line case, we also observe that the BS-on-line case is more efficient than the BS-off-line case in terms of communication and memory overhead when both are possible.

Finally, we consider the addition of a new cluster head when the BS is off-line. The same set of symbols as in the on-line case is used and the corresponding Protocol 4 is illustrated in Figure 5.

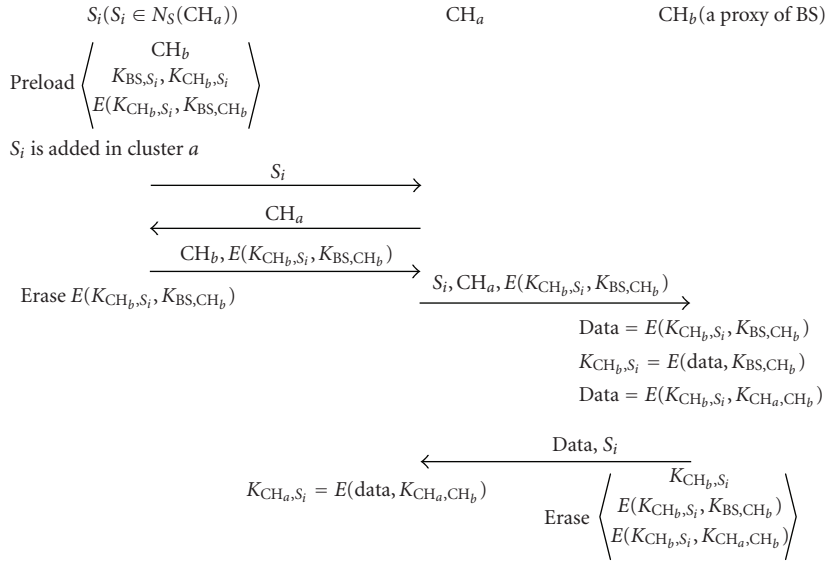


Figure 4: Protocol illustration of adding a new sensor node when BS is off-line.

Protocol 4 (CH addition when BS is off-line).

(1) The following secret information is created and preloaded into $\mathrm{CH}_{a'}$:

   (i) the pairwise key with base station $K_{\mathrm{BS},\mathrm{CH}_{a'}}$,
   (ii) for each sensor $S_i \in N_S(\mathrm{CH}_a)$, its Id, the key $K_{\mathrm{CH}_{a'},S_i} = H(f_{\text{CH-NS}}(\mathrm{CH}_{a'}, S_i))$ and encrypted key $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$,
   (iii) for each cluster head $\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a)$, its Id, the key $K_{\mathrm{CH}_{a'},\mathrm{CH}_b} = H(f_{\text{CH-NCH}}(\mathrm{CH}_{a'}, \mathrm{CH}_b))$ and the encrypted key $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$,

(2) The new cluster head $\mathrm{CH}_{a'}$ is then deployed physically to the cluster area where the old cluster CH_a is located.

(3) $\mathrm{CH}_{a'}$ exchanges Ids with each sensor node $S_i \in N_S(\mathrm{CH}_a)$ and then sends S_i the corresponding encrypted key $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$. After that the new cluster head $\mathrm{CH}_{a'}$ erases $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$ immediately. Each sensor node S_i then decrypts the received information to recover the key $K_{\mathrm{CH}_{a'},S_i}$.

(4) $\mathrm{CH}_{a'}$ exchanges Ids with each neighboring cluster head $\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a)$ and then sends CH_b the corresponding encrypted key $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$. After that the new cluster head $\mathrm{CH}_{a'}$ erases $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$ immediately. Each cluster head CH_b decrypts the received information to recover the key $K_{\mathrm{CH}_{a'},\mathrm{CH}_b}$.

5. Security and Performance Evaluation

In this section, we will analyze the security and evaluate the performance of our proposed scheme by comparing with IKDM [9] and LEKM [10].

We note that neither of IKDM and LEKM protocols supports cluster head addition process. Regarding the sensor node addition process, we have the following observations. Recall that in the IKDM scheme, the polynomial functions to be used for key generation are stored in CHs all the time and thus no on-line BS is required. As we shall later, while it simplifies the process by avoiding the involvement of BS, potential security problem has been neglected. In the LEKM scheme, the preloaded key at each sensor node must be stored in some cluster head as well. If the key assigned to the new sensor node has not been preloaded to some CH at very initial deployment of the network, such key must be distributed to a CH as well by the on-line BS. Therefore, in the following evaluation, we only consider the off-line BS case and on-line BS case for the IKDM and LEKM protocols, respectively, in the senor node addition process.

5.1. Security Analysis. The security is analyzed in terms of the ability to defend from the node capture attack, which means the capture of some nodes may compromise the communication between other noncaptured nodes. This is recognized as the major threat in wireless sensor networks. In particular, we consider the security property of all these schemes in two typical scenarios: the fractions of compromised keys in noncaptured sensor nodes as a function of the number of compromised cluster heads and the number of sensor node, respectively.

Because only pairwise keys are remained in the sensor nodes for all schemes after deployment the network, that is, all security parameters that will not be used in the future have been already erased from the network, any sensor node's compromising will not endanger the secret communications of other noncaptured nodes. In other words, all these schemes have full ability to defense the node capture attack at sensor nodes.

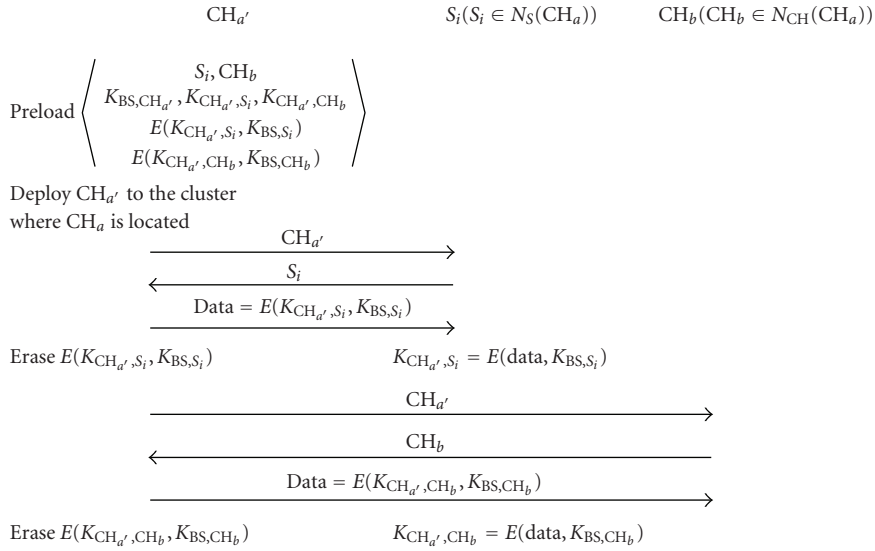


Figure 5: Protocol illustration of adding a new cluster head when BS is off-line.

Table 2: Storage cost comparison over various distribution schemes.

Schemes		Our protocols	IKDM	LEKM
On-line	Cluster head	$\lambda_S + \lambda_{CH}$ Ids	N/A	$\lambda_S + m$ keys
		$\lambda_S + \lambda_{CH} + 1$ keys		
	Sensor node	One key		One Id
		One s.p.f.		Two keys
Off-line	Cluster head	$\lambda_S + \lambda_{CH}$ Ids	One key	N/A
		$2\lambda_S + 2\lambda_{CH} + 1$ keys	Two s.p.f.	
	Sensor node	ℓ Ids	ℓ Ids	
		$2\ell + 1$ keys	Two keys	

Now we consider the security property when some cluster heads are compromised. In our key distribution protocols, because the pairwise keys in CHs are unique and hashed, they cannot be used to obtain the corresponding polynomial, that is, all the coefficients of the polynomial, reversely. We conclude that our scheme has full ability to defense the node capture attack. This conclusion applies to LEKM as well because all unrelated keys are removed at CHs after network deployment. On the other hand, the IKDM scheme has the t-security problem because all preloaded t-degree polynomials at each CH will not be removed after network deployment. Once a group of CHs, exceeding t, are captured, all the keys in noncaptured nodes will also be compromised.

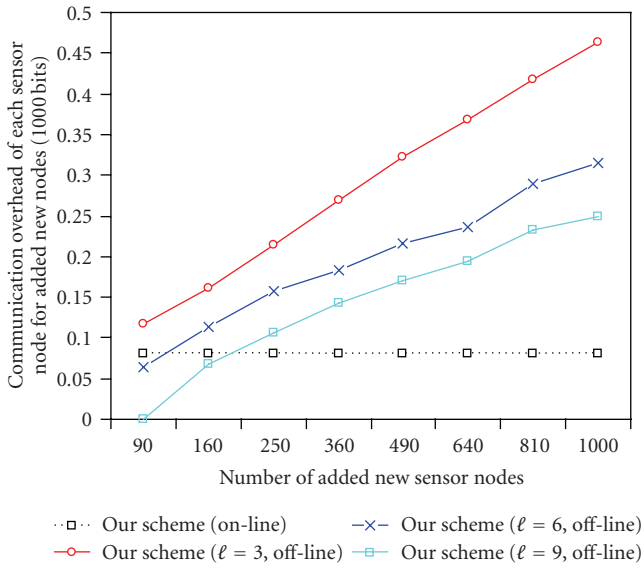
5.2. Performance Evaluation. Now we turn our attention to evaluate the performance of this group of key distribution schemes in hierarchical WSNs. The performance metrics are storage and communication overhead.

To supports a large-scale WSN, a feasible solution of key distribution should be scalable in terms of storage cost. In the scheme LEKM [10], the number of keys stored in each CH is linearly proportional to the number of clusters. The IKDM scheme has fixed storage overhead for sensor nodes and cluster heads. Our scheme has fixed storage cost for sensor nodes. The storage requirement $O(\lambda_S + \lambda_{CH})$ for cluster head is also reasonable because it requires to communicate with at least $\lambda_S + \lambda_{CH}$ number of nodes. The performance comparison in various network sizes is summarized in Table 2.

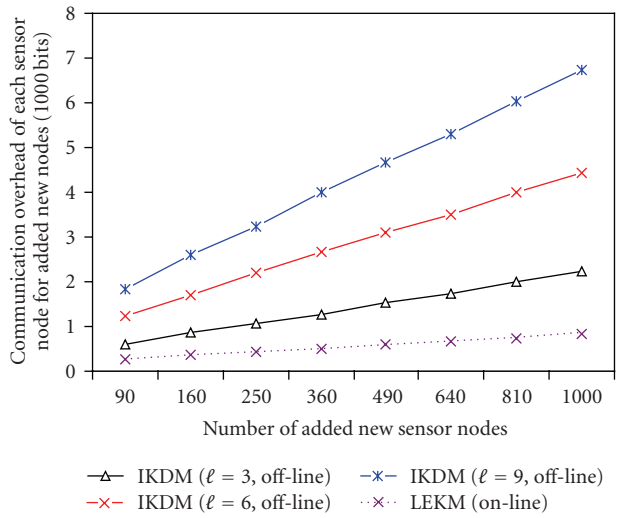
As shown in Figures 3 and 5 for the cluster head addition processes, the communication overhead of Protocols 2 and 4 is both fixed under the condition that λ_S and λ_{CH} are constant numbers, which is true for a uniform node deployment. This feature shows the scalability of our scheme in terms of message complexity. They are also the first solution for key management in WSNs with renewable cluster heads.

In the following, we conduct a simulation study on the communication overhead for the sensor node addition process. We have implemented a simulation tool using Java for the special purpose of evaluating the performance of this group of protocols while the lower MAC layer is assumed to be ideal.

A hierarchical wireless sensor network was simulated with different sizes of n sensor nodes and m clusters. In order to study the scalability of these protocols, we have considered the scenarios with a specified a cluster size m ($m = 9, 16, 25, 36, 49, 64, 81$, and 100) and a sensor node size n ($n = 100$ m). For each example, the whole network



(a) Our scheme



(b) Other schemes

Figure 6: Communication overhead comparison.

is regularly organized as $\sqrt{m} \times \sqrt{m}$ number of clusters, and there are exactly 100 sensor nodes in each $R \times R$ cluster. The transmission range of each cluster head is set as $\sqrt{5}R$, and the communications between CHs may be made in a multihop manner if they are separated far away from each other. To simulate the sensor node addition process, we consider 10 new sensor nodes to be added to each cluster. In each message interaction for all protocols, the length of each Id and key takes up 32 and 80 bits, respectively.

The performance comparison is made in terms of communication overhead. It is evaluated in the number of bits transmitted for key establishment between a sensor node and a cluster head. In all cases, that is, a sensor node size n, a cluster size m, and a specific key distribution scheme, we randomly generated 50 different instances and we present here the average over those 50 instances.

As shown in Figure 6(a), our scheme has the fixed and lowest communication overhead for the on-line scenario. The experimental results also comply with our protocol design for the off-line scenario, in which multiple candidate proxies can improve the performance, that is, the communication overhead is a decreasing function of ℓ under fixed network size. In summary, our scheme in both scenarios can significantly outperform other proposals as shown in Figure 6(b).

6. Conclusion

In this paper, we present an efficient and flexible key distribution scheme based on three-tier renewable wireless sensor networks. Our scheme can defend against node capture attack and support dynamic membership management. To our best knowledge, the solution of the key establishment for new cluster heads under both the BS off-line and on-line cases is proposed by the first time. Furthermore, our scheme is efficient and scalable in terms of communication and storage costs, which is particularly beneficial to support large-scale and resource constrained WSNs.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[3] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[4] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[5] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, Washington, DC, USA, November 2002.

[6] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '03)*, pp. 52–61, Washington, DC, USA, October 2003.

[7] W. Du, Y. S. Han, J. Deng, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '03)*, pp. 42–51, Washington, DC, USA, October 2003.

[8] Y. Cheng and D. P. Agrawal, "Efficient pairwise key establishment and management in static wireless sensor networks," in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '05)*, pp. 544–550, Washington, DC, USA, November 2005.

[9] Y. Cheng and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 35–48, 2007.

[10] G. Jolly, M. C. Kuscu, P. Kokate, and M. Yuonis, "A low-energy management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC '03)*, pp. 335–340, Kemer-Antalya, Turkey, June-July 2003.

[11] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, pp. 378–389, Urbana-Champaign, Ill, USA, May 2005.

[12] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," Lecture Notes in Computer Science, pp. 471–486, 1993.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 718318, 16 pages
doi:10.1155/2009/718318

Research Article

Cautious Rating for Trust-Enabled Routing in Wireless Sensor Networks

Ismat Maarouf,[1] Uthman Baroudi,[1] and A. R. Naseer[2]

[1] *Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia*
[2] *JITS, Nustalapur, K.N. District, AP-505481, India*

Correspondence should be addressed to A. R. Naseer, dr_arnaseer@hotmail.com

Received 30 January 2009; Revised 13 July 2009; Accepted 20 October 2009

Recommended by Hui Chen

Trust aware routing in Wireless Sensor Network (WSN) is an important direction in designing routing protocols for WSN that are susceptible to malicious attacks. The common approach to provide trust aware routing is to implement an efficient reputation system. Reputation systems in WSN require a good rating approach that can model the information on the behavior of nodes in a way that represents different sources of this information. In some WSN applications, nodes need to be more cautious in rating other nodes since it may be in a very hostile environment or it may be very intolerant to malicious behavior. Moreover, to prove the creditability of a reputation system or its related rating components, a global and system-independent technique is required that can evaluate the proposed solution. In this paper, a new rating approach called Cautious RAting for Trust Enabled Routing (CRATER). CRATER is introduced which provides a rating model that takes into account the cautious aspect of WSN nodes. Further, a promising evaluation mechanism for reputation systems called REputation Systems-Independent Scale for Trust On Routing (RESISTOR). RESISTOR is presented which can be used to evaluate and compare reputation and rating systems in a global, simple, and independent manner.

1. Introduction

Sensor networks are susceptible to attacks at the routing layer that are related to the node behavior. The most familiar attacks are nonforwarding attacks in which a compromised node will drop packets it receives instead of forwarding them. Such attacks cannot be detected or avoided by identity checking mechanisms. Hence, behavior trust should be implemented in order to defend against these attacks. Trust shall facilitate the cooperation among these nodes, though "trust" is a complex concept and it is difficult to define it precisely [1, 2]. The trust has several characteristics that can be summarized with the following six features: subjectivity, transitivity, temporalness, contextualness and dynamicity, and nonmonotonicity [3].

In this work, we adopt the following definition for the "trust": *the level of confidence that a node has in its neighbor's cooperation* [4]. This trust can be attained following two broad approaches: centralized or distributed. The centralized approach assumes a central agent that can assess the "credibility" of each node and then disseminate this information to all "real" nodes. It is obvious that such approach is difficult to realize in practice. On the other hand, the distributed approach is a localized scheme where each node assesses the credibility of its neighboring nodes and accordingly it builds its trust-aware routing.

Reputation is another complex concept and is closely linked to that of trustworthiness [1]. A reputation system is a type of cooperative filtering algorithm which attempts to determine ratings for a collection of entities that belong to the same community. Every entity rates other entities of interest based on a given collection of opinions that those entities hold about each other [5]. In [1], the main differences between trust and reputation systems are summarized as follows. First, trust systems rely on the subjective view of an entity to produce a score of an entity's trustworthiness whereas a score is produced by reputation systems as seen by the whole community. Transitivity is the second difference which is an explicit component in trust systems, whereas reputation systems usually only take transitivity implicitly

into account. Thirdly, trust systems usually depend on subjective and general measures of (reliability) trust as input, whereas objective information or ratings about specific events, such as transactions, are used as input in reputation systems.

In the context of MANET and WSN, the reputation of a node is the amount of trust the other nodes grant to it regarding its cooperation and participation in forwarding packets [6]. Hence, each node keeps track of each other's reputation according to the behavior it observes, and the reputation information may be exchanged between nodes to help each other to infer the accurate values.

Any reputation system in this context should, generally, exhibit the following three main functions [6, 7].

(i) Monitoring: this function is responsible for observing the activities of the nodes of its interest set, for example, the set of its neighbors [8].

(ii) Rating: based on the node's own observation, other nodes' observations that are exchanged among themselves and the history of the observed node, a node will rate other nodes in its interest set.

(iii) Response: once a node builds knowledge on others' reputations, it should be able to decide about different possible reactions it can take, like, avoiding bad nodes or even punishing them.

The rating component of a reputation system is a very critical part since it is responsible for providing the reputation of nodes. Thus, it can be considered as the heart of any reputation system. To illustrate the rating operation, assume that node A wants to evaluate a reputation value for a node B that may or may not be directly monitored by A. Then, the reputation value of B evaluated by A is a number that reflects how good or bad node B behaves from the perspective of node A, considering what follows.

(i) Monitoring results of all types of routing activities.

(ii) Monitoring results obtained by direct observations from A as first hand information (FHI).

(iii) Monitoring results gathered from other nodes observing B and shared with A as second-hand information (SHI).

In this work, we are proposing a new rating technique called Cautious Rating for Trust Enabled Routing (CRATER). Basically, this technique identifies three rating factors: FHI, SHI, and Neutral Behavior period during which a node is not doing any activity. The new contribution in CRATER is its mathematical approach that is used to rate nodes based on what we call cautious assumptions, which are very true in most WSN. Moreover, we are proposing a new promising mechanism to evaluate different reputation systems and their corresponding rating components called Reputation Systems-Independent Scale for Trust On Routing (RESISTOR). RESISTOR is based on the analogy of the resistance phenomenon in electric circuits. It defines a metric called "resistance" to represent how much a node is resisting its malicious neighbors. Then, based on that figure, the reputation system performance is being analyzed for evaluation.

The rest of this paper is organized as follows. In Section 2, we provide an overview of our proposed reputation system. After that, the monitoring approach is described in Section 3. Then, a detailed description of CRATER is given in Section 4 along with RESISTOR with some validation experiments results and analysis. Section 5 then describes the response (routing) component of our reputation system. In Section 6, we show system performance evaluation with the focus on system resistance behavior. This is followed by literature review in Section 7. Finally, we conclude our paper with the main findings of this research and future suggested work in Section 8.

2. Reputation System Overview

2.1. Network Model. In this work, the nodes in our WSN are deployed randomly or in a grid topology inside a square area. It is assumed that nodes communicate via bidirectional links so that they can monitor each other. Moreover, all nodes have equivalent power transmission capabilities; that is, all have equivalent transmission range. It is also assumed that the consumed power during the simulation time does not impact the transmission range of nodes. This assumption is made to keep the focus of our work on security issues and not on power control. To demonstrate the power consumption under the proposed scheme, we assume that the transmission and reception power are 1000 times more than the processing power per transmission, reception, or monitoring operation [9] (in our computation we used 1 Watt, 1 milli-watt; resp.). In this work, we care more about the overall performance and not the absolute values of the consumed power as the focus here is on securing our routes. RF channel is assumed to be ideal and collision free. Moreover, we assume a static WSN. Mobile WSN can be an interesting subject of a future research work.

Regarding communication discipline, we assume that each node in the system can initiate a routing operation. Thus, any node can be a source. Moreover, any node can be a destination for that node. The selection of source-destination pair is done randomly.

2.2. Attack Model. The existence of the reputation system does not imply a complete solution for all security problems. Our proposed solution tries to solve a particular security problem that is related to nodal behavior in the routing operation, as has been discussed earlier. Thus, some reasonable assumptions are made to make the work more focused on our problem.

(i) The system assumes always suspicious nodes. This means that a node cannot be fully trusted. Every node is assumed to have a minimum risk value that can be encountered if that node is used as a router.

(ii) The system assumes collusion-free attacks. The design of the system, however, can be easily modified to handle collusion based attacks since we adopt modular design. Changes need to be done in the

rating component. This can be considered for future work.

(iii) The system treats only one type of behavior related attacks, that is, nonforwarding attack. In this attack, when a malicious node receives a packet to forward, it drops this packet with a certain probability that will represent its actual risk value.

(iv) The system assumes honesty in treating information exchange about nodes energy levels or risk values. Honesty can be accounted for in the rating component. However, we left this aspect for future studies.

2.3. Reputation System Model. Our reputation system consists of three main components, that is, monitoring component, rating component, and response component.

2.3.1. Monitoring Component. The monitoring component observes packet forwarding events. A monitoring node will apply a watchdog mechanism by which it will be continuously monitoring other neighboring nodes for possible non forwarding attacks. When a misbehaving event is detected, it is counted and stored until an update time T_{update} is due. Then a report is sent to the rating component, CRATER.

2.3.2. Rating Component: CRATER. The rating component, CRATER, evaluates the amount of risk an observed node would provide for the routing operation. The risk value is a quantity that represents previous misbehaving activities that a malicious node (a node that drops packet) obtained. This value is used as an expectation for how much risk would be suffered by selecting that malicious node as a router. It is calculated based on first hand information (FHI) and second hand information (SHI). FHI is achieved by the direct observation done by the node of concern. Risk values are updated based on the FHI every time a new misbehavior report is received from the monitoring component. Moreover, if an observed node shows an idle behavior during a certain period, its risk value is reduced. A monitor also updates the risk values of its neighbors by SHI received periodically from some announcers.

2.3.3. Response Component. The response component in our system is a trust aware version of the GEAR routing protocol [10]. Our protocol incorporates risk values computed by rating component along with distance and energy information to choose the best next hop for the routing operation. A node will only try to avoid malicious nodes. We call this as a defensive approach. A future possible enhancement is to allow a node not to forward packets initiated from a malicious node as a response. However, we are not considering such a mechanism in this current work.

3. Routing Events Monitoring

In monitoring operation, a node will record any new packet transmission that it can overhear. The following algorithm is used to identify misbehavior events.

(i) Record each overheard packet transmission.

(ii) Search for a match for that packet in a monitoring queue.

(iii) If a match is found, delete the packet from the monitoring queue. A match here corresponds to a match in source ID, destination ID, and previous hop ID.

(iv) If the match is not found, then if the next hop node in the packet is a neighbor, that is, it can be monitored, add the recorded packet as a new entry to the monitoring queue; otherwise, ignore the packet.

(v) If an update period T_{update} passes, clear the monitoring queue. This step provides a maximum period (T_{update}) allowed to validate that a node has forwarded a packet.

(vi) After each T_{update}, report the number of misbehaving events for each monitored node to the rating component.

4. Rating Component: CRATER

In this work, our proposed rating technique is called Cautious Rating for Trust Enabled Routing (CRATER). Basically, this technique identifies three rating factors: first hand information (FHI), second hand information (SHI), and neutral behavior period (NBP). FHI is the information gathered by direct monitoring and interaction between the monitoring and monitored node. SHI is the opinion of other nodes about a monitored node. NBP is a period during which a node is not doing any routing activity. The new contribution in CRATER is its mathematical approach that is used to rate nodes based on what we call cautious assumptions.

4.1. Cautious Assumptions. Rating methodology proposed in CRATER assumes what we call "the cautious assumptions." These assumptions are the following.

(i) Pessimistic start: the default status of a node joining the WSN network is to be untrustworthy. However, its reputation, or what we will call later the risk value, will not be at the extreme level.

(ii) Unreliable SHI: a node tries to be as much independent from SHI as possible to avoid dishonesty issues.

(iii) Rejecting good news: announcing "good news" about other nodes in SHI can be a trial from the announcer to relieve itself from routing duties and put the burden on the others or it can be thought as collusion between the announcer and an attacker. Thus, nodes are not interested in hearing good news. On the other hand, "bad news" is very much welcomed. The differentiation between these good or bad announcements is realized by a threshold.

(iv) Local interest: this means that a node is only interested in rating its immediate neighbors.

In CRATER, each node rates its neighbor by assigning a risk value to the corresponding monitored node. The risk value of node j assigned by node i, $r_{i,j}$ is defined as a quantity that represents how much risk the node i will encounter when it uses node j as a next hop to route its packets. This value ranges from 0 to 1 where 0 represents the minimum risk and 1 represents the maximum risk. The reputation of node j as per node i is then computed as

$$\text{rep}_{i,j} = 1 - r_{i,j}. \tag{1}$$

CRATER operation is based on rating the nodes on the risk notion. Each node evaluates the risk values of its neighbors and takes the proper action based on the values it obtains. Risk values calculations are affected by the three factors, that is, FHI, SHI and NBP. Each node in the system continuously and periodically updates the risk values of its neighbors based on the information collected during these update periods. The general algorithm that a node i follows to rate its neighbor j is what follows.

(i) node i monitors node j for the duration of the update period, T_{update}.

(ii) at the end of each update period, do the following:

  (a) calculate $r_{i,j,\text{FHI}}$ using the new FHI
  (b) update the old risk value, $r_{i,j,\text{old}}$ using the new calculated $r_{i,j,\text{FHI}}$ to get $r_{i,j}$
  (c) calculate the $r_{i,j,\text{SHI}}$ using the SHI
  (d) update $r_{i,j}$ using the $r_{i,j,\text{SHI}}$
  (e) update $r_{i,j}$ if neutral behavior periods are realized.

4.2. Rating on First Hand Information. During an update period, node i monitors its neighbor j. Based on the outputs of this monitoring operation, the value of $r_{i,j,\text{FHI}}$ is calculated. All risk evaluation formulas are based on the frequency of misbehaviors (the number of packets that are dropped over a period of time regardless of the total transmitted packets, assuming error free channel). Adopting such approach instead of considering the rate (i.e., dropped/transmitted) as a measure of trustworthiness will prevent forwarder nodes from taking advantage of their status and starts dropping more packets and eventually, it deceives the overall system. This is another interesting feature of our reputation system.

Let us define the following quantities

(i) $c_{i,j}$: the occurrence count of node j misbehavior that is monitored by node i.

(ii) T_{update}: the length of the update period during which the misbehavior of node j monitored by i occurs.

(iii) $f_{i,j}$: the frequency of node j misbehavior that is monitored by node i. Thus, $f_{i,j}$ can be calculated as follows:

$$f_{i,j} = \frac{c_{i,j}}{T_{\text{update}}}. \tag{2}$$

(iv) $f_{\max}$: a maximum misbehavior frequency value that can be tolerated by the reputation system. In fact, $f_{\max}$ can be used to account for false positives, that is, drops that are not related to attacks. In some practical scenarios, if the channel is known to have lots of collisions or if we allow node mobility in the system, $f_{\max}$ can be used to tolerate these factors. For example, if we estimate that a channel would have a collision rate of 2 packets/second; $f_{\max}$ should be designed to be greater than 2 since we know that we will encounter some drops due to collisions. However, modeling $f_{\max}$ with these factors requires much more in-depth analysis. In this work, we just focus on looking at its effect as an input to the rating system.

Given the previous parameters, the risk value $r_{i,j,\text{FHI}}$ assigned by node i to j on FHI is calculated and normalized as follows:

$$r_{i,j,\text{FHI}} = \frac{f_{i,j}}{f_{\max}}. \tag{3}$$

However, $r_{i,j,\text{FHI}}$ in (3) can be greater than 1. Thus, to ensure that $r_{i,j,\text{FHI}} \in [0,1]$, the quantity $f_{i,j}/f_{\max}$ should be less than 1. Thus (3) is rewritten conditionally as follows:

$$r_{i,j,\text{FHI}} = \frac{f_{i,j}}{f_{\max}}, \quad \text{where } \frac{f_{i,j}}{f_{\max}} < 1. \tag{4}$$

In fact, the case where $f_{i,j}/f_{\max} > 1$ indicates a serious misbehavior event that cannot be tolerated by the reputation system, since $f_{\max}$ represents the maximum tolerable misbehavior. In that case, the node will be assigned the maximum risk value, that is, 1. Now, once $r_{i,j,\text{FHI}}$ is obtained, node i should update the old risk value $r_{i,j,\text{old}}$.

It is well known that the trust is originally a social value and it is a very complex issue. Hence, the proposed approach tried to tackle the trust problem thoroughly via identifying the different cases and find a way to characterize each case uniquely and then propose a method to assess the risk/trust properly. In this work, CRATER updates $r_{i,j,\text{old}}$ differently based on the value of $r_{i,j,\text{FHI}}$. We can consider the following three cases.

Case 1 ($r_{i,j,\text{FHI}} = 0$). If $r_{i,j,\text{FHI}}$ is equal to zero, it means that node j has proved a good behavior during the update period (Remember that if node j was idle, it will be considered as a neutral behavior period and $r_{i,j,\text{FHI}}$ will not have a value, hence, no update to $r_{i,j}$ will be done at this step). In this case of $r_{i,j,\text{FHI}} = 0$, $r_{i,j,\text{old}}$ should be updated to have a new value smaller than the old one because node j has proved a good behavior . The updated value of $r_{i,j}$ will be recalculated as

$$r_{i,j,\text{new}} = r_{i,j,\text{old}} \times \left(1 - \theta_{i,j}\right), \tag{5}$$

where $\theta_{i,j}$ is a reduction factor $\in [0, \theta_{\max}]$ and $\theta_{\max}$ is a global maximum reduction factor allowed by the whole reputation

system and $\theta_{\max} < 1$. We can notice that $\theta_{i,j}$ differs according to the monitored node. The reason is that $\theta_{i,j}$ should reflect the trust relationship between node i and j, that is, $\text{Trust}_{i,j}$.

We define the trustworthiness of a node j with respect to i as follows:

$$\text{Trust}_{i,j} = 1 - \frac{r_{i,j}}{r_{i,\text{th}}}, \tag{6}$$

where $r_{i,\text{th}}$ is the maximum risk level a node can exhibit beyond which it cannot build a trust relationship with node i. If $\text{Trust}_{i,j} = 1$, node j is fully trusted. If $0 \leq \text{Trust}_{i,j} < 1$, node j is trusted with some risk as $\text{Trust}_{i,j}$ decreases towards 0. When $\text{Trust}_{i,j} \leq 0$, j is never trusted.

Given this trust notion, $\theta_{i,j}$ in (5) can be calculated as follows:

$$\theta_{i,j} = \theta_{\max}\text{Trust}_{i,j}. \tag{7}$$

Since the reputation system assumes an always suspicious environment, $r_{i,j}$ cannot reduce indefinitely. Thus, a reduction will be allowed as long as the new value of $r_{i,j}$ will be greater than or equal to a minimum allowed value $r_{\min}$. We can notice here that the better the reputation of a node (i.e., the lower its risk value is), the more reduction it will acquire.

If $r_{i,j,\text{FHI}}$ is not equal to zero, we look at the following other two cases.

Case 2 ($r_{i,j,\text{FHI}} > r_{i,j,\text{old}}$). In this case, the new risk value will be updated and biased to the current value, that is, $r_{i,j,\text{FHI}}$. This is to punish the misbehaving node according to how much it misbehaves more than the expectation of staying at $r_{i,j,\text{old}}$. The update methodology used here in CRATER is similar to the average exponential weighting. The equation used to calculate the new risk $r_{i,j,\text{new}}$ given the old value $r_{i,j,\text{old}}$ and the current FHI risk value $r_{i,j,\text{FHI}}$ is as follows:

$$r_{i,j,\text{new}} = \lambda r_{i,j,\text{FHI}} + (1-\lambda) r_{i,j,\text{old}}. \tag{8}$$

Here, λ is a real number $\in (0.5, 1]$ that represents a preference parameter to indicate the importance of the history of FHI embedded in $r_{i,j,\text{old}}$ and the current $r_{i,j,\text{FHI}}$. In CRATER, λ is a tunable design parameter that depends on the difference between the current and old risk values, that is,

$$r_{\text{diff}} = r_{i,j,\text{FHI}} - r_{i,j,\text{old}}. \tag{9}$$

If the difference between the two risk values is insignificant, λ should be moderate to the value 0.5. As the difference increases, λ should increase because the current risk value is more and it predicts more about the future than the history. So, λ is modeled by the following equation:

$$\lambda = 0.5(1 + r_{\text{diff}}). \tag{10}$$

Case 3 ($r_{i,j,\text{FHI}} \leq r_{i,j,\text{old}}$). Here, although j has equal or better current observation results than previous observations, it is still misbehaving. Thus, we still should punish node j and increase its risk value. However, this time the increase will depend on a discouragement and attraction strategy. If a node has a low risk value, it will be punished more compared to a node with higher risk. This is to discourage any further trials from the lower risk node. In the same time, the higher risk node will be attracted to behave better in the future by increasing its risk value slightly. This will not affect the rating fairness because the higher risk node is already in a very serious situation and increasing its risk value greatly or slightly will not have a significant difference.

Mathematically, the increment of the risk value should decrease as $r_{i,j,\text{old}}$ increases. Since $r_{i,j,\text{old}} \in [0,1]$, we can relate the increment to $(1 - r_{i,j,\text{old}})$. Then, the increment ε can be modeled as

$$\varepsilon = \varepsilon_0\left(1 - r_{i,j,\text{old}}\right), \tag{11}$$

where ε_0 is a value representing the relation constant. However, it is better to reflect this constant in the lights of the old and current FHI so that if the current value is very close to the old value, the increment should increase. So, ε_0 should be related to the ratio between the current and the old risk values. Moreover, if the current value itself is large, the increment should also be more. Thus ε_0 should be also related to the current value. As a result, ε_0 can be modeled by:

$$\varepsilon_0 = r_{i,j,\text{FHI}} \times \frac{r_{i,j,\text{FHI}}}{r_{i,j,\text{old}}} = \frac{r_{i,j,\text{FHI}}^2}{r_{i,j,\text{old}}}. \tag{12}$$

Then, (11) is rewritten as

$$\varepsilon = \frac{r_{i,j,\text{FHI}}^2}{r_{i,j,\text{old}}} \times \left(1 - r_{i,j,\text{old}}\right) = \frac{r_{i,j,\text{FHI}}^2}{r_{i,j,\text{old}}} - r_{i,j,\text{FHI}}^2. \tag{13}$$

Notice that ε is guaranteed to be always positive since $r_{i,j,\text{old}} < 1$. Finally, the updated value $r_{i,j,\text{new}}$ is the old value incremented by ε

$$r_{i,j,\text{new}} = r_{i,j,\text{old}} + \varepsilon = r_{i,j,\text{old}} + \frac{r_{i,j,\text{FHI}}^2}{r_{i,j,\text{old}}} - r_{i,j,\text{FHI}}^2. \tag{14}$$

4.2.1. Discussion. The proposed approach as mentioned in several places in the paper is a suspicious approach. Therefore, when a node tries to show "good" behavior, the system will be suspicious and its new risk value gets worse. On the same direction, when the node's FHI is higher than the old value, its new risk value will be higher but not with the same rate as the case where the FHI is greater than the old risk value (i.e., Case 2). On the other hand, the trust theorem still applies but not immediately. The node should show this "good" behavior for sufficient time and then its risk value will get lower (more trusted).

4.3. Rating on Second Hand Information. Due to the assumption of rejecting good news, accepting SHI is governed by a threshold value. When a node k wants to announce to node i the risk value it obtained about j, it sends its current first hand observation risk value, that is, $r_{i,j,\text{FHI}}$. When node i receives $r_{k,j,\text{FHI}}$, it will compare it with the SHI acceptance

threshold, that is, $r_{k,j,\mathrm{SHI}}$. If $r_{k,j,\mathrm{FHI}} > r_{th,\mathrm{SHI}}$, it will accept this SHI announcement. Otherwise, it will ignore it.

When node i receives all SHI regarding node j, it calculates the corresponding rating of node j based on SHI, that is, $r_{i,j,\mathrm{SHI}}$. This step should account for the concept of accuracy of the reported information. Accuracy is the term used to represent how much a reported information deviates from the actual reading. There are many ways to account for accuracy when calculating $r_{i,j,\mathrm{SHI}}$. One approach that we use in CRATER is to take the average of the reported SHI. Thus, $r_{i,j,\mathrm{SHI}}$ is calculated as

$$r_{i,j,\mathrm{SHI}} = \frac{\sum_{\forall k} r_{i,k,\mathrm{FHI}}}{K}, \tag{15}$$

where K is the number of accepted reporters or announcers. If $K = 0$, no SHI update will be done.

Once $r_{i,j,\mathrm{SHI}}$ is calculated, the risk value $r_{i,j}$ will be updated to get $r_{i,j,\mathrm{new}}$ by considering the old value $r_{i,j,\mathrm{old}}$ and $r_{i,j,\mathrm{SHI}}$. The update methodology will follow a similar approach to the exponential average weighting approach by the following equation:

$$r_{i,j,\mathrm{new}} = \omega r_{i,j,\mathrm{old}} + (1-\omega) r_{i,j,\mathrm{SHI}}. \tag{16}$$

Here, ω is a real number $\in [0, 1]$ that represents a preference parameter to indicate the importance of the history of the node rating and the SHI. In our system, ω is a tunable design parameter that depends on the difference between the old rating risk value and SHI risk value, that is,

$$r_{\mathrm{diff}} = r_{i,j,\mathrm{old}} - r_{i,j,\mathrm{SHI}}. \tag{17}$$

If the difference between the two risk values is insignificant, ω should be moderate to the value 0.5. As the difference increases positively or negatively, ω should increase because we want to rely on the old experience due to the unreliable SHI assumption, which is one of the previously mentioned cautious assumptions. Since we want the preference to be always associated with the old rating over the SHI, we consider the absolute value of the difference rather than the signed difference. So, ω can be modeled by the following equation:

$$\omega = 0.5(1 + |r_{\mathrm{diff}}|). \tag{18}$$

4.3.1. Example. Let us assume $r_{i,j,\mathrm{old}} = 0.1$ and $r_{i,j,\mathrm{SHI}} = 0.4$, then using (16), $r_{i,j,\mathrm{new}} = 0.205$. If however $r_{i,j,\mathrm{SHI}} = 0.9$, then $r_{i,j,\mathrm{new}} = 0.18$. This appears as a paradoxical; how can a very negative SHI (risk of 0.9) have a smaller impact than a less negative SHI (risk of 0.4)? This issue can be explained as follows. In our approach, we do not want to make SHI to deviate our measurements far from old values. Therefore, the SHI measurements that deviate new risk measurements far away from the old ones are not well respected. Using such approach should minimize the bad mouthing nodes.

4.4. Rating on Neutral Behavior. When node j is observed by i for n consecutive update periods to be idle in its behavior, node i will give node j a chance to be more trusted by reducing its current risk value. A node is considered to be in idle behavior if it does not perform any routing operation. The reduction procedure follows exactly the same methodology explained in rating based on FHI when $r_{i,j,\mathrm{FHI}} = 0$. The only difference here is that in the case of neutral behavior the update is done after we observe such behavior during n consecutive update periods whereas it is done immediately after an update period in the case of $r_{i,j,\mathrm{FHI}} = 0$. The choice of n is a design parameter that depends on how much a network is tolerable against attacks. High values of n mean that we are not willing to forgive malicious nodes quickly.

4.5. CRATER Evaluation Using RESISTOR. As any rating mechanism, CRATER needs to be evaluated to see how various rating factors affect trust evolution and risk evaluation. One approach is to see how the risk value is evolving during network operation. In this work, we enhance this evolution mechanism using a new technique that we call *REputaion Systems-Independent Scale for Trust On Routing* (RESISTOR).

In RESISTOR, we introduce a new metric called the resistance metric. The resistance between node i and a malicious node j in the direction from i to j is denoted by $\mathrm{RES}_{i,j}$. It is defined as the ratio of the risk value $r_{i,j}$ to the number of packets that flow from node i to j; $P_{i,j}$. Mathematically:

$$\mathrm{RES}_{i,j} = \frac{r_{i,j}}{P_{i,j}}. \tag{19}$$

Thus, a good reputation system must provide high resistance. A perfect reputation system should provide an infinite resistance since $P_{i,j} = 0$.

For reputation systems evaluation purpose, RESITOR works as follows.

(i) For each node i in the network, do the following steps at the end of each update period, T_{update}:

  (a) at the end of each update period, node i computes $r_{i,j}$ for all neighbors,

  (b) at the end of each update period, node i knows how many packets have been forwarded to its neighbor, j,

  (c) for each malicious neighbor, node i will compute its resistance against that malicious node j as

$$\mathrm{RES}_{i,j} = \frac{r_{i,j} - r_{i,\min}}{P_{i,j}}, \tag{20}$$

where $r_{i,\min}$ is the minimum risk value among its neighbors and $P_{i,j} \neq 0$. Please notice that when $r_{i,\min} = r_{i,j}$, the node i is either completely surrounded by malicious nodes or it has only one neighbor who is malicious. In either case, if $P_{i,j} \neq 0$, $\mathrm{RES}_{i,j} = 0$ which reflects that i is not able to resist node j.

(i) If $P_{i,j} = 0$; i will not compute $\mathrm{RES}_{i,j}$. This is because j will be considered as if it does not exist.

(ii) Compute the average resistance of node i against its neighborhood $\text{RES}_{i,\text{avg}}$ as the arithmetic mean of all $\text{RES}_{i,j}$, that is,

$$\text{RES}_{i,j} = \frac{\sum_{\forall j} \text{RES}_{i,j}}{m}, \tag{21}$$

where m is the number of malicious neighbors and j is neighboring malicious nodes. If $m = 0$, $\text{RES}_{i,\text{avg}}$ is set to 0.

(iii) Repeat all the previous steps, but this time assume that $r_{i,j}$ is the expected theoretical value $r_{i,j,\text{theoritical}}$. In the case of nonforwarding attack, like in this work, we can model $r_{i,j,\text{theoritical}}$ as the probability of dropping a packet. Compute then the corresponding $\text{RES}_{i,\text{avg,theoritical}}$. Notice that $P_{i,j}$ is the same in the theoretical or actual calculations. The rational behind this step is to weigh the short-term resistance value to the long-term resistance value and this what we called Resistance Figure.

(iv) Compute the resistance figure $\text{RES}_{i,\text{fig}}$ of a node i as:

$$\text{RES}_{i,\text{fig}} = \frac{\text{RES}_{i,\text{avg}}}{\text{RES}_{i,\text{avg,theoritical}}}. \tag{22}$$

(v) Compute the average resistance figure of all nodes $\text{RES}_{\text{avg,fig}}$ as the arithmetic mean of all $\text{RES}_{i,\text{fig}}$, that is,

$$\text{RES}_{i,\text{fig}} = \frac{\sum_{\forall i} \text{RES}_{i,\text{avg}}}{\text{Number of nodes in the network}}. \tag{23}$$

(vi) Plot the obtained values of $\text{RES}_{\text{avg,fig}}$ versus their corresponding update times and analyze the behavior of the curve.

4.6. Validation Experiments. Before analyzing out reputation system performance, we need to make sure that CRATER is working as required. Thus, we provide some validation tests to investigate following points

(i) The effective role of FHI rating, SHI rating, and neutral behavior related rating. The purpose is to see how much these factors affect CRATER.

(ii) The effect of the frequency of rating updates, that is to see if very frequent updates can improve the resistance significantly or not.

(iii) The effect of changing some threshold parameters on the resistance of the system so that better choices can be adopted for those that provide higher resistance.

Table 1 summarizes all experiments' parameters.

Figure 1 shows the resistance figure for CRATER versus time for two cases. In the first case, the thick curve, CRATER rates nodes based on FHI only. In the second case, the thin dotted curve, CRATER rates nodes on FHI and allows a reduction of the risk level of nodes if a neutral behavior period (NBP) is observed for 10 consecutive update periods.

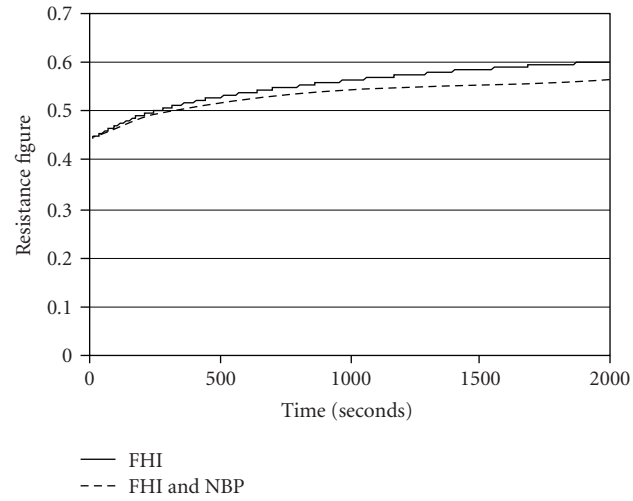


Figure 1: The resistance figure for FHI with and without neutral behavior period (NBP).

The figure shows that when CRATER implements FHI only, the resistance is higher than the case when it allows for NBP. The reason is that when NBP is allowed, its main role is to provide a chance for those idle malicious nodes to be more engaged in the routing operations by reducing their risk values. The lower resistance of that case proves that CRATER works as expected in terms of NBP.

Another important point to note here is the curve convergence issue. We can see that the curves are strictly increasing in a nonlinear trend with time. If the curves will converge, they have to converge at a value close to one, as explained earlier. However, it seems from curves behavior that the curve is very slowly converging since it increases from 0.45 at $t = 0$ to 0.6 at $t = 2000$ seconds in case of FHI. This slow convergence is due to the choice of rating parameters, as will be discussed later.

In Figure 2, we are studying the effect of adding SHI as a rating factor in CRATER. The same rating parameters used for FHI in Figure 1 are used here. The left side of the figure shows the resistance in compressed scale, while the right hand side shows the same figure magnified on a detailed scale.

Before analyzing the curves, we should highlight the role of SHI in CRATER. SHI should assist in rating a certain node in a way that makes everyone has similar opinion about that node. To illustrate this point, assume that nodes A and B are interested in rating node C. Assume also that initially, $r_{A,C} = 0.9$ and $r_{B,C} = 0.5$. If SHI is not allowed, A and B may still have the same gap in their ratings for node C. However, when SHI is allowed, A and B will exchange their knowledge about C and adjust their ratings accordingly. Ultimately, both of them will have risk values on C that are close to each other.

Now, back to Figure 2, we can see in the left side that the resistance is almost constant. A constant resistance implies a convergence situation, which should happen when the

Table 1: Simulation parameters for CRATER experiments.

Parameter	Value	Parameter	Value
$f_{\max}$	5 dps (drops per second) if it is not changing as per the simulation objective	Simulation period	2000 seconds
$r_{i,\text{th}}$	0.9	Number of nodes	100
Default risk value	0.5	Deployment	random
Minimum risk value	0.1	Network size	100∗100 squared units
SHI acceptance threshold	0.5	Node transmission range	15 units
T_{update}	5 seconds if it is not changing as per the simulation objective	Monitoring mode	Promiscuous
$\theta_{\max}$	0.01 if it is not changing as per the simulation objective	Attack type	Nonforwarding with probability of dropping = 1
Mean arrival rate	1 pps	Attacker percentage	50%
Mean service rate	500 pps	Attackers deployment	Random
Queuing model	M/M/1	NBP consecutive periods	10 periods
Routing protocol	GEAR	$P_{i,j}$	1

resistance figure is equal to 1. However, the curve shows that this convergence happens at a value around 0.4475, which is much less than 1. This can happen only if FHI is suppressed by another factor that is trying to reduce FHI-related resistance, while at the same time; it tries to keep the ratings at a "global opinion" level. This is exactly what SHI role is supposed to be. This effect of SHI is much clearer in the right side of Figure 2 where we can see how the resistance curve is alternating around an average of 0.4475 as if SHI is competing FHI in a trial to keep the resistance around that value. The convergence at the value 0.4475 is not the ideal case. Where to converge is actually related to the rating parameters.

Figure 3 shows the resistance curve for CRATER considering all rating factors, that is, FHI, SHI, and NBP. The same parameters used for Figures 1 and 2 are used here. The left side provides a compressed scale while the right one gives the same curve in a detailed scale. If we compare Figure 2 with Figure 3, we can notice that there is no big difference between the two situations. This is because Figure 3 differs from Figure 2 by the addition of NBP in rating calculations. As we have seen in the analysis of Figure 1, NBP does not affect the FHI rating very much. As a result, NBP has transparent effect on CRATER under these settings and conditions.

Figure 4 studies the impact of the frequency of rating updates on the system resistance. The figure studies the resistance of CRATER considering FHI. Three cases are provided here, that is, when the updates are done every 2 seconds, 5 seconds, and 10 seconds. We can notice that as the updates are done more frequently the resistance gets higher values and converges faster towards 1. For example, with the updates done every 2 seconds, the resistance is 0.8 at $t = 1000$ seconds, whereas it is equal to 0.45 when they are done every 10 seconds. Although the rate of attack is still the same, with frequent updates, CRATER punishes the malicious nodes in smaller increments in their risk values, but more frequently. This accumulates at a larger risk value as compared with less frequent updates. As a result, fast convergence and high resistance can be achieved with more frequent updates. However, remember that we are working in WSN environment where this can be an unnecessary overhead that consumes resources.

Figure 5 analyzes the effect of varying $f_{\max}$ on the resistance of CRATER as FHI rating is concerned. Remember that $f_{\max}$ was defined as the maximum misbehavior frequency

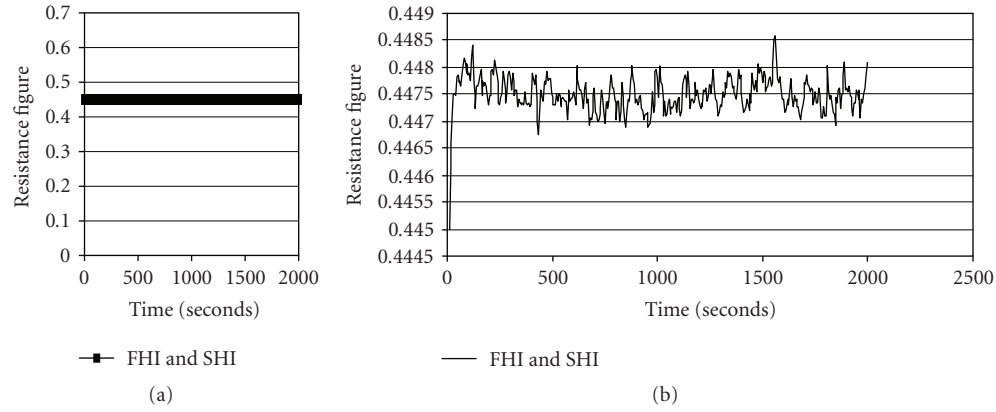


Figure 2: The effect of SHI on resistance figure: (a) compressed scale, (b) detailed scale.

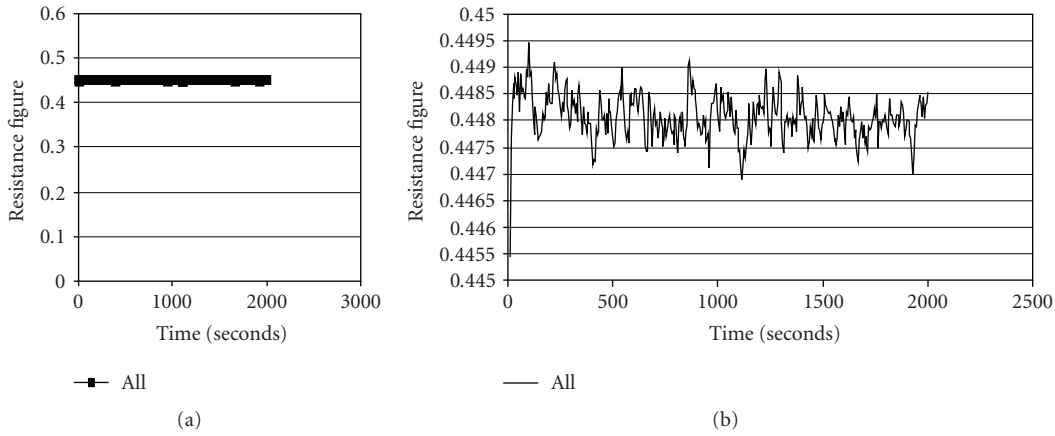


Figure 3: The RESISTOR curve for CRATER with all rating factors, that is, FHI, SHI, and neutral behavior: (a) compressed scale, (b) detailed scale.

value that can be tolerated by the reputation system. So, when we decrease the value of $f_{\max}$ we should expect a very sensitive system that will assign much higher risk values for malicious nodes as compared to high $f_{\max}$ value case. Thus, we expect to have higher resistance with low values of $f_{\max}$.

Figure 5 shows that as we decrease $f_{\max}$ from 10 dropped packets per second (dps) to 0.5 dps, the resistance is improving in terms of the convergence value and the convergence speed as well. For example, with $f_{\max} = 10$ dps, the resistance is very slowly increasing and it is operating around 0.43, whereas with $f_{\max} = 0.5$ dps, the system very early jumps to 0.85 at around $t = 500$ seconds. Although the $f_{\max} = 0.5$ dps provides better resistance, it can cause a situation where we overestimate the misbehaving nodes. In such cases, the resistance may exceed 1. This can happen, for example, if the attacker drops the packet with probability less than 1. In that case, $\text{RES}_{i,\text{avg,theoritical}}$ can be less than $\text{RES}_{i,\text{avg}}$ due to $f_{\max}$. However, in this section, we are studying the non forwarding attack with dropping probability = 1. Thus, the system does not overestimate nodes' behavior as they are all at their maximum risk value when calculating $\text{RES}_{i,\text{avg,theoritical}}$. Thus, $\text{RES}_{i,\text{avg,theoritical}}$ will be always greater than or equal to $\text{RES}_{i,\text{avg}}$, and, consequently, the resistance figure will be always less than or equal to 1.

5. Response

Once a node obtains risk information about its neighbors, a routing decision should be made regarding its future transaction. In our system, we modify GEAR protocol, which is geographic and energy aware routing protocol, to have the additional feature of trust awareness. Trust awareness is achieved by the rating functionality that will feed the routing protocol with the trust metric, which is basically the risk values, $r_{i,j}$. The risk value $r_{i,j}$, as discussed earlier, is a quantity that reflects, to some extent, the expectation that a node j will not forward the packet received from node i, assuming non forwarding attack.

The risk value metric, along with distance and energy metrics, is used to compute a learned cost function for each neighbor. The concerned node, then, makes the routing decision by selecting the neighbor of the lowest cost. The cost function that will be used to select the best router is as follows:

$$t(j,R) = \beta\left(r_{i,j}\right) + (1-\beta)\left[\alpha d(j,R) + (1-\alpha)e(j,R)\right], \tag{24}$$

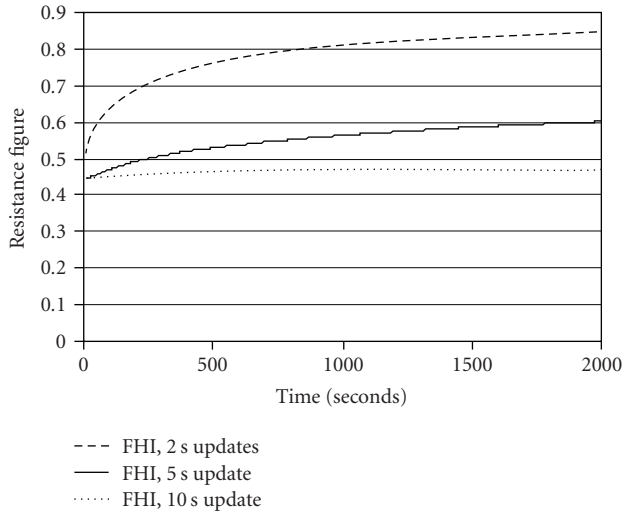


Figure 4: Studying the effect of update periods frequency on the resistance figure considering FHI factor.

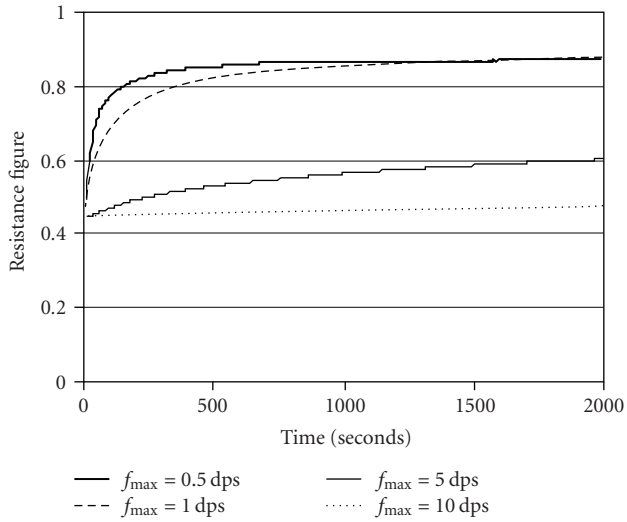


Figure 5: Studying the effect of f_{max} on the resistance figure considering FHI factor.

where

(i) $t(j, R)$ is the *trust-aware* cost of using the node j by node i as a router to the destination R. $r_{i,j}$ is the *risk value* that node i so far knows about node j.

(ii) $d(j, R)$ is the normalized distance from j to R (the distance from j to R divided by the distance from the farthest neighbor of i to R).

(iii) $e(j, R)$ is the so far normalized consumed energy at node j which is announced periodically every T_{update}.

(iv) α is a tunable parameter $\in [0, 1]$ to give more preference to distance or energy.

(v) $[\alpha d(j, R) + (1 - \alpha)e(j, R)]$ is the GEAR component of the routing decision.

(vi) β is a tunable parameter $\in [0, 1]$ to give more or less preference to trust as opposed to other resources.

If we are concerned about trust more than other resources, β should be close to 1. When β equals 1, the trust-aware cost will consider only the trust part of (24) and the next hop will be the most trusted one. Setting β to zero, however, turns the protocol to pure GEAR without any security considerations from the routing protocol perspective.

Different than GEAR, our routing operation involves only packet forwarding and does not implement dissemination. This is because in the dissemination phase in GEAR, packets are intended to be forwarded to all nodes in the target region. However, when we consider trust awareness, a misbehaving node should not be given a chance to have the packet since it will not forward the packet. Thus, our protocol continues to forward packets based on the routing decisions made by the learned cost function.

Finally, regarding the problem of void regions, which is the case when a node finds itself the closest to the destination among its neighbors, there is no change in the escaping operation proposed by GEAR. The only difference here is that the reason of being in a void region can be related to the existence of misbehaving nodes in the proximity of the node of interest.

6. Reputation System Resistance Evaluation

In this part of the work, our simulation experiments are set to study the impact of adopting CRATER as a monitoring procedure on the performance on the reputation system. This will be done by studying the evolution of the resistance figure after allowing real interaction between CRATER and our trust-aware routing. The main difference between these experiments and the ones presented in Section 4.6 is that the system was trust unaware in Section 4.6. Thus, packet flow was governed by trust aware decision. Whereas in this section, our routing protocol is trust aware. Thus, rating and packet flow will be definitely impacted by routing decisions. Simulation settings and parameters are provided in Table 2. In this simulation, we will focus on the effect of T_{update} and f_{max} since they represent the key parameters in risk and resistance evolution.

6.1. Varying T_{update}. T_{update} represents the periodicity of information update regarding cost functions and risk evaluation. The more frequent the system is updated, the faster the system can reach the actual risk values of nodes. However, since our trust aware version of GEAR makes relative routing decisions, system performance in terms of delivery ratio (number of successfully delivered packets/total generated packets) cannot be directly related to T_{update} values. This is because each node will ultimately reach the same conclusion about its neighbors in terms of who is more risky than others. If this conclusion is reached at very early stages of the simulation time, the effect of T_{update} will not appear

on routing performance. The investigation of this problem, however, is left for a future work.

In this part of simulation analysis, we are interested in seeing how responsive is our reputation system in relation to T_{update} variation as well as inspecting the stability issues. CRATER parameters used in this experiment are presented in Table 3.

Figure 6 shows the number of dropped packets per a previous T_{update} versus simulation time. We can notice that as T_{update} increases, the dropped packets increase, which is an intuitive result. However, what is important for this analysis is the time at which the number of dropped packets starts to stabilize around the average. The simulation shows the following observation: (after applying initial data deletion technique).

It is very noticeable that as the system gets updated very frequently, that is, as T_{update} gets smaller, the system reaches a stable state much faster, as shown in Table 4.

Moreover, the resistance figure in Figure 7 shows that as T_{update} gets smaller, the stable value of the resistance figure increases. The increase in the resistance figure should be analyzed using the resistance definition, that is, $\text{RES}_{i,j} = (r_{i,j} - r_{i,\min})/P_{i,j}$. Now, $\text{RES}_{i,j}$ gets higher as $r_{i,j}$ increases and $P_{i,j}$ decreases. However, $r_{i,j}$ is mostly affected by FHI calculations as, $r_{i,j,\text{FHI}} = f_{i,j}/f_{\max}$, where, $f_{i,j}$ is given by $f_{i,j} = c_{i,j}/T_{\text{update}}$. However, the ratio $c_{i,j}/T_{\text{update}}$ is fixed and not affected by T_{update} values for the assumption of fixed rate, noncollusion attack. Thus, $r_{i,j}$ is almost unaffected by T_{update} for initial interactions. On the other hand, $P_{i,j}$ gets smaller with T_{update} as it is evident from Figure 6. Thus, $\text{RES}_{i,j}$ becomes higher with smaller values of T_{update}.

The benefit of having high values of resistance is not reflected on the performance of routing protocol, as we explained earlier. However, this trend of resistance figure with T_{update} values has an important application, if we adopt offensive and dismissal response mechanisms. For example, we can apply thresholds to start punishing nodes based on reaching certain resistance values by the whole system. If we have a sever situation where we require fast punishment and critical threshold values, small values of T_{update} like 2 seconds will be the best choice. Of course, this will be at the expense of more overhead, which is beyond the scope of the work objective. Since our routing protocol does not implement such advanced mechanisms, and since changing T_{update} does not have a direct impact on routing performance, the best choice for T_{update} is the one that provides the least overhead, that is, $T_{\text{update}} = 10$ seconds. However, in the remaining simulations we use $T_{\text{update}} = 5$ seconds for the sake of consistency with other simulations.

One last observation to notice here is that the value of the resistance figure in these experiments can exceed 1. This is actually due to the fact that we are allowing the attacker to drop packets with probabilities less than 1. As explained earlier in Section 4.6, this leads to overestimating the risk level of nodes. However, considering cautious assumptions, overestimating in CRATER is acceptable according to these assumptions.

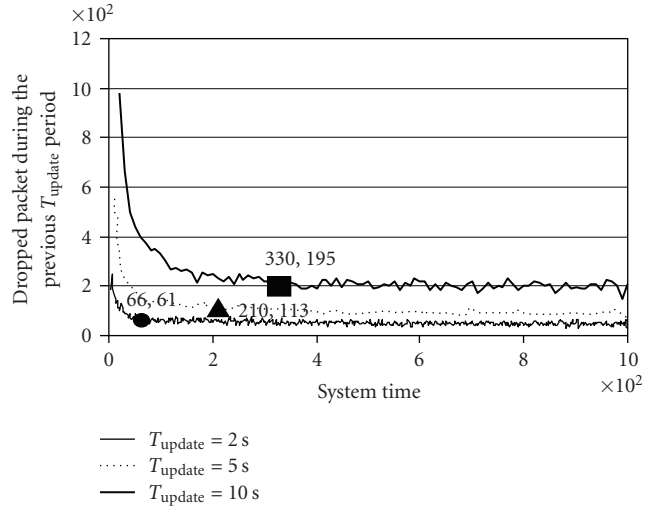


Figure 6: Dropped packets per T_{update} for different T_{update} values.

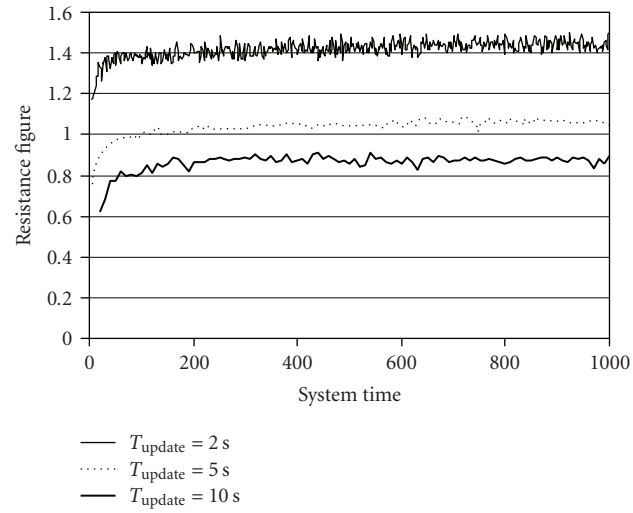


Figure 7: Resistance figure under different values of T_{update}.

6.2. Varying $f_{\max}$. For experiments regarding varying $f_{\max}$, we used the same parameters in Table 3 except that T_{update} is set to 5 seconds and $f_{\max}$ varies as 1, 5, and 10.

As in the analysis of T_{update} impact on routing performance, the same argument is applied here with the variation of $f_{\max}$ (maximum misbehavior frequency value that can be tolerated by the reputation system). Routing performance in terms of delivery ratio is not influenced by changing $f_{\max}$ because the concept of routing decision relativity is still maintained. Figure 8 clearly indicates that aspect since it shows that the number of dropped packets is the same during the simulation time irrespective of $f_{\max}$ value.

However, as $f_{\max}$ decreases $\text{RES}_{i,j}$ increases. That is why the resistance figure becomes higher as $f_{\max}$ decreases in Figure 9. Again, these absolute values of the resistance under the lights of $f_{\max}$ can be utilized to design threshold for advanced response techniques as discussed earlier in the analysis of T_{update}. For example, we can set the value of $f_{\max}$

Table 2: Simulation parameters for repuation system experemints.

Parameter	Value	Parameter	Value
Number of nodes	100 nodes	Queuing model	M/M/1
Network dimensions	Square 90 units $*$ 90 units	Simulation platform	Event driven simulation using Java programming language
Transmission range	15 units	Simulation duration	1000 seconds
Network deployment	Random topology	Retransmission timeout	Explicit retransmission request
Power consumption	1 Watt per reception, 1 Watt per sending, 1 milli-Watt per processing operation	Retransmission trials	Unlimited
Mean arrival rate	1 pps	Update strategy	Periodic, every 5 seconds
Mean service rate	500 pps	α	0.5 (GEAR parameter)
Outsider attackers deployment	Random	Communication discipline	Random source to random destination
Escaping void	Using GEAR part and then distance	Void failure: max number of hops	100
% of attackers	50%	Attackers deployment	Random

Table 3: Simulation parameters for T_{update} variation experiments.

Parameter	Value	Parameter	Value
T_{update}	2, 5, 10 seconds	$f_{\max}$	10
% of attackers	50%	Simulation time	1000 seconds
Number of nodes	100 nodes	Attackers deployment	Random
NMA	$P_{ON1} = P_{ON2} = 1$	β	0.5

Table 4: Packet drops information with different T_{update}.

T_{update}	Stabilization time	Average number of dropped packets
2 seconds	66 seconds	61
5 seconds	210 seconds	113
10 seconds	330 seconds	195

to 1 to have high resistance in sever applications in order to apply isolation mechanisms in an offensive response.

6.3. The Effect of Attacker Population in the Network. It is trivial to conclude that as the attackers' percentage increases in the system, the delivery ratio degrades. However, the purpose of this simulation is to show how much improvement is expected by being exposed to less number of attackers under the lights of various values of β.

In Figure 10, we tested three attackers' percentages, that is, 10, 30 and 50%. We did not go beyond 50% since after that the network is mostly owned by the attacking community. Two important observations can be extracted from Figure 10.

(i) The impact of β (the trust aware preference parameter) on delivery ratio starts to appear significantly after $\beta = 0.4$, which is beyond the value 1/3 that provides equal preference for all factors in routing cost function with $\alpha = 0.5$. This implies that any good system design should consider β values greater than 1/3, irrespective of the attackers' percentage.

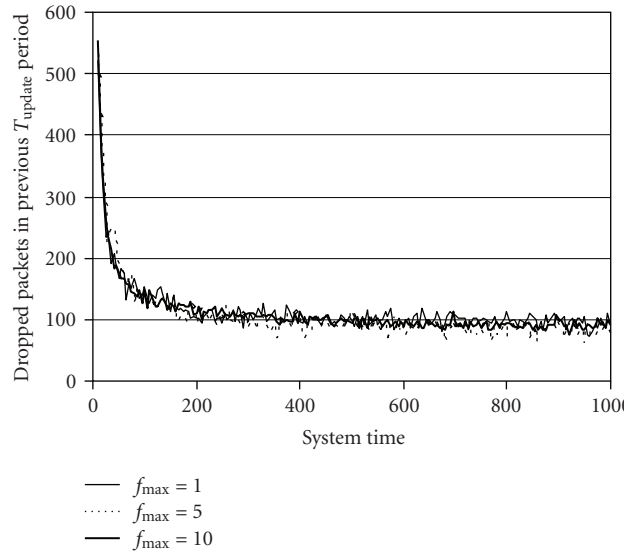


Figure 8: Packet dropping per T_{update} for different $f_{\max}$ values.

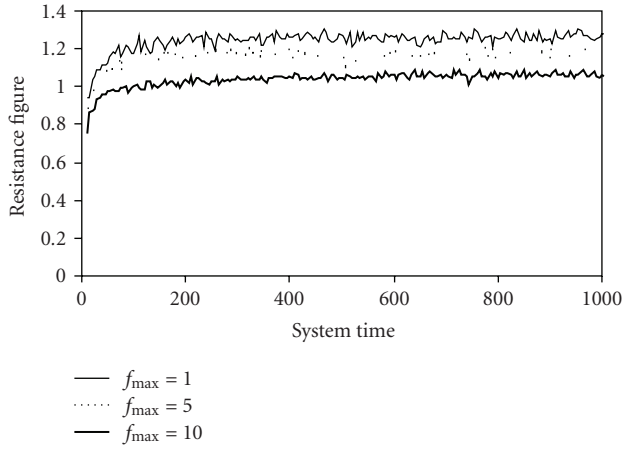


Figure 9: Resistance figure under different values of f_{max}.

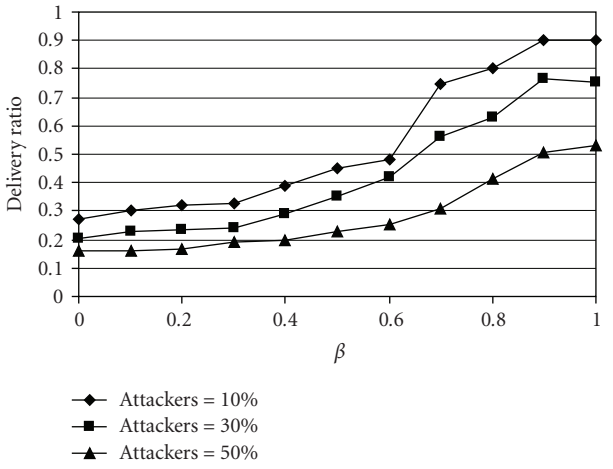


Figure 10: Delivery ratio with various percentages of attackers.

(ii) The delivery ratio improves significantly by reducing the percentage of attackers in the system. For example, at $\beta = 0.9$, the delivery ratio improves from 0.49 to 0.9. Since WSN can be dynamically redeployed, one trick can be used here is to decrease the number of attacker by deploying more "fresh" nodes. However, this guarantees that better nodes will exist in the vicinity of other nodes and they will be more qualified to be routers as opposed to the malicious ones.

Coming to resistance analysis, Figure 11 shows an interesting phenomenon of our RESISTOR tool. That is, the more exposure to attacks the system is, the more resistant the system should be. When the number of attackers is high, more packets will be dropped initially. This is because the alternative routers are also malicious. This implies that the victim node will have better updates on the risk value as it will experience more interactions with malicious nodes. As a result, the risk values will get higher. In a later time, yet not so much late, fewer packets will be delivered per malicious node due to the discovery of its malicious behavior. Thus,

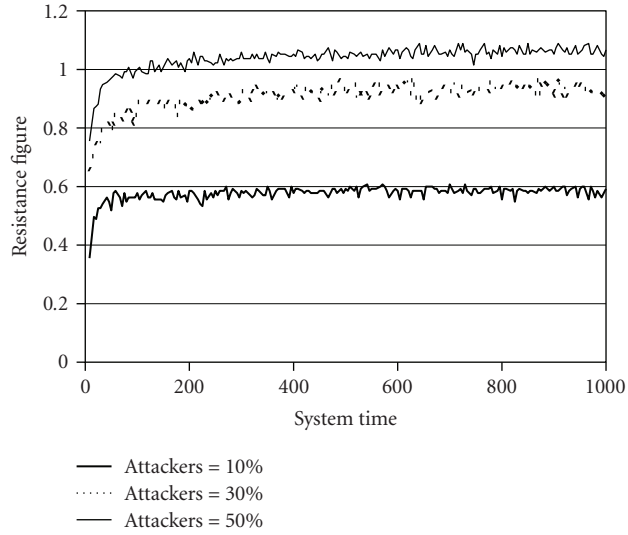


Figure 11: Resistance figure with various percentages of attackers in the integrated system.

ultimately we will have high risk values with few delivered packets per malicious node that implies high resistance. However, although we deliver fewer packets per malicious node in high percentage of attackers, the collective drops due to the population of the attackers sums up to larger drop counts than what is encountered when we have less percentage of attackers where more packets are mistakenly delivered to malicious nodes. This is evident from the delivery ratio results in Figure 10.

7. Related Work

In literature, several famous work deals with behavioral related routing security problems using different approaches. For example, Intrusion-tolerant Routing in Wireless Sensor Networks (INSENS) [11] constructs tree-structured routing for wireless sensor networks (WSNs). It aims to tolerate damage caused by an intruder who has compromised deployed sensor nodes and is intent on injecting, modifying, or blocking packets. INSENS incorporates distributed lightweight security mechanisms, including one-way hash chains and nested keyed message authentication codes to defend against routing attacks such as wormhole attack. Adapting to WSN characteristics, the design of INSENS also pushes complexity away from resource-poor sensor nodes towards resource-rich base stations.

Another work is SeFER [12], which stands for secure, flexible, and efficient routing protocol for sensor networks. It is based on random key predistribution mechanism. This mechanism aims to provide an easy way for managing the keys in WSN without using public key cryptography. The protocol assumes nonsymmetric communication architecture in which a tree of sensor nodes delivers information to a controller according to an inquiry sent into the network. Two nodes may communicate indirectly, but securely over a multiple hop path where each pair of nodes on this path

shares a common key. The protocol provides the methods for nodes to securely share their keys and communicate directly so that the efficiency of communication is increased.

The two previously mentioned protocols are crypto-based solutions. They can successfully fight against attacks in which an intruder falsifies his identity to be a relay for the source such as sybil attack. However, other attacks like selective forwarding, blackhole and HELLO flooding [13] are still possible especially when the attack is performed by an insider node or a node compromised by an intruder. Moreover, any misbehavior due to selfishness or faulty operational nodes cannot be prevented or even detected.

The authors of [8, 14] introduced a mechanism that includes two parts: watchdog and pathrater. The watchdog is the monitoring part that is designed to be responsible for detecting only non forwarding misbehavior. This is accomplished by overhearing the transmission of the next node. The node thus is assumed to be in a continuous promiscuous mode. When the attack is detected, the observing node informs the source of the concerned path.

The pathrater is the component used for reputation. Ratings are kept about every node in the network based on its routing activity and they are updated periodically. Nodes select routes with the highest average node rating. Thus, nodes can avoid misbehaving nodes in their routes as a response. However, misbehaving nodes can still transmit their packets as there is no punishment mechanism adopted here. Moreover, no SHI propagation view is considered which limits the cooperativeness among nodes.

In SORI (Secure and Objective Reputation-Based Incentive Scheme for Ad Hoc Networks) [15], the authors target only the non forwarding attack, as we have implemented in this work. SORI monitors the number of forwarded packets from neighborhood and the number of forwarded packets to neighborhood. Reputation ratings are then acquired by computing the ratio between the two numbers with a consideration for the confidence in the rating proportional to the number of packets that are initially requested for forwarding. SHI is delivered only to the immediate neighbors. This rating source, however, is weighted by what is called credibility, which is derived from the rating ratio. The delivery of the SHI is achieved by hash-chain based authentication.

An important reference for reputation systems in ad hoc networks is Cooperation Of Nodes—Fairness In Dynamic Ad-hoc Networks (CONFIDANT) [16]. It is a reputation-based secure routing framework in which nodes monitor their neighborhood and detect different kinds of misbehavior by means of an enhanced PACK mechanism. The nodes use the second-hand information from others as a resource of rating, as well. The protocol is based on Bayesian estimation that aims to classify other nodes as misbehaving or normal. The observing node excludes misbehaving nodes from the network as a response, by both avoiding them for routing and denying them cooperation. The protocol assumes a DSR operational routing protocol and lacks a provision on WSN constraints and conditions as it is designed for general ad hoc networks.

Another famous reputation mechanism in literature is CORE protocol (Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks) [5]. It is a complete reputation mechanism that differentiates between subjective reputations or observations, indirect reputation which includes only the positive reports by others (SHI), and functional reputation, also referred as task-specific behavior, which are weighted according to a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. The system assumes a DSR routing in which nodes can be requesters or providers. The rating is done by comparing the expected result with the actually obtained result of a request.

The authors of [7] proposed a robust reputation system for P2P and mobile ad hoc networks. Their main contribution is its proposal for a distributed reputation system that can handle false disseminated information. Every node maintains a reputation rating and a trust rating about every node that is of interest. The authors use a modified Bayesian approach so that they will accept only an SHI set that is compatible with the current reputation rating. Also, Trust ratings are updated based on the compatibility of second-hand reputation information with prior reputation ratings. The work avoids exploitation of good behavior that can be incorrectly built over time by introducing a concept of re-evaluation and reputation fading.

The work in [17] is an integrated approach that provides energy, efficiency, reliability, scalability, and support for QoS. It applies TRAP; a trust-aware routing protocol that derives its routes based on link quality and echo ratio (node packets forwarded by j that belong to i to the total broadcasted packets by i) and from both components a reputation system is developed.

An algebraic approach is adopted in [18], where the trust inference problem is modeled as a generalized shortest path problem on a weighted directed graph. A weighted edge from vertex to vertex corresponds to the *opinion* that entity, also referred to as the *issuer*, has about entity. Each opinion consists of two numbers: the *trust* value, and the *confidence* value. To enhance the reliability of the system, multiple trust paths are utilized to compute the trust distance from the source to the destination. The essence of this approach is the two operators used to combine opinions. One operator combines opinions along a path, while the other operator combines opinions across paths. Eventually, we end with solving path problems in graphs, provided that they satisfy certain mathematical properties, that is, form an algebraic structure called a semiring.

Another recent work on developing a reputation system is the one presented in [19]. It incorporates a measure of uncertainty based on subjective logic into the reputation system to reflect the confidence in such system.

The closest work in literature that tackles WSN specifically is RFSN [20]. This work proposed a reputation-based framework for sensor networks, where nodes maintain reputation for other nodes and use it to evaluate their trustworthiness. The authors tried to focus on an abstract view that provides a scalable, diverse, and a generalized approach hoping to tackle all types of misbehaviors. They also designed a system within this framework and employed

a Bayesian formulation, using a beta distribution model for reputation representation.

The system starts the operation by monitoring. Monitoring mechanism follows the classic watchdog methodology in which a node is assumed to be in a promiscuous mode to overhear neighbors' packets. Monitoring behavioral events can result in either cooperative event, α, in which a node is behaving well or noncooperative behavior, β, in which a node misbehaves. The count of each type is injected into the beta distribution formula as the distribution parameters to calculate the node reputation R. This formula calculates node's reputation based on FHI. The reputation is updated based on the new monitoring events, SHI received and according to the age of the current reputation value. Any response action is based on selecting the most trusted node. The trust value of a node that is used for decision making is calculated as the statistical expectation of the reputation value.

RFSN, however, lacks some important points.

(i) The monitoring mechanism uses a normal watchdog mechanism that assumes a promiscuous mode operation for every node. This is not suitable for the WSN conditions in terms of energy scarcity as discussed earlier.

(ii) The work does not propose a response methodology, for example, a routing algorithm. Instead, it leaves it as an open issue. Therefore, the work lacks performance figures that can show the efficiency and security gain and benefits in routing operation that can be obtained in adopting this solution.

8. Conclusion and Future Work

In this paper we proposed a new rating approach for reputation systems in WSN called CRATER. CRATER evaluates nodes reputation by a risk representation. This risk value is computed based on first hand information (FHI), second hand information (SHI), and idle behavior (NBP). The mathematical modeling of CRATER assumes a set of conditions that we define as cautious assumptions in which a node is very cautious in dealing with other's information. Our proposed approach is robust against nonforwarding attack and we hint in our discussion for its ability to mitigate bad mouthing attack. Also, CRATER is a modular approach that can easily be modified to tackle other attacking such as colluding attacks.

CRATER has been evaluated using our novel evaluation technique RESISTOR. Simulation results proved our expectation on how CRATER should behave. CRATER parameters variations were directly reflected in our proposed resistance figure and trust-awareness knowledge evolution.

As a future work, we suggest the following important research directions.

(i) *Using RESISTOR for comparisons among different rating methods*: to further prove the efficiency of RESISTOR as an appropriate tool to measure different reputation systems, we can use other rating techniques and trust evolution algorithms instead of CRATER and then use RESISTOR to evaluate these different systems and compare the results with other evaluation techniques. This can help in achieving standardized mechanisms to evaluate reputation systems.

(ii) *Modifying routing protocol to have offensive and dismissal response*: in our reputation system, our response part performs a defensive function in the sense that it only avoids malicious nodes without any further actions against them. However, we can make use of the obtained risk values and the trust relations to enhance system response to function in offensive manner (e.g., not forwarding malicious nodes' packets) or dismissal manner (e.g., total isolation of malicious nodes). This will be done by designing certain thresholds that determine how and when such actions should be taken. These thresholds will be set by the network operator by the aid of RESISTOR.

Acknowledgments

The authors would like to thank King Fahd University of Petroleum and Minerals for its support. Also, special thank is due to the anonymous reviewers for their valuable comments and input that help in presenting our work better.

References

[1] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[2] A. Jøsang, E. Gray, and M. Kinateder, "Simplification and analysis of transitive trust networks," *Web Intelligence and Agent Systems*, vol. 4, no. 2, pp. 139–161, 2006.

[3] A. Boukerch, L. Xu, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2413–2427, 2007.

[4] A. Rezgui and M. Eltoweissy, "TARP: a trust-aware routing protocol for sensor-actuator networks," in *Proceedings of the IEEE Internatonal Conference on Mobile Ad hoc and Sensor Systems (MASS '07)*, pp. 1–9, Pisa, Italy, October 2007.

[5] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proceedings of the International Conference of Communication and Multimedia Security*, pp. 26–27, Portoroz, Slovenia, September 2002.

[6] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference, e-Reality: Constructing the e-Economy*, Bled, Slovenia, June 2002.

[7] S. Buchegger and J.-Y. Le Boudec, "A robust reputation system for peer-to-peer and mobile ad hoc networks," in *Proceedings of the Workshop on Economics of Peer-to-Peer Systems (P2PEcon '04)*, Harvard University, Cambridge, Mass, USA, June 2004.

[8] S. Buchegger and J.-Y. Le Boudec, "Self-policing mobile ad hoc networks by reputation systems," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 101–107, 2005.

[9] http://www.xbow.com/.

[10] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for

wireless sensor networks," Tech. Rep. UCLA/CSD-TR-01-0023, 2001.

[11] J. Deng, R. Han, and S. Mishra, "Insens: intrusion-tolerant routing in wireless sensor networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS '03)*, Providence, RI, USA, May 2003.

[12] C. C. Oniz, S. E. Tasci, E. Savas, O. Ercetin, and A. Levi, "SeFER: secure, flexible and efficient routing protocol for distributed sensor networks," in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN '05)*, pp. 246–255, Istanbul, Turkey, January-February 2005.

[13] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, no. 2-3, pp. 293–315, 2003, special issue on Sensor Network Applications and Protocols.

[14] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 255–265, Boston, Mass, USA, August 2000.

[15] Q. He, D. Wu, and P. Khosla, "SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, pp. 825–830, Atlanta, Ga, USA, March 2004.

[16] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the CONFIDANT protocol: cooperation of nodes—fairness in dynamic ad-hoc networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 226–236, Lausanne, Switzerland, June 2002.

[17] A. Rezgui and M. Eltoweissy, "μRACER: a reliable adaptive service-driven efficient routing protocol suite for sensor-actuator networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 5, pp. 607–622, 2009.

[18] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 318–328, 2006.

[19] K. Kane and J. C. Browne, "Using uncertainty in reputation methods to enforce cooperation in ad-hoc networks," in *Proceedings of the 5th ACM Workshop on Wireless Security (WiSE '06)*, vol. 2006, pp. 105–113, Los Angeles, Calif, USA, September 2006.

[20] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, article 15, pp. 1–37, 2008.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 946493, 13 pages
doi:10.1155/2009/946493

Research Article

On Multipath Routing in Multihop Wireless Networks: Security, Performance, and Their Tradeoff

Lin Chen and Jean Leneutre

Department of Computer Science and Networking, LTCI-UMR 5141 laboratory, CNRS-Telecom Paris Tech, 46 Rue Barrault, 75013 Paris, France

Correspondence should be addressed to Lin Chen, lchen@enst.fr

Received 29 January 2009; Accepted 1 June 2009

Recommended by Hui Chen

Routing amid malicious attackers in multihop wireless networks with unreliable links is a challenging task. In this paper, we address the fundamental problem of how to choose secure and reliable paths in such environments. We formulate the multipath routing problem as optimization problems and propose algorithms with polynomial complexity to solve them. Game theory is employed to solve and analyze the formulated multipath routing problem. We first propose the multipath routing solution minimizing the worst-case security risk (i.e., the percentage of packets captured by attackers in the worst case). While the obtained solution provides the most security routes, it may perform poorly given the unreliability of wireless links. Hence we then investigate the multipath routing solution maximizing the worst-case packet delivery ratio. As a natural extension, to achieve a tradeoff between the routing security and performance, we derive the multipath routing protocol maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under given threshold. As another contribution, we establish the relationship between the worst-case security risk and packet delivery ratio, which gives the theoretical limit on the security-performance tradeoff of node-disjoint multipath routing in multihop wireless networks.

1. Introduction

It is widely recognized that the intrinsic nature of wireless networks, such as the broadcast nature of the wireless channel and the limited resources of network nodes, makes them extremely attractive and vulnerable to attackers. Routing amid malicious attackers in such environments is a challenging task. On one hand, the most secure route(s) should be chosen such that the percentage of packet captured by attackers is as small as possible. On the other hand, given the unreliability of wireless links, the most reliable route(s) should be selected such that the packet delivery ratio at destination is as high as possible.

A natural approach is to use multiple paths to increase the fault tolerance and the resilience to attackers. However, how to choose the secure and reliable paths among exponentially many candidates and how to allocate traffic among them remain a difficult but crucial problem.

1.1. Paper Overview. In this paper, we address the above fundamental routing problem by focusing on two metrics: route security and performance. We start with the single-attacker case and extend our work to the multiple-attacker case in Section 7.

We first study the multipath routing solution minimizing the worst-case security risk; that is, the percentage of packets captured by the attacker under the condition that the attacker makes all its efforts to maximize this percentage. We model such multipath routing problem as a minimaximization problem and formulate it as the maximum flow problem in lossy networks based on which a routing algorithm with polynomial time complexity being derived to solve it.

While the obtained solution provides the most security routes, which is crucial for security sensitive applications, performance is another important issue that definitively cannot be ignored, especially in wireless networks with unreliable links. To this end, we investigate the multipath

routing solution maximizing the packet delivery ratio under the condition that the attacker makes all its efforts to minimize this ratio. Noticing that solving this problem requires exponential time complexity, we propose a heuristic algorithm computing the optimal path set with polynomial time complexity. In our study, we also apply game theory as a systematic tool to solve and analyze the formulated multipath routing problems.

Next, we extend our efforts to study a natural problem: how to achieve a tradeoff between the route security and performance. In this perspective, we derive the routing solution maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under given threshold. Furthermore, as a theoretical limit on the security-performance tradeoff of node-disjoint multipath routing, we establish the relationship between the worst-case packet delivery ratio a^* and the security risk r^*:

$$a^* \le r^* \left(\left| \mathcal{P}^{\mathrm{nd}} \right| - 1 \right), \tag{1}$$

where $|\mathcal{P}^{\mathrm{nd}}|$ is the maximum number of node-disjoint paths in the network.

By simulation, we evaluate the performance of the proposed multipath routing protocols. The results show that our solutions show the best worst-case security and performance among the simulated multipath routing protocols.

1.2. Background and Motivation. Multipath routing, as mentioned above, is a promising way to improve route reliability and security. Past work on multipath routing in wireless networks mainly consists of evaluating the possible paths via reputation metrics based on security or reliability and distributing traffic among the routes with the highest reputation ratings.

In [1], Papadimitratos et al. proposed an algorithm, called Disjoint Path-set Selection Protocol (DPSP), to find the maximum number of paths between a source and destination with the highest reliability. DPSP tries to find maximum number of node-disjoint paths based on the reliability metric to improve the reliability of communication by increasing the number of used paths.

In [2], Lou et al. proposed another solution for calculating the maximum number of the most secure paths called Security Protocol for REliable dAta Delivery (SPREAD). Their solution relies on previous knowledge of security level of each node and calculates the link costs according to them. It also exploits secret sharing to spread data over multiple paths and proposes a security-optimized share allocation method.

In [3], Papadimitratos and Haas proposed and analyzed a routing protocol named Secure Message Transmission Protocol (SMT) which improves security and reliability of data transmission through diversity coding of data into multiple symbols and transmitting each symbol over one path by uniform loading. SMT employs a rating mechanism to select the most reliable paths based on end-to-end feedback.

Our work in this paper differs with existing work in that we base our work on the worst-case scenarios and provide multipath routing solutions with guaranteed security and performance properties. Our motivation is twofold: first, in most of the proposed solutions, each path is rated according to its past performance, and the paths with high rate are selected to carry traffic. In such reputation-based mechanism, the computation of the reputation rates is not trivial at all; furthermore, this mechanism may fail to provide good paths when facing strategic attackers. For example, assume that three paths are available and each time the two paths with the highest rates are selected. A strategic attacker can itself do the same rating estimation and attack the two paths with the highest rate. The problem is that the rating mechanism implicitly assumes that there exists correlation between the history and future performance. With this correlation, one can predict the attacker's action to some extent. Unfortunately, a strategic attacker will certainly not take predictable actions. Instead, in some cases it can even take the advantage of the rating mechanism to cause more severe damage to the networks. Motivated by the above observation, we believe that it is crucial to study multipath routing solutions with guaranteed worst-case security and performance properties, which is the focus of our work.

In terms of the underlying methodology, our work is also related to the min-max optimization and routing games [4–7]. In fact, our work can be seen as the application of this tools in hostile wireless networks with unreliable/lossy links absent in classical context which pose significant difficulties in solving the problem, as shown in later sections.

2. System Model and Assumptions

In our work, we consider a multihop wireless network, modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes and m edges. For the wireless links, we consider a model in which any link is either "good" (i.e., error-free) or "bad" otherwise. We refer to the probability that link $e \in \mathcal{E}$ is "good" as the reliability factor of e, denoted by r_e. We assume that different links are independent. (This assumption holds in the case where different wireless links use channels that are well separated in time and frequency via the MAC protocol or some channel coordination mechanism. The extension of our analysis to alleviate this assumption to consider the correlated-link case (the correlation between wireless links highly depends on the underlying MAC protocol) is left for future work.)

We consider a data session between a single source S and destination T. S routes its packets along path $P_i \in \mathcal{P}$ (let $\mathcal{P}$ be the set of paths between S and T) with probability q_i. An attacker M attacks the node $v \in \mathcal{V} \setminus \{S, T\}$ with probability p_v to disrupt the communication between S and T. (We assume that S and T are not attacked by M during the communication. Multiple-attacker case is discussed in Section 7.) If node v is attacked, all the traffic passing by it is captured by M during the attack period.

In this paper, we assume that each node knows the link reliability factors $\{r_e\}$. References [8, 9] address the issue of how to estimate and collect this information. We also assume that each node has the knowledge of network topology.

This information can be acquired from any secure link-state routing protocol, for example, [10]. These assumptions allow us to concentrate on the essential theoretical properties of the multipath routing problem and the resulting solutions. In the case where link reliability factors and network topology change frequently, the update of the multipath set should be performed periodically or triggered by the change.

3. Multipath Routing with Minimum Worst-Case Security Risk

In this section, we study the multipath routing solution minimizing the worst-case security risk. We quantify the worst-case security risk by the percentage of packets captured by the attackers under the condition that the attackers make all their efforts to maximize this percentage (or equivalently, the probability that a packet is captured by the attackers under the condition that the attackers make all their efforts to maximize this probability). We start with the case of single attacker M. In such a routing problem, the objective of S is to calculate $\mathbf{q} = \{q_i\}$ to minimize the maximum security risk caused by M. Mathematically, the multipath routing problem can be formulated as the following minimaximization problem $\mathbf{MP_1}$:

$$\begin{aligned} r^* = \min_{\mathbf{q}} \max_{\mathbf{p}} \sum_{v\in\mathcal{V}} \left[\sum_{v\in P, P\in\mathcal{P}} q(P)\tau(P,v)\varphi(P,v) \right] p_v \\ \text{Subject to} \sum_{v\in\mathcal{V}} p_v \le 1, \quad p_v \ge 0, \ \forall v \in V \\ \sum_{P\in\mathcal{P}} q(P) = 1, \quad q(P) \ge 0, \ \forall P \in \mathcal{P}, \end{aligned} \tag{2}$$

where $\tau(P,v) = \prod_{e\in P, e\succ v} r_e$, $\varphi(P,v) = \prod_{b\in P, b\succ v}(1 - p_b)$. $a \succ b$ denotes that packets encounter node/edge a before node/edge b when routed along P. $r = \sum_{v\in\mathcal{V}}[\sum_{v\in P,P\in\mathcal{P}} q(P)\tau(P,v)\varphi(P,v)]p_v$ is the expected probability that the packet is captured by M. Let $r' = \sum_{v\in\mathcal{V}}[\sum_{v\in P,P\in\mathcal{P}} q(P)\tau(P,v)]p_v$. If M attacks at most one node per path, then $r = r'$. In general case, it always holds that $r \le r'$. Noticing that $\mathbf{MP_1}$ is a nonlinear optimization problem, we focus on solving $\mathbf{MP_1'}$:

$$(r')^* = \min_{\mathbf{q}} \max_{\mathbf{p}} r', \tag{3}$$

which is a linear optimization problem. Later in Section 3.2 we will show that $r^* = (r')^*$.

Consider the inner maximization problem of $\mathbf{MP_1'}$ for fixed $\mathbf{q}$:

$$\begin{aligned} \max_{\mathbf{p}} \sum_{v\in\mathcal{V}} \left[\sum_{v\in P, P\in\mathcal{P}} \tau(P,v)q(P) \right] p_v \\ \text{Subject to} \sum_{v\in\mathcal{V}} p_v \le 1, \quad p_v \ge 0, \ \forall v \in \mathcal{V}. \end{aligned} \tag{4}$$

Associating a dual variable y, we obtain the following dual optimization problem:

$$\begin{aligned} \min \ & y \\ \text{Subject to } & y \ge \sum_{v\in P, P\in\mathcal{P}} \tau(P,v)q(P), \quad \forall v \in \mathcal{V}. \end{aligned} \tag{5}$$

Substituting this minimization problem in $\mathbf{MP_1'}$ leads to the following linear optimization problem $\mathbf{LP_1'}$:

$$\begin{aligned} \min \ & y \\ \text{Subject to } & \sum_{v\in P, P\in\mathcal{P}} \tau(P,v)q(P) \le y, \quad \forall v \in \mathcal{V}, \\ & \sum_{P\in\mathcal{P}} q(P) = 1, \quad q(P) \ge 0, \ \forall P \in \mathcal{P}. \end{aligned} \tag{6}$$

The size of $\mathbf{LP_1'}$ grows with the number of possible paths between S and T and can be exponentially large. For this reason we reformulate $\mathbf{LP_1'}$ as the maximum flow problem in lossy networks which can be solved in a polynomial number of steps.

In $\mathbf{LP_1'}$, we can interpret $q(P)$ as a flow on P and y as the capacity of node v. Thus the constraint $\sum_{v\in P,P\in\mathcal{P}} \tau(P,v)q(P) \le y$ restricts the flow on node v. The constraint $\sum_{P\in\mathcal{P}} q(P) = 1$ states that one unit of flow is sent from S to T. Assume that the capacity of each node v in the network is 1. $\mathbf{LP_1'}$ equals to determine the smallest scaling factor y on the network nodes such that one unit of flow can be sent from S to T. In this way $\mathbf{LP_1'}$ can be mapped to the *maximum flow* problem.

Here we would like to emphasize that the maximum flow problem in our context differs from the classical maximum flow problem due to the packet loss factor $\tau(P,v)$. Indeed our problem can be seen as the maximum flow problem in lossy networks [11]. Each link has unlimited capacity $+\infty$, but has a reliable factor r_e. If $r_e = 1$, for all $e \in \mathcal{V}$, our problem degenerates to the standard maximum flow problem with node capacity constraint.

3.1. Solving the Multipath Routing Problem. We first give the stretch of the solution.

(i) Perform *node splitting* to transform the maximum flow problem with node capacity constraint into the maximum flow problem with link capacity constraint.

(ii) Calculate the maximum flow f^* in the transformed network after the node splitting procedure. Decompose the maximum flow into subflow on paths P_1, P_2,..., P_l from S to T with flow f_i on P_i, respectively.

(iii) S should route its packets along path P_i with probability $q_i = f_i/f^*$ to minimize the security risk. The minimum security risk r^* is $1/f^*$.

(iv) Perform the inverse procedure of node splitting. Map the paths and flows in transformed graph into the correspondent paths and flows in the original graph.

In the following, we detail the core part of the solution.

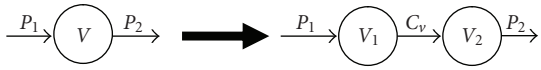


Figure 1: Node splitting.

3.1.1. Node Splitting. The objective of *node splitting* is to transform the maximum flow problem with node capacity constraint into the standard maximum flow problem with link capacity constraint. The key idea is to replace a node with capacity c with two virtual nodes with a link of capacity c between them. The detailed transformation procedure is as follows.

(i) Split each node $v \in \mathcal{V}$ of capacity c_v into two virtual nodes v_1 and v_2. Add a link (v_1, v_2) with the same capacity c_v and the reliable factor 1.

(ii) For each link $(v, v') \in \mathcal{E}$ of reliability p, replace (v, v') by a link (v_2, v') with the same reliability p and the capacity $+\infty$. For each link $(v'', v) \in \mathcal{E}$ of reliability p, replace (v'', v) by a link (v, v_1) with the same reliability p and the capacity $+\infty$.

Figure 1 illustrates the node splitting procedure. After the procedure, node v_1 receives all the input flows of node v; the output flows of node v are sent by the node v_2; the added virtual link (v_1, v_2) carries the flow from input to the output which is restricted by its capacity c_v. Let $\mathcal{G}'$ denote the resulting network after applying the node splitting process on the original network $\mathcal{G}$. It is clear that each flow in $\mathcal{G}$ is one-to-one mapped into a flow with the same quantity in $\mathcal{G}'$. Hence it holds that f^* is the maximum flow in $\mathcal{G}$ if and only if f^* is the maximum flow in $\mathcal{G}'$.

3.1.2. Finding Maximum Flow. Our discussion in this subsection relies on the maximum flow problem in lossy networks. Given a lossy network, the maximum flow problem is to determine the maximum flow that can be sent from a source node S to a sink node T subject to the capacity constraints (i.e., each link has flow bounded by the link capacity) [11].

Such maximum flow problem in lossy networks is a generalized case of the classical maximum flow problem. To solve this generalized problem, we run the most improving augmenting path algorithm described in [11], which generalizes the maximum capacity augmenting path algorithm for the traditional maximum flow problem [12].

In Algorithm 1, the augmenting path has a value, defined as the maximum amount of flow that can reach the sink, while respecting the capacity limits, by sending excess from the first node of the path to the sink. A most improving augmenting path is an augmenting path with the highest value. The algorithm repeatedly sends flow along the most improving augmenting paths. Since these may not be the highest gain augmenting paths, this may creates residual flow-generating cycles. After each augmentation, the algorithm cancels all residual flow-generating cycles in CancelCycles(), so that computing the next most improving path can be done efficiently. Intuitively, canceling flow-generating cycles can be interpreted as rerouting flow from its current paths to the highest-gain paths.

```
1: Input: transformed network G'
2: Output: maximum flow f*
3: repeat
4:     f ← CancelCycles(G')
5:     f* ← f* + f
6:     Find a most improving augmenting path P in G'
7:     Augment flow along P and update f*
8: until f* is maximum
```

Algorithm 1: Max-flow: most Improving Augmenting Path.

An efficient algorithm for computing a most improving augmenting path based on Dijkstra's shortest path algorithm is proposed in [12] with time complexity $O(m+n\log n)$ when implemented using Fibonacci heaps. We refer readers to [11] for detailed algorithm and [13] for a completed survey on the generalized maximum flow problem in lossy networks.

3.2. A Game Theoretic Interpretation. In this subsection, to gain a more in-depth insight of the internal structure of the obtained multipath routing solution, we study the multipath routing problem from a game theoretic perspective by modelling it as a noncooperative game between S and M, denoted as G_1. The strategy of S and M is $\mathbf{q}$ and $\mathbf{p}$, respectively. The objective of S is to determine $\mathbf{q}$ to minimize its utility function $U_s = r$, which is the security risk. The objective of M, on the other hand, is to determine $\mathbf{p}$ to maximize its utility function $U_a = r$.

G_1 is a classical two-person zero-sum game with finite strategy set. Following [14, Proposition 33.1], a Nash equilibrium (mixed strategy) is guaranteed to exist. Based on the result on the two-person zero-sum game [14, Proposition 22.2], we have the following theorem on the NE (Nash equilibrium) of the multipath routing game G_1.

Theorem 1. *At the NE of $G_1(\mathbf{p}^*, \mathbf{q}^*)$, it holds that*

$$\begin{aligned} U_s(\mathbf{p}^*, \mathbf{q}^*) &= U_a(\mathbf{p}^*, \mathbf{q}^*) \\ &= \min_{\mathbf{q}} \max_{\mathbf{p}} r = \max_{\mathbf{p}} \min_{\mathbf{q}} r \end{aligned} \tag{7}$$

Theorem 1 shows that the solution of $\mathbf{MP_1}$ is the most secure routing strategy minimizing the security risk. The minimized security risk from S's point is, on the other hand, the upper bound of the payoff that M can get. Hence, at the NE, the two players reach a compromise through self-optimization such that neither has incentive to deviate.

We now investigate the attacker's strategy at the NE. We consider the maximum flow f^* on the lossy network $\mathcal{G}'$ which is obtained from $\mathcal{G}$ applying the node splitting. Let f_e^* be the flow of f^* on the edge e. It follows from [15] that there exists a cut $\mathcal{C}$ separating S and T such that $\sum_{e\in S} f_e^* = \sum_{e\in S} C_e$. In our case, $\mathcal{C}$ consists of a subset of virtual links added in the node splitting process with capacity 1. This

can be shown by the fact that the capacity of all other links is $+\infty$. These virtual links correspond to a set of nodes in the original network, denoted as $\mathcal{V}^c$. As a dual part of the maximum flow problem, at the NE, M attacks every node $v \in \mathcal{V}^c$ with probability $1/|\mathcal{V}^c|$ where $|\mathcal{V}^c|$ denotes the cardinality of $\mathcal{V}^c$. At the NE, the probability that a packet passes the node $v \in \mathcal{V}^c$ is $1/f^*$; thus the probability of the packet captured can be computed as

$$r^* = \frac{1}{f^*} \times \frac{1}{|\mathcal{V}^c|} \times \left|\mathcal{V}^c\right| = \frac{1}{f^*}, \tag{8}$$

which confirms the previous analytical results. Furthermore, it follows that at such NE, M attacks at most one node per path. This leads to $r^* = (r')^*$, which justifies our operation of solving $\mathbf{MP}_1'$ instead of $\mathbf{MP}_1$.

3.3. Complexity Analysis. In the solution of the previous multipath routing problem, the complexity of the node splitting and the inverse procedure is $O(n)$. We now investigate the complexity of Algorithm 1 in the following theorem.

Theorem 2. *Let ϵ_0 be the smallest positive number describing all possible values in Algorithm 1; Algorithm 1 terminates within at most $\lfloor \log_{m/(m-1)}(f^*/\epsilon_0) \rfloor + 1$ iterations, where $\lfloor n \rfloor$ denotes the largest integer not larger than n.*

Proof. The key idea of the proof is to notice that the maximum flow in lossy networks can be decomposed into at most m augmenting paths. Algorithm 1 selects the path that generates the maximum amount of excess at the sink. Thus, each iteration captures at least a $1/m$ fraction of the remaining flow. Please refer to appendix for the detail of the proof. □

Note that in Algorithm 1, the time complexity of the CancelCycles subroutine is $O(mn^2 \log(1/\epsilon_0))$ and that of finding the most augmenting path is $O(m + n \log n)$. Generally, ϵ_0 is sufficiently small. The total time complexity of the algorithm is thus $O(mn^2 \log(1/\epsilon_0) \log(f^*/\epsilon_0))$.

In reality, it is often more practical for S to find the quasioptimal solution of $\mathbf{MP}_1$, that is, the flow $\widetilde{f^*} = (1 - \epsilon) f^*$ where ϵ is sufficiently small. In such cases, the time complexity of finding $\widetilde{f^*}$ is $O(mn^2 \log(1/\epsilon) \log(f^*/\epsilon))$ applying the proof of Theorem 2. As a result, the proposed solution offers the flexibility for the source node to balance between the time complexity of the algorithm and the optimality of the result by tuning the parameter ϵ.

3.4. Discussion. The multipath routing problem investigated in this section is related to the work of inspection point deployment in [16] and intrusion detection via sampling in [17] which root from the drug interdiction problem. Our work differs from theirs in the following. Firstly, in [16, 17], the strategy of the police and the service provider is to inspect and sample the edges, while in our problem, the attack is on the nodes, which is more efficient from the attacker's point of view. Secondly, in [16, 17], the network is lossless, while we work on the lossy network, which is more adapted for wireless networks where packet loss and link instability is one of the major concerns. Thirdly, since finding the maximum flow in lossy networks is by nature much more complex to solve than in classical lossless networks, we choose a solution providing the flexibility for the source node to balance between the time complexity of the algorithm and the optimality of the result by tuning the parameter ϵ.

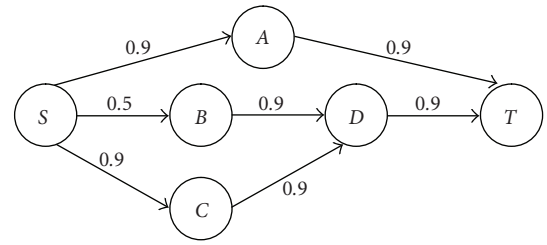


Figure 2: Limitation.

One limitation of the obtained multipath routing solution is that it minimizes the security risk by choosing appropriate multipaths without taking into account the performance of the selected path set. Figure 2 (the number beside the edge is the reliability of the link) provides an illustrative example. Based on the proposed solution, S should select the path SAT and $SBDT$, but it is clear that the path $SCDT$ is more efficient than $SBDT$. The problem is that in previous solution, in some cases, the security is obtained at the price of performance (characterized by the packet delivery ratio). This limitation may pose problem for the applications where the performance of the paths is as important as the security or even more, such as ad hoc networks for emergency rescue. In such scenarios, it is more important for S to find the paths of which the packet delivery ratio at T is maximized even at the presence of M. This motivates us to investigate the multipath routing solution maximizing the worst-case packet delivery ratio. In Section 6, we extend our work to derive the multipath routing solution to achieve a tradeoff between route security and performance.

4. Multipath Routing with Maximum Worst-Case Packet Delivery Ratio

In this section, we study the multipath routing solution to maximize the worst-case packet delivery ratio (or equivalently, the probability that a packet arrives at T under the condition that the attacker makes all its efforts to minimize this probability). In such context, S solves the following maximinimization problem $\mathbf{MP}_2$:

$$\begin{aligned} a^* = \max_{\mathbf{q}} \min_{\mathbf{p}} & \sum_{P \in \mathcal{P}} q(P) \tau(P, T) \prod_{v \in P} (1 - p_v) \\ \text{Subject to } & \sum_{v \in \mathcal{V}} p_v \le 1, \quad p_v \ge 0, \ \forall v \in \mathcal{V}, \\ & \sum_{P \in \mathcal{P}} q(P) = 1, \quad q(P) \ge 0, \ \forall P \in \mathcal{P}, \end{aligned} \tag{9}$$

where $a = \sum_{P\in\mathcal{P}} q(P)\tau(P,T)\prod_{v\in P}(1-p_v)$ is the expected probability that a packet arrives at T.

4.1. Solving the Maximinimization Problem $\mathbf{MP_2}$. The maximinimization problems such as $\mathbf{MP_2}$ are usually hard to solve directly. In our study, in order to make the problem more tractable, we apply game theory by modelling the multipath routing problem $\mathbf{MP_2}$ as a game G_2 by following the similar way as in Section 3.2. What differs here is that the objective of S is to maximize its utility function defined as $U_s = a$ and that the objective of M is to minimize $U_a = a$. Following the same argument, the following theorem is immediate.

Theorem 3. *G_2 admits at least one NE $(\mathbf{p}^*, \mathbf{q}^*)$, at which it holds that*

$$U_s(\mathbf{p}^*,\mathbf{q}^*) = U_a(\mathbf{p}^*,\mathbf{q}^*) = \max_{\mathbf{q}}\min_{\mathbf{p}} a = \min_{\mathbf{p}}\max_{\mathbf{q}} a. \tag{10}$$

Under the game theoretic formulation, solving $\mathbf{MP_2}$ consists of solving the multipath routing game G_2, more specifically, finding the NE of G_2.

Before delving into the solution, we prove the following useful theorems on the choice of strategy at the NE for the players S and M.

Theorem 4. *There exists an NE where the source node S chooses only node-disjoint paths between S and T.*

Proof. The proof consists of showing that if there exists an NE where S routes its traffic on the paths with common nodes, we can always construct an NE where the source node S chooses only node-disjoint paths. Please refer to appendix for the detailed proof. □

In the following, we focus ourselves on finding the NE with node-disjoint paths.

Theorem 5. *At the NE with only node-disjoint paths, the attacker M attacks at most one node per path.*

Proof. If at such NE, M attacks node $V_1,\ldots,V_n$ on the same path P with probability $p_1,\ldots,p_n$, then the payoff M gets on the path P is

$$U_P = \tau(P,T)(1-p_1)\cdots(1-p_n). \tag{11}$$

If M uses the same resource to attack only one node on P, say V_1, then the payoff it gets on P is

$$U'_P = \tau(P,T)(1-p_1-\cdots-p_n) < U_P \tag{12}$$

which implies that the strategy of attacking more than one node on the same path cannot be an NE. □

Now we are ready to solve the NE. We cite the following well-known lemma [14] to conduct further analysis.

Lemma 1. *Every action in the support of any player's mixed strategy NE yields that player the same payoff.*

Let $\mathcal{P}^*$ denote the multipath set chosen by S at the NE, and q_i the probability that S chooses path $P_i \in \mathcal{P}^*$ to route its traffic at the NE, p_i the probability that M attacks P_i at the NE, $\tau_i = \tau(P_i,T) = \prod_{e\in P_i} r_e$. Applying Lemma 1 , we have

$$\tau_i(1-p_i) = \tau_j\left(1-p_j\right), \quad \forall P_i,P_j\in P, \qquad q_i\tau_i = q_j\tau_j. \tag{13}$$

The packet delivery ratio $a = \sum_{P_i\in\mathcal{P}^*} q_i\tau_i(1-p_i)$. Noticing $\sum_{P_i\in\mathcal{P}^*} p_i = 1$, we have $a = (|\mathcal{P}^*|-1)/\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)$, where $|\mathcal{P}^*|$ is the number of paths in $\mathcal{P}^*$. Noticing that a is the packet delivery ratio that S wants to maximize, solving the NE consists of finding the multipath set $\mathcal{P}^*$ such that $(|\mathcal{P}^*|-1)/\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)$ is maximized. The maximized value is the solution of $\mathbf{MP_2}$. The strategy of S and M at the NE can be solved as follows.

(i) S's strategy: route the packet along path P_i with probability $q_i^* = 1/\tau_i\sum_{P_j\in\mathcal{P}^*}(1/\tau_j)$.

(i) A's strategy: attack path P_i with probability $p_i^* = 1 - ((|\mathcal{P}^*|-1)/\tau_i\sum_{P_j\in\mathcal{P}^*}(1/\tau_j))$.

It follows from $p_i^* \le 1$, for all $P_i\in\mathcal{P}^*$ that $\tau_i \ge (|\mathcal{P}^*|-1)/(\sum_{P_j\in\mathcal{P}^*}(1/\tau_j))$. This implicates that M only focuses on a subset of routes to minimize a. Interestingly, S also has incentive to only route its packets on these paths even though other paths are attack free due to the fact that the attack-free paths are very poor in terms of performance. In summary, S should solve the following optimization problem $\mathbf{MP_2'}$ to find the NE:

$$a^* = \max_{\mathcal{P}^*}\frac{|\mathcal{P}^*|-1}{\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)} \quad \text{Subject to } \tau_i \ge \frac{|\mathcal{P}^*|-1}{\sum_{P_j\in\mathcal{P}^*}\left(1/\tau_j\right)} \quad \forall P_i\in\mathcal{P}^*. \tag{C_1}$$

4.2. Heuristic Path Set Computation Algorithm. Although solving $\mathbf{MP_2'}$ is more tractable than solving $\mathbf{MP_2}$, yet it requires searching all possible node-disjoint paths between S and T, which leads to exponential time complexity. In the following, we propose a heuristic algorithm computing $\mathcal{P}^*$ with polynomial time complexity.

The goal of the heuristic algorithm is to find the optimal multipath set $\mathcal{P}^*$ such that $a = (|\mathcal{P}^*|-1)/\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)$ is maximized. We first introduce the two intuitions of the algorithm. Firstly, if we define τ_i as the reliability of path P_i, then choosing more reliable paths leads to higher global packet delivery ratio. Secondly, if we include more paths in $\mathcal{P}^*$, then $|\mathcal{P}^*|$ increases. However, the denominator of a also increases, especially when τ_i is small. Thus, the key point of our heuristic path set computation algorithm is to find as many node-disjoint paths as possible while at the same time as reliable as possible under the condition that the paths in the multipath set satisfy the constraint (C_1) such that the global packet delivery ratio a is maximized.

In order to change the path reliability from a multiplicative to an additive form, each edge $e\in\mathcal{E}$ is assigned

1: **Input:** network $\mathcal{G}$
2: **Output:** multipath set $\mathcal{P}^*$ maximizing $a = (|\mathcal{P}^*| - 1)/\sum_{P_i \in \mathcal{P}^*}(1/\tau_i)$
3: Find the most reliable path P_1 by Dijkstra algorithm, select P_1; Set $\mathcal{P}^*(1) = \{P_1\}$, $k = 1$, $a = 0$.
4: **for** each path $P_i \in \mathcal{P}^*(k)$ **do**
5: Inverse the direction of each edge on P_i, and make its length negative of the original link cost.
6: Split each node v on P_i (except S and T) into two nodes v_1 and v_2; Add an edge (v_2, v_1) of cost 0. Replace each edge $(v', v) \in \mathcal{E}$ by the edge (v', v_1) without changing its reliability, replace each edge $(v, v'') \in \mathcal{E}$ by the edge (v_2, v'') without changing its reliability.
7: **end for**
8: Run the Dijkstra algorithm, find the most reliable path P' with reliability τ' in the transformed graph.
9: If $\tau' < |\mathcal{P}^*(k)|/(1/\tau') + \sum_{P_j \in \mathcal{P}^*(k)}(1/\tau_j)$, halt by returning $\mathcal{P}^*$.
10: Transform back to the original graph; erase any interlacing edges; group the remaining edges to form the new path set $\mathcal{P}^*(k+1)$.
11: If $a < (|\mathcal{P}^*(k+1)| - 1)/\sum_{P_i \in \mathcal{P}^*(k+1)}(1/\tau_i)$, then $\mathcal{P}^* = \mathcal{P}^*(k+1)$, $a = (|\mathcal{P}^*(k+1)| - 1)/\sum_{P_i \in \mathcal{P}^*(k+1)}(1/\tau_i)$.
12: If no more path can be found in the transformed graph, halt by returning $\mathcal{P}^*$, else $k = k + 1$ and go to 2.

Algorithm 2: Heuristic path set computation algorithm.

a weight $w_e = -\log p_e$. Then the conventional shortest path algorithm such as Dijkstra algorithm can be applied to find the most reliable path.

The heuristic path set computation algorithm, shown as above, is based on the K-node-disjoint shortest path algorithm [18]. The basic idea of the K-node-disjoint shortest path algorithm is to add a path in each iteration using graph transformation and link interlacing removal such that the total cost is minimized. We refer readers to [18] for a detailed description of the algorithm.

Algorithm 2 is a greedy approach finding the most reliable path at each iteration. The iteration continues as long as: (1) there exist paths in the transformed graph, implying that there exist node-disjoint paths in the original graph; (2) the constraint (C_1) is satisfied. At the end of the algorithm, the multipath set $\mathcal{P}^*$ maximizing a is returned. Once $\mathcal{P}^*$ is found, S routes its traffic along P_i with probability q_i^*.

One point concerning the correctness of the heuristic algorithm is that if the most reliable path found in the transformed graph satisfies the constraint (C_1) (in the transformed graph), then after erasing the interlacing edges, all the paths in the newly formed multipath set $\mathcal{P}^*(k+1)$ satisfy (C_1). This can be shown by recursively applying the following lemma.

Lemma 2. *If P_2 is the most reliable path in the transformed graph that satisfies the constraint (C_1) (in the transformed graph), then after erasing an interlacing edge with another path $P_1 \in \mathcal{P}^*$, the resulting path P_1' and P_2' satisfy (C_1).*

Proof. Please refer to appendix for the detailed proof. □

We conclude this subsection by addressing the complexity of Algorithm 2. The worst-case complexity of the heuristic algorithm is $O(n^3)$ in that there are at most d_s node-disjoint paths between S and T, where d_s is the number of outgoing edges from S. Since $d_s \leq n-1$, the algorithm iterates $n-1$ times in the worst case (S can reach all nodes in the graph in one hop). In each iteration we run a minimum weight node-disjoint paths algorithm whose complexity is $O(n^2)$. The result is an overall worst-case complexity of $O(n^3)$.

5. Achieving Security-Performance Tradeoff

In Sections 3 and 4, we focus on the multipath routing solution minimizing the worst-case security risk and maximizing the worst-case packet delivery ratio. In fact, security and performance are two important aspects, of which neither should be ignored. Unfortunately, these two aspects sometimes lead to divergent routing solutions. Hence a natural next step is to investigate the multipath routing solution for multihop wireless networks that achieves a good tradeoff between the route security and performance. We formulated the routing problem in such context as the following maximinimization problem $\mathbf{MP_3}$:

$$\begin{aligned}
&\max_{\mathbf{q}} \min_{\mathbf{p}} \sum_{P \in \mathcal{P}} \sum_{v \in P} q(P)\tau(P,T) \prod_{v \in P}(1 - p_v) \\
&\text{Subject to} \sum_{v \in \mathcal{V}} \left[\sum_{v \in P, P \in \mathcal{P}} q(P)\tau(P,v)\varphi(P,v) \right] p_v \leq r_0, \\
&\qquad \sum_{v \in \mathcal{V}} p_v \leq 1, \quad p_v \geq 0, \ \forall v \in \mathcal{V}, \\
&\qquad \sum_{P \in \mathcal{P}} q(P) = 1, \quad q(P) \geq 0, \ \forall P \in \mathcal{P}.
\end{aligned} \tag{14}$$

In $\mathbf{MP_3}$, S wants to maximize the worst-case packet delivery ratio in the presence of attacker M, while limiting the worst-case security risk at most r_0. Directly solving $\mathbf{MP_3}$ needs an algorithm of exponential time complexity. In this section, we propose a heuristic solution based on Algorithm 2 to solve $\mathbf{MP_3}$. As discussed in Section 4, maximizing the worst-case packet delivery ratio equals to solve $\max_{\mathcal{P}^*}(|\mathbf{P}^*| - 1)/\sum_{P_i \in \mathcal{P}^*}(1/\tau_i)$ under the constraint (C_1). The routing strategy for S is to route the packets along path P_i with probability $q_i^* = 1/\tau_i \sum_{P_j \in \mathcal{P}^*}(1/\tau_j)$. In such context, it is easy to compute the worst-case security risk as $r = \max_{P_i \in \mathcal{P}^*}(r_{e_1^i}/\tau_i \sum_{P_j \in P}(1/\tau_j))$ where $r_{e_1^i}$ is the reliability

of the first edge of P_i, since $\max_{\mathbf{p}}\min_{\mathbf{q}} r = \min_{\mathbf{q}}\max_{\mathbf{p}} r$, and the first constraint of $\mathbf{MP_3}$ on the security risk can be transformed into

$$\tau_i \geq \frac{r_{e_1^i}}{r_0 \sum_{P_j \in \mathcal{P}^*} \left(1/\tau_j\right)}, \quad \forall P_i \in \mathcal{P}^*. \tag{C_2}$$

Our heuristic solution is extended form Algorithm 2. The key idea is to include enough number of reliable paths in $\mathcal{P}^*$ to limit the security risk. The intuition behind is that distributing the traffic among more paths helps limit the security risk. With this in mind, we modify Algorithm 2 such that the iteration stops until the constraints (C_1) and (C_2) are both satisfied or there is no more node-disjoint path available. In the latter case, the heuristic algorithm fails to find the multipath routing solution to $\mathbf{MP_3}$. This failure may due to the fact that the constraint on the security risk is too stringent such that no possible multipath set can meet the constraint, or alternatively, the heuristic algorithm itself cannot find the solution though it does exist. In such cases, possible solutions include secret sharing and information dispersion in which the key idea is to divide the packet to N parts, and the recovery of the packet is possible only with at least T parts. These techniques can further decrease the security risk and improve the performance. We refer readers to [3, 19] since they are out of the scope of our work.

6. Theoretical Security-Performance Limit of Node-Disjoint Multipath Routing

In this section, we establish the relationship between the worst-case packet delivery ratio a^* and the worst-case security risk r^* in node-disjoint multipath routing. The relationship gives one important security-performance limit of the node-disjoint multipath routing with the presence of an attacker in the sense that we cannot find better routing solutions with node-disjoint paths whose security and performance can go beyond the limit.

Let $\mathcal{P}^{\mathrm{nd}}$ be the node-disjoint multipath set selected by S to route traffic; we have shown in Section 4 that

$$a^* = \frac{\left|\mathcal{P}^{\mathrm{nd}}\right| - 1}{\sum_{P_i \in \mathcal{P}^{\mathrm{nd}}} (1/\tau_i)}. \tag{15}$$

On the other hand, let $q_k^0 = 1/\tau_k \sum_{P_j \in \mathcal{P}^{\mathrm{nd}}} (1/P_j)$. We have $\sum_{P_k \in \mathcal{P}^{\mathrm{nd}}} q_k^0 = 1 = \sum_{P_k \in \mathcal{P}^{\mathrm{nd}}} q_k$, where q_k is the probability of routing packets along P_k. From the Pigeon Hole Principle, there exists at least one path $P_m \in \mathcal{P}^{\mathrm{nd}}$ such that $q_m \geq q_m^0$. It follows that

$$\begin{aligned} r^* &= \min_{\mathbf{q}} \max_{\mathbf{p}} = \max_{\mathbf{p}} \min_{\mathbf{q}} \\ &\geq q_m r_{e_1^m} = \frac{r_{e_1^m}}{\tau_m \sum_{P_j \in \mathcal{P}^{\mathrm{nd}}} \left(1/\tau_j\right)}, \end{aligned} \tag{16}$$

where $r_{e_1^m}$ is the reliability of the first edge on P_m.

As a result, we get

$$\frac{a^*}{r^*} = \left(\left|\mathcal{P}^{\mathrm{nd}}\right| - 1\right)\frac{\tau_m}{r_{e_1^m}} \leq \left|\mathcal{P}^{\mathrm{nd}}\right| - 1 \leq \left|\mathcal{P}^{\mathrm{nd}}\right|_{\max} - 1, \tag{17}$$

where $|\mathcal{P}^{\mathrm{nd}}|_{\max}$ is the maximum number of node-disjoint path between S and T.

As a limit of node-disjoint multipath routing, the above relationship shows the intrinsic constraint of minimizing r and maximizing a at the same time. More specifically, if we want to limit the worst-case security risk as low as r, it is impossible to achieve $a > (|\mathcal{P}^{\mathrm{nd}}|_{\max} - 1)r$; if we want to guarantee the worst-case packet delivery ratio as high as a, then we should expect the worst-case security risk of at least $r/(|\mathcal{P}^{\mathrm{nd}}|_{\max} - 1)$. Moreover, given the requirement on the route security and performance, one can check if it is realizable or too stringent by using the above formula before searching for the routing solution.

7. Multipath Routing with Multiple Attackers

In this section, we extend our efforts to investigate the case where there are n $(n > 1)$ attackers in the network.

7.1. Minimizing Worst-Case Security Risk. There are various formulations of the multipath routing problem under n attackers to minimize the worst-case security risk, among which we are interested in two typical formulations. In the first formulation, let r_i be the probability that a packet is captured by attacker i, and S wants to minimize $\sum r_i$. This case can be regarded as the case where S plays the multipath routing game G_1 with each of the attackers. Hence, the solution of $\mathbf{MP_1}$ can be applied here. The only difference is that the resulting minimum worst-case security risk is nr^*. However, this does not influence routing strategy of S; in other words, no matter how many attackers are there, the routing strategy of $\mathbf{MP_1}$ provides the most secure routing strategy minimizing the worst-case security risk in this case.

In the second formulation, the security risk is defined as the probability that a packet is captured by at least one attacker. In this context, the attackers will arrange their attacks such that no more than one attacker will attack the same node simultaneously; that is, they try to coverage the most nodes possible to maximize the probability of capturing the packet. Similar as in Section 3.2, we can show that the attackers attack at most one node per path to maximize the security risk. For S, to minimize the worst-case security risk is to solve the following optimization problem $\mathbf{MP_4}$:

$$\begin{aligned} &\min_{\mathbf{q}} \max_{\mathbf{p}} \sum_{v \in \mathcal{V}} \left[\sum_{v \in P, P \in \mathcal{P}} q(P)\tau(P, v) \right] p_v \\ &\text{Subject to} \sum_{v \in \mathcal{V}} p_v \leq n, \quad 0 \leq p_v \leq 1, \ \forall v \in \mathcal{V}, \\ &\qquad \sum_{P \in \mathcal{P}} q(P) = 1, \quad q(P) \geq 0, \ \forall P \in \mathcal{P}, \end{aligned} \tag{18}$$

where p_v is the probability that a node v is attacked by any of the n attackers.

$\mathbf{MP_4}$ is a linear optimization problem and can be solved by classical linear programming techniques. However, due to additional constraints $p_v \leq 1$, $\mathbf{MP_4}$ cannot be transformed into maximum flow problem in lossy networks as $\mathbf{MP_1}$ that

can be solved in polynomial time. As a result, solving $\mathbf{MP_4}$ may require an algorithm with exponential time complexity.

In the following, we give the upper bound of the worst-case security risk under n attackers. To this end, we relax the constraint $p_v \le 1$ and perform variable transformation by letting $p'_v = p_v/n$. $\mathbf{MP_4}$ after the transformation becomes $\mathbf{MP_4'}$:

$$\min_{\mathbf{q}} \max_{\mathbf{p}} n \sum_{v\in\mathcal{V}} \left[\sum_{v\in P, P\in\mathcal{P}} q(P)\tau(P,v) \right] p'_v$$
$$\text{Subject to } \sum_{v\in\mathcal{V}} p'_v \le 1, \quad 0 \le p'_v \le 1, \ \forall v \in \mathcal{V} \tag{19}$$
$$\sum_{P\in\mathcal{P}} q(P) = 1, \quad q(P) \ge 0, \ \forall P \in \mathcal{P}.$$

$\mathbf{MP_4'}$ is identical to $\mathbf{MP_1'}$ except for a constant coefficient n. It follows immediately that its solution is n/f^* where $1/f^*$ is the maximum flow in $\mathbf{MP_1'}$. Let $r\prime$ be the worst-case security risk under n attackers; following the fact that $\mathbf{MP_4'}$ is obtained by relaxing the constraint $p_v \le 1$ in $\mathbf{MP_4}$, it holds that $r' \le n/f^*$. In summary, by increasing the number of attackers from 1 to n, the worst-case security risk increases at most n times.

7.2. Maximizing Worst-Case Packet Delivery Ratio. We consider the multipath routing game between S and the attacker side consisting of n attackers. S tries to maximize the packet delivery ratio and the attacker side tries to minimize it. It can be shown that at the NE of the game, no more than one attacker attacks the same node at the same time. This is because attacking the same node at the same time gives the attacker side the same payoff as the case where only one attacker attacks the node, which gives the attacker side less payoff than the case where the attacker side arranges the attack to cover the most number of nodes possible. With this in mind, by conducting the similar analysis as in Section 4.1, the optimization problem S should solve in multiple-attacker case $\mathbf{MP_5}$

$$\max_{\mathcal{P}^*} \frac{|\mathcal{P}^*| - n}{\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)}$$
$$\text{Subject to } \tau_i \ge \frac{|\mathcal{P}^*| - n}{\sum_{P_j\in\mathcal{P}^*}\left(1/\tau_j\right)} \quad \forall P_i \in \mathcal{P}^*, \tag{C_3}$$

where $\mathcal{P}^*$ consists of node-disjoint paths. The extension of Algorithm 2 to solve $\mathbf{MP_5}$ is straightforward.

We now investigate the case where S also wants to limit the worst-case security risk as low as r_0 at the same time, as in Section 5. Recall that $r_{e_1^i}$ denotes the reliability of the first edge of P_i, and we sort the path by $r_{e_1^i}/\tau_i$, that is, $r_{e_1^i}/\tau_i \le r_{e_1^j}/\tau_j \Leftrightarrow i \le j$. The worst-case security risk in multiple-attacker case is $\sum_{i=1}^{n}(r_{e_1^i}/\tau_i \sum_{P_j\in P}(1/\tau_j))$, which is achieved when the n attackers attack the n most profitable paths. To limit the worst-case security risk, the constraint $\sum_{i=1}^{n}(r_{e_1^i}/\tau_i \sum_{P_j\in P}(1/\tau_j)) \le r_0$ should be added to $\mathbf{MP_5}$. Algorithm 2 can be extended in a similar way as Section 5 solves it. In the multiple-attacker case, if $|\mathcal{P}^{\mathrm{nd}}|_{\max} \le n$, the communication between S and T is paralyzed by the attackers.

Table 1: Simulation parameters.

Simulation time	1000 s
Number of nodes	100, randomly distributed
Network dimension	1000 m × 1000 m
Transmission range	200 m
Node speed	4 m/s, Random waypoint model
Data traffic	CBR 4 pkt/s 64 bytes per pkt

Table 2: Simulation results: single-attacker case.

	Scenario 1		Scenario 2	
	r	p_s	r	p_s
MinSR	15.2%	54.2%	13.1%	50.3%
MaxDR	19.1%	62.2%	16.8%	59.0%
MaxDR-SR	15.8%	58.2%	15.3%	54.4%
SMT	32.3%	48.5%	39.8%	36.5%
DPSP	24.1%	49.7%	22.8%	45.3%

8. Performance Evaluation

In this section, we evaluate the performance of proposed multipath routing solutions through simulation using Network Simulator (NS 2). Table 1 shows the simulation setting. The link reliability of each link is generated from a normal distribution $\sigma(0.7, 0.2)$ trunked in $[0, 1]$ interval.

8.1. Single-Attacker Case. We start with single-attacker case. Two scenarios are simulated: the attacker launches its attack to maximize the packet capture probability (scenario 1) or minimize the packet delivery ratio (scenario 2). In both scenarios, we assume that the attacker knows the routing strategy of S.

We compare our solutions with SMT [3] and DPSP [1]. To focus on the multipath routing solution itself and perform a fair comparison, we do not implement the message dispersion in SMT. Since SMT and DPSP do not specify how to balance traffic among the paths, we let S chose randomly in the multipath set when having a packet to send.

Let MinSR denote the multipath routing algorithm minimizing the worst-case security risk, MaxDR denote the heuristic multipath routing algorithm maximizing the worst-case packet delivery ratio, and MaxDR-SR denote the heuristic multipath routing algorithm maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under certain threshold (the threshold is set to 16% in out simulation). In MinSR, to balance the complexity of the algorithm and the solution optimality, we set $\epsilon = 0.05$. Table 2 shows the simulation results.

The simulation results show that SMT performs poorly in both scenarios. This is due to the fact that in our simulation, different from the scenarios simulated in literatures [3, 20], we simulate the worst-case scenarios where the attacker

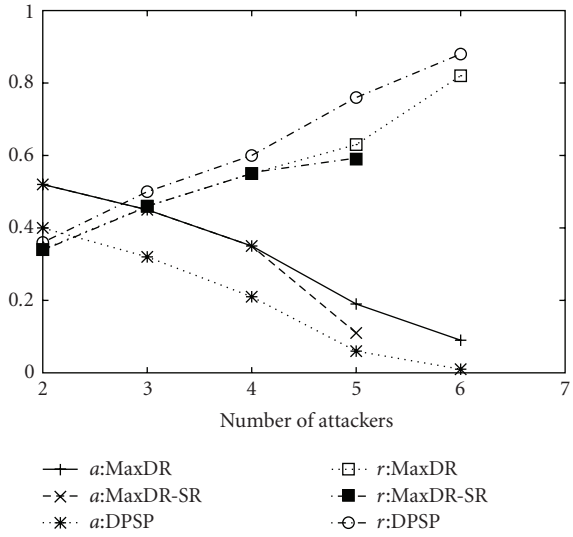


Figure 3: Multiple-attacker case: scenario 1.

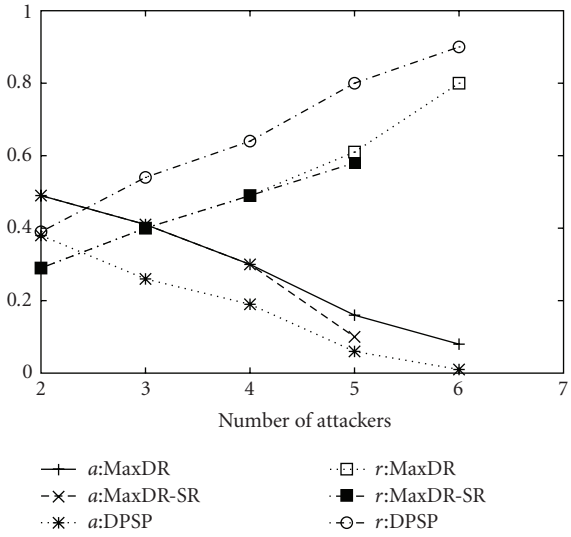


Figure 4: Multiple-attacker case: scenario 2.

launches its attack in the unpredictable way which is not correlated with the history rating. In such context, the attacker can actually take the advantage of the path rating mechanism to cause more severe damage. DSDP performs almost the same in two scenarios in that it selects the most reliable multipath set without taking into consideration of attackers. The resilience to attacks of DPSP is purely due to its multipath nature.

For our solution MinSR, it achieves the minimum security risk in scenario 2, which confirms the analytical result in that the upper bound of the security risk r^* is achieved in scenario 1. However, the packet delivery ratio in MinSR is less than that in MaxDR. This is due to the limitation of MinSR discussed in Section 3.4. From the simulation, we can see that the suboptimality of MinSR in terms of performance can be rather important compared to MaxDR, which achieves the best performance among all the simulated multipath routing solutions. MaxDR-SR, on the other hand, achieves a tradeoff between the route security and performance, which is shown by the simulation results that MaxDR-SR lies between MinSR and MaxDR in terms of route security and performance. Furthermore, we observe the fact that the number of maximum node-disjoint paths in our simulation is around 6. From this observation, we can verify the relation between the route security and performance using the formula derived in Section 6 on the theoretical limit of node-disjoint multipath routing.

8.2. Multiple-Attacker Case. We then evaluate the performance of MaxDR and MaxDR-SR (the security risk threshold r_0 is set to 0.55) in cooperative multiple-attacker case where the attacker side arranges their attacks on a subset of paths so as to maximize the security risk in scenario 1 and to minimize the packet delivery ratio in scenario 2. Figures 3 and 4 plot a and r as a function of the number of attackers. SMT is not plotted here since the worst-case packet delivery ratio of SMT drops below 20% even with 2 attackers. MinSR is not simulated here in that according to our analysis in Section 7.1, the first formulation is simply the aggregated case of the single-attacker case; in the second formulation, no polynomial routing algorithm exists minimizing the worst-case security risk.

The results show that the performance degrades significantly with the increase of the number of attackers. The communication is almost paralyzed with 5 attackers. At the presence of 6 attackers, MaxDR-SR cannot find routing solution whose security risk is not more than 0.55. Once again, our results seem very different from those obtained from literatures. This is because we focus on the worst-case scenarios throughout this paper. Unlike the traditional simulation where a percentage of nodes is assumed to be compromised, we implement much more powerful attackers with perfect knowledge of the network and the routing strategies. These attackers are able to launch the most severe attacks which are not predictable nor correlated in time or space. In such context, our results reflect the lower bound of performance of the simulated routing solutions. We argue that maximizing this lower bound, as discussed in our work, is of great importance since the attackers cannot be underestimated in any case. Meanwhile, we can see from the results that our solutions perform substantially better than DPSP in terms of both route security and performance.

In summary, the simulations show that the proposed multipath routing solutions achieve the design objective of providing the best security and/or performance in the worst-case scenarios.

9. Conclusion

In this paper, we address the fundamental problem of how to choose secure and reliable paths in wireless networks. We formulate the multipath routing problem as optimization problems and propose algorithms with polynomial complexity to solve them. Three multipath routing solutions are

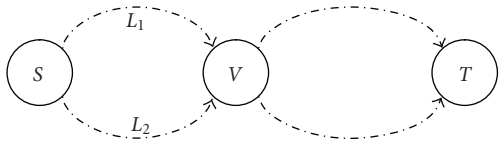


Figure 5: Two paths forms a cycle.

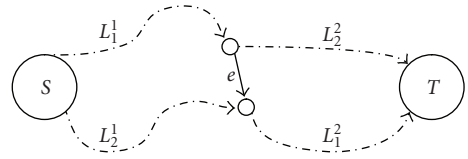


Figure 6: P_1, P_2 shares the edge e.

proposed: MinSR minimizes the worst-case security risk, MaxDR maximizes the worst-case packet delivery ratio, and MaxDR-SR achieves a tradeoff between them by maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under given threshold. We also establish the relationship between the worst-case security risk and packet delivery ratio, which gives the theoretical security-performance limit of node-disjoint multipath routing.

The analytical and simulation results in the paper lead us to the following conclusion.

(i) Solutions based on path rating which work well in the presence of time or location correlated attacks may fail to provide secure and reliable paths facing strategic attackers with unpredictable attack patterns.

(ii) Two issues are crucial in multipath routing. Firstly, both the security and performance should be taken into account when choosing the optimal paths, as in [2] and our work. Secondly, the traffic should be balanced among paths such that they are equally "attractive" to attackers.

(iii) Among the proposed multipath solutions, MaxDR-SR achieves good security-performance tradeoff by choosing sufficient number of mutually disjoint paths with high reliability and balancing the traffic in the optimal way.

Appendix

A. Proof of Theorem 2

By [11, Corollary 2.3.4], the maximum flow in lossy networks can be decomposed into at most m augmenting paths. Algorithm 1 selects the path that generates the maximum amount of excess at the sink. Thus, each iteration captures at least a $1/m$ fraction of the remaining flow. Let f_k be the flow after iteration k, and we have

$$\begin{aligned} f_1 &\geq \frac{1}{m} f^*, \\ f_2 &\geq f_1 + \frac{1}{m}(f^* - f_1), \\ &\cdots \\ f_k &\geq f_{k-1} + \frac{1}{m}(f^* - f_{k-1}). \end{aligned} \tag{A.1}$$

Injecting $f_{k-1}, \ldots, f_2, f_1$ into f_k, we have

$$\begin{aligned} f_k &\geq f_{k-1} + \frac{1}{m}(f^* - f_{k-1}) \\ &= \frac{1}{m} f^* + \frac{m-1}{m} f_{k-1} \\ &\geq \frac{1}{m} f^* + \frac{m-1}{m}\left(\frac{1}{m} f^* + \frac{m-1}{m} f_{k-2}\right) \\ &= \frac{1}{m}\left(1 + \frac{m-1}{m}\right) f^* + \left(\frac{m-1}{m}\right)^2 f_{k-2} \\ &\geq \frac{1}{m}\left(1 + \frac{m-1}{m}\right) f^* + \left(\frac{m-1}{m}\right)^2 \left(\frac{f^*}{m} + \frac{m-1}{m} f_{k-3}\right) \\ &= \frac{1}{m}\left(1 + \frac{m-1}{m} + \left(\frac{m-1}{m}\right)^2\right) f^* + \left(\frac{m-1}{m}\right)^3 f_{k-3} \\ &\geq \cdots \\ &\geq \frac{1}{m}\left[\sum_{i=0}^{k-2} \left(\frac{m-1}{m}\right)^i\right] f^* + \left(\frac{m-1}{m}\right)^{k-1} f_1 \\ &\geq \left[1 - \left(\frac{m-1}{m}\right)^{k-1}\right] f^* + \left(\frac{m-1}{m}\right)^{k-1} \frac{1}{m} f^* \\ &= \left[1 - \left(\frac{m-1}{m}\right)^k\right] f^*. \end{aligned} \tag{A.2}$$

Algorithm 1 terminates if $f^* - [1 - ((m-1)/m)^k] f^* < \epsilon_o$, that is, $k > \log_{m/(m-1)}(f^*/\epsilon_0)$.

B. Proof of Theorem 4

We have shown that there exists at least one NE in G_2. We now show that if the NE consists of overlapped paths with common nodes, we can construct another NE with node-disjoint paths.

We first give some definitions. For two paths sharing nodes A, B with $(A, B) \neq (S, T)$, let Q_1 and Q_2 be the node sequence of the two paths between A and B. Q_1, Q_2 can be empty, but they cannot both be empty. Let $l(Q)$ denote the number of nodes in the sequence Q, we call the node sequence AQ_1BQ_2A a *cycle*, and define the *diameter* of the cycle AQ_1BQ_2A as $\min\{l(Q_1), l(Q_2)\}$.

Assume that at the NE, there exists paths with common nodes. We now study the cycle containing S with the common nodes S and V with the smallest diameter. Suppose that this cycle is formed by paths P_1 and P_2 with the node

sequence $L_1 \in P_1$ and $L_2 \in P_2$ between S and V, as shown in Figure 5 . Without loss of generality, we assume that $l(L_1) \leq l(L_2)$. It follows that at the NE, any node $V_n \in L_1$ does not belong to the multipath set chosen by the source except P_1; otherwise we find a cycle with smaller diameter, which contradicts our assumption. It then holds that, at the NE, the attacker has no incentive to attack any nodes on L_1 because if it attacks any node on L_1 with probability p, it gets less payoff if it uses the same resource attacking V. From the definition of NE, routing the packets on L_1 gives S the same payoff as routing them on L_2. Hence, we can switch all the traffic from L_1 to L_2 without changing the payoff of S. Moreover, since the attacker does not attack any node on L_1 at the NE, this operation does not change the payoff of the attacker, either. Therefore, it is easy to verify that the multipath set after the above operation is also an NE of G_2. However, the number of cycles decreases by one. As a result, by recursively repeating the above process, we can transfer any NE to an NE where the number of cycles is 0. Such NE consists of only node-disjoint paths between S and T.

C. Proof of Lemma 2

The lemma holds evidently if P_2 does not intercross P_1. In the following we prove the case where P_2 intercrosses with P_1. As illustrated in Figure 6 , P_1 is composed of L_1^1, e, L_1^2, and P_2 is composed of L_2^1, e, L_2^2 before erasing the interlacing edge e. Here L_i^j $(i, j = 1, 2)$ denotes a sequence of edges. Since P_2 satisfies the constraint (C_1), we have

$$r_2^1 \frac{1}{r_e} r_2^2 \geq \frac{|\mathcal{P}^*(k)|}{1/r_1^1 r_e r_1^2 + r_e/r_2^1 r_2^2 + \Gamma}, \tag{C.1}$$

where $\Gamma = \sum_{P_j \in \mathcal{P}^*(k), P_j \neq P_1} (1/\tau_j)$ and $r_i^j = \prod_{e \in L_i^j} r_e$ $(i, j = 1, 2)$. At this moment, P_2 has not been added into $\mathcal{P}^*(k)$ yet, and so the numerator of the above inequality and that in step 7 in Algorithm 2 is $|\mathcal{P}^*(k)|$, not $|\mathcal{P}^*(k)| - 1$. Note that the cost of e is $-\log(r_e)$ in P_1 and $\log(r_e)$ in P_2 in the transformed graph.

Since the Dijkstra algorithm is applied on the graph with link cost $w_e = -\log r_e$, it follows that $r_1^1 r_e \geq r_2^1$ and $r_e r_1^2 \geq r_2^2$. Hence, we have

$$\begin{aligned}
&\frac{1}{r_2^1 r_1^2} \geq \frac{1}{r_1^1 r_e r_1^2},\ r_1^1 r_2^2 \geq \frac{r_2^1 r_2^2}{r_e} \\
&\Longrightarrow 1 + \frac{r_1^1 r_2^2}{r_2^1 r_1^2} + r_1^1 r_2^2 \Gamma \\
&\geq 1 + \frac{r_2^1 r_2^2}{r_1^1 (r_e)^2 r_1^2} + \frac{r_2^1 r_2^2}{r_e} \Gamma \\
&\Longrightarrow r_1^1 r_2^2 \left(\frac{1}{r_1^1 r_2^2} + \frac{1}{r_2^1 r_1^2} + \Gamma \right) \\
&\geq \frac{r_2^1 r_2^2}{r_e} \left(\frac{1}{r_1^1 r_e r_1^2} + \frac{r_e}{r_2^1 r_2^2} + \Gamma \right) \\
&\Longrightarrow r_1^1 r_2^2 \left(\frac{1}{r_1^1 r_2^2} + \frac{1}{r_2^1 r_1^2} + \Gamma \right) \geq |\mathcal{P}^*(k)| \\
&\Longrightarrow \tau_1' = r_1^1 r_2^2 \geq \frac{|\mathcal{P}^*(k)|}{1/r_1^1 r_2^2 + 1/r_2^1 r_1^2 + \Gamma}.
\end{aligned} \tag{C.2}$$

In the same way, we can show that $\tau_2' = r_2^1 r_1^2 \geq |\mathcal{P}^*(k)|/(1/r_1^1 r_2^2 + 1/r_2^1 r_1^2 + \Gamma)$. Noticing that P_1', P_2' consist of $r_1^1 r_2^2$ and $r_2^1 r_1^2$, respectively, it follows that both P_1' and P_2' satisfy (C_1), which concludes our proof.

References

[1] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 1–11, Lausanne, Switzerland, June 2002.

[2] W. Lou, W. Liu, and Y. Fang, "SPREAD: enhancing data confidentiality in mobile ad hoc networks," in *Proceedings of the Conference on IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2404–2413, Hong Kong, April 2004.

[3] P. Papadimitratos and Z. J. Haas, "Secure data communication in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 343–356, 2006.

[4] J. P. Brumbaugh-Smith and D. R. Shier, "Minimax models for diverse routing," *INFORMS Journal on Computing*, vol. 14, no. 1, p. 8195, 2002.

[5] J. P. Hespanha and S. Bohacek, "Preliminary results in routing games," in *Proceedings of the American Control Conference (ACC '01)*, vol. 3, pp. 1904–1909, Arlington, Va, USA, June 2001.

[6] P. P. C. Lee, V. Misra, and D. Rubenstein, "Distributed algorithms for secure multipath routing," in *Proceedings of the Conference on IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1952–1963, Miami, Fla, USA, April 2005.

[7] S. Bohacek, J. Hespanha, J. Lee, C. Lim, and K. Obraczka, "Enhancing security via stochastic routing," in *Proceedings of the International Conference on Computer Communications and Networks (ICCCN '02)*, Miami, Fla, USA, October 2002.

[8] Y. Wang, M. Martonosi, and L. Peh, "A new scheme on link quality prediction and its applications to metric-based routing," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SENSYS '05)*, San Diego, Calif, USA, November 2005.

[9] S. Zhong, L. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks—an integrated approach using game theoretical and cryptographic techniques," in *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 117–131, Cologne, Germany, August 2005.

[10] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in *Proceedings of the IEEE Workshop on Security and Assurance in Ad Hoc Networks*, 2003.

[11] K. D. Wayne, *Generalized maximum flow algorithms*, Ph.D dissertation, Cornell University, 1999.

[12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[13] M. Shigeno, "A survey of combinatorial maximum flow algorithms on a network with gains," *Journal of the Operations Research Society of Japan*, vol. 47, no. 4, pp. 244–264, 2004.

[14] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, Cambridge, Mass, USA.

[15] W. Mayeda and M. Van Valkenburg, "Properties of lossy communication nets," *IEEE Transactions on Circuits and Systems*, vol. 12, no. 3, pp. 334–338, 1965.

[16] A. Washburn and K. Wood, "Two-person sum games for network interdiction," *Operations Research*, vol. 43, pp. 243–251, 1995.

[17] M. Kodialam and T. V. Lakshman, "Detecting network intrusions via sampling: a game theoretic approach," in *Proceedings of the Conference on IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1880–1889, San Francisco, Calif, USA, April 2003.

[18] R. Bhandari, "Optimal physical diversity algorithms and survivable networks," in *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 433–441, Alexandria, Egypt, July 1997.

[19] J. Yang and S. Papavassiliou, "Improving network security by multipath traffic dispersion," in *Proceedings of IEEE Military Communications Conference on Communications for Network-Centric Operations: Creating the Information Force (MILCOM '01)*, Washington, DC, USA, October 2001.

[20] M. Kefayati, H. R. Rabiee, S. G. Miremadi, and A. Khonsari, "Misbehavior resilient multi-path data transmission in mobile ad-hoc networks," in *Proceedings of the 4th ACM Workshop on Security of ad hoc and Sensor Networks (SASN '06)*, pp. 91–100, Alexandria, Va, USA, October 2006.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 256714, 8 pages
doi:10.1155/2009/256714

Research Article

Minimizing Detection Probability Routing in Ad Hoc Networks Using Directional Antennas

Xiaofeng Lu,[1] Don Towsley,[2] Pietro Lio',[3] Fletcher Wicker,[4] and Zhang Xiong[1]

[1] *School of Computer Science, Beijing University of Aeronautics and Astronautics, Beijing 100191, China*
[2] *Department of Computer Science, University of Massachusetts at Amherst, Amherst, MA 01003-9264, USA*
[3] *Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, UK*
[4] *Communication Network Architectures Subdivision, The Aerospace Corporation, CA 90245-4691, USA*

Correspondence should be addressed to Xiaofeng Lu, luxf@cse.buaa.edu.cn

Received 31 January 2009; Revised 1 April 2009; Accepted 3 May 2009

Recommended by Shuhui Yang

In a hostile environment, it is important for a transmitter to make its wireless transmission invisible to adversaries because an adversary can detect the transmitter if the received power at its antennas is strong enough. This paper defines a detection probability model to compute the level of a transmitter being detected by a detection system at arbitrary location around the transmitter. Our study proves that the probability of detecting a directional antenna is much lower than that of detecting an omnidirectional antenna if both the directional and omnidirectional antennas provide the same Effective Isotropic Radiated Power (EIRP) in the direction of the receiver. We propose a Minimizing Detection Probability (MinDP) routing algorithm to find a secure routing path in ad hoc networks where nodes employ directional antennas to transmit data to decrease the probability of being detected by adversaries. Our study shows that the MinDP routing algorithm can reduce the total detection probability of deliveries from the source to the destination by over 74%.

1. Introduction

In a wireless network, nodes communicate with others through shared wireless medium, which makes the communications more susceptible to passive eavesdropping and malicious traffic analysis [1]. An adversary may eavesdrop network in order to discover the location of the transmitter. These adversaries are referred as detection systems. If the power received by a detection system is strong enough, the detection system can distinguish the transmission signals from the electromagnetic noise, and it becomes aware of the existence of a transmitter. If more than two detection systems detect a transmitter in a synchronous manner, they are able to compute the transmitter's position with localization algorithms and go to find the transmitter and catch it. Hence, transmission with low detection probability is very important in an untrustworthy network.

Typically, the assumption for ad hoc networks is that nodes are equipped with omnidirectional antennas, which can transmit and receive signals in all horizontal directions [2, 3]. However, a directional antenna can get antenna gain in the main lobe direction, thus transmitters can use the directional antenna to transmit signals farther away than omnidirectional antennas with the same transmit power, or transmit signals to a receiver while using less transmit power [2, 4].

The work in [5–8] mentioned that directional antennas can reduce the detection probability, but no study has been conducted to compare the detection probability of directional and omnidirectional antennas. On the other hand, using directional antennas to achieve secure routing has not been studied yet.

Researchers in the past have done much fundamental research on directional antennas in wireless networks that focused on medium-access control, spatial reuse, efficient power consumption, network capacity, and so forth. The work in [9–13] proposed adaptive Medium-Access Control (MAC) protocols to improve IEEE 802.11. These adaptive MAC protocols attempted to limit the disadvantages of IEEE 802.11 in spatial use. Power is another constrained source

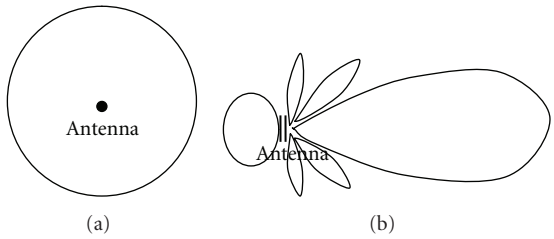


Figure 1: Transmission region of omnidirectional antenna and directional antenna.

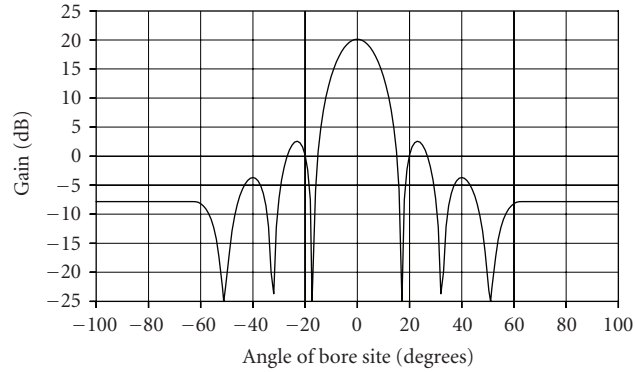


Figure 2: A directional antenna gain function.

in some ad hoc network scenarios because in these cases the power for the antenna comes from batteries, which are energy-constrained. Sometimes, nodes equipped with batteries-powered antennas cannot recharge frequently. This is another reason for using directional antennas. Authors of [14, 15] described the advantages of using directional antennas to reduce power consumption in ad hoc networks. As directional antennas can increase spatial use [16], more than one directional antenna can send data at the same time. Directional antennas can also increase network capacity [17, 18].

In this paper, we address the work we have done on routing path selection to reduce the transmitter's probability of being detected by adversaries in ad hoc networks. This paper is organized as follows. Section 2 introduces the antenna model. We introduce the detection probability model in Section 3 and our minimizing detection probability routing algorithm in Section 4. In Section 5, we review some related work about anonymous routing and secure routing protocols. Finally, we conclude our work in Section 6.

2. Antenna Model

Antennas are either omnidirectional mode or directional mode [2, 3]. Omnidirectional antennas cover 360 degrees and send data in all directions. All nodes in the radiation region can receive the communication signals [2, 3]. Omnidirectional antennas spread the electromagnetic energy over a large region, while only small portion is received by the desired receivers, so the omnidirectional transmissions waste a large portion of the transmit power and the network capacity.

Directional transmission can overcome this disadvantage. A directional antenna can form a directional beam pointing at the receiver by concentrating its transmit power into that direction. By pointing the main lobe at the receiver, a directional antenna can get more antenna gain in the direction of the receiver. Directional antennas strongly reduce signal interference in unnecessary directions.

In our antenna model, we assume that an antenna can work in two modes: omnidirectional mode and directional mode. It can send and receive data in both these two modes [2]. If nodes have nothing to transmit, their antennas work in omnidirectional mode to detect signals. A receiver and a transmitter can communicate over a larger distance when both antennas are in directional mode than just one of them is in directional mode while another is in omnidirectional mode.

Effective Isotropic Radiated Power (EIRP) is the gain of a transmitting antenna multiplied by the net power accepted by the antenna from the connected transmitter in a given direction [19]. As the gain and received power are measured in dB, EIRP can be calculated as

$$\text{EIRP} = P_t + G_t, \tag{1}$$

where P_t is the transmit power in dBW, and G_t is the antenna gain in dBi (dB $= 10\log_{10}(x)$).

Antenna gain refers to an antenna's ability to direct its radiated power in a desired direction, or to receive energy preferentially from a desired direction [4]. It is defined as the ratio of the radiation intensity of an antenna in a given direction to the intensity of the same antenna as it radiates in all directions (isotropically) and has no losses [20]. Antenna gain is expressed in dBi.

For an omnidirectional antenna, because the ratio of the radiation intensity is 1, the antenna gain is $10\log_{10}(1) = 0$. As a directional antenna concentrates the transmit power into the main lobe direction, the radiation intensity in the main lobe direction is larger than that in other directions and its G_t in that direction is much larger than zero. Therefore, the directional antenna can provide the same EIRP in the main lobe direction as that an omnidirectional antenna provides while using much less transmit power than that the omni-direction antennas uses.

No directional antenna is able to radiate all of its energy in one preferred direction. Some is inevitably radiated in other directions. These smaller peaks in Figure 1(b) are referred to as side lobes, commonly specified in dBi down from the main lobe. Figure 2 shows a case directional antenna gain in main lobe, side lobes, and back lobe.

As different antennas have different antenna structures and physical characteristic, their antenna gain functions are different. We use an approximate gain function to fit the directional antenna gain function. This approximate gain function is showed in Figure 3.

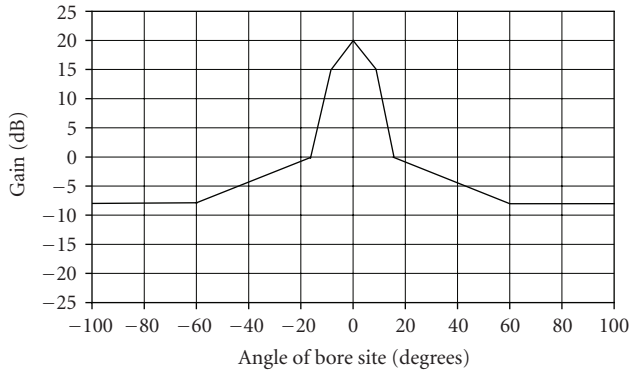


Figure 3: An approximate directional antenna gain function.

3. Detection Probability Model

3.1. Link Budget Equation. If the power received by a detection system is strong enough, the detection system can distinguish the transmission signals from electromagnetic noise. The ratio of the total received signal power to the total noise which includes thermal and system noise plus total interference is denoted as SNIR [21]. Hence, the detection event occurs if and only if the SNIR is larger than a threshold λ at a detection system.

The equation to compute the total received signal level at the receiver antenna is the following [22]:

$$S = P_t + G_t + G_r - C_t - C_r - \widetilde{Pl}, \tag{2}$$

where P_t(dBW) is the transmitter's power level, G_t(dBi) is the transmitter's antenna gain in the direction towards the receiver, G_r(dBi) is the receiver's antenna gain in the direction of the transmitter, C_t is the transmitter's cable attenuation, C_r is the receiver's cable attenuation, and $\widetilde{Pl}$ is adaptive transmission path loss, which we will discuss carefully later. C_t and C_r are assumed to be zero here.

The total noise level at the receiving unit is

$$N = k + \mathrm{dB}(T_r + T_e) + \mathrm{dB}(\mathrm{BW}) + I \tag{3}$$

where k is Boltzmann constant equal to -228.6 dB(Watts/(Hertz $*$ Degree Kelvin)). T_r is noise temperature at the receiver's antenna and T_e is environment noise temperature at the receiver's antenna [22]. The receiving bandwidth is of course matched to communication signal's bandwidth BW. The final term I is the total interference power level. The impact of interference is assumed to be zero in our study.

Free-space path loss (FSPL) is the loss in signal strength of an electromagnetic wave that would result from a line of sight path through free space, with no obstacles nearby to cause reflection or diffraction [23]. This loss is calculated using the following formula:

$$pl(d, f, n) = c + 20\log_{10}(d) + 20\log_{10}(f), \tag{4}$$

where d is the distance from the transmitter to the receiver, the radio frequency is f, and c is a constant that depends of the units of measure for d and f. With the units of measure for d and f listed in Table 1, $c = -27.55$.

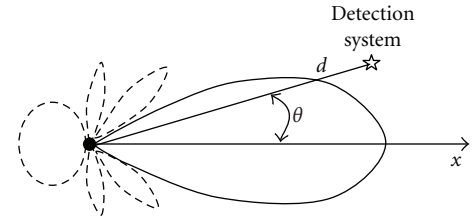


Figure 4: Illustration of d and θ.

Past line of sight, communications is still possible, but there is additional attenuation due to shadowing. Additionally it is well know that the average receive power level, measured in dBW, around a circle at a constant distance from the transmitter and beyond the line of sight is a lognormally distributed random variable. Let $\widetilde{Pl}(d, f, n)$ be the path loss when the distance from the receiver to the transmitter is larger than the line of sight distance. We modify the FSPL formula and propose an adaptive path loss formula:

$$\widetilde{Pl}(d, f, n) = -27.55 + n10\log_{10}(d) + 20\log_{10}(f), \tag{5}$$

where n is determined by the terrain type.

In our analysis, the coefficient n is a random variable that depends of the type of terrain, that is, how rugged the terrain is to radio frequency waves. Typical terrain types include open rural, rural trees and rolling hills, suburban, and urban. For each of the terrain types there is an average distance to the edge of the unobstructed line of sight given. Beyond this limit, the value of n is drawn uniformly random between the values listed in Table 2 with the possibility that there are locations that have direct line of sight beyond this average.

3.2. Detection Probability Model. Now we study the issue of the probability that a detection system detects a transmitter. Let the direction of the directional antenna's peak radiation intensity lie on the positive x axis and the star node be a detection system in Figure 4. The distance from the transmitter to the detection system is d and the angle between the direction of the detection system and the direction of the positive x axis is θ. We will use d and θ in the following sections of this paper with the same meanings defined here. We assume that the detection system's antenna works in omnidirectional mode.

The detection event occurs at a detection system if and only if the SNIR is larger than the threshold λ:

$$\Pr(\text{Detection}) = \Pr(\text{SNIR} > \lambda),$$

$$\text{SNIR} = S - N. \tag{6}$$

Substitute (2), (3), and (5) into (6).

$$\begin{aligned}\text{SNIR} = {} & P_t + G_t(\theta) + 27.55 - n10\log_{10}(d) - 20\log_{10}(f) \\ & - k - \mathrm{dB}(T_r + T_e) - \mathrm{dB}(\mathrm{BW}) + G_r,\end{aligned} \tag{7}$$

Table 1: Variable definitions for link budget equations.

Symbol	Meaning	Value	Units
P_t	Transmitter power level	–	dBW
G_t	Transmit antenna gain in the direction of the hostile antenna	Figure 3	dB
f	Radio frequency	2500	MHz
d	Distance between the transmitter and hostile node	Calculated	M
G_r	Receiver antenna gain in the direction of the transmit antenna	0	dB
S	Total received signal level after receive antenna	Equation (2)	dBW
BW	Hostile receiver's Bandwidth	1000000	Hertz
T_r	Noise temperature of hostile antenna	500	Degrees Kelvin
T_e	Environment noise temperature at hostile antenna	300	Degrees Kelvin
T	Total system noise temperature at hostile antenna	$T_r + T_e$	Degrees Kelvin
N	Total noise level in signal bandwidth at hostile antenna	Equation (3)	dB Watts

Table 2: Terrain type parameters.

Terrain type	Distance to horizon (m)	Range of n
Rural-open	1000	2 to 2.5
Rural-trees	300	2 to 4.0
Suburban	200	2 to 5.0
Urban	100	2 to 6.0

where $G_t(\theta)$ is the transmitter's antenna gain function as Figure 3 shows, and $G_r = 0$. As P_t, $20\log_{10} f$, dB$(T_r + T_e)$, and dB(BW) are constants, let

$$K = P_t + 256.15 - 20\log_{10} f - \text{dB}(T_r + T_e) + \text{dB(BW)}. \quad (8)$$

Substitute (8) into the definition of the SNIR, the probability of the detection event occuring is

$$\begin{aligned}\Pr(\text{SNIR} > \lambda) &= \Pr\left(K + G_t(\theta) - n10\log_{10} d > \lambda\right) \\ &= \Pr\left(\frac{K + G_t(\theta) - \lambda}{10\log_{10} d} > n\right).\end{aligned} \quad (9)$$

Now we discuss the value of n, for each of the terrain types listed in Table 1, there is an average distance to the edge of the unobstructed line of sight given, which we defined as d_0. When the distance d is smaller than d_0, we set n equal to 2. If the distance to the transmitter is greater than d_0, the value of n is a random variable between the values listed in Table 1:

$$\Pr(\text{SNIR} > \lambda) = f(d,\theta) = \begin{cases} \dfrac{K + G_t(\theta) - \lambda}{10\log_{10} d} > 2, & d \le d_0, \\ \dfrac{K + G_t(\theta) - \lambda}{10\log_{10} d} > n, & d > d_0, \end{cases} \quad (10)$$

where K is given by (8).

3.3. Model Analysis. Assume both the directional and omnidirectional antennas provide the same EIRP in the direction of the receiver. Assume that the omnidirectional antenna's transmit power is 3 watt and the directional antenna's gain function is as Figure 3 shows, so the directional antenna's transmit power is 0.03 watt. We assume that the operational area Ω is a finite area 100 kilometers $\times$ 100 kilometers and the terrain is rural-open. We place the transmitter at the center of the operational area.

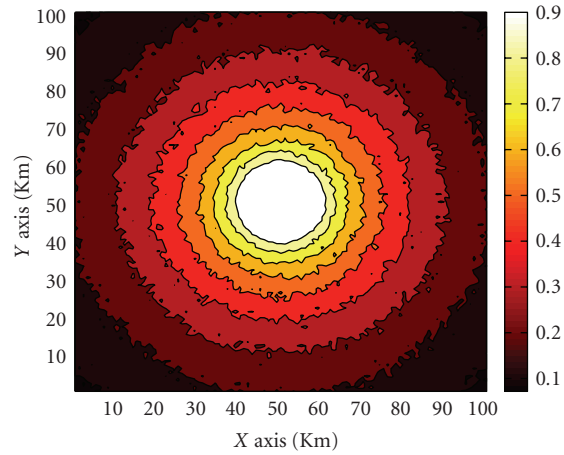


Figure 5: An omnidirectional antenna's detection probability map.

Figure 5 shows the detection probability map of an omnidirectional antenna in the operational area. In this figure, different colors mean different probability values. As omnidirectional antennas radiate signals in all directions equally, the contour lines are almost circles in Figure 5. The detection probability becomes lower and lower with the increase of the distance d. Figure 6 shows the detection probability map of a directional antenna. Only locations in the main lobe direction of the directional antenna have high probabilities to detect the transmitter, the detection probabilities at other directions are very low.

Let $A_1, \ldots, A_n$ be a partition of the operational area Ω. Assume that there is only one detection system that is in

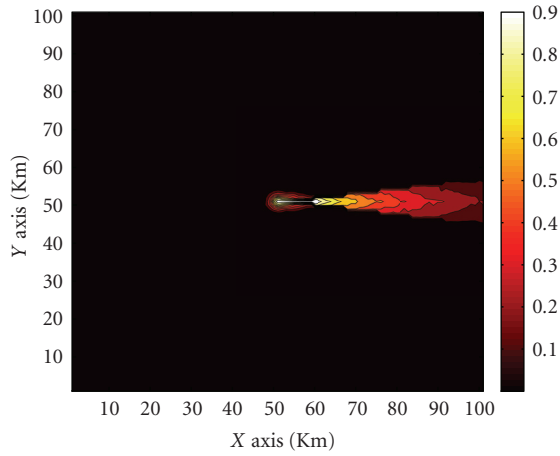


Figure 6: A directional antenna's detection probability map.

one of $\{A_i\}$. According to the total probability theorem, the probability of detecting the transmitter is

$$\text{dp} = \Pr(\text{Detection}) = \sum_{i=1}^{n} \Pr(A_i)\Pr(\text{Detection} \mid A_i), \quad (11)$$

where $\Pr(A_i)$ is the probability of the detection system being in region A_i. We assume that the probability of the detection system being in A_i are even, $\Pr(A_1) = \Pr(A_2) = \cdots = \Pr(A_n)$. Then the probability of detecting the transmitter is

$$\text{dp} = \Pr(\text{Detection}) = \sum_{i=1}^{n} \frac{\Pr(\text{Detection} \mid A_i)}{n}. \quad (12)$$

Here we assume that each A_i is 1 km × 1 km, which is a small region for directional transmissions. Normally, if two locations are very near, the detection probabilities at these two locations should be almost equal, so we can assume $\Pr(\text{Detection} \mid A_i)$ to be the detection probability at the center of A_i. Using equation (10), we can calculate the probability of detecting a transmitter at the center of A_i.

The dp of Figure 5 is 0.36 and dp of Figure 6 is 0.012. This indicates that directional antennas can reduce the detection probability by over 96.7%. Comparing these two figures, we can find that the area where the detection probability being zero in Figure 6 is much larger than that in Figure 5 and the colorful area where the detection probabilities being larger than 0.1 in Figure 6 is much less than that area in Figure 5. This can explain why a directional antenna has the lower detection probability than an omnidirectional antenna if they provide the same EIRP in the direction of receiver.

4. Minimizing Detection Probability Routing Algorithm

4.1. Definition. We model adversaries as passive. Adversaries in this model are assumed to be able to receive any transmitter's signals but are not able to modify these signals. If a set of adversaries detect a transmitter in a synchronous manner, they may be able to compute the transmitter's position with localization algorithms. It is dangerous to reveal the position information to adversaries, because adversaries may find the transmitter and catch it according to its position.

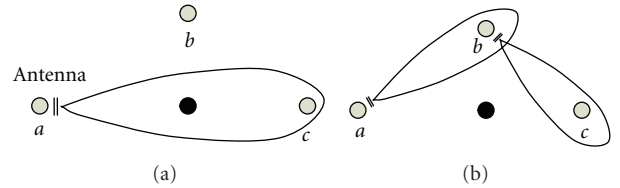


Figure 7: An illustration of using directional antennas to bypass a detection system.

As directional antennas can transmit signals towards a specific direction, we can employ several directional antennas as relays to bypass a detection system. In Figure 7, node a, b, and c are three network nodes and the black node is a detection system. Assume that node a wants to send data to node c. If node a transmits data to node c directly using directional antenna, as the detection system happens to lie in main lobe direction of node a, it can detect node a with 100% probability. Or, node a can send data to node c via node b as Figure 7(b) shows. As the detection system is not in the main lobe direction of these two directional antennas, the probability of detecting the transmissions at the detection system is very low as Figure 6 indicates.

Assume detection systems and network nodes are scattered within the operational area. To make the relay transmission from the source to the destination more secure, the strategy of our routing algorithm is to Minimize Detection Probability (MinDP) by selecting a routing path with the lowest detection probability rather than the shortest distance or the least power consumption. In Figure (8), the relay transmission path ($a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$) is more secure than the path ($a \rightarrow b \rightarrow c \rightarrow e$). If network nodes know the locations of detection systems, they can use equation (10) to calculate the detection probability. If network nodes do not know the locations of detection systems, they can use equation (12) to calculate the detection probability.

The goal of our routing protocol is to find a secure routing path which has the lowest detection probability throughout the whole delivery process from the source to the destination. Assume that a packet would be delivered from the source to the destination through N hops. If any of these N hops deliveries is detected by a detection system, the detection event occurs. Let TDP be the total detection probability from the source to the destination

$$\text{TDP} = 1 - \prod_{i=1}^{N} (1 - P_i) \quad (13)$$

where P_i is the probability of the i hop delivery being detected by all detection systems.

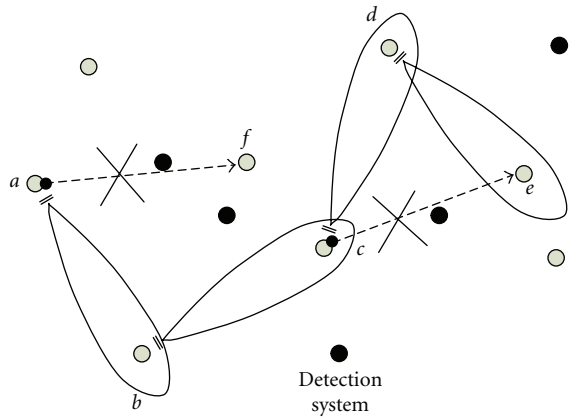


Figure 8: An illustration of anonymous routing using directional antennas.

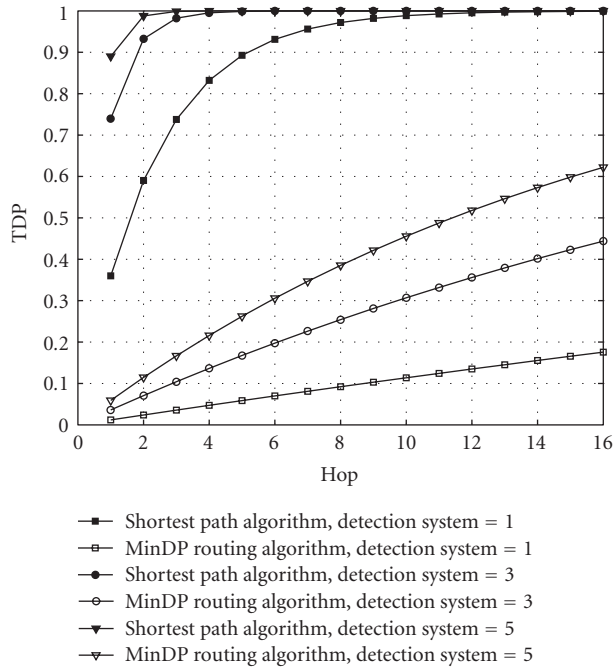


Figure 9: Total detection probability function of hops.

Some assumptions for this routing algorithm are as follows.

(1) Assume that there are k network nodes and all of them employ directional antennas to transmit data.

(2) The transmit power of a transmitter varies based on the distance from the transmitter to the receiver and the transmit rate.

The formal definition of MinDP routing algorithm is shown in Algorithm 1.

4.2. Evaluation. Assume the experimental area is 100 km $\times$ 100 km and detection systems and network nodes are scattered within the operational area randomly. We compare the total detection probability of MinDP routing algorithm using directional antennas with that of shortest path rouging using omnidirectional antennas. We randomly select two nodes as the source and the destination of each routing.

Figure 9 shows the TDP function of hops. In this figure, the TDP of Shortest path routing using omni-direction antennas increases rapidly, while the TDP of MinDP routing algorithm increases adagio. In a scenario where the number of detection systems is given, the TDP of Shortest path routing is much higher than that of MinDP routing algorithm. It is reasonable that the more detection systems are within the experiment area, the higher total detection probability is. We can know from this figure that the transmission from the source to the destination using omni-directional antennas will be detected by detection systems definitely when the number of detection systems is larger than 3 and the number of hops is larger than 2. The average TDP of Shortest path routing is 0.953 and the average TDP of MinDP routing algorithm is 0.244. Hence, the MinDP routing algorithm using directional antennas can reduce the total detection probability by over 74%.

5. Related Work

Many protocols have been proposed to provide anonymity in Internet, such as Crowds [24], Onion [25]. For ad hoc networks, although a number of papers about secure routing have been proposed, such as SEAD [26], ARAN [27], AODV-S [28], only a few papers are about anonymous routing issue and few of them talk about directional antennas and locations.

Zhu et al. proposed a secure routing protocol ASR for MANET [29] to realize anonymous data transmission. ASR makes sure that adversaries are not able to know the source and the destination from data packets. ASR considers the anonymity of addresses of the source and the destination in a packet but not the physical location of the source. In ASR, their solution make use of the shared secrets between any two consecutive nodes. The goal of ASR is to hide the source and destination information from data packets but not to protect the transmission from being detected by hostile detection systems.

ANODR is an secure protocol for mobile Ad hoc networks to provide route anonymity and location privacy [30]. For route anonymity, ANODR prevents strong adversaries from tracing a packet flow back to its source or destination; for location privacy, ANODR ensures that adversaries cannot discover the real identities of local transmitters. However, the location privacy ANODR provides is the identity of sender, not the physical location privacy.

Zhang et al. proposed an anonymous on-demand routing protocol, MASK, for MANET [31]. In MASK, nodes authenticate their neighboring nodes without revealing their identities to establish pairwise secret keys. By utilizing the secret keys, MASK achieves routing and forwarding task without disclosing the identities of participating nodes.

Most secure routing protocols and anonymous routing protocols employ authentication and secret key approaches

Let $PATH$ note the selected path and $AvailablePath$ save all possible routing paths
$Min = 1$
for $i = 1$ to k
for $j = 1$ to k
  if $i != j$
  Calculate dp(node$_i$ → node$_j$)
  end if
end for
end for
  /*Generate all available routing paths and save routing paths to $AvailablePath$. A $path$ is nodes sequence like $path_1 \rightarrow path_2 \rightarrow \cdots \rightarrow path_x^*$/
  GeneratePath($AvailablePath$)
while $AvailablePath != Empty$
  $path$ = GetPath($AvailablePath$)
  /*Calculate the total detection probability (TDP) of $path^*$/
  TDP = $1 - (1 - \text{dp}(path_1 \rightarrow path_2)) \cdots (1 - \text{dp}(path_{\{x-1\}} \rightarrow path_x))$
  if TDP < Min **then**
    Min = TDP
    $PATH = path$
  end if
  DeletePath($AvailablePath$,$path$)
  /*delete $path$ from $AvailablePath^*$/
end while
$PATH$ is the selected routing path

Algorithm 1

to ensure the security. In a real wireless network, there is no clear transmission range, hostile detection systems can detect the transmitter's signals even if it is very far away from the transmitter. In this scenario, the detection system does not need to pass the authentication, they just detect signals. Hence, authentication cannot thwart hostile detection.

6. Conclusions

In an untrustworthy network, it is very important for the transmitter to avoid being detected by adversaries. In this paper, we propose a detection probability model to calculate the probability of detecting a transmitter at any location around the transmitter. Since signals from omnidirectional antennas are radiated in all directions, hostile nodes at any location can receive these electromagnetic waves, they have probabilities to tell signals from noises. A directional antenna could form a directional beam pointing to the receiver, and only nodes in the main lobe beam region can receive signals well. If a directional antenna employs less transmit power than an omnidirectional antenna but provides the same EIRP to the receiver, the directional antenna can reduce the detection probability by over 96.7%. Therefore, we prefer to employ directional antennas to relay data from the source to the destination. Minimizing Detection Probability (MinDP) routing algorithm we proposed can select a routing path that has the lowest total detection probability. The simulation results show that the MinDP routing algorithm can reduce the TDP by over 74% so as to provide high security and concealment for transmitters.

Acknowledgments

We would like to gratefully acknowledge ITA Project. Our research was sponsored by the US Army Research Laboratory and the U.K. Ministry of Defence.

References

[1] J.-F. Raymond, "Traffic analysis: protocols, attacks, design issues, and open problems," in *Designing Privacy Enhancing Technologies*, H. Federath, Ed., Lecture Notes in Computer Science, Springer, Berlin, Germany, 2001.

[2] G. W. Stimson, *Introduction to Airborne Radar*, SciTech, Raleigh, NC, USA, 1998.

[3] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.

[4] J. E. Hill, "Gain of Directional Antennas," Watkins-Johnson Company, Tech-notes,1976.

[5] Z. Huang and C.-C. Shen, "A comparison study of omnidirectional and directional MAC protocols for ad hoc networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 57–61, Taipei, Taiwan, November 2002.

[6] A. Spyropoulos and C. S. Raghavendra, "Energy efficient communications in ad hoc networks using directional antennas," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 1, pp. 220–228, New York, NY, USA, June 2002.

[7] M. E. Steenstrup, "Neighbor discovery among mobile nodes equipped with smart antennas," in *Proceedings of the Swedish Workshop on Wireless Ad-Hoc Networks (ADHOC '03)*, 2003.

[8] Z. Zhang, "Pure directional transmission and reception algorithms in wireless ad hoc networks with directional antennas," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, vol. 5, pp. 3386–3390, Seoul, Korea, May 2005.

[9] A. Nasipuri, S. Ye, J. You, and R. E. Hiromoto, "A MAC protocol for mobile ad hoc networks using directional antennas," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '00)*, pp. 1214–1219, Chicago, Ill, USA, September 2000.

[10] Y.-B. Ko, V. Shankarkumar, and N. H. Vaidya, "Medium access control protocols using directional antennas in ad hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 1, pp. 13–21, Tel Aviv, Israel, March 2000.

[11] M. Takai, J. Martin, A. Ren, and R. Bagrodia, "Directional virtual carrier sensing for directional antennas in mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '02)*, pp. 183–193, Lausanne, Switzerland, June 2002.

[12] L. Bao and J. J. Garcia-Luna-Aceves, "Transmission scheduling in ad hoc networks with directional antennas," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MOBICOM '02)*, pp. 48–58, Atlanta, Ga, USA, September 2002.

[13] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, "Using directional antennas for medium access control in ad hoc networks," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MOBICOM '02)*, pp. 59–70, Atlanta, Ga, USA, September 2002.

[14] A. Spyropoulos and C. S. Raghavendra, "Energy efficient communications in ad hoc networks using directional antennas," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 1, pp. 220–228, New York, NY, USA, June 2002.

[15] A. Nasipuri, K. Li, and U. R. Sappidi, "Power consumption and throughput in mobile ad hoc networks using directional antennas," in *Proceedings of the 11th International Conference on Computer Communications and Networks (IC3N '02)*, October 2002.

[16] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, "Ad hoc networking with directional antennas: a complete system solution," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 496–506, 2005.

[17] S. Yi, Y. Pei, and S. Kalyanaraman, "On the capacity improvement of ad hoc wireless networks using directional antennas," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 108–116, Annapolis, Md, USA, June 2003.

[18] B. Liu, Z. Liu, and D. Towsley, "On the capacity of hybrid wireless networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 2, pp. 1543–1552, San Francisco, Calif, USA, March-April 2003.

[19] IEEE Std, *100 The Authoritative Dictionary of IEEE Standards Terms*, The Institute of Electrical and Electronics Engineers, New York, NY, USA, 7th edition, 2000.

[20] C. Balanis, *Antenna Theory*, John Wiley & Sons, New York, NY, USA, 3rd edition, 2005.

[21] G. Breed, "Bit error rate: fundamental concepts and measurement issues," *High Frequency Electronics*, vol. 2, no. 1, pp. 46–47, 2003.

[22] Breeze Wireless Communications Ltd, Radio Signal Propagation, http://www.breezecom.com.

[23] Federal Standard 1037C, "Telecommunications: Glossary of Telecommunication Terms," National Communication System Technology & Standards Division, 1991.

[24] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *Communications of the ACM*, vol. 42, no. 2, pp. 32–48, 1999.

[25] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 482–493, 1998.

[26] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiHoc '02)*, pp. 12–23, Atlanta, Ga, USA, September 2002.

[27] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP '02)*, Paris, France, November 2002.

[28] H. Yang, X. Meng, and S. Lu, "Self-organized network-layer security in mobile ad hoc networks," in *Proceedings of the ACM Workshop on Wireless Security*, pp. 11–20, Atlanta, Ga, USA, September 2002.

[29] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng, "Anonymous secure routing in mobile ad-hoc networks," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 102–108, Tampa, Fla, USA, November 2004.

[30] J. Kong and X. Hong, "ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 291–302, Annapolis, Md, USA, June 2003.

[31] Y. Zhang, W. Liu, and W. Lou, "Anonymous communications in mobile ad hoc networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1940–1951, Miami, Fla, USA, March 2005.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 945943, 13 pages
doi:10.1155/2009/945943

Research Article

Mobility and Cooperation to Thwart Node Capture Attacks in MANETs

Mauro Conti,[1] Roberto Di Pietro,[2,3] Luigi V. Mancini,[4] and Alessandro Mei[4]

[1] *Department of Computer Science, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands*
[2] *UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, 43700 Tarragona, Spain*
[3] *Dipartimento di Matematica, Università di Roma Tre, 00146 Roma, Italy*
[4] *Dipartimento di Informatica, Università di Roma "Sapienza", 00198 Roma, Italy*

Correspondence should be addressed to Mauro Conti, conti@di.uniroma1.it

Received 22 February 2009; Revised 13 June 2009; Accepted 22 July 2009

Recommended by Hui Chen

The nature of mobile ad hoc networks (MANETs), often unattended, makes this type of networks subject to some unique security issues. In particular, one of the most vexing problem for MANETs security is the node capture attack: an adversary can capture a node from the network eventually acquiring all the cryptographic material stored in it. Further, the captured node can be reprogrammed by the adversary and redeployed in the network in order to perform malicious activities. In this paper, we address the node capture attack in MANETs. We start from the intuition that mobility, in conjunction with a reduced amount of local cooperation, helps computing effectively and with a limited resource usage network global security properties. Then, we develop this intuition and use it to design a mechanism to detect the node capture attack. We support our proposal with a wide set of experiments showing that mobile networks can leverage mobility to compute global security properties, like node capture detection, with a small overhead.

1. Introduction

Ad hoc network can be deployed in harsh environments to fulfil law enforcement, search-and-rescue, disaster recovery, and other civil applications. Due to their nature, ad hoc networks are often unattended, hence prone to different kinds of novel attacks. For instance, an adversary could eavesdrop all the network communications. Further, the adversary might capture (i.e., remove) nodes from the network. These captured nodes can then be reprogrammed and deployed within the network area, for instance, to subvert the data aggregation or the decision making process in the network [1]. Also, the adversary could perform a *sybil attack* [2], where a single node illegitimately claims multiple identities also stolen from previously captured nodes. Another type of attack is the *clone* attack, where the node is first captured, then tampered with, reprogrammed, and finally replicated in the network. The former attack can be efficiently addressed with mechanism based on RSSI [3] or with authentication based on the knowledge of a fixed key set [4], while recent solutions have been proposed also for the detection of the clone attack [5, 6].

To think of a foreseeable application for node capture detection, note that recently the US Defense Advanced Research Projects Agency (DARPA) initiated a new research program to develop so-called LANdroids [7]: Smart robotic radio relay nodes for battlefield deployment. LANdroid mobile nodes are supposed to be deployed in hostile environment, establish an ad-hoc network, and provide connectivity as well as valuable information for soldiers that would later approach the deployment area. LANdroids might retain valuable information for a long time, until soldiers move close to the network. In the interim, the adversary might attempt to capture one of these nodes. We are not interested in the goals of the capture (that could be, e.g., to reprogram the node to infiltrate the network, or simply extracting the information stored in it); but on the open problem of how to detect the node capture that represents, as shown by the above-cited examples, a possible first step to jeopardize an ad hoc network. Indeed, an adversary has often

to capture a node to tamper with—that is, to compromise its key set, or to reprogram it with malicious code—before being able to launch other more vicious, and may be still unknown, attacks. Node capture is one of the most vexing problems in ad hoc network security [8]. In fact, it is a very powerful attack and its detection is still an open issue. We believe that any solution to this problem has to meet the following requirements: (i) to detect the node capture as early as possible; (ii) to have a low rate of false positives—nodes which are believed to be captured and thus subject to a revocation process, but which were not actually taken by the adversary; (iii) to introduce a small overhead.

The solutions proposed so far are not satisfactory as for efficiency [8]. Also, while naïve centralized solutions can be applied to generic ad-hoc networks, they presents drawbacks like single point of failure and nonuniform energy consumption. These drawbacks do not make them appealing for ad hoc networks. Moreover, these networks often operates without the support of a base station. Efficient and distributed solutions to the node capture attack are of particular interest in this context.

To the best of our knowledge, there are no distributed solutions for the problem of detecting the node capture attack in Mobile Ad Hoc Networks (MANETs). Following a new interesting research thread that focuses on leveraging mobility to enforce security properties for wireless sensor and ad hoc networks [9, 10], we propose a new capture detection framework that leverages node mobility. We show that this approach can provide better performance compared to traditional solutions. Also, we show that using node cooperation in conjunction with node mobility can still improve the capture detection performance within specific network requirements.

The contribution of this paper is to provide a proof of concept: it is possible to leverage the emergent properties of mobile ad hoc networks via node mobility and node cooperation to design a node capture detection protocol. To this aim, we use the Random Waypoint Mobility Model (RWM) [11], an ideal mobility model which is simple and general enough (at least for some application scenarios) to explore our ideas. Furthermore, the result on any particular mobility model should depend not only from the model but also from the network setting, as pointed out in [12] for the delay-capacity tradeoff. Indeed, providing specific settings and evaluations for other models is out of the scope of this work.

Our solution is based on the simple observation that if node a will not *remeet* node b within a period λ, then it is possible that node b has been captured. This observation is based on the fact that some time is required to the adversary to tamper with a sensor node. The time required by the adversary to perform such a type of attack was not investigated in the context of sensor network, until the work in [13]. In [13], the authors found out that node capture attacks (that give the adversary full control over a sensor node) are not so easy to implement, contrary to what was usually assumed in literature—indeed, among other requirements (e.g., expert knowledge and costly equipment), node tampering requires the removal of nodes from the network for a nonnegligible amount of time. In particular, while *short attacks* such as using plug-in devices can be performed in some 5 minutes, *medium attacks* that require (de-)soldering requires more than 30 minutes, and *long attacks* and *very long attacks* (e.g., erasing the security protection bits by UV light or invasive attack on electronic component) can require even some hours.

We will build upon this intuition to provide a protocol that makes use of local cooperation and mobility to locally decide, with a certain probability, whether a node has been captured or not. Our proposed solution does not rely on any specific routing protocol: we resort to one-hop communications and to a sparing use of a message broadcasting primitive. These distinguished features help keep our protocol simple, efficient, and practically deployable, avoiding the use of sophisticated routing that can introduce complexity and overhead in the mobile setting. Furthermore, our experimental results demonstrate the effectiveness and the efficiency of our proposal. For instance, for a given energy budget, while the reference solution requires about 4000 seconds to detect node capture, our proposal requires less than 2000 seconds. We remark that the solution proposed in this paper is completely tunable: the capture detection time can be set as small as desired. However, a smaller detection time would imply an higher energy consumption.

The paper is organized as follows. Section 2 presents the related work in this area. Section 3 introduces the motivation and the framework of our proposal based on simple ad hoc network capabilities like node mobility and message broadcasting. Our specific proposal, the CMC Protocol, is then presented in Section 4, while in Section 5 we discuss the simulation results that give a qualitative idea of how mobility and node cooperation can be leveraged in order to decrease the node capture detection time. Finally, Section 6 reports some concluding remarks.

2. Related Work and Background

Mobility as a means to enforce security in mobile networks has been considered in [9]. Further, mobility has been considered in the context of routing [14] and of network property optimization [15]. In particular, the work in [14] leverages node mobility in order to disseminate information about destination location without incurring any communication overhead. In [15], the sink mobility is used to optimize the energy consumption of the whole network. A mobility-based solution for detecting the sybil attack has been recently presented in [10]. Finally, note that a few solutions exist for node failure detection in ad hoc networks [16–19]. However, such solutions assume a static network, missing a fundamental component of our scenario, as shown in what follows.

In this work, we use node mobility to cope with the node capture attack. As described in the following section, we specifically rely on the meeting frequencies between honest nodes to gather information about the absence of captured nodes. A property similar to that of node "remeeting" has been already considered in [20]. However, in [20], the

authors investigate the time needed for a node to meet (for the first time) a fixed number of other nodes. This analysis is then used together with node mobility to achieve noninteractive recovery of missed messages. To the best of our knowledge no distributed solution leveraging node mobility has been proposed to detect the node capture attack in mobile ad-hoc and sensor networks.

While node capture attack is considered as major threat in many security solutions for WSN, to the best of our knowledge, it has not been directly addressed yet. However, some interest has been shown in modeling the node capture attack. In particular, in [21], both oblivious and smart node capture is considered for the design of a key management scheme for WSN. A deeper analysis on the modeling of the capture attack has been presented [22, 23]. In [22], it is shown how different greedy heuristics can be developed for node capture attacks and how minimum cost node capture attacks can be prevented in particular setting. In [23], the authors formalize node capture attacks using the vulnerability metric as a nonlinear integer programming minimization problem.

We recently published [24, 25]; the former arguments that mobility models have a relevant effect on the properties of the proposed algorithms, while the latter is a short contribution on the possibility to leverage network mobility for node capture detection. In particular, in [25] we presented the rationales for this type of approach and a preliminary solution to the problem. However, while the results given in [25] are encouraging, the specific solution proposed requires a high overhead to bound the number of false positives (wrongly revoked nodes). Note that, without this bounding mechanism, the number of false positives would be unacceptable. Furthermore, in [25] we did not study the feasibility of the new approach compared with other ones. In the present work, we leverage the intuition proposed in [25], which is the "remeeting" time between nodes, to design an efficient solution that leverages different levels of cooperation between nodes. In particular, we introduce a presence-proving mechanism used by allegedly captured nodes to show their actual presence in the network (i.e., eliminating the possibility of revoking a node which is present within the network). Further, we introduce a reference solution in order to quantify the quality of the proposed solutions. The proposed solutions are compared between them and with the reference solution. In particular, to have a fair comparison, we observed the detection time provided by the different protocols using the same energy budget. The result of our study confirms the intuition provided in [25]. Furthermore, it proves that within certain scenarios of node mobility, the proposed solutions provide a sensitive improvement over other possible approaches, such as the one based on classical message exchange.

Node mobility and node cooperation in a mobile ad hoc setting have been considered already in Disruption Tolerant Networks (DTNs) [26, 27]. However, such a message passing paradigm has not been used, so far, to support security. We leverage the concept introduced with DTN to cooperatively control the presence of a network node. Mobility to recover the secret state of a node has been recently introduced in [28, 29]. In this paper, we use one of the most common mobility patterns in literature, the Random Waypoint Mobility Model [11]. In this model, it is assumed that each node in the network acts independently: it selects a geographic destination in the deployment area (the *way-point*), it selects a speed uniformly at random in a given interval $[s_{\min}, s_{\max}]$, and then it moves toward the destination on a straight route at the selected speed. When at the way-point, it waits for some time, again selected uniformly at random from a given interval, and then the node repeats the process by choosing the next way-point. Some researchers have shown some problems related to this mobility model. One of the problems is that the average speed of the network tends to decrease during the life of the network itself and, if the minimum speed that can be selected by the nodes is zero, then average speed of the system converges to zero [30]. In the same paper, it is suggested to set the minimum speed to a value strictly greater than zero. In this case, the average speed of the system continues decreasing, but it converges to a nonzero asymptotic value. Other problems related to spatial node distribution have been considered by different authors [30, 31]. In the analysis presented in [14], "human speeds" are claimed to be a reasonable practical choice for mobile nodes. Note that the RWM might not be the best model to capture a "realistic" mobility scenario, as highlighted in [12]; however, the results achieved in this paper are meaningful as they are a proof of concept that mobility can be leveraged to enforce security properties; the provided protocols could be used in, and adapted to, more realistic mobility models.

In our proposed approach every node maintains its own clock. However, we require that clocks among nodes are just loosely synchronized. Note that there are a few solutions proposed in literature to provide loose time synchronization, like [32]. Therefore, in the following we will assume that skew and drift errors are negligible.

In our proposal, we also need to take into consideration the cost of broadcasting a message to all the nodes in the network. In [33], a classification of the different solutions for broadcasting scheme is provided: (i) Simple Flooding; (ii) probabilistic-based schemes; (iii) area-based schemes that assume location awareness; (iv) neighbor knowledge schemes that assume knowledge of two hop neighborhood.

Analyzing or comparing broadcasting cost is out of the scope of this paper. However, for a better comparison of the solutions proposed in this paper, we need to set a broadcast cost that will be expressed in terms of unicast messages. In fact, the overhead associated to the broadcasting varies with different network parameters (e.g., node density and communication radius). A deeper analysis on the overhead generated for different broadcasting protocols is presented in [34]. Also, note that probabilistic-based and neighbor-based protocols require a big overhead for a mobile network in order to know the network topology and neighborhood, respectively. Furthermore, the same argument can be considered for the localization protocol that is used in the area-based schemes. In the following, to embrace the more general case, we assume that nodes are not equipped with localization devices, like GPS. Finally, note that a

message could be received more than once, for instance, because the receiver is in the transmission range of different rely nodes. However, in the following, we assume that a broadcasted message is received (then counted) only once for each node. A similar assumption is used, for example, in [34].

3. Node Capture Detection through Mobility and Cooperation

The aim of a capture detection protocol is to detect as soon as possible that a node has been removed from the network. In the following, we also refer to this event as a node capture. The protocol should be able to identify which is the captured node, so that its ID could be revoked from the network. Revocation is a fundamental feature—if the adversary reintroduces the captured (and possibly reprogrammed) node in the network, the node should not be able to take part to the network operations.

In the following, we first describe a simple distributed solution that does not exploit neither mobility nor cooperation among nodes; we use this solution as a reference solution to compare with our proposal. Then, we introduce the rationals we leverage to develop our protocol for node capture detection, detailed in the following section.

3.1. Reference Solution. To the best of our knowledge, no efficient and distributed solution leveraging mobility was proposed so far to cope with the node capture detection problem in Mobile Ad Hoc Network. However, a naïve solution that makes use of node communication capabilities can be easily figured out. We first describe this solution assuming the presence of a base station (BS); then, we will show how to relax this assumption. In the BS-based solution, each node periodically sends a message to the BS carrying some evidence of its own presence. In this way, the base station can witness for the presence of the claiming nodes. If a node does not send the claim of its presence to the BS within a given time range, the base station will revoke the corresponding node ID from the network (e.g., flooding the network with a revocation message). To remove the centralization point given by the presence of the BS, we require each node to notify its presence to any other node in the network. To achieve this goal, every t seconds a node sends a claim message advertising its presence to all the network nodes through a broadcast message. A node receiving this claim would restart a timeout set to $t + \sigma$ where σ accounts for network propagation delay. Should the presence claim not be received before the timeout elapses, the revocation procedure would be triggered. However, note that if a node is required to store the ID of any other node as well as the receiving time of the received claim message, $O(n)$ memory locations would be needed in every node. To reduce the memory requirement on node, it is possible to assume that the presence in the network of each node is tracked by a small subset of the nodes of the network. Hence, if a node is absent from the network for more than t seconds, its absence can still be detected by a set of nodes.

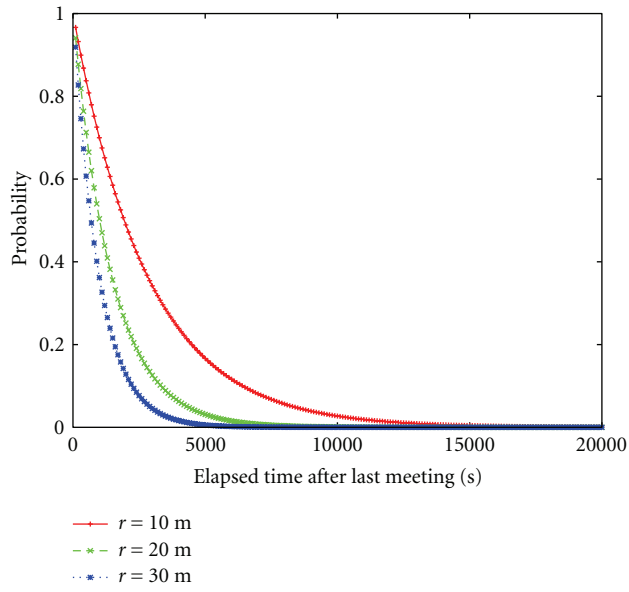


Figure 1: Noncooperative approach: the probability for two nodes not to remeet again: $n = 100$, $s_{\min} = 5$ m/s, $s_{\max} = 15$ m/s.

3.2. Our Approach. Our approach is based on the intuition that leveraging node mobility and cooperation helps node capture detection. We start from the following observation: if node a has detected a transmission originated by node b, at time t, we will say that a *meeting* occurred. Now, nodes a and b are mobile, so they will leave the communication range of each other after some time. However, we expect these two nodes to remeet again within a certain interval of time, or at least within a certain time interval with a certain probability. The solution can also be thought of as an exploitation of the opportunistic communication concept [27], like contact-based message delivery, to wireless ad hoc network security. In [25], the authors investigated how mobility can be used to detect a node capture and investigated the feasibility of mobility-based solutions. As a starting point, we analysed the remeeting probability through network simulation: the results comply with previous studies on delay in mobile ad hoc networks [12]. In Figure 1, we report on the simulation results on the probability that two nodes that had a meeting would not have a meeting again after x seconds. This probability has been evaluated for different values of the communication radius. In particular, we assume that the nodes are randomly deployed in a square area of 1000 m $\times$ 1000 m and that they move according to the random waypoint mobility model. While the x-axis indicates the time after the last meeting, the y-axis indicates the probability that the two nodes have not remet yet. For example, assume that node a meets node b at time t, then the probability that these two nodes have not met again after 5000 seconds is very close to 0 (for a sensing radius $r = 30$).

In the following section, we propose a protocol that leverages node mobility to enhance node capture detection probability.

Table 1: Time-related notation.

Symbol	Meaning
σ	Message propagation delay.
λ	Alarm time used in CMC (our proposal).
δ	Time available to the allegedly captured node to prove its presence.

3.3. Assumptions and Notation. In the remaining of the paper, we assume a "smart" attacker model: it knows the detection protocol implemented in the network. This implies, for the reference solution, that a node a is captured just after node a has broadcasted its presence claim message. The assumption at the base of our protocol is that if a node has been absent from the network for a given interval time (i.e., none can prove its presence in that interval) the node has been captured. It is worth noticing that also if a node is temporarily disconnected, a DTN-like routing mechanism [35] can be used to deliver a message to that node with some delay. For the aim of our protocol, we do not explicitly consider that interval time.

In the following we define a *false-positive alarm* as an alarm raised for a node that is actually present. One or more false-positive alarms can imply a *false-positive detection*, which corresponds to the revocation of a not captured node. Further, we refer to a *false-negative detection* as a captured node not actually revoked. However, we observe that using the presence-proving mechanism introduced in this paper (later discussed in Section 4), a node that is accused by a false-positive alarm would prove its presence, hence neutralizing the revoke. Furthermore, we observe that accordingly to our protocol, a node no longer active (e.g., destroyed or with run out batteries) would be revoked. However, there would be no false alarms and the overhead paid for the protocol would be just one network flooding. The flooding would allow every node in the network to be aware of the absence of the failed node—having a beneficial effect for other protocols such as routing. In general, we cannot distinguish if a node is not able to communicate with the other network nodes for a nonmalicious reason, or because it has been actually captured—our solution is conservative in this way, revoking such a node. It is out of the scope of this paper, and left as future work, to address the recovery of the former type of revoked nodes.

Another issue is Denial of Service (DoS). Indeed, since alarms are flooded in the network, it could be possible for a corrupted node to trigger false alarms so as to generate a DoS. This issue is out of the scope of this paper, however, for the sake of completeness, we sketch in the following a possible solution. The impact of false positives can be mitigated noticing that it could be possible, once the recovery mechanism detects a false alarm, to associate a failure tally to the node that raised the false alarm. If the tally exceeds a certain threshold, the appropriate action to isolate the misbehaving node could be take.

Further, we assume the existence of a failure-free node broadcasting mechanism [36]; and, finally, we point out that addressing node-to-node secure communications properties such as confidentiality, integrity, privacy, and authentication are out of the scope of this paper. However, note that a few solutions explicitly addressing these issues can be found in literature [4, 37, 38].

Table 1 resumes the intervals time notation used in this paper.

4. The Protocol

In this section, we describe our proposal for a node Capture detection protocol that leverages Mobility and Cooperation (CMC Protocol). Basically, each node a is given the task of witnessing for the presence of a specific set T_a of other nodes (we will say that a is *tracking* nodes in T_a). For each node $b \in T_a$ that a gets into the communication range of, a sets a new time-out for b with the value of the a's internal clock; the time out will expire after λ seconds. The meeting nodes can also cooperate, exchanging information on the meeting time of nodes of interests, that is, nodes that are tracked by both a and b. Note that node cooperation is an option that can be enabled or disabled in our protocol. If the time-out expires (i.e., a and b did not remeet within λ seconds), a floods the network with an alarm message. If node b does not prove its presence within δ seconds after the broadcasted alarm is flooded, every node in the network will revoke node b. The detailed description of the CMC protocol follows.

4.1. Protocol Description. The CMC protocol is event-based; in particular, it is executed when the following holds.

(i) Node a and node b meet: this event triggers node a and node b to execute *CMC_Meeting*$(ID_b, \text{false}, -)$ and *CMC_Meeting* $(ID_a, \text{false}, -)$, respectively, if the cooperation parameter is set to false. Otherwise, node a executes *CMC_Meeting* $(ID_b, \text{true}, -)$ and node b executes *CMC_Meeting* $(ID_a, \text{true}, -)$. The function *CMC_Meeting* is also used in the cooperative scenario as a *virtual* meeting in order to update node presence information.

(ii) The time-out related to node ID_x expires on node a: node a executes the procedure *CMC_TimeOut* (ID_x).

(iii) Node a eavesdrops a message m: node a executes the procedure *CMC_Receive(m)*.

Algorithms 1, 2, and 3 show the corresponding pseudocode. The procedure *CMC_Meeting*, shown in Algorithm 1, is executed by both nodes involved in a meeting. In the case of a real meeting, the time is not specified, then the current node time t_a is used. However, when the procedure is invoked as a *virtual* meeting, a reference time (t_x) is also considered (lines 2, 3, and 4). When node a meets node b, node a checks if it is supposed to trace node b (that is if $b \in T_a$). This check is performed using the `Trace` function (line 5). It takes in input two node IDs, and provides a result pseudouniformly distributed in $[1 \cdots \lceil n/|T| \rceil]$—where n is the size of the wireless ad hoc network and $|T|$ is the number of nodes tracked by each node. Node b is to be tracked if and only if the result of the `Trace` function is one. A simple and efficient implementation of the function Trace can be found

in [39], where it has been used in the context of pairwise key establishment. Assume now that $b \in T_a$, then a further check on node b is performed (line 6). Indeed, node b could be already revoked. Hence, each node stores a Revocation Table (RT_a) that lists the revoked nodes. If both previous tests (lines 5 and 6) succeed, then a calls the function `Update` that updates the information about the last meeting with node b (line 7). For example, if node a meets b at a given time t_a, the function `Update` sets the information $\langle ID_b, t_a \rangle$ in the CT_a (a Check Table stored in node a memory). Node a uses a Time-out Table TT_a to store and signal the following time-outs:

(i) ALARM time-out, which is triggered after λ seconds are elapsed without remeeting node b.,

(ii) REVOKE time-out, which is triggered after δ seconds are elapsed from receiving/triggering a node revocation for node b—assuming that in these δ seconds no presence claim from b are received.

Then, for each meeting with non-revoked nodes in T_a, node a removes any previous time-out for the met node and sets a new ALARM time-out for that node (line 8). Note that both the update functions (lines 7 and 8) do not perform any operation if the time argument t_x is lower than the currently stored meeting time for the node ID_x: . This could happen in the case of a *virtual* meeting.

If the cooperation option is set ($COOP\_opt$=*true* in line 11), also the following steps are performed. For each not revoked node x traced by both node a and b (lines 12, 13, and 14), node a sends a CLAIM message to b carrying the meeting time between a and x. Each CLAIM message has the following format: $\langle ID_a, CLAIM, ID_x, \text{elapsed time} \rangle$, where ID_a is the sender of the claim message, CLAIM is the message type, ID_x is the ID of node x the claim is related to, and the last parameter indicates the meeting time between a and x. Another message type is ALARM, described in the following.

CMC_TimeOut (Algorithm 2) is triggered when a time-out expires. If on node a an ALARM time-out expires for node ID_b, this means that node a did not meet node ID_b for a time λ. Then, node a floods the network with an alarm (Algorithm 2, line 3) and a new REVOKE time-out for node b is set. Each ALARM message has the following format: $\langle ID_a, ALARM, ID_b \rangle$, where ID_a is the sender of the claim message, ALARM notifies the message type, and ID_b is the ID of node b the alarm is related to. When a REVOKE time-out expires, this means that after δ seconds elapsed from the alarm triggering, no evidence of the presence in the network of the suspected captured node appeared. In this latter case, a node revocation procedure for node b is invoked by node a.

CMC_Receive (Algorithm 3) is invoked when a message *MSG* is received. The fields of the message are assigned to local variables (line 2) and the type of the message is checked (line 3). Assume the message is of type ALARM: the executing node checks if the alarm is related to itself (line 4).

If the latter test fails, a further check is performed: the node checks whether the node ID_x is not already revoked (line 5). If the check succeeds, a REVOKE time-out is set through an `UpdateTimeOut` procedure. Note that a REVOKE time-out for node b already should be in place, this procedure does not override the existing REVOKE time-out and simply returns. If the ALARM is related to the executing node itself (test performed at line 4 fails) node a will flood the network with a presence CLAIM message (line 9). This measure prevents *false-positive detection,* that is, the revocation of nodes that are active in the network.

If the received message is of type CLAIM, this means that a node that was the target of an ALARM message is proving its presence; this message triggers a *virtual* meeting between a and the wrongly accused nodes (line 13). The overall result is that node a disables the REVOKE time-out for that node while restarting the ALARM time-out for the same node. These activities are also triggered when the $COOP\_opt$ is set (in fact, a CLAIM message is also sent in line 16, Algorithm 1). The objective of this invocation is to update the information on traced nodes via an information exchange with the met nodes.

Finally, when a receives a message issued by node b which is not originated within the protocol (e.g., it can be originated by the application layer), this message can be interpreted by the protocol as an evidence of the presence of node b. Therefore, this can be interpreted as a special case

Input: ID_a : ID of the executing node. ID_b : ID of the met node. t_a : Current time of node a. CT_a : Check Table stored in node a memory. RT_a : Revoked nodes table stored in node a memory. TT_a : Time out table stored in node a memory. λ : Alarm time. δ : Time for the accused node to prove its presence. $COOP\_opt$: Boolean variable for cooperation option.

```
1 begin
2   if NotSpecified (t_x) then
3       t_x = t_a;
4   end
5   if Trace (ID_a, ID_b)=1 then
6     if Is-Not-Revoked ( RT_a, ID_b) then
7       Update (CT_a,⟨ID_b, t_x⟩);
8       UpdateTimeOut (TT_a,
          ⟨ID_b, t_x + λ, ALARM⟩);
9     end
10  end
11  if COOP_opt = true then
12    foreach ⟨ID_x, t_x⟩ ∈ CT_a do
13      If Is-Not-Revoked (RT_a,ID_b) then
14        If Trace (ID_b, ID_x) = 1 then
15          ⟨t_old⟩ ←Look-Up (CT_a, ID_x);
16          ⟨ID_a, CLAIM, ID_x, t_old⟩ → b;
17        end
18      end
19    end
20  end
21 end
```

Algorithm 1: **CMC_Meeting($\mathbf{ID_x}$, $\mathbf{COOP\_opt}$, $\mathbf{t_x}$)**. Node meeting event handler.

Input: ID_a : ID of the executing node. ID_b : ID of the node which time-out is expired. t_a : Current time of node a. RT_a : Revoked nodes table stored in node a memory. TT_a : Time out table stored in node a memory. δ : Time for the accused node to prove its presence.

```
1 begin
2   if TimeOutKind(ALARM) then
3       Flooding (⟨ID_a, ALARM, ID_b⟩);
4       UpdatingTimeOut (TT_a,
        ⟨ID_b, t_a + δ, REVOKE⟩);
5   else
6      RevokeNode (RT_a,ID_x)
7   end
8 end
```

Algorithm 2: **CMC_TimeOut($\mathbf{ID_x}$)**. Node Time Out event handler.

Input: ID_a : ID of the executing node. t_a : Current time of node a. MSG : Received message. RT_a : Revocation Table stored in node a memory. δ : Time for the accused node to prove its presence.

```
1 begin
2   ⟨ID_b, msg_type, ID_x, t_x⟩ ← MSG;
3   if (msg_type = ALARM) then
4         if (ID_x ≠ ID_a) then
5            if Is-Not-Revoked (RT_a,ID_x) then
6               UpdateTimeOut (TT_a,
                ⟨ID_b, t_a + δ, REVOKE⟩);
7            end
8         else
9            Flooding (⟨ID_a, CLAIM, −, −⟩);
10        end
11  end
12  if(msg_type = CLAIM)then
13      CMC_Meeting(ID_x, false, t_x);
14  end
15  CMC_Meeting(ID_b, false, −);
16 end
```

Algorithm 3: **CMC_Receive(MSG)**. Received message event handler.

of a node meeting, and the appropriate actions are triggered (line 15).

5. Simulations and Discussion

We performed simulations using a self-developed discrete event simulator. The simulator is written in C++ and implements the Random Waypoint Mobility Model. The events (nodes meeting, node arrival at its selected destination, and alarms time-out) are pushed to and pulled from an ideal time-line. Initially, nodes are assumed to be randomly deployed over a network area. Then, until the simulation ends, for each node, a random speed and destination location are randomly chosen (within the bounds set by the user): this implies to analyze and to order all the meeting events and the node arrival events with reference to the time-line. While the time goes by, the events on the time-line are processed. The events corresponding to node arrival are processed as previously described (choosing a destination, a node speed, and analyzing the new generated events). The node meeting events are processed as the core part of our detection protocol, for example, updating the time-out or sharing information with the met nodes. The alarms time-out expiring event generates the network flooding.

As for the energy model, we adopted the one proposed in [40]. To plot each point in the following graphs (as well as for Figure 1), we performed a set of experiments and reported the averaged results; the number of experiments has been set to achieve a confidence interval of 98%.

The comparison on the detection time between our protocol and the reference solution has been performed considering the energy cost. In particular, the energy cost has been expressed as a frequency of network flooding, as explained later.

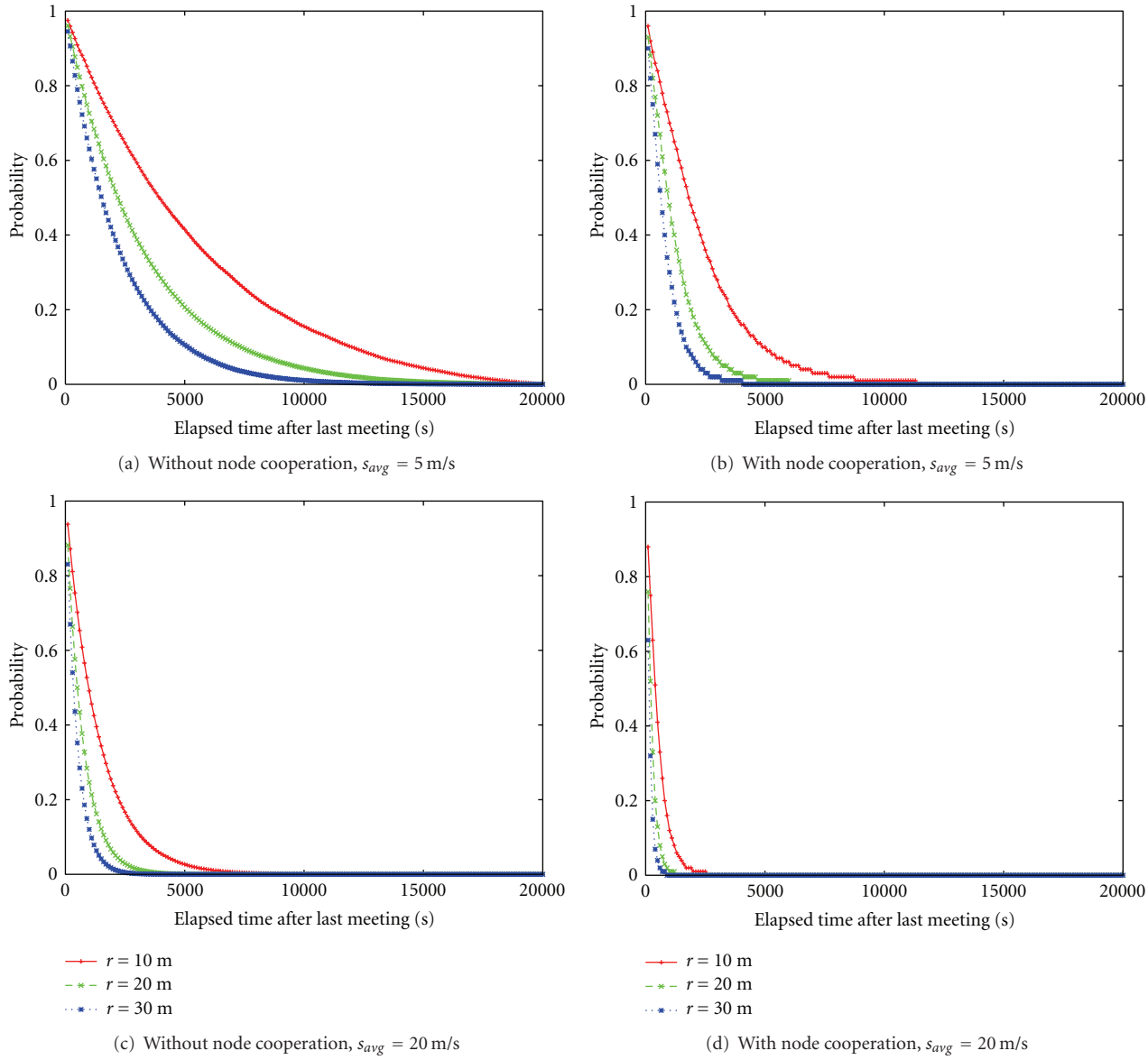
5.1. Node Remeeting. In order to better understand how mobility and cooperation can speed up the capture detection process, we performed a first set of simulations to assess the frequency of node-to-node meetings. We considered a network of $n = 100$ nodes randomly deployed over a square area of $1000\,\text{m} \times 1000\,\text{m}$. We used the random waypoint mobility model as the node mobility pattern. In particular, in our simulations we set the value for the minimum node speed greater than zero—this is a way to solve the decreasing average node speed problem of the random waypoint mobility model [8].

The experiment was set in this way: we choose two nodes a and b; when they meet, we set time at $t = 0$ and continued following these nodes thorough their network evolution to experimentally determine how long it takes for these two nodes to meet again, in both the noncooperative and in the cooperative case. Crucially, in the cooperative scenario, if node c meets node a and sends to it all the information c received during its last meeting with node b, this also accounts as a meeting between a and b.

We performed the simulation for different values of sensing radius and average node speed both for the non-cooperative and the cooperative scenario. The results are shown in Figure 2. The experiments support the following, simple intuitions: node cooperation increases the meeting probability; the higher is the sensing radius, the higher is the meeting probability; and the higher is the average node speed, the higher is the meeting probability. We used these results also to propose a reasonable value for the variable λ to be used in the implementation of our proposal, for both the cooperative and noncooperative case.

5.2. Experimental Results.

Parameters Tuning. As observed in previous work [25], all the protocols parameters are correlated, for example,



(a) Without node cooperation, $s_{avg} = 5$ m/s

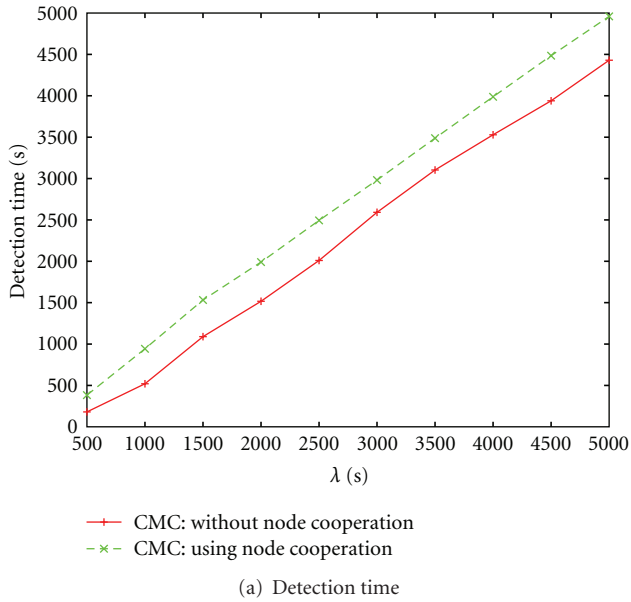
(b) With node cooperation, $s_{avg} = 5$ m/s

(c) Without node cooperation, $s_{avg} = 20$ m/s

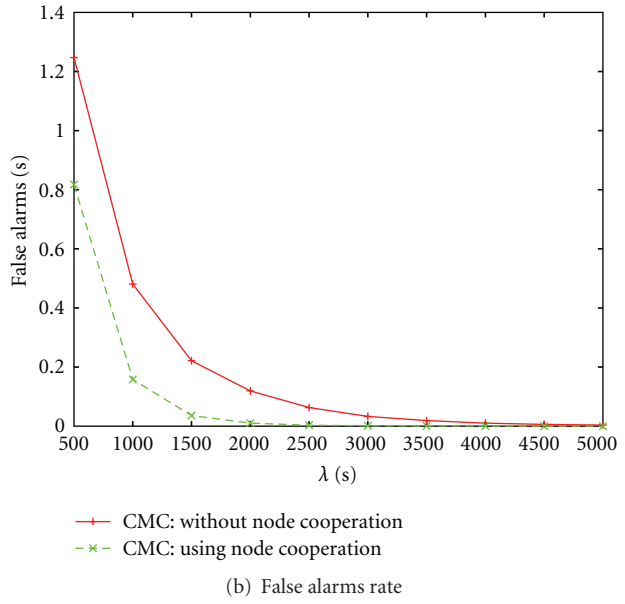
(d) With node cooperation, $s_{avg} = 20$ m/s

Figure 2: Probability for two nodes not to remeet: $n = 100$.

increasing the average speed of the network would increase the number of meetings between nodes, hence reducing the number of false alarms. However, if we assume that parameters such as the network size, the nodes' mobility, and the network area are given, the main parameters that the network administrator can set is the alarm time λ. In Figures 3(a) and 3(b) we show the influence of λ over the detection time and the rate of false positive alarms. We notice that increasing the alarm time also increases the detection time while decreasing the number of false positives. In particular, from Figure 3(a), we observe that the detection time increases linearly with λ. Furthermore, we observe that the detection time using node cooperation is higher than the one without node cooperation. The motivation follows from the fact that without node cooperation nodes have stale information about the presence of the traced nodes. So, when a node is really captured, in the noncooperative scenario there will be some nodes that are already not meeting the captured node for a while. These nodes would raise the capture alarm before λ seconds elapses after the real node capture, hence decreasing the detection time with respect to the cooperative protocol. From Figure 3(a), we observe that the false alarms rate decreases exponentially with λ. Comparison between Figures 3(a) and 3(b) suggests that there is a tradeoff between the detection time and the number of false alarms. In order to give a straight and fair comparison between the proposed solutions (cooperative and noncooperative) and also with the reference solution, in the following section, we compare the detection time of the solutions on the basis of the overall energetic cost.



(a) Detection time



(b) False alarms rate

FIGURE 3: Influence of λ on CMC performances: $n = 100$, $r = 20$ m, *Avg speed* = 15 m/s.

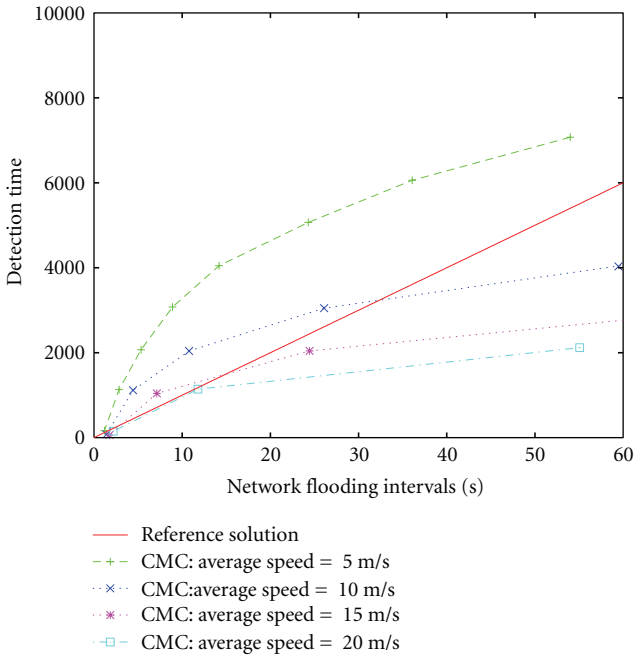
Energy-Driven Comparison. One of the key issues in ad hoc and sensor network is the energy consumption. Hence, we compared our proposal with the reference solution focusing on energy consumption. To provide an evaluation of our protocols in a manner that is device-independent, we chose to express the energy consumption in terms of generated messages. As for the energy devoted to computation, we considered the cost be negligible, as in [40].

The main communication cost of both our protocol and the reference solution is the number of flooding. The reference solution uses the flooding as a presence claim message while our protocol uses the flooding for both alarm broadcast and alarm-triggered presence notification; the latter flooding occurs when a node that has been erroneously advertised as possibly compromised sends (floods) a claim of its actual presence. To simplify our discussion, we assume that a network flooding corresponds to sending and to receiving a message by each network node. This is not always the case; actually, the load for broadcasting varies with different network parameters and the specific broadcasting protocol used [34]. However, this approximation is good enough to achieve our goal, that is, to show the qualitative improvement of our solution over the reference solution. To better appreciate the comparison with the reference solution—where a flooding occurs every time interval—in the following graphs, we report on the x-axis the time interval between two subsequent flooding, instead of the flooding frequency. Note that once the flooding interval is fixed, also the amount of required energy is fixed, and we can plot the performance of our protocol when using the same amount of energy, that is, the same amount of messages.

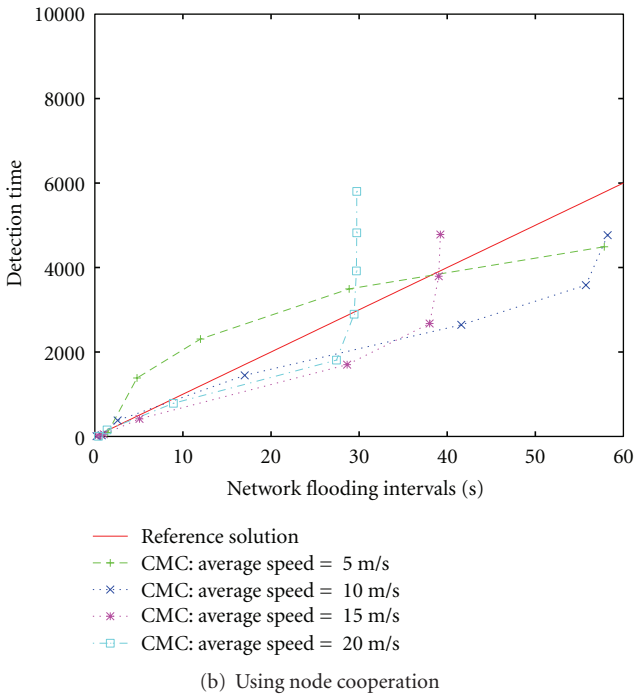
In our simulation, we analyze how increasing the energy overhead affects the detection time. In other words, we fix the energy overhead at the same level for both protocols under evaluation, and measure which protocol achieves the best detection time.

Performance. To compare the performance of the proposed solution with the reference solution presented in Section 3.1, we implemented our protocol. In what follows, we fix a sensing radius of $r = 20$ m. Since nodes in ad hoc settings could have strict memory constraints (e.g., in sensor network), in our simulations, we assume that each node traces a small number of other nodes. In fact, as a result of the pseudorandom function Trace (Algorithm 1, line 2) each node traces exactly 5 other network nodes. For the cooperative scenario, when two nodes a and b meet, they exchange the information concerning the nodes tracked by both a and b; we assume that this information can be contained in one message. Indeed, the number of shared traced nodes can be up to 5 (number of nodes traced by each node), but in practice, it turns out to be much smaller, on average (0.25 in our setting). We simulated our protocol with and without node cooperation, varying the alarm time from 250 to 8000 seconds and the average node speed from 5 m/s to 20 m/s. Figures 4(a) and 4(b) show the results of the simulation of our protocol without and with cooperation, respectively.

Figure 4(a) shows the results when cooperation is switched off, for the two protocols and different speeds. On the x-axis, we fix the flooding interval for the reference solution protocol. In this way, the detection time is also fixed for the reference solution and it does not change when changing the speed. The quality of the detection for the reference solution is just linear: by doubling the flooding interval also the detection time doubles, while the energy cost halves. Figure 4(a) confirms our intuition: mobility with local cooperation can help computing global properties incurring in a small overhead.



(a) Without node cooperation



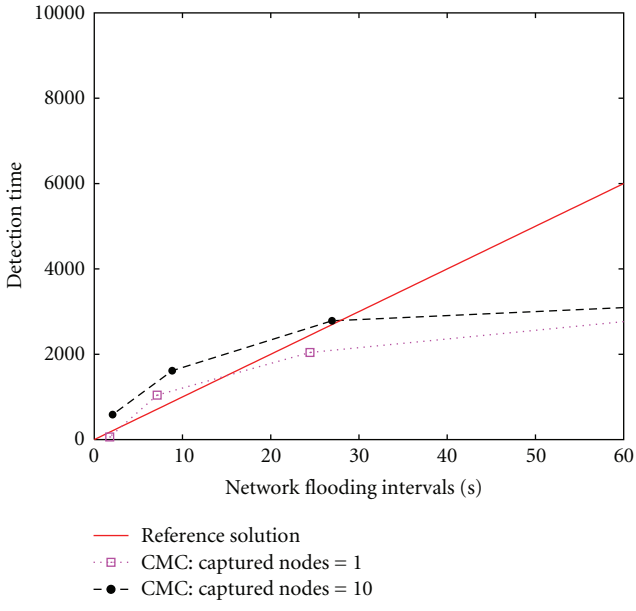
(b) Using node cooperation

FIGURE 4: CMC Detection time: $n = 100$, $r = 20$ m.

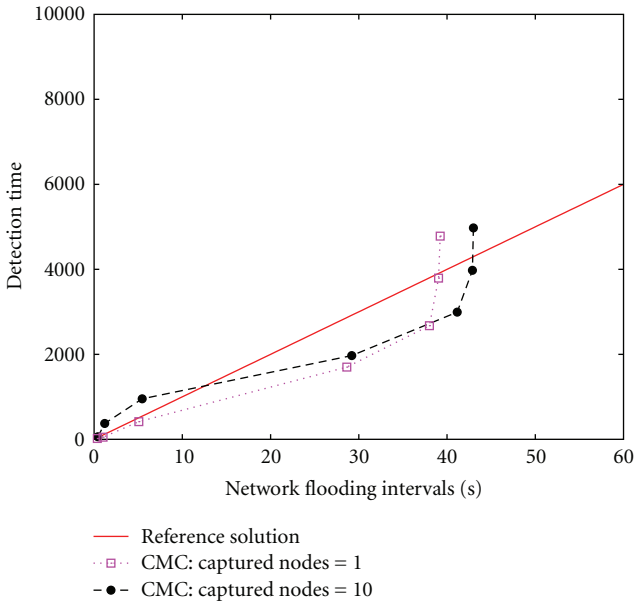
In this simulation scenario, for a reasonable speed of nodes, our protocol outperforms the reference solution. Take, as an example, a flooding interval of 50 seconds. From Figure 4(a), we can see that the detection time of the reference solution protocol is 5000 seconds. The performance of our protocol depends on the average speed of the system. If the average speed of the system is slow, for example, 5 m/s, then the detection time is more than 6000 seconds. However, if the network nodes move faster, then our solution improves over the reference solution. For instance, when the average speed is 20 m/s, the detection time is as low as 1600 seconds, much faster than the reference solution. From this experiment, it is also clear that the performance of our protocol depends on the average speed in the network: the faster the better. While the reference solution is an excellent solution for slow networks, for example, where nodes are carried by humans walking, our solution is the best for faster networks, and it is always the best when the energy overhead must be low. Now, we will switch cooperation on, and see that the performance of our protocol increases considerably, even though with some drawbacks when the energy budget is small.

Figure 4(b) describes the performance of our protocol when using cooperation. When the network flooding frequency is high, that is, network flooding interval is small, cooperation is very effective. Further, with cooperation, the performance of our protocol improves as the average speed of the nodes increases. In this case, our protocol is better than the reference solution even when starting from very high flooding frequency, that is, starting from systems that are very fast in detecting the node capture attack and that, consequently, have very high energy requirements. What is less intuitive is that cooperation is not useful when we move to more energy-saving systems. Take, as an example, a network where the average speed is 15 m/s. Our protocol is better than the reference solution whenever the design goal is to have a network with more energy available and to achieve a small detection time, that is, in Figure 4(b), whenever the flooding interval is smaller than 38 seconds. However, when considering a network with more stringent energy requirements, for example, when the flooding interval is 50 seconds, then it is simply not possible to reach such low energy costs by using cooperation. Cooperation has a cost, which is higher when the network is faster—indeed, in a faster network, the nodes meet more frequently, and thus cooperation is higher. In this case, the correct design guideline is to use our protocol with cooperation, if the objective is to have a system that is fast in detecting the node capture attack, though using more energy—in particular, in our example until a flooding interval of 38 seconds—and then to switch cooperation off, to get a cheaper protocol that can be used when the flooding interval can be larger.

As described in Figure 4(b), the limits of cooperation appear sooner in faster networks. This is intuitive, cooperation is more costly when nodes meet more often, and so the tradeoff moves toward noncooperation earlier. The implications of using mobility and local communications to compute global properties are not self-evident. If the network is fast enough, it is always better to use protocols like the one we propose rather than using static approaches like the reference solution. However, node cooperation flavored techniques, which appears to be effective in any case, have the result of making the information in the network spread faster, but at a cost.



(a) Without node cooperation



(b) Using node cooperation

FIGURE 5: CMC Detection time under massive attack: $n = 100$, $r = 20$ m, $s_{avg} = 15$ m/s.

5.3. Massive Attacks. In order to investigate the behavior of our protocol under a massive attack, we simulated the capture of 10% of the network nodes (10 out of 100) at the same time. We fixed the average speed at 15 m/s. Simulation results are shown in Figures 5(a) and 5(b) for the noncooperative and cooperative scenarios, respectively. For both cases, the figures show the result for one captured node and 10 captured nodes in a network of 100 nodes. From both figures, we can see that all the protocols, both the reference solution and our solution, with or without cooperation, are robust against massive attacks. Indeed, the small differences in performance do not justify a change in the defense strategy but for small intervals.

5.4. Other Mobility Patterns. We stress once again that the aim of this work is to give a proof of concept that both node mobility and node cooperation can help thwarting the node capture attack. Hence, to abstract from mobility details we choose to use the Random Waypoint Mobility Model. Mobility models based on randomly moving nodes may, for example, provide useful analytical approximations to the motion of vehicles that operate in dispatch mode or delivery mode [41]. It is important to note that the results obtained in this work are not directly applicable to others scenario-inspired mobility models [12]; for instance, while intermeeting time follows an exponential distribution under the RWM, intermeeting time is shown to be better approximated by a power-law distribution in some scenarios [12, 42]. However, it is also interesting to note that our solution allows the network to let autonomously emerge the subgroups of nodes that meet with higher frequency (communities). In fact, this can be done leveraging the false-positive alarm: if node a sends a high number of false alarms (further revoked by the accused node) related to node b, this implies that a actually does not meet with b with "high" frequency. This information can be interpreted as if a and b do not belong to the same community.

6. Conclusions

In this paper we have proposed, to the best of our knowledge, the first distributed solution to a major security threat in MANETs: the node capture attack. Our solution is based on the intuition that node mobility, together with local node cooperation, can be leveraged to design security protocols that are extremely effective and energy-efficient. We have also developed a protocol that, increasing the level of cooperation among nodes, makes global information flow faster in the network, even if at a cost in terms of energy. The experiments clearly show that leveraging mobility and cooperation helps in designing effective and efficient protocols. In particular, we also pointed out that there is critical speed necessary to induce enough information flow to make these new protocols outperform traditional ones, designed for static networks.

We believe that the ideas and protocols introduced in this paper pave the way for further research in the area; furthermore, even if specifically suited to address a major security threat, they could be also adopted in other scenarios to support other emergent properties as well.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments that helped to improve the quality of this paper. The authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of supporting Organizations. This work

was partly supported by: (1) the Spanish Ministry of Education through projects TSI2007-65406-C03-01 "E-AEGIS" and CONSOLIDER INGENIO 2010 CSD2007-0004 "ARES," (2) the Government of Catalonia under grant 2005 SGR 00446, and (3) the project APPLICAZIONI GOVERNATIVE LEGATE ALL'USO DEL PRS GALILEO (PRESAGO)—contract ASI I/030/07/0 starting September 6, 2007.

References

[1] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P '03)*, September 2003.

[2] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *Proceedings of the 3rd International Conference on Information Processing in Sensor Networks (IPSN '04)*, April 2004.

[3] M. Demirbas and Y. Song, "An RSSI-based scheme for sybil attack detection in wireless sensor networks," in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, pp. 564–568, New York, NY, USA, June 2006.

[4] R. Di Pietro, L. V. Mancini, and A. Mei, "Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks," *Wireless Networks*, vol. 12, no. 6, pp. 709–721, 2006.

[5] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 80–89, 2007.

[6] B. Parno, A. Perrig, and V. D. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P '05)*, 2005.

[7] Information Processing Technology Office (IPTO) Defense Advanced Research Projects Agency (DARPA), BAA 07-46 LANdroids Broad Agency Announcement, 2007, http://www.darpa.mil/index.html.

[8] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commununications of ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[9] S. Capkun, J.-P. Hubaux, and L. Buttyán, "Mobility helps security in ad hoc networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 46–56, 2003.

[10] C. Piro, C. Shields, and B. N. Levine, "Detecting the sybil attack in mobile ad hoc networks," in *Proceedings of the 2nd International Conference on Security and Privacy in Communication Networks (SecureComm '06)*, Baltimore, Md, USA, 2006.

[11] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, pp. 85–79, 1998.

[12] G. Sharma, R. Mazumdar, and N. B. Shroff, "Delay and capacity trade-offs in mobile ad hoc networks: a global perspective," in *Proceedings of the 25th Conference on Computer Communications (INFOCOM '06)*, 2006.

[13] A. Becher, E. Becher, Z. Benenson, and M. Dornseif, "Tampering with motes: real-world physical attacks on wireless sensor networks," in *Proceeding of the 3rd International Conference on Security in Pervasive Computing (SPC '06)*, pp. 104–118, 2006.

[14] M. Grossglauser and M. Vetterli, "Locating nodes with EASE: last encounter routing in ad hoc networks through mobility diffusion," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, Calif, USA, 2003.

[15] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, Miami, Fla, USA, March 2005.

[16] C. fan Hsin and M. Liu, "A distributed monitoring mechanism for wireless sensor networks," in *Proceedings of the Workshop on Wireless Security (WiSe '02)*, pp. 57–66, 2002.

[17] C. fan Hsin and M. Liu, "Self-monitoring of wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 462–476, 2006.

[18] N. Hayashibara, A. Cherif, and T. Katayama, "Failure detectors for large-scale distributed systems," in *Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems (SRDS '02)*, Suita, Japan, October 2002.

[19] S. Ranganathan, A. D. George, R. W. Todd, and M. C. Chidester, "Gossip-style failure detection and distributed consensus for scalable heterogeneous clusters," *Cluster Computing*, vol. 4, no. 3, pp. 197–209, 2001.

[20] R. Curtmola and S. Kamara, "A mechanism for communication-efficient broadcast encryption over wireless ad hoc networks," *Electronic Notes in Theoretical Computer Science*, vol. 171, no. 1, pp. 57–69, 2007.

[21] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 29–42, Washington, DC, USA, November 2004.

[22] P. Tague and R. Poovendran, "Modeling adaptive node capture attacks in multi-hop wireless networks," *Ad Hoc Network*, vol. 5, no. 6, pp. 801–814, 2007.

[23] P. Tague, D. Slater, J. Rogers, and R. Poovendran, "Vulnerability of network traffic under node capture attacks using circuit theoretic analysis," in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM '08)*, pp. 161–165, 2008.

[24] M. Conti, R. Di Pietro, A. Gabrielli, L. V. Mancini, and A. Mei, "The quest for mobility models to analyse security in mobile ad hoc networks," in *Proceedings of the 7th International Conference on Wired/Wireless Internet Communications (WWIC '09)*, pp. 85–96, May 2009.

[25] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "Emergent properties: detection of the node-capture attack in mobile wireless sensor networks," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 214–219, 2008.

[26] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 32–40, September 2007.

[27] J. P. G. Sterbenz, R. Krishnan, R. R. Hain, et al., "Survivable mobile wireless networks: issues, challenges, and research directions," in *Proceedings of the 1st ACM Workshop on Wireless Security (WiSe '02)*, pp. 31–40, Atlanta, Ga, USA, 2002.

[28] R. Di Pietro, L. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Data security in unattended sensor networks," *IEEE Transactions on Computers*, vol. 58, no. 11, pp. 1500–1511, 2009.

[29] R. Di Pietro, L. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Playing hide-and-seek with a focused mobile adversary in unattended wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1463–1475, 2009.

[30] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 1312–1321, San Franciso, Calif, USA, March 2003.

[31] E. Hyytiä, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 680–694, 2006.

[32] K. Sun, P. Ning, and C. Wang, "Fault-tolerant cluster-wise clock synchronization for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 177–189, 2005.

[33] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 194–205, 2002.

[34] L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti, "Localized techniques for broadcasting in wireless sensor networks," in *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC '04)*, Philadelphia, Pa, USA, October 2004.

[35] B. Burns, O. Brock, and B. N. Levine, "MORA routing and capacity building in disruption-tolerant networks," *Ad Hoc Networks*, vol. 6, no. 4, pp. 600–620, 2008.

[36] H. Liu, P.-J. Wan, X. Liu, and F. Yao, "A distributed and efficient flooding scheme using 1-hop information in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 5, pp. 658–671, 2007.

[37] S. M. M. Rahman, N. Nasser, A. Inomata, T. Okamoto, M. Mambo, and E. Okamoto, "Anonymous authentication and secure communication protocol for wireless mobile ad hoc networks," *Security and Communication Networks*, vol. 1, no. 2, pp. 179–189, 2008.

[38] M. Striki, J. Baras, and K. Manousakis, "A robust, distributed TGDH-based scheme for secure group communications in MANET," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, May 2004.

[39] R. Di Pietro, L. V. Mancini, and A. Mei, "Efficient and resilient key discovery based on pseudo-random key pre-deployment," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 2991–2998, 2004.

[40] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '05)*, 2005.

[41] S. Bandyopadhyay, E. J. Coyle, and T. Falck, "Stochastic properties of mobility models in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 11, pp. 1218–1229, 2007.

[42] A. Chaintreau, P. Hui, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 692654, 11 pages
doi:10.1155/2009/692654

Review Article

Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures

Jing Liu,[1] Yang Xiao,[1] Kaveh Ghaboosi,[2] Hongmei Deng,[3] and Jingyuan Zhang[1]

[1] *Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, USA*
[2] *The Centre for Wireless Communications, University of Oulu, P.O. Box 4500, FI-90014, Finland*
[3] *Intelligent Automation, Inc., Rockville, MD 20855, USA*

Correspondence should be addressed to Yang Xiao, yangxiao@ieee.org

Received 25 December 2008; Revised 17 June 2009; Accepted 19 July 2009

Recommended by Yi-Bing Lin

Botnets become widespread in wired and wireless networks, whereas the relevant research is still in the initial stage. In this paper, a survey of botnets is provided. We first discuss fundamental concepts of botnets, including formation and exploitation, lifecycle, and two major kinds of topologies. Several related attacks, detection, tracing, and countermeasures, are then introduced, followed by recent research work and possible future challenges.

1. Introduction

The untraceable feature of coordinated attacks is just what hackers/attackers demand to compromise a computer or a network for their illegal activities. Once a group of hosts at different locations controlled by a malicious individual or organization to initiate an attack, one can hardly trace back to the origin due to the complexity of the Internet. For this reason, the increase of events and threats against legitimate Internet activities such as information leakage, click fraud, denial of service (DoS) and attack, E-mail spam, etc., has become a very serious problem nowadays [1]. Those victims controlled by coordinated attackers are called zombies or bots which derives from the word "robot." The term of bots is commonly referred to software applications running as an automated task over the Internet [2]. Under a command and control (C2, or C&C) infrastructure, a group of bots are able to form a self-propagating, self-organizing, and autonomous framework, named botnet [3]. Generally, to compromise a series of systems, the botnet's master (also called as herder or perpetrator) will remotely control bots to install worms, Trojan horses, or backdoors on them [3]. The majority of those victims are running Microsoft Windows operating system [3]. The process of stealing host resources to form a botnet is so called "scrumping" [3].

Fortunately, botnet attacks and the corresponding preventive measures or tracking approaches have been studied by industry and academia in last decades. It is known that botnets have thousands of different implementations, which can be classified into two major categories based on their topologies [4]. One typical and the most common type is Internet Relay Chat-(IRC-) based botnets. Because of its centralized architecture, researchers have designed some feasible countermeasures to detect and destroy such botnets [5, 6]. Hence, newer and more sophisticated hackers/attackers start to use Peer to Peer (P2P) technologies in botnets [4, 7]. P2P botnets are distributed and do not have a central point of failure. Compared to IRC-based botnets, they are more difficult to detect and take down [4]. Besides, most of its existing studies are still in the analysis phase [4, 7].

Scholars firstly discovered botnets due to the study on Distributed DoS (DDoS) attacks [8]. After that, botnet features have been disclosed using probing and Honeypots [9–11]. Levy [12] mentioned that spammers increasingly relied on bots to generate spam messages, since bots can hide their identities [13]. To identify and block spam, blacklists are widely used in practice. Jung and Sit [14] found that 80% of spammers could be detected by blacklists of MIT in 2004. Besides, blacklists also impact on other hostile actions. Through examining blacklist abuse by botnet's

masters, Ramachandran et al. [15] noted that those masters with higher premiums on addresses would not present on blacklists. Thus, only deploying blacklists may be not enough to address the botnet problem.

So far, industry and much of academia are still engaged in damage control via patch-management rather than fundamental problem solving. In fact, without innovative approaches to removing the botnet threat, the full utility of the Internet for human beings will still be a dream. The major objective of this paper is to exploit open issues in botnet detection and preventive measures through exhaustive analysis of botnets features and existing researches.

The rest of this paper is organized as follows. In Section 2, we provide a background introduction as well as the botnet classification. Section 3 describes the relevant attacks. Section 4 elaborates on the detection and tracing mechanisms. We introduce preventive measures in Section 5. The conclusion and future challenges are discussed in Section 6.

2. Classification

Botnets are emerging threats with billions of hosts worldwide infected. Bots can spread over thousands of computers at a very high speed as worms do. Unlike worms, bots in a botnet are able to cooperate towards a common malicious purpose. For that reason, botnets nowadays play a very important role in the Internet malware epidemic [16]. Many works try to summarize their taxonomy [17, 18], using properties such as the propagation mechanism, the topology of C2 infrastructure used, the exploitation strategy, or the set of commands available to the perpetrator. So far, botnet's master often uses IRC protocol to control and manage the bots. For the sake of reducing botnet's threat efficiently, scholars and researchers emphasize their studies on detecting IRC-based botnets. Generally speaking, the academic literature on botnet detection is sparse. In [19], Strayer et al. presented some metrics by flow analysis on detecting botnets. After filtering IRC session out of the traffic, flow-based methods were applied to discriminate malicious from benign IRC channels. The methods proposed by [20, 21] combined both application and network layer analysis. Cooke et al. [22] dealt with IRC activities at the application layer, using information coming from the monitoring of network activities. Some authors had introduced machine learning techniques into botnet detection [23], since they led a better way to characterize botnets. Currently, honeynets and Intrusion Detection System (IDS) are two major techniques to prevent their attacks. Honeynets can be deployed in both distributed and local context [9]. They are capable of providing botnet attacking information but cannot tell the details such as whether the victim has a certain worm [9]. The IDS uses the signatures or behavior of existing botnets for reference to detect potential attacks. Thus, to summarize the characteristics of botnets is significant for secure networks. To the best of our knowledge, we have not found any other work about anomaly-based detection for botnets. Before going to the discussion of botnet attacks and preventive measures, we will introduce some relevant terms and classification of bots in the rest of this section.

2.1. Formation and Exploitation. To illustrate the formation and exploitation, we take a spamming botnet as an example. A typical formation of botnet can be described by the following steps [3], as shown in Figure 1.

(1) The perpetrator of botnet sends out worms or viruses to infect victims' machines, whose payloads are bots.

(2) The bots on the infected hosts log into an IRC server or other communications medium, forming a botnet.

(3) Spammer makes payment to the owner of this botnet to gain the access right.

(4) Spammer sends commands to this botnet to order the bots to send out spam.

(5) The infected hosts send the spam messages to various mail servers in the Internet.

Botnets can be exploited for criminally purposes or just for fun, depending on the individuals. The next section will go into the details of various exploitations.

2.2. Botnet Lifecycle. Figure 2 shows the lifecycle of a botnet and a single bot [16].

2.3. IRC-Based Bot. IRC is a protocol for text-based instant messaging among people connected with the Internet. It is based on Client/Server (C/S) model but suited for distributed environment as well [18]. Typical IRC severs are interconnected and pass messages from one to another [18]. One can connect with hundreds of clients via multiple servers. It is so-called multiple IRC (mIRC), in which communications among clients and a server are pushed to those who are connected to the channel. The functions of IRC-based bots include managing access lists, moving files, sharing clients, sharing channel information, and so on [18]. Major parts of a typical IRC bot attack are showed in Figure 3 [18].

(i) *Bot* is typically an executable file triggered by a specific command from the IRC sever. Once a bot is installed on a victim host, it will make a copy into a configurable directory and let the malicious program to start with the operating system. Consider Windows as an instance, the bots sized no more than 15 kb are able to add into the system registry (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVerssion\Run\) [18]. Generally, bots are just the payload of worms or the way to open a backdoor [18].

(ii) *Control channel* is a secured IRC channel set up by the attacker to manage all the bots.

(iii) *IRC Server* may be a compromised machine or even a legitimate provider for public service.

(iv) *Attacker* is the one who control the IRC bot attack.

The attacker's operations have four stages [16].

(1) *The first one is the Creation Stage*, where the attacker may add malicious code or just modify an existing one out of numerous highly configurable bots over the Internet [16].

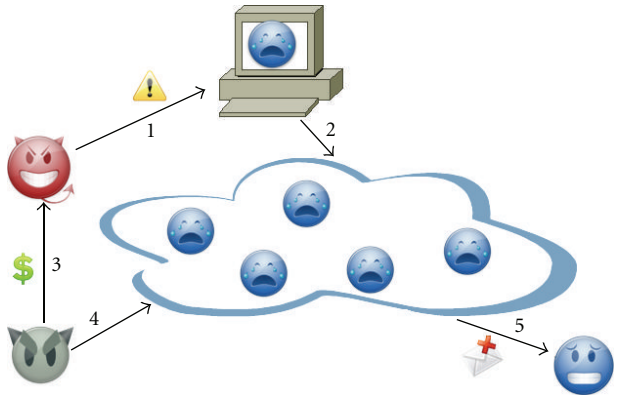


Figure 1: Using a botnet to send spam [3].

(2) *The second one is the Configuration Stage*, where the IRC server and channel information can be collected [16]. As long as the bot is installed on the victim, it will automatically connect to the selected host [16]. Then, the attacker may restrict the access and secure the channel to the bots for business or some other purpose [16]. For example, the attacker is able to provide a list of bots for authorized users who want to further customize and use them for their own purpose.

(3) *The third one is the Infection Stage*, where bots are propagated by various direct and indirect means [16]. As the name implies, direct techniques exploit vulnerabilities of the services or operating systems and are usually associated with the use of viruses [16]. While the vulnerable systems are compromised, they continue the infection process such that saving the time of attacker to add other victims [16]. The most vulnerable systems are Windows 2000 and XP SP1, where the attacker can easily find unpatched or unsecured (e.g., without firewall) hosts [16]. By contrary, indirect approaches use other programs as a proxy to spread bots, that is, using distributed malware through DCC (Direct Client-to-Client) file exchange on IRC or P2P networks to exploit the vulnerabilities of target machines [16].

(4) *The forth one is the Control Stage*, where the attacker can send the instructions to a group of bots via IRC channel to do some malicious tasks.

2.4. P2P-Based Bot. Few papers focus on P2P-based bots so far [4, 24–30]. It is still a challenging issue. In fact, using P2P ad hoc network to control victim hosts is not a novel technique [26]. A worm with a P2P fashion, named Slapper [27], infected Linux system by DoS attack in 2002. It used hypothetical clients to send commands to compromised hosts and receive responses from them [27]. Thereby, its network location could be anonymous and hardly be monitored [27]. One year after, another P2P-based bot appeared, called Dubbed Sinit [28]. It used public key cryptography for update authentication. Later, in 2004, Phatbot [29] was created to send commands to other compromised hosts using a P2P system. Currently, Storm Worm [24] may be the most wide-spread P2P bot over the Internet. Holz et al. have analyzed it using binary and network tracing [24]. Besides, they also proposed some techniques to disrupt the communication of a P2P-based botnet, such as eclipsing content and polluting the file.

Nevertheless, the above P2P-based bots are not mature and have many weaknesses. Many P2P networks have a central server or a seed list of peers who can be contacted for adding a new peer. This process named bootstrap has a single point of failure for a P2P-based botnet [25]. For this reason, authors in [25] presented a specific hybrid P2P botnet to overcome this problem.

Figure 4 presents the C2 architecture of the hybrid P2P-based botnet proposed by [25]. It has three client bots and five servant bots, who behave both as clients and servers in a traditional P2P file sharing system. The arrow represents a directed connection between bots. A group of servant bots interconnect with each other and form the backbone of the botnet. An attacker can inject his/her commands into any hosts of this botnet. Each host periodically connects to its neighbors for retrieving orders issued by their commander. As soon as a new command shows up, the host will forward this command to all nearby servant bots immediately. Such architecture combines the following features [25]: (1) it requires no bootstrap procedure; (2) only a limited number of bots nearby the captured one can be exposed; (3) an attacker can easily manage the entire botnet by issuing a single command. Albeit the authors in [25] proposed several countermeasures against this botnet attack, more researches on both architecture and prevention means are still needed in the future. The relevant future work will be discussed in Section 6.

2.5. Types of Bots. Many types of bots in the network have already been discovered and studied [9, 16, 17]. Table 1 will present several widespread and well-known bots, together with their basic features. Then, some typical types will be studied in details.

2.5.1. Agobot. This well-known bot is written in C/C++ with cross-platform capabilities [9]. It is the only bot so far that utilizes a control protocol in IRC channel [9]. Due to its standard data structures, modularity, and code documentation, Agobot is very easy for attacker to extend commands for their own purposes by simply adding new function into the CCommandHandler or CScanner class [9]. Besides, it has both standard and special IRC commands for harvesting sensitive information [17]. For example, it can request the bot to do some basic operations (accessing a file on the compromised machine by "bot.open" directive) [17]. Also, Agobot is capable of securing the system via closing NetBIOS shares, RPC-DCOM, for instance [17]. It has various commands to control the victim host, for example, using "pctrl" to manage all the processes and using "inst" to manage autostart programs [17]. In addition, it has the following features [17]: (1) it is IRC-based C2 framework,

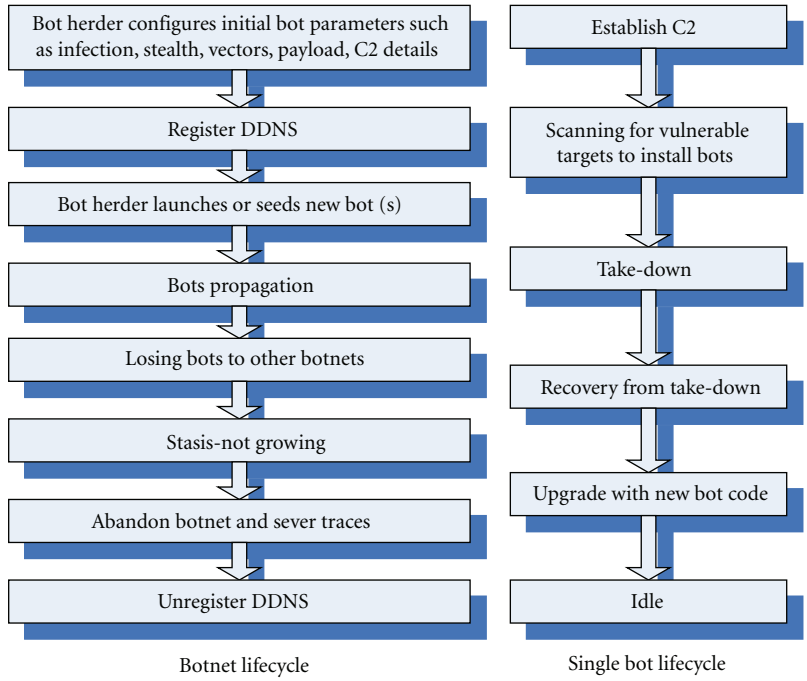


Figure 2: Lifecycle of a Botnet and of a single Bot [16].

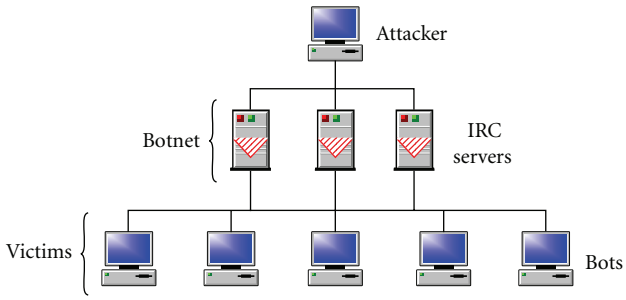


Figure 3: Major parts of a typical IRC Bot attack [18].

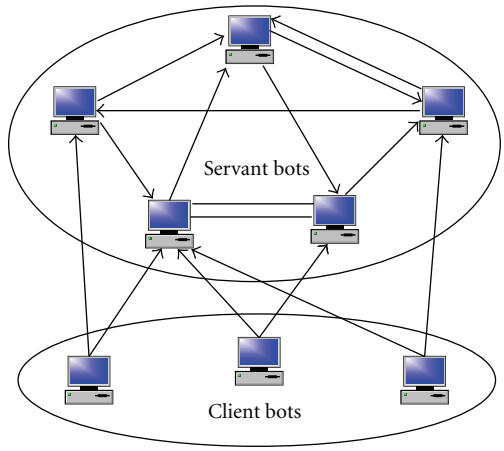


Figure 4: The C2 architecture of a hybrid P2P botnet proposed by [25].

(2) it can launch various DoS attacks, (3) it can attack a large number of targets, (4) it offers shell encoding function and limits polymorphic obfuscations, (5) it can harvest the sensitive information via traffic sniffing (using libpcap, a packet sniffing library [9]), key logging or searching registry entries, (6) it can evade detection of antivirus software either through patching vulnerabilities, closing back doors or disabling access to anti-virus sites (using NTFS Alternate Data Stream to hide its presence on victim host [9]), and (7) it can detect debuggers (e.g., SoftIce and Ollydbg) and virtual machines (e.g., VMware and Virtual PC) and thus avoid disassembly [9, 17].

To find a new victim, Agobot just simply scans across a predefined network range [17]. Nevertheless, it is unable to effectively distribute targets among a group of bots as a whole based on current command set [17].

2.5.2. SDBot. SDBot's source code is not well written in C and has no more than 2500 lines, but its command set and features are similar to Agobot [9, 17]. It is published under GPL [9, 17]. Albeit SDBot has no propagation capability and only provides some basic functions of host control, attackers still like this bot since its commands are easy to extend [17]. In addition, SDBot has its own IRC functions such as spying and cloning [17]. Spying is just recording the activities of a specified channel on a log file [17]. Cloning means that the bot repeats to connect one channel [17]. At present, SDBot may be the most active bot used in the wild [9]. There are plenty of auxiliary patches available on the Internet, including non-malicious ones [17].

Table 1: Types of bots.

Types	Features
Agobot Phatbot Forbot Xtrembot	They are so prevalent that over 500 variants exist in the Internet today. Agobot is the only bot that can use other control protocols besides IRC [9]. It offers various approaches to hide bots on the compromised hosts, including NTFS Alternate Data Stream, Polymorphic Encryptor Engine and Antivirus Killer [16].
SDBot RBot UrBot UrXBot	SDBot is the basis of the other three bots and probably many more [9]. Different from Agobot, its code is unclear and only has limited functions. Even so, this group of bots is still widely used in the Internet [16].
SpyBot NetBIOS Kuang Netdevil KaZaa	There are hundreds of variants of SpyBot nowadays [17]. Most of their C2 frameworks appear to be shared with or evolved from SDBot [17]. But it does not provide accountability or conceal their malicious purpose in codebase [17].
mIRC-based GT-Bots	GT (Global Threat) bot is mIRC-based bot. It enables a mIRC chat-client based on a set of binaries (mainly DLLs) and scripts [16]. It often hides the application window in compromised hosts to make mIRC invisible to the user [9].
DSNX Bots	The DSNX (Data Spy Network X) bot has a convenient plug-in interface for adding a new function [16]. Albeit the default version does not meet the requirement of spreaders, plugins can help to address this problem [9].
Q8 Bots	It is designed for Unix/Linux OS with the common features of a bot, such as dynamic HTTP updating, various DDoS-attacks, execution of arbitrary commands and so forth. [9].
Kaiten	It is quite similar to Q8 Bots due to the same runtime environment and lacking of spreader as well. Kaiten has an easy remote shell, thus it is convenient to check further vulnerabilities via IRC [9].
Perl-based bots	Many variants written in Perl nowadays [9]. They are so small that only have a few hundred lines of the bots code [9]. Thus, limited fundamental commands are available for attacks, especially for DDoS-attacks in Unix-based systems [9].

SDBot's is essentially a compact IRC implementation [17]. To contact the IRC server, it first sends identity information, for example, USER and NICK [17]. As long as it gets an admission message (PING) from the server, the bot will acknowledge this connection with a PONG response [17]. While the bot receives the success code (001 or 005) for connection, it can request a hostname by USERHOST and join the channel by JOIN message [17]. Once it receives a response code 302, this bot has successfully participated in the IRC channel and the master can control it via some IRC commands (e.g., NOTICE, PRIVMSG, or TOPIC) [17].

With the help of many powerful scanning tools, SDBot can easily find the next victim [17]. For instance, using NetBIOS scanner, it can randomly choose a target located in any predefined IP range [17]. Since the SDBot is able to send ICMP and UDP packets, it is always used for simple flooding attacks [17]. Moreover, a large number of variants capable of DDoS attack are available in the wild [17].

2.5.3. SpyBot. SpyBot is written in C with no more than 3,000 lines, and has pretty much variants nowadays as well [17]. As a matter of fact, SpyBot is enhanced version of SDBot [17]. Besides the essential command language implementation, it also involves the scanning capability, host control function, and the modules of DDoS attack and flooding attack (e.g., TCP SYN, ICMP, and UDP) [17]. SpyBot's host control capabilities are quite similar to Agobot's in remote command execution, process/system manipulation, key logging, and local file manipulation [17]. Nevertheless, SpyBot still does not have the capability breadth and modularity of Agobot [17].

2.5.4. GT Bot. GT (Global Threat) Bot, as known as Aristotles, is supposed to stand for all mIRC-based bots which have numerous variants and are widely used for Windows [9, 17]. Besides some general capabilities such as IRC host control, DoS attacks, port scanning, and NetBIOS/RPC exploiting, GT Bot also provides a limited set of binaries and scripts of mIRC [9, 17]. One important binary is *HideWindow* program used to keep the mIRC instance invisible from the user [9, 17]. Another function is recording the response to each command received by remote hosts [17]. Some other binaries mainly extend the functions of mIRC via DDL (Dynamic Link Library) [9]. These scripts often store in files with "*.mrc*" extension or in "*mirc.ini*" [9, 17]. Although the binaries are almost all named as "*mIRC.exe*", they may have different capabilities due to distinct configuration files [17]. Compared to the above instances, GT Bot only provides limited commands for host control, just capable of getting local system information and running or deleting local files [17].

3. Botnet Attacks

Botnets can serve both legitimate and illegitimate purposes [6]. One legitimate purpose is to support the operations of IRC channels using administrative privileges on specific individuals. Nevertheless, such goals do not meet the vast number of bots that we have seen. Based on the wealth of data logged in Honeypots [9], the possibilities to use botnets for criminally motivated or for destructive goals can be categorized as follows.

3.1. DDoS Attacks. Botnets are often used for DDoS attacks [9], which can disable the network services of victim system by consuming its bandwidth. For instance, a perpetrator may order the botnet to connect a victim's IRC channel at first, and then this target can be flooded by thousands of service requests from the botnet. In this kind of DDoS attack, the victim IRC network is taken down. Evidence reveals that most commonly implemented by botnets are TCP SYN and UDP flooding attacks [31].

General countermeasure against DDoS attacks requires: (1) controlling a large number of compromised machines; (2) disabling the remote control mechanism [31]. However, more efficient ways are still needed to avoid this kind of attack. Freiling et al. [31] have presented an approach to prevent DDoS attack via exploring the hiding bots in Honeypots.

3.2. Spamming and Spreading Malware. About 70% to 90% of the world's spam is caused by botnets nowadays, which has most experienced in the Internet security industry concerned [32, 33]. Study report indicates that, once the SOCKS v4/v5 proxy (TCP/IP RFC 1928) on compromised hosts is opened by some bots, those machines may be used for nefarious tasks, for example, spamming. Besides, some bots are able to gather email addresses by some particular functions [9]. Therefore, attackers can use such a botnet to send massive amounts of spam [34].

Researchers in [35] have proposed a distributed content independent spam classification system, called Trinity, against spamming from botnets. The designer assumes that the spamming bots will send a mass of e-mails within a short time. Hence, any letter from such address can be a spam. It is a little bit unexpected that we do not know the effectiveness of Trinity since it is still under experiment.

In order to discover the aggregate behaviors of spamming botnet and benefit its detection in the future, Xie et al. [36] have designed a spam signature generation framework named AutoRE. They also found several characteristics of spamming botnet: (1) spammer often appends some random and legitimate URLs into the letter to evade detection [36]; (2) botnet IP addresses are usually distributed over many ASes (Autonomous Systems), with only a few participating machines in each AS on average [36]; (3) despite that the contents of spam are different, their recipients' addresses may be similar [36]. How to use these features to capture the botnets and avoid spamming is worth to research in the future.

Similarly, botnets can be used to spread malware too [9]. For instance, a botnet can launch Witty worm to attack ICQ protocol since the victims' system may have not activated Internet Security Systems (ISS) services [9].

3.3. Information Leakage. Because some bots may sniff not only the traffic passing by the compromised machines but also the command data within the victims, perpetrators can retrieve sensitive information like usernames and passwords from botnets easily [9]. Evidences indicate that, botnets are becoming more sophisticated at quickly scanning in the host for significant corporate and financial data [32]. Since the bots rarely affect the performance of the running infected systems, they are often out of the surveillance area and hard to be caught. Keylogging is the very solution to the inner attack [9, 16]. Such kind of bots listens for keyboard activities and then reports to its master the useful information after filtering the meaningless inputs. This enables the attacker to steal thousands of private information and credential data [16].

3.4. Click Fraud. With the help of botnet, perpetrators are able to install advertisement add-ons and browser helper objects (BHOs) for business purpose [9]. Just like Google's AdSense program, for the sake of obtaining higher click-through rate (CTR), perpetrators may use botnets to periodically click on specific hyperlinks and thus promote the CTR artificially [9]. This is also effective to online polls or games [9]. Because each victim's host owns a unique IP address scattered across the globe, every single click will be regarded as a valid action from a legitimate person.

3.5. Identity Fraud. Identity Fraud, also called as Identity Theft, is a fast growing crime on the Internet [9]. Phishing mail is a typical case. It usually includes legitimate-like URLs and asks the receiver to submit personal or confidential information. Such mails can be generated and sent by botnets through spamming mechanisms [9]. In a further step, botnets also can set up several fake websites pretending to be an official business sites to harvest victims' information. Once a fake site is closed by its owner, another one can pop up, until you shut down the computer.

4. Detection and Tracing

By now, several different approaches of identifying and tracing back botnets have been proposed or attempted. First and the most generally, the use of Honeypots, where a subnet pretends to be compromised by a Trojan, but actually observing the behavior of attackers, enables the controlling hosts to be identified [22]. In a relevant case, Freiling et al. [31] have introduced a feasible way to detect certain types of DDoS attacks lunched by the botnet. To begin with, use honeypot and active responders to collect bot binaries. Then, pretend to join the botnet as a compromised machine by running bots on the honeypot and allowing them to access the IRC server. At the end, the botnet is infiltrated by a "silent drone" for information collecting, which may be useful

in botnet dismantling. Another and also commonly used method is using the information form insiders to track an IRC-based botnet [11]. The third but not the least prevalent approach to detect botnets is probing DNS caches on the network to resolve the IP addresses of the destination servers [11].

4.1. Honeypot and Honeynet. Honeypots are well-known by their strong ability to detect security threats, collect malwares, and to understand the behaviors and motivations of perpetrators. Honeynet, for monitoring a large-scale diverse network, consists of more than one honeypot on a network. Most of researchers focus on Linux-based honeynet, due to the obvious reason that, compared to any other platform, more freely honeynet tools are available on Linux [6]. As a result, only few tools support the honeypots deployment on Windows and intruders start to proactively dismantle the honeypot.

Some scholars aim at the design of a reactive firewall or related means to prevent multiple compromises of honeypots [6]. While a compromised port is detected by such a firewall, the inbound attacks on it can be blocked [6]. This operation should be carried on covertly to avoid raising suspicions of the attacker. Evidence shows that operating less covertly is needed on protection of honeypots against multiple compromises by worms, since worms are used to detect its presence [6]. Because many intruders download toolkits in a victim immediate aftermath, corresponding traffic should be blocked only selectively. Such toolkits are significant evidences for future analysis. Hence, to some extent, attackers' access to honeypots could not be prevented very well [6].

As honeypots have become more and more popular in monitoring and defense systems, intruders begin to seek a way to avoid honeypot traps [37]. There are some feasible techniques to detect honeypots. For instance, to detect VMware or other emulated virtual machines [38, 39], or, to detect the responses of program's faulty in honeypot [40]. In [41], Bethencourt et al. have successfully identified honeypots using intelligent probing according to public report statistics. In addition, Krawetz [42] have presented a commercial spamming tool capable of anti-honeypot function, called "Send-Safe's Honeypot Hunter." By checking the reply form remote proxy, spammer is able to detect honeypot open proxies [42]. However, this tool cannot effectively detect others except open proxy honeypot. Recently, Zou and Cunninqham [37] have proposed another methodology for honeypot detection based on independent software and hardware. In their paper, they also have introduced an approach to effectively locate and remove infected honeypots using a P2P structured botnet [37]. All of the above evidences indicate that, future research is needed in case that a botnet becomes invisible to honeypot.

4.2. IRC-based Detection. IRC-based botnet is wildly studied and therefore several characteristics have been discovered for detection so far. One of the easy ways to detect this kind of botnets is to sniff traffic on common IRC ports (TCP port 6667), and then check whether the payloads march the strings in the knowledge database [22]. Nevertheless, botnets can use random ports to communicate. Therefore, another approach looking for behavioral characteristics of bots comes up. Racine [43] found IRC-based bots were often idle and only responded upon receiving a specific instruction. Thus, the connections with such features can be marked as potential enemies. Nevertheless, it still has a high false positive rate in the result.

There are also other methodologies existing for IRC-based botnet detection. Barford and Yegneswaran [17] proposed some approaches based on the source code analysis. Rajab et al. [11] introduced a modified IRC client called IRC tracker, which was able to connect the IRC sever and reply the queries automatically. Given a template and relevant fingerprint, the IRC tracker could instantiate a new IRC session to the IRC server [11]. In case the bot master could find the real identity of the tracker, it appeared as a powerful and responsive bot on the Internet and run every malicious command, including the responses to the attacker [11]. We will introduce some detection methods against IRC-based botnets below.

4.2.1. Detection Based on Traffic Analysis. Signature technology is often used in anomaly detection. The basic idea is to extract feature information on the packets from the traffic and march the patterns registered in the knowledge base of existing bots. Apparently, it is easy to carry on by simply comparing every byte in the packet, but it also goes with several drawbacks [44]. Firstly, it is unable to identify the undefined bots [44]. Second, it should always update the knowledge base with new signatures, which enhances the management cost and reduces the performance [44]. Third, new bots may launch attacks before the knowledge base are patched [44].

Based on the features of IRC, some other techniques to detect botnets come up. Basically, two kinds of actions are involved in a normal IRC communication. One is interactive commands and another is messages exchanging [44]. If we can identify the IRC operation with a specified program, it is possible to detect a botnet attack [44]. For instance, if the private information is copied to other places by some IRC commands, we claim that the system is under an attack since a normal chatting behavior will never do that [44]. However, the shortcomings also exist. On the one hand, IRC port number may be changed by attackers. On the other hand, the traffic may be encrypted or be concealed by network noises [21]. Any situation will make the bots invisible.

In [44], authors observed the real traffic on IRC communication ports ranging from 6666 to 6669. They found some IRC clients repeated sending login information while the server refused their connections [44]. Based on the experiment result, they claimed that bots would repeat these actions at certain intervals after refused by the IRC server, and those time intervals are different [44]. However, they did not consider a real IRC-based botnet attack into their experiment. It is a possible future work to extend their achievements.

In [33], Sroufe et al. proposed a different method for botnet detection. Their approach can efficiently and automatically identify spam or bots. The main idea is to extract the shape of the Email (lines and the character count of each line) by applying a Gaussian kernel density estimator [33]. Emails with similar shape are suspected. However, authors did not show the way to detect botnet by using this method. It may be another future work worth to study.

4.2.2. Detection Based on Anomaly Activities. In [21], authors proposed an algorithm for anomaly-based botnet detection. It combined IRC mesh features with TCP-based anomaly detection module. It first observed and recorded a large number of TCP packets with respect to IRC hosts. Based on the ratio computed by the total amount of TCP control packets (e.g., SYN, SYNACK, FIN, and RESETS) over total number of TCP packets, it is able to detect some anomaly activities [21]. They called this ratio as the TCP work weight and claimed that high value implied a potential attack by a scanner or worm [21]. However, this mechanism may not work if the IRC commands have been encoded, as discussed in [21].

4.3. DNS Tracking. Since bots usually send DNS queries in order to access the C2 servers, if we can intercept their domain names, the botnet traffic is able to be captured by blacklisting the domain names [45, 46]. Actually, it also provides an important secondary avenue to take down botnets by disabling their propagation capability [11].

Choi et al. [45] have discussed the features of botnet DNS. According to their analysis, botnets' DNS queries can be easily distinguished from legitimate ones [45]. First of all, only bots will send DNS queries to the domain of C2 servers, a legitimate one never do this [45]. Secondly, botnet's members act and migrate together simultaneously, as well as their DNS queries [45]. Whereas the legitimate one occurs continuously, varying from botnet [45]. Third, legitimate hosts will not use DDNS very often while botnet usually use DDNS for C2 servers [45]. Based on the above features, they developed an algorithm to identify botnet DNS queries [45]. The main idea is to compute the similarity for group activities and then distinguish the botnet from them based on the similarity value. The similarity value is defined as 0.5 (C/A+C/B), where A and B stand for the sizes of two requested IP lists which have some common IP addresses and the same domain name, and C stands for the size of duplicated IP addresses [45]. If the value approximated zero, such common domain will be suspected [45].

There are also some other approaches. Dagon [46] presented a method of examining the query rates of DDNS domain. Abnormally high rates or temporally concentrated were suspected, since the attackers changed their C2 servers quite often [47]. They utilized both Mahalanobis distance and Chebyshev's inequality to quantify how anomalous the rate is [47]. Schonewille and van Helmond [48] found that when C2 servers had been taken down, DDNS would often response name error. Hosts who repeatedly did such queries could be infected and thus to be suspected [48]. In [47], authors evaluated the above two methods through experiments on the real world. They claimed that, Dagon's approach was not as effective since it misclassified some C2 server domains with short TTL, while Schonewille's method was comparatively effective due to the fact that the suspicious name came from independent individuals [47].

In [49], Hu et al. proposed a botnet detection system called RB-Seeker (Redirection Botnet Seeker). It is able to automatically detect botnets in any structure. RB-Seeker first gathers information about bots redirection activities (e.g., temporal and spatial features) from two subsystems. Then it utilizes the statistical methodology and DNS query probing technique to distinguish the malicious domain from legitimate ones. Experiment results show that RB-Seeker is an efficient tool to detect both "aggressive" and "stealthy" botnets.

5. Preventive Measures

It takes only a couple of hours for conventional worms to circle the globe since its release from a single host. If worms using botnet appear from multiple hosts simultaneously, they are able to infect the majority of vulnerable hosts worldwide in minutes [7]. Some botnets have been discussed in previous sections. Nevertheless, there are still plenty of them that are unknown to us. We also discuss a topic of how to minimize the risk caused by botnets in the future in this section.

5.1. Countermeasures on Botnet Attacks. Unfortunately, few solutions have been in existence for a host to against a botnet DoS attack so far [3]. Albeit it is hard to find the patterns of malicious hosts, network administrators can still identify botnet attacks based on passive operating system fingerprinting extracted from the latest firewall equipment [3]. The lifecycle of botnets tells us that bots often utilize free DNS hosting services to redirect a sub-domain to an inaccessible IP address. Thus, removing those services may take down such a botnet [3]. At present, many security companies focus on offerings to stop botnets [3]. Some of them protect consumers, whereas most others are designed for ISPs or enterprises [3]. The individual products try to identify bot behavior by anti-virus software. The enterprise products have no better solutions than nullrouting DNS entries or shutting down the IRC and other main servers after a botnet attack identified [3].

5.2. Countermeasures for Public. Personal or corporation security inevitably depends on the communication partners [7]. Building a good relationship with those partners is essential. Firstly, one should continuously request the service supplier for security packages, such as firewall, anti-virus tool-kit, intrusion detection utility, and so forth. [7]. Once something goes wrong, there should be a corresponding contact number to call [7]. Secondly, one should also pay much attention on network traffic and report it to ISP if there is a DDoS attack. ISP can help blocking those malicious IP addresses [7]. Thirdly, it is better to establish

Table 2: Rules of prevention by home users [18].

Type	Strategies
Personal Habits	Attention while downloading
	Avoid to install useless things
	Read carefully before you click
Routine	Use anti-virus/trojan utilities
	Update system frequently
	Shutdown PC when you leave
Optional Operations	Back-up all systems regularly
	Keep all software up-to-date
	Deploy personal firewall

accountability on its system, together with a law enforcement authority [7]. More specifically, scholars and industries have proposed some strategies for both home users and system administrators, to prevent, detect and respond botnet attacks [16, 18]. Here we summarize their suggestions.

5.2.1. Home Users. To prevent attacks from a botnet, home users can follow the rules described in Table 2. They are classified into three categories: (1) Personal Habits, (2) Routine, and (3) Optional Operations. As personal habits, people should pay attention when downloading, especially for those programs coming from unscrupulous sites. Besides, try to avoid installing useless things on personal computer, which will minimize the possibility of bots infection. If necessary, read the License Agreement and the notes carefully before click the button on the web site. As a routine, use anti-virus software and anti-trojan utilities while system is on. Scan and update system regularly, especially for Windows. When leaving the PC, shutdown the system or it may be remotely controlled by hackers. As the optional operations, home users are recommended to backup system regularly, to keep all software up-to-date and to deploy personal firewall by all means. By doing so, home PCs are shielded from unauthorized accesses, and thus bots cannot compromise them.

To detect an abnormal behavior, taking Windows operating system as an instance, a home user can check the IRC port range from 6000 to 7000 (typically 6667) by command "C:\Windows\netstat-an" [16, 18]. The result can reveal the connection of current IRC client. However, bots may use some other TCP ports [18]. If unusual behavior occurs on a home PC, such as slow network response, unknown ports being used, and something like that, there is possibly a bot attack [16, 18]. Also, home users can use anti-virus software or online services to detect attacks [16, 18]. Once the computer has been compromised, there are strategies to recover it. The following procedure (Figure 5) is a good example for home users.

5.2.2. System Administrator. Similarly, there are corresponding rules for system administrators to prevent, detect, and respond botnet attacks [16, 18]. For a prevention method, administrators should follow vendor guidelines for updating the system and applications [18]. Also, keep informed of latest vulnerabilities and use access control and log files to achieve accountability [18]. As illustrated in Table 3, the following measures can help the system administrator to minimize the possibilities of botnet attacking.

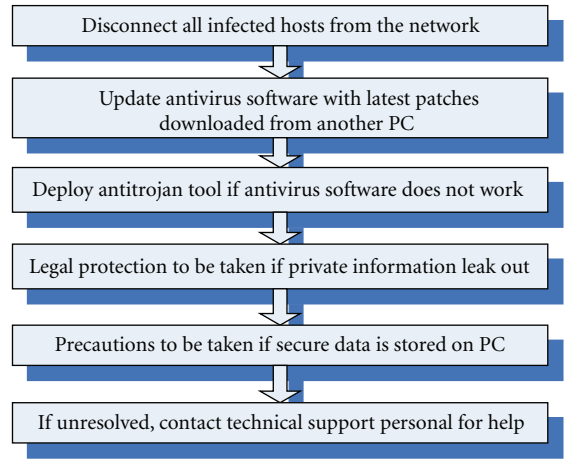


Figure 5: Home users' response to botnet attacks [18].

Table 3: Rules of detection by system administrators [18].

Rules	Notes
Monitor logs regularly	Analyze the internet traffic for anomalies
Use network packet sniffer	Identify the malicious traffic in intranet
Isolate the malicious subnet	Verify IRC activity on host
Scan individual machine	They may contain malware

Once an attack is detected, a system administrator should isolate those compromised hosts and notify the home users [16]. Then preserve the data on those infected hosts including the log files [16]. Besides, identify the number of victims via sniffer tools [16]. Finally, report the infection to security consultant [16].

6. Conclusion and Future Challenges

To better understand the botnet and stop its attack eventually, we provide a survey on existing research on botnets. The survey first discussed botnet formation and exploitation, the lifecycle, and two typical topologies. Aiming at the IRC-based and P2P-based botnet, we give a tutorial of current research on their attacks and countermeasures.

According to the discussion in Section 2, we propose several ideas on different topologies as follows. For IRC-based botnets, the thorny problem is that we cannot get the source code of most of bots. Hence, in-depth analysis at networking level and system level for bots' behaviors are hardly carried out. For P2P-based botnets, the following practical challenges should be further considered: (1) maintaining the rest of bots after some have been taken down by defenders; (2) hiding the botnet topology while some bots are captured by defenders; (3) managing the botnet more easily; (4) changing the traffic patterns more often and making it harder for detection.

Detecting and tracking compromised hosts in a botnet will continue to be a challenging task. Traffic fingerprinting is useful for identifying botnets. Nevertheless, just like previous signature technologies discussed in Section 3, its drawbacks are obvious. We need an up-to-date knowledge base for all released bots in the world, which seems to be an impossible mission. Anomaly detection is another feasible approach. However, when infected hosts do not behave as unusual, it may be unable to detect such a potential threat. Since current detecting technology depends on the happened attacking event, no guarantee for us to find every possible compromised hosts. One interesting issue about anomaly detection is the time efficiency. If an attack occurs and we can capture the anomaly in the first place and fix the relevant problems before it is used for malicious purposes, we say this anomaly detection is time efficient. We need to focus on its time efficiency in the future work.

In wireless context, especially for an ad hoc network, there is not much related research conducted on either attacking or defending. There are lots of open issues: (1) How to find the shortest route to attack a target; (2) How to prevent the compromised hosts from being detected in the wireless network; (3) How to propagate the bots in the wireless network, especially before some compromised hosts become off line.

There are also some other interesting open issues that need to be considered. To the best of our knowledge, DDoS attack derived from botnets cannot be avoided. Even if the attacking has been detected, there is no effective way to trace back or fight against it. Instead, one can only shut down the compromised hosts or disconnect with the network, waiting for further command such as scanning virus or reinstalling the operating system. As a matter of fact, what we need indeed is to avoid the propagation of bots in the first place. Perhaps the only effective approach to eliminate botnets is to deploy new protocols on routers worldwide. It is really a huge and beyond reality project. Then, why not consider installing them on a local gateway? If the gateway could block the communication of bots between several domains, the attacker could not easily manage the compromised hosts worldwide. In the meantime, the gateway could give us information about where the malicious command came from. Based on the available evidence over the network, it would be possible to trace back the initial attack source. Nevertheless, it is very difficult to implement such an idea due to the following reasons: (1) It is hard to distinguish the malicious packages from the regular traffic flow; (2) Cooperating among domains is not very easy, and sometimes even gateways can be compromised; (3) How to trace the potential attack and who should be noticed for further analysis need to be studied.

Acknowledgment

We would like to thank the editor and anonymous reviewers for their useful comments on earlier versions of this paper. This work was supported in part by the National Science Foundation (NSF) under grants CNS-0716211, CNS-0737325, and CCF-0829827.

References

[1] K. Ono, I. Kawaishi, and T. Kamon, "Trend of botnet activities," in *Proceedings of the 41st Annual IEEE Carnahan Conference on Security Technology (ICCST '07)*, pp. 243–249, Ottawa, Canada, October 2007.

[2] Wikipedia, "Internet bot," http://en.wikipedia.org/wiki/Internet_bot.

[3] Wikipedia, "Botnet," http://en.wikipedia.org/wiki/Botnet.

[4] B. Thuraisingham, "Data mining for security applications: mining concept-drifting data streams to detect peer to peer botnet traffic," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI '08)*, Taipei, Taiwan, June 2008.

[5] C. Mazzariello, "IRC traffic analysis for botnet detection," in *Proceedings of the 4th International Symposium on Information Assurance and Security (IAS '08)*, pp. 318–323, Napoli, Italy, September 2008.

[6] B. McCarty, "Botnets: big and bigger," *IEEE Security and Privacy*, vol. 1, no. 4, pp. 87–90, 2003.

[7] G. P. Schaffer, "Worms and viruses and botnets, oh my!: rational responses to emerging internet threats," *IEEE Security and Privacy*, vol. 4, no. 3, pp. 52–58, 2006.

[8] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP '02)*, pp. 312–321, Paris, France, November 2002.

[9] P. Bacher, T. Holz, M. Kotter, and G. Wicherski, "Know your Enemy: Tracking Botnets," http://www.honeynet.org/papers/bots.

[10] T. Holz, S. Marechal, and F. Raynal, "New threats and attacks on the world wide web," *IEEE Security and Privacy*, vol. 4, no. 2, pp. 72–75, 2006.

[11] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the 6th ACM SIGCOMM Internet Measurement Conference (IMC '06)*, pp. 41–52, Rio de Janeriro, Brazil, October 2006.

[12] E. Levy, "The making of a spam zombie army: dissecting the sobig worms," *IEEE Security and Privacy*, vol. 1, no. 4, pp. 58–59, 2003.

[13] D. Cook, J. Hartnett, K. Manderson, and J. Scanlan, "Catching spam before it arrives: domain specific dynamic blacklists," in *Proceedings of the Australasian Workshops on Grid Computing and E-Research*, pp. 193–202, Hobart, Australia, January 2006.

[14] J. Jung and E. Sit, "An empirical study of spam traffic and the use of DNS black lists," in *Proceedings of the 4th ACM SIGCOMM Internet Measurement Conference (IMC '04)*, pp. 370–378, Taormina, Italy, October 2004.

[15] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing botnet membership using DNSBL counter-intelligence," in *Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '06)*, vol. 2, p. 8, San Jose, Calif, USA, 2006.

[16] J. Govil, "Examining the criminology of bot zoo," in *Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS '07)*, pp. 1–6, Singapore, December 2007.

[17] P. Barford and V. Yegneswaran, "An inside look at botnets," in *Proceedings of the ARO-DHS Special Workshop on Malware Detection*, Advances in Information Security, Springer, 2006.

[18] R. Puri, "Bots and botnets: an overview," Tech. Rep., SANS Institute, 2003.

[19] W. T. Strayer, R. Walsh, C. Livadas, and D. Lapsley, "Detecting botnets with tight command and control," in *Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06)*, pp. 195–202, Tampa, Fla, USA, November 2006.

[20] M. Akiyama, T. Kawamoto, M. Shimamura, T. Yokoyama, Y. Kadobayashi, and S. Yamaguchi, "A proposal of metrics for botnet detection based on its cooperative behavior," in *Proceedings of the International Symposium on Applications and the Internet Workshops*, p. 82, Washington, DC, USA, January 2007.

[21] J. R. Binkley and S. Singh, "An algorithm for anomaly-based botnet detection," in *Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '06)*, p. 7, San Jose, Calif, USA, 2006.

[22] E. Cooke, F. Jahanian, and D. Mcpherson, "The zombie roundup: understanding, detecting, and disrupting botnets," in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '05)*, p. 6, Cambridge, Mass, USA, 2005.

[23] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using machine learning techniques to identify botnet traffic," in *Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06)*, pp. 967–974, Tampa, Fla, USA, November 2006.

[24] T. Holz, M. Steiner, F. Dahl, E. W. Biersack, and F. Freiling, "Measurement and mitigation of peer-to-peer-based botnets: a case study on storm worm," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pp. 1–9, San Francisco, Calif, USA, April 2008.

[25] P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," in *Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets*, p. 2, Cambridge, Mass, USA, July 2008.

[26] R. Lemos, "Bot software looks to improve peerage," http://www.securityfocus.com/news/11390.

[27] I. Arce and E. levy, "An analysis of the slapper worm," *IEEE Security & Privacy Magazine*, vol. 1, no. 1, pp. 82–87, 2003.

[28] J. Stewart, "Sinit P2P Trojan analysis," http://www.secureworks.com/research/threats/sinit/.

[29] J. Stewart, "Phatbot Trojan analysis," http://www.secureworks.com/research/threats/phatbot/?threat=phatbot.

[30] C. Langin, H. Zhou, and S. Rahimi, "A model to use denied Internet traffic to indirectly discover internal network security problems," in *Proceedings of the IEEE International Performance, Computing, and Communications Conference (IPCCC '08)*, pp. 486–490, Austin, Tex, USA, December 2008.

[31] F. C. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks," in *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS '05)*, vol. 3679 of *Lecture Notes in Computer Science*, pp. 319–335, Springer, Milan, Italy, September 2005.

[32] K. Pappas, "Back to basics to fight botnets," *Communications News*, vol. 45, no. 5, p. 12, 2008.

[33] P. Sroufe, S. Phithakkitnukoon, R. Dantu, and J. Cangussu, "Email shape analysis for spam botnet detection," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC '09)*, pp. 1–2, Las Vegas, Nev, USA, January 2009.

[34] K. Chiang and L. Lloyd, "A case study of the restock rootkit and spam bot," in *Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets*, p. 10, Cambridge, Mass, USA, 2007.

[35] A. Brodsky and D. Brodsky, "A distributed content independent method for spam detection," in *Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets*, p. 3, Cambridge, Mass, USA, 2007.

[36] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming botnets: signatures and characteristics," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '08)*, vol. 38, pp. 171–182, Seattle, Wash, USA, August 2008.

[37] C. C. Zou and R. Cunningham, "Honeypot-aware advanced botnet construction and maintenance," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN '06)*, pp. 199–208, Philadelphia, Pa, USA, June 2006.

[38] J. Corey, "Advanced honey pot identification and exploitation," 2004, http://www.ouah.org/p63-0x09.txt.

[39] K. Seifried, "Honeypotting with VMware basics," 2002, http://www.seifried.org/security/index.html.

[40] Honeyd security advisory 2004–001, "Remote detection via simple probe packet," 2004, http://www.honeyd.org/adv.2004-01.asc.

[41] J. Bethencourt, J. Franklin, and M. Vernon, "Mapping internet sensors with probe response attacks," in *Proceedings of the 14th Conference on USENIX Security Symposium*, pp. 193–208, Baltimore, Md, USA, August 2005.

[42] N. Krawetz, "Anti-Honeypot technology," *IEEE Security and Privacy*, vol. 2, no. 1, pp. 76–79, 2004.

[43] S. Racine, *Analysis of internet relay chat usage by DDoS zombies*, M.S. thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, April 2004.

[44] Y. Kugisaki, Y. Kasahara, Y. Hori, and K. Sakurai, "Bot detection based on traffic analysis," in *Proceedings of the International Conference on Intelligent Pervasive Computing (IPC '07)*, pp. 303–306, Jeju Island, South Korea, October 2007.

[45] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in DNS traffic," in *Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT '07)*, pp. 715–720, Fukushima, Japan, October 2007.

[46] D. Dagon, "Botnet detection and response, the network is the infection," 2005, http://www.caida.org/workshops/dns-oarc/200507/slides/oarc0507-Dagon.pdf.

[47] R. Villamarin-Salomon and J. C. Brustoloni, "Identifying botnets using anomaly detection techniques applied to DNS traffic," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, pp. 476–481, Las Vegas, Nev, USA, January 2008.

[48] A. Schonewille and D. J. van Helmond, *The domain name service as an IDS*, M.S. thesis, University of Amsterdam, Amsterdam, The Netherlands, February 2006.

[49] X. Hu, M. Knyz, and K. G. Shin, "RB-Seeker: auto-detection of redirection botnets," in *Proceedings of 16th Annual Network & Distributed System Security Symposium (NDSS '09)*, February 2009.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 806563, 16 pages
doi:10.1155/2009/806563

Research Article

Pre-Authentication Schemes for UMTS-WLAN Interworking

Ali Al Shidhani and Victor C. M. Leung

Department of Electrical and Computer Engineering, University of British Columbia, 2332 Main Mall, Vancouver, BC, Canada V6T 1Z4

Correspondence should be addressed to Ali Al Shidhani, alia@ece.ubc.ca

Received 31 January 2009; Accepted 30 April 2009

Recommended by Yang Xiao

Interworking Universal Mobile Telecommunication System (UMTS) and IEEE 802.11 Wireless Local Area Networks (WLANs) introduce new challenges including the design of secured and fast handover protocols. Handover operations within and between networks must not compromise the security of the networks involved. In addition, handovers must be instantaneous to sustain the quality of service (QoS) of the applications running on the User Equipment (UE). There is a need to design fast and secured handover protocols to operate in UMTS-WLAN interworking architectures. This paper proposes two secured pre-authentication protocols in the UMTS-WLAN interworking architectures. Performance analysis of the proposed protocols show superior results in comparison to existing protocols in terms of authentication signaling cost, authentication delay and load on critical nodes involved in the authentication procedure. Additionally, the security of the proposed protocols was verified by the Automated Validation of Internet Security Protocols and Applications (AVISPA) security analyzer.

1. Introduction

UMTS-WLAN interworking is being widely considered by cellular service providers because of its advantages for both end users and service providers. The 3rd Generation Partnership Project (3GPP) has recently published specifications detailing suggested UMTS-WLAN interworking architecture [1]. A simplified architecture following a nonroaming reference model [1] is shown in Figure 1. Interworking UMTS and WLAN introduces new handover and security challenges. Handovers in general are classified into horizontal and vertical handovers [2]. Horizontal Handovers (HH) occur when roaming within a network employing the same wireless technology while Vertical Handovers (VH) occur when roaming between networks employing different wireless technologies. Handovers are further subdivided into link-layer (L2) handovers and Internet Protocol (IP)-layer (L3) handovers [2].

Link-layer handover handles association and authentication of the WLAN User Equipment (UE) to a target attachment point. IP-layer handover is generally based on Mobile IP (MIP) functionalities and aims to register a new UE IP address in the visited network. This paper discusses the authentication operation during link-layer HH within WLANs when operating in a UMTS-WLAN interworking architecture. In such architecture, the UE must be initially authenticated by servers in the UMTS Home Network (UHN) such as the Home Location Register (HLR), Home Subscriber Server (HSS), and Home Authentication, Authorization, and Accounting (HAAA) server [3].

Several UMTS-WLAN authentication schemes have been proposed in the literature. Kambourakis et al. [4], Prasithsangaree and Krishnamurthy [5], and Chen et al. [6] proposed using Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) [7, 8], EAP-Tunneled TLS (EAP-TTLS) [9], and Protected EAP (PEAP) [10], respectively, to authenticate a UE in the UMTS-WLAN interworking architecture. These authentication protocols are based on public key cryptography and require digital certificate management to operate properly.

3GPP recommends invoking EAP with Authentication and Key Agreement (EAP-AKA) to authenticate a UE in the UMTS-WLAN interworking architecture [3, 11]. EAP-AKA relies on pre-shared secrets held by the UE and HSS and does not require public key cryptography or digital certificate management. In EAP-AKA, the UE, and the

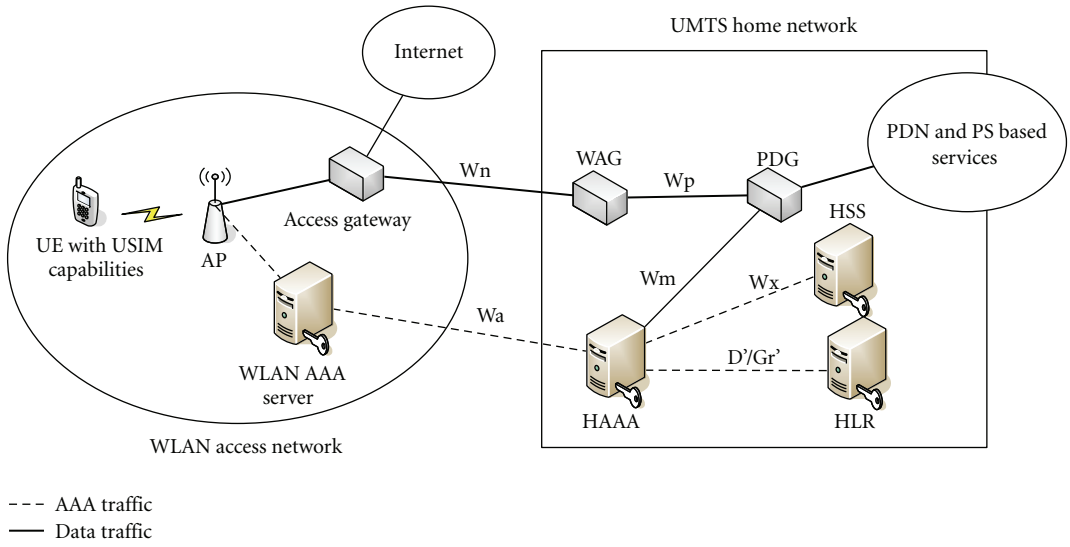


Figure 1: Simplified UMTS-WLAN interworking architecture.

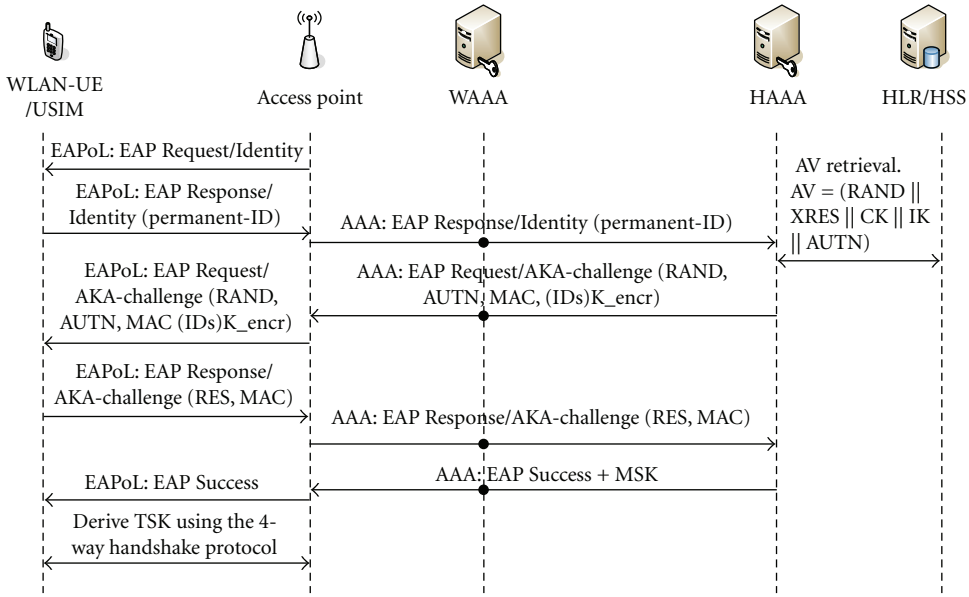


Figure 2: EAP-AKA authentication protocol.

HAAA exchange series of EAP messages to request/respond authentication information. HAAA communicates with the HSS to obtain Authentication Vectors (AVs) as shown in Figure 2. The UMTS Subscriber Identity Module (USIM) application on the UE and the HSS execute special message authentication and key generation functions known as "f1–f5" functions [12] to generate AVs. On successful mutual authentication, the UE and HAAA derive important security keys like the Master Session Key (MSK), Extended MSK (EMSK), and Transient EAP Key (TEK) [11]. The integrity and confidentiality of EAP messages are protected by K_auth and K_encr keys derived from TEK. The UE and its associated WLAN Access Point (AP) use MSK to derive a new session key, the Transient Session Key (TSK), which is later used to secure communications between them. TSK is derived from MSK using the 4-way handshake protocol introduced in IEEE 802.11i [13]. Due to the need to retrieve authentication information from authentication servers in the UHN, EAP-AKA authentication protocol is prone to high authentication delays and introduces redundant signaling traffic between the WLAN network and the UHN.

Generally, handover delay caused by roaming between and within WLANs is composed of delays like AP scanning delays [14], authentication delays, and MIP registration delays in the case of L3 handover. Several proposals reported in the literature focused on minimizing authentication delays during HH in autonomous WLAN networks [15–20]. However, the problem of reducing authentication

delays during HH when the UE operates in UMTS-WLAN interworking architecture remains mostly unexplored. In such architecture, authentication delay largely contributes to the overall handover delay because the UE needs to communicate with the UHN to successfully complete the authentication procedure. In practice, the UHN could be far away from the UE and separated by multiple networks and proxy AAA servers, resulting in high authentication and handover delays. Due to these reasons, invoking EAP-AKA protocol whenever WLAN HH takes place in UMTS-WLAN interworking architecture is unfavorable.

In our preliminary work, we have proposed two protocols to reduce authentication delays during WLAN HH in UMTS-WLAN interworking architecture. The proposed protocols were immature and initial and limited performance and security discussion were presented [21]. In this paper, we present improvements to the protocols and conduct extensive and thorough performance and security analysis on them. The comprehensive performance analysis considers important metrics like authentication signaling cost, authentication delay, and resource optimization of critical nodes involved in the authentication procedure. The thorough security analysis employs widely-accepted formal security verification tools to confirm that our protocols can withstand all forms of authentication and key secrecy attacks. In comparison with EAP-AKA protocol, our protocols achieve outstanding performance while preserving adequate security. The rest of this paper is organized as follows. In Section 2 we report some related works. In Section 3 we give detailed descriptions of our proposed protocols. In Section 4 we evaluate the performance of our protocols. In Section 5 we analyze the security of our proposed protocols. In Section 6 we present some conclusions.

2. Related Work

Research to reduce authentication delay during HH in WLANs in the context of UMTS-WLAN interworking architecture is in its initial stages. 3GPP did not specify protocols specific to UMTS-WLAN interworking to support WLAN HH. Thus, EAP-AKA protocol is invoked whenever HH takes place. On the other hand, many research studies are focusing on WLAN HH in autonomous WLANs architecture. In terms of network architecture, a major difference between authenticating a roaming UE in autonomous WLANs architecture in contrast to UMTS-WLAN interworking architecture is that authentication servers reside in the WLAN network in the former case and they reside in the UHN in the latter case. Another difference is that IEEE recommends invoking EAP-TLS protocols in autonomous WLANs, while 3GPP recommends invoking EAP-AKA authentication protocols in UMTS-WLAN interworking architecture. Therefore, existing HH authentication protocols designed specifically for autonomous WLANs architecture are not directly applicable over the UMTS-WLAN interworking architecture. Besides, several HH authentication protocols proposed for WLANs attain reduction in authentication delay at the cost of operational and security problems like introducing extra signaling overhead in the WLAN network [15–17] or demonstrating high dependency on UE mobility patterns [18, 19].

The rudimentary handover and security support in the base IEEE 802.11 protocol [22] has been enhanced in IEEE802.11i [13], IEEE802.11f [23], and IEEE802.11r [24]. Handover protocols in IEEE802.11i are optional and have seen limited implementation and deployment support [25]. Handover protocols in IEEE802.11f are not suitable for UMTS-WLAN interworking environments because strong trust agreements are required between WLAN administration domains for secure inter-Extended Service Set (inter-ESS) HH across these WLAN domains. On the other hand, IEEE802.11r supports only intra-ESS HH within specific WLAN domain but not inter-ESS HH.

Many papers in the literature proposed mechanisms to reduce intra- or inter-ESS HH delays in autonomous WLAN architecture. Some papers achieved this goal by preauthenticating the UE before handover, predistributing security keys, predicting UE's next move, introducing public key cryptography, or adopting hybrid techniques combining more than one method. Mishra et al. [15], Kassab et al. [16], and Hur et al. [17] proposed proactive key distribution using neighbor graphs to predict potential Target AP (TAP). These schemes utilize EAP-TLS and may result in unnecessary distribution of keys and increase signaling overhead in the WLAN as the number of UEs increases. Pack and Choi [18] and Mukherjee et al. [19] proposed mechanisms to predict UE mobility and hence preauthenticating the UE with the TAP before handover. The protocols share similar drawbacks as in [15–17] and their operations are restricted to intra-ESS HH. In the context of UMTS-WLAN interworking architecture, the UE roams between WLANs belonging to different administration and security domains, which imply that protocols designed to work in autonomous WLAN architectures like in [15–19] cannot be simply migrated to operate in the UMTS-WLAN interworking architecture.

Techniques to reduce delays in the event of WLAN HH in UMTS-WLAN interworking architecture have been proposed in [20, 26, 27]. Long et al. [20] proposed localized UE authentication for inter-ESS HH, in an architecture similar in concept to the UMTS-WLAN interworking architecture. The proposed mechanism requires that the UE should be authenticated by its home network while roaming. This protocol achieves fast inter-ESS HH by means of public key cryptography. Lee et al. [26] proposed a location-aware handover protocol. Location-aware service brokers are introduced in the interworking architecture to predict UE movement and perform fast authentication during handover. This scheme aims at offloading the 3G AAA servers from handling authentication whenever the UE moves, thus reducing authentication and handover delays. The drawback of this approach is that it requires major modifications to the existing 3G-WLAN interworking architecture. Lim et al. [27] proposed a protocol to reduce probing/scanning delays of the target AP. The downside to this solution is that APs must perform some of the functionalities of UMTS base station and share some control channels with it.

In comparison with protocols in [4–6, 15–20, 26, 27], our proposed protocols enjoy unique characteristics which make them first in their kind. Firstly, they are designed to operate in the 3GPP-specified UMTS-WLAN interworking architecture and adopt a variation of EAP-AKA protocols according to 3GPP recommendations unlike [4–6, 15–17]. Secondly, they are independent of UE movement pattern or TAP predictions contrasting protocols in [18, 19, 26]. Thirdly, they do not rely on public key cryptography like protocols in [4–6, 15–17, 20], which might require substantial processing resources that may not be available in mobile UEs. Fourthly, they do not require major modifications to APs or the introduction of new servers in the UMTS-WLAN interworking architecture as the case in [26, 27]. Finally they avoid unnecessary generation and pre-distribution of keys to TAPs and are therefore more efficient and secure.

3. Proposed Protocols

Novel pre-authentication protocols are proposed to improve intra- and inter-ESS WLAN HH when operating in a UMTS-WLAN interworking architecture. Intra- and Inter-WLAN ESS Fast Pre-authentication protocols (Intra/Inter-WLAN FP) preauthenticate the UE locally before handover takes place which results in reduction in the handover delay. To realize our proposed protocols, simple modifications are required to the standard EAP-AKA authentication protocol.

3.1. Assumptions. Firstly, some general assumptions are outlined which are similar in part to the assumptions made by 3GPP for authenticating a UE in UMTS-WLAN Interworking architecture [3].

(i) A WLAN AAA (WAAA) server exists in every WLAN. WAAA controls multiple APs forming a "WLAN domain." The WAAA and all APs in its domain must share a Long Term Security Association (LTSA).

(ii) WAAAs belonging to different WLAN domains must have LTSA and roaming agreements with the HAAA in the UHN.

(iii) WAAA and UE must maintain a WLAN counter (WC) which indicates the number of times pre-authentications has been performed. They are incremented by both corresponding nodes after every successful pre-authentication.

(iv) The HAAA or WAAA must supply a new UE local identity to the UE during authentication session to be used in future pre-authentications.

3.2. Modifications to EAP-AKA Protocol. In the standard EAP-AKA protocol, the UE and the HAAA must generate MSK and EMSK after a successful authentication [3, 11]. MSK is transported to the AP to be used in generating a TSK. EMSK is generated but its usage is not yet specified. We propose using EMSK to derive additional keys to achieve faster pre-authentication without compromising security. We extended the key hierarchy in EAP-AKA protocol by introducing WLAN domain-level and local-level keys derived from MSK and EMSK. Domain-level keys are unique keys derived by the HAAA and the UE per WLAN domain. Local-level keys are unique keys derived by the WAAA and the UE per AP within the WLAN domain. The local-level keys are later used to derive TSKs.

MSK is used to derive additional keys to speed UE's reauthentication operations only, that is, without handover. Usage of MSK to speed reauthentication operation in UMTS-WLAN interworking is described in [28]. We propose using EMSK as the root key for handover pre-authentications. The keys derived from EMSK are the Handover Root Key (HOK), the Domain-level Handover key (DHOK) and the Local-level handover key (LHOK). LHOK is ultimately used to derive TSK in Intra- and Inter-WLAN FP. To derive the required additional keys we suggest the following modifications to EAP-AKA authentication protocol as depicted in Figure 3.

(i) The HAAA generates the next local ID, $\mathrm{ID}_{\mathrm{WLAN}}$, to be used by the UE in the next pre-authentication and a nonce value (HN). The HAAA should indicates the permitted number of pre-authentications (n_{pre}) the UE can perform before falling back to standard EAP-AKA authentication. The WAAA and UE adjust the maximum value WC can reach according to n_{pre}. In addition, the UE generates a nonce, UN.

(ii) Five new keys are generated.

(a) Root handover key, HOK. This key is derived from EMSK by the HAAA and the UE only. Both nodes use a special Pseudorandom Function (PRF) similar to the one used in generating MSK in the standard EAP-AKA protocol [11]

$$\mathrm{HOK} = \mathrm{PRF}(\mathrm{EMSK}, \text{EAP-AKA session ID} \mid \text{HAAA ID} \mid \mathrm{UEM}, 256), \tag{1}$$

where "|" denotes concatenation and,

$$\text{EAP-AKA session ID} = (\text{EAP Type Code}|\mathrm{RAND}|\mathrm{AUTN}) \tag{2}$$

see, [29].
UEM is the UE address in the medium access control layer. HAAA ID is the identity of the HAAA server.

(b) The domain-level handover key, DHOK. It is derived from HOK by HAAA and UE only

$$\mathrm{DHOK} = \mathrm{PRF}(\mathrm{HOK}, \mathrm{HN} \mid \text{WAAA ID} \mid \mathrm{UEM}, 256), \tag{3}$$

where WAAA ID is the identity of the WAAA.

(c) The domain-level and local-level reauthentication keys, DRK and LRK. Their derivation and usage are detailed in [28].

(d) A key used to secure traffic between the UE and WAAA, $\mathrm{K}_{\mathrm{WAAA\text{-}UE}}$. This key is only derived by the UE and WAAA

$$\mathrm{K}_{\mathrm{WAAA\text{-}UE}} = \mathrm{PRF}(\mathrm{DHOK} \oplus \mathrm{DRK} \mid \text{WAAA ID} \mid \mathrm{UEM}, 256). \tag{4}$$

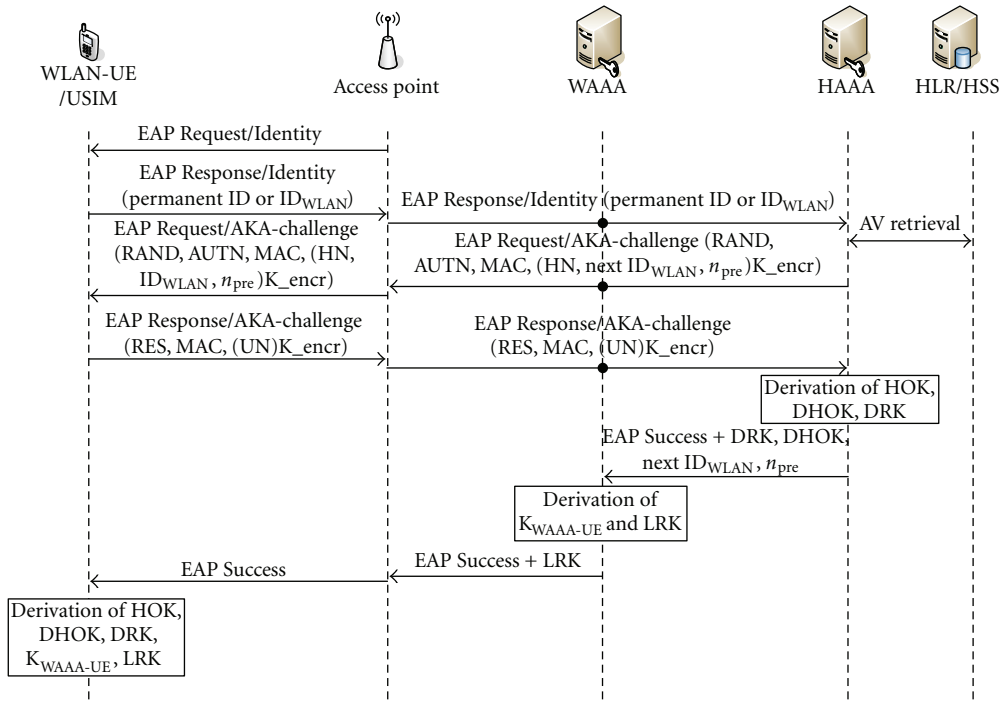


Figure 3: Modified EAP-AKA authentication protocol.

(iii) Secure delivery of DRK, DHOK, n_{pre} and ID_{WLAN} by the HAAA to the WAAA.

(iv) Secure delivery of LRK by the WAAA to the AP.

(v) Derivation of HOK, DHOK, DRK, LRK, and $\text{K}_{\text{WAAA-UE}}$ by the UE.

3.3. Intra/Inter-WLAN Fast Pre-authentication. A UE roams to a neighbor AP when experiencing poor signal-strength from the currently associated AP. The Target AP (TAP) might be in the same WLAN domain or belong to a different WLAN domain. Due to the lack of WLAN HH authentication protocol support by 3GPP in UMTS-WLAN interworking architecture and inadaptability of autonomous WLAN HH authentication protocols, we designed Intra- and Inter-WLAN Fast Pre-authentication protocols (Intra/Inter-WLAN FP) to minimize authentication delay and signaling overhead during intra- and inter-ESS HH. The proposed protocols utilize EAP-AKA messages and can efficiently operate in the UMTS-WLAN interworking architecture. Intra-WLAN FP is locally executed when the currently associated AP and the TAP reside in the same WLAN domain. Inter-WLAN FP is executed when the currently associated AP and the TAP reside in different WLAN domains. Intra/Inter-WLAN FP minimizes the dependency on HSS and HAAA to authenticate the UE which results in improved performance without compromising security.

The UE needs to supply target AP and target WAAA identities it requires to handover to, TAP ID and TWAAA ID. Therefore we propose adjusting IEEE 802.11 *Probe Response* management frames transmitted by the TAP to include its identity and the identity of WAAA it is associated with as Information Elements (IEs). Element IDs 7–15 and 32–255 are reserved for future use and can be used for this purpose [22]. Handover related decisions like handover triggers and best TAP selection is out of the scope of the paper. Figure 4 depicts Intra-WLAN FP operation.

In Intra-WLAN FP, the WAAA handles UE authentication instead of the HSS and HAAA. Intra-WLAN FP protocol proceeds as follows.

(1) When the UE recognizes the need for handover, it sends an *EAPoL-start* message to the currently associated AP, not shown in Figure 4. The AP replies with an identity request message.

(2) UE responds to the request with ID_{WLAN}, TWAAA ID and TAP ID.

(3) Receiving TWAAA ID and TAP ID indicates a handover pre-authentication request. The WAAA classifies this request as an Intra-WLAN if the received TWAAA ID matches its identity and the TAP ID matches the identity of one of the APs in the WLAN domain. The WAAA then consults WC and prepares a challenge message that includes a fresh nonce, WN, and the next ID_{WLAN} as well as WC and $\text{MAC1}_{\text{Intra}}$ calculated using $K_{\text{WAAA-UE}}$,

$$\text{MAC1}_{\text{Intra}} = \text{SHA-1}(\text{K}_{\text{WAAA-UE}}, \text{WC} \mid \text{ID}_{\text{WLAN}} \mid \text{WN}), \quad (5)$$

where SHA-1 is the Secure Hash Algorithm.

(4) In the UE's side, WC stored in the UE's database is matched with WC recently received. Then a new $\text{MAC1}_{\text{Intra}}$ is calculated and compared with the received $\text{MAC1}_{\text{Intra}}$. If both checks are positive,

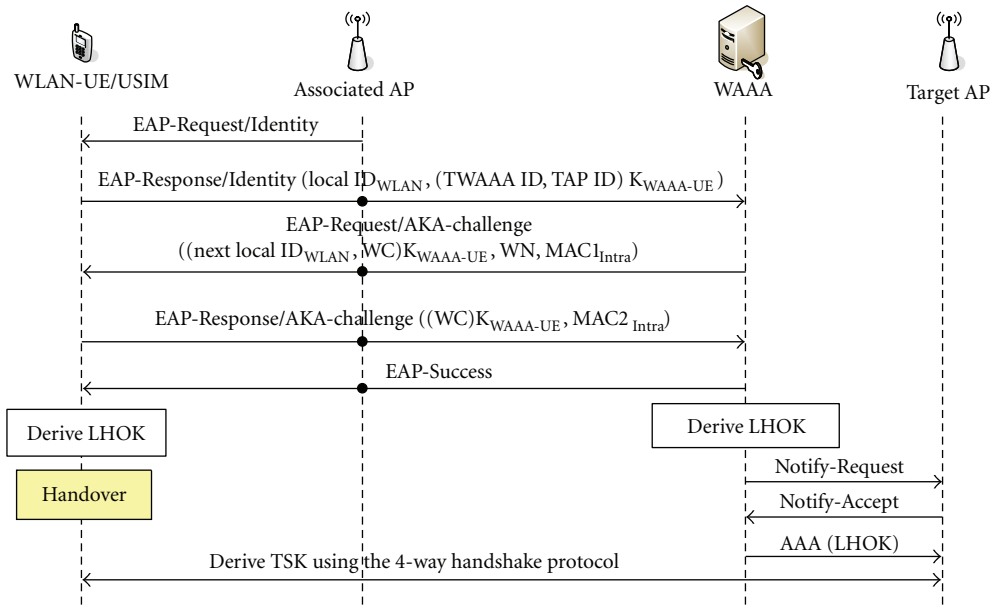


Figure 4: Intra-WLAN Fast Pre-authentication protocol.

the UE stores ID_{WLAN} and replies with WC and $MAC2_{Intra}$,

$$MAC2_{Intra} = \text{SHA-1}(K_{WAAA-UE}, WC \mid WN). \quad (6)$$

(5) The WAAA then derives a local-level handover key, LHOK, from DHOK as follows:

$$LHOK = PRF(DHOK, WC \mid TAP\ ID \mid UEM, 512). \quad (7)$$

The WAAA also increments WC and sends *EAP success* message to the UE. Consequently, the UE derives LHOK and increments WC. WAAA and TAP exchange *Notify-Request* and *Notify-Accept* RADIUS AAA message to confirm handover operation [30]. Finally LHOK is pushed to the TAP in *RADIUS Access-Accept* message with *MS-MPPE-Recv-Key* attribute [11].

In Inter-WLAN FP, authentication procedure is completed without the need to retrieve security keys from the HSS as shown in Figure 5. The protocol proceeds as follows:

(1) The UE replies to the identity request message with ID_{WLAN}, TWAAA ID, and TAP ID.

(2) The handover pre-authentication request is classified as Inter-WLAN by the WAAA if the TWAAA ID does not match its identity and TAP ID does not match any of the AP identities in the WLAN domain. The WAAA retrieves the UE permanent ID and forwards it along with the TAP ID and TWAAA ID to the HAAA.

(3) Upon receiving the IDs, the HAAA recognize that an Inter-WLAN FP is requested and prepares an authentication challenge. The challenge includes the next ID_{WLAN}, UN, newly generated HN and $MAC1_{Inter}$

$$MAC1_{Inter} = \text{SHA1}(K\_\text{auth}, UN \mid ID_{WLAN} \mid \text{new HN}). \quad (8)$$

UN was previously received by the HAAA in the modified EAP-AKA protocol.

(4) Upon receiving the authentication challenge, the UE checks UN, calculates a new $MAC1_{Inter}$ and compares it with the received $MAC1_{Inter}$. If all verification returns positive, ID_{WLAN} is stored and a reply message is prepared. The reply message includes the new HN, newly generated UN, WC, and $MAC2_{Inter}$,

$$MAC2_{Inter} = \text{SHA-1}(K\_\text{auth}, \text{new UN} \mid \text{new HN} \mid \text{last HN} \mid WC). \quad (9)$$

(5) Upon receiving the message, the HAAA consults WC to verify that pre-authentication limit is not exceeded and verifies $MAC2_{Inter}$. If all verifications are successful, the HAAA validates HOK lifetime, generates a new DHOK and DRK and *EAP Success* message is sent to the UE.

(6) Upon receiving *EAP success* message, the UE derives a new DHOK, DRK, $K_{TWAAA-UE}$, and LHOK. It also increments WC.

(7) AAA message that includes DHOK, DRK, WC, n_{pre}, UE permanent ID, ID_{WLAN}, and TAP ID is sent to the TWAAA by the HAAA. As a result, $K_{TWAAA-UE}$ and LHOK are generated and WC is incremented by TWAAA. Lastly, TWAAA confirms handover with TAP by exchanging RADIUS AAA *Notify-Request* and *Notify-Accept* message and forwards LHOK in *Access-Accept* message.

At the conclusion of a successful Intra- or Inter-WLAN FP, a fresh LHOK is held by the UE and the TAP. The LHOK is used to generate TSK, which is then used to

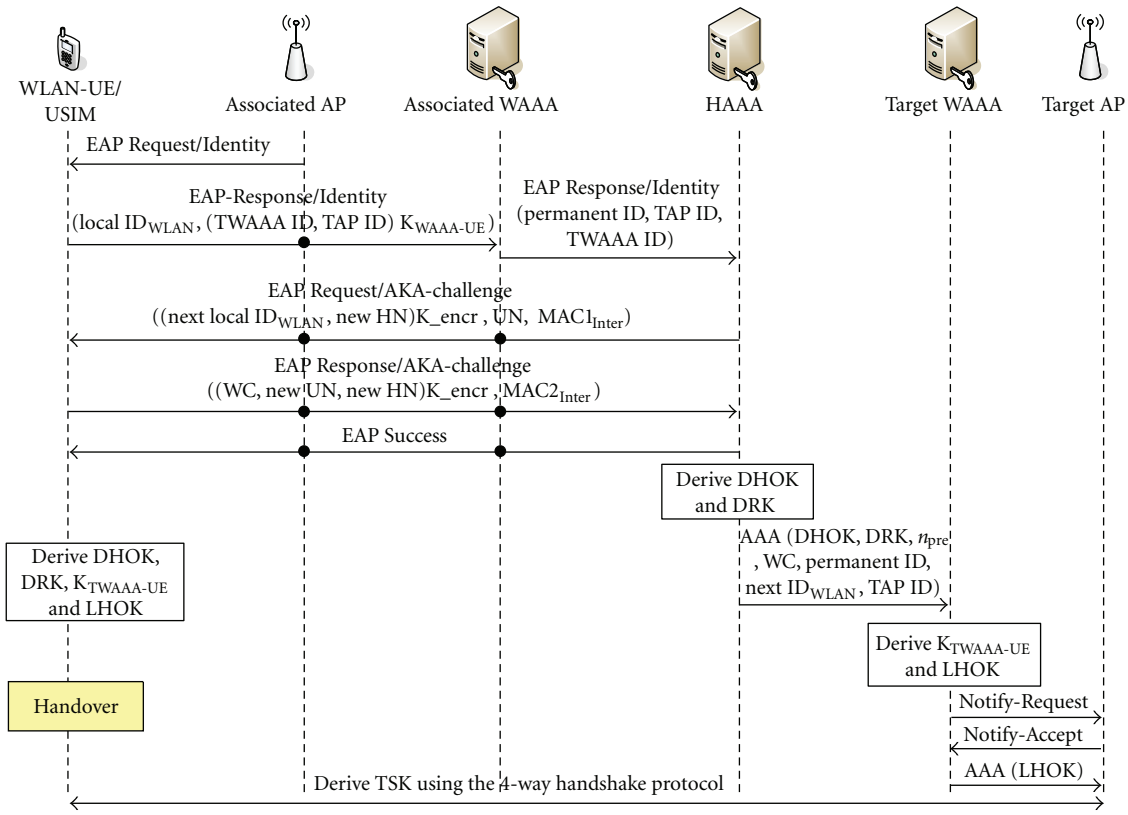


Figure 5: Inter-WLAN fast pre-authentication protocol.

derive additional keys that are needed to secure the link between the UE and the TAP. EAP-AKA highly depends on IEEE802.1X [31] protocol implemented in the AP to successfully control UE's network access. IEEE802.1X is a port-based access control protocol. When an EAP session completes successfully between the UE and the AP, normal communications is permitted by the latter to pass through an authorized port. Therefore, simultaneous exchange of normal communications and EAP session is disallowed. We propose two classes of Intra/Inter-WLAN FP execution depending on the implementation of IEEE802.1X protocol in the AP. The two classes differ on whether IEEE802.1X protocol in the AP permits single or multiport communications. Based on this, each class imposes different effect on the authentication delay. Single-port communication implies that normal communications between the UE and the AP is disallowed when EAP session is executed. Multiport communications implie that the AP can still handle normal communications while processing EAP messages. Multiport communications are achievable by simple modifications to the IEEE802.1X protocol in the AP. In studying the performance of our proposed protocols, both single-port and multiport communications are considered.

4. Performance Evaluation

In this section we evaluate the performance of our proposed pre-authentication protocols against EAP-AKA protocol. Performance evaluation against protocols in the literature like [15–19] is not reasonable because of the difference in the network architecture. We considered three performance metrics in our study, they are authentication signaling cost, authentication delay, and the load on critical nodes in the UMTS-WLAN interworking architecture.

4.1. UE Movement and Authentication Scenarios. Performance evaluations are studied based on a fixed path UE movement. This movement might not reflect realistic UE paths but it is considered here for performance evaluation purposes only. Initially, the UE is connected to AP1 in WLAN1 as depicted in Figure 6. The UE then performs two intra-ESS HH to APs 2 and 3 in WLAN1, respectively. Later, it performs an inter-ESS HH to AP1 in WLAN2 followed by two intra-ESS HH to AP2 and AP3 in WLAN2, respectively.

Three authentication scenarios are considered in the performance study.

Scenario 1 (Sc1). This scenario adopts authentication protocols specified by 3GPP [3]. The UE performs EAP-AKA authentication whenever it starts communicating with an AP regardless whether HH was performed or not.

Scenario 2 (Sc2). This scenario executes our proposed modifications to EAP-AKA protocols and Intra/Inter-WLAN FP protocols. The IEEE802.1X protocol in the APs in this scenario supports single-port communications.

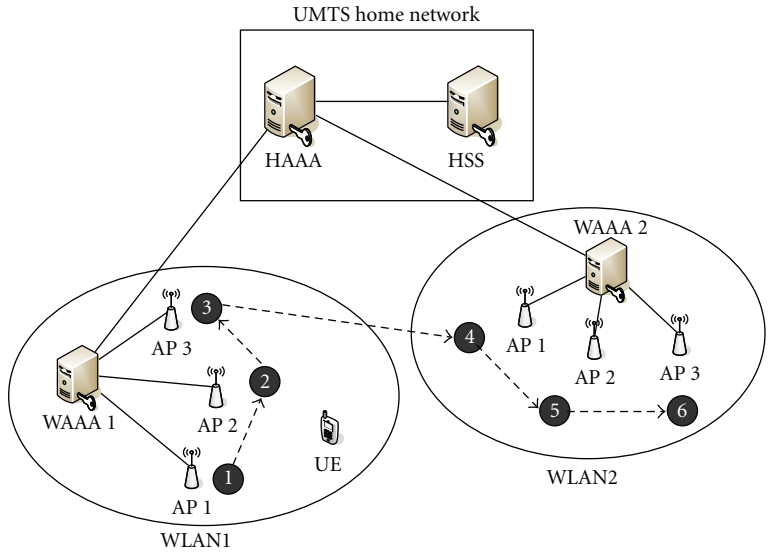


Figure 6: UE movement.

Scenario 3 (Sc3). This scenario is identical to Sc2 in terms of message signaling, however, IEEE802.1X protocol in the APs supports multiport communications. Therefore, the UE and APs are capable of handling normal communications while processing EAP messages for pre-authentication purposes.

Our proposed pre-authentication protocols represented by Sc2 and Sc3 are expected to show similar results in terms of authentication, signaling cost, and the load on critical nodes, however, authentication delay experienced by these scenarios should distinctly differ. Authentication protocols invoked in Sc2 and Sc3 depend on the number of permitted pre-authentications (n_{pre}). For example, setting n_{pre} to 1, 3, and 5 mean that our modified EAP-AKA protocol is going to be invoked thrice, twice, and once, respectively. The value of n_{pre} should be carefully chosen by the service provider; very high value might negatively affect security because of frequent reuse of HOK and DHOK while very low values might negatively affect performance due to contacting UHN repeatedly for authentication. Figure 7 depicts the authentication protocols in Sc1 and Sc2 when $n_{\text{pre}} = 5$.

4.2. Authentication Signaling Cost. Studying the signaling cost produced by an authentication protocol is an important metric in evaluating its performance. Authentication signaling cost is the accumulative traffic load introduced in the network by exchanging authentication signaling during a communication session [32]. For simplicity, all nodes are a single hop (H) apart except between WAAA and HAAA. The authentication signaling cost (C) for the authentication scenarios when $n_{\text{pre}} = 5$ are calculated as follows:

$$\begin{aligned} C_{\text{Sc1}} &= \left(6\, M_{\text{EAP-AKA(stnd)}}\right) \times S \times Nm, \\ C_{\text{Sc2}} &= C_{\text{Sc3}} = \left(M_{\text{EAP-AKA(}\,\text{mod}\,\text{)}} + 4\, M_{\text{Intra}} + M_{\text{Inter}}\right) \times S \times Nm, \end{aligned} \tag{10}$$

where (M) is the number of messages exchanged in each authentication protocol, S is the average message size, it is set to 100 bytes. Nm is the average number of UE movements during a session, $Nm = Ts/Tr$. Ts is the average session time, it is set to 1000 seconds. Tr is the average WLAN resident time, it varies from 10 to 40 seconds. Figure 8 shows the authentication signaling cost against UE resident time when $H_{\text{WAAA-HAAA}} = 3$ for different n_{pre} values.

Generally the higher the UE resident time the less authentication signaling is generated. It is clear from the figure that the authentication signaling cost of Sc2 is less than Sc1. Our proposal reduces signaling cost by 13% when compared to Sc1 when $n_{\text{pre}} = 1$. Improved performance results are achieved when increasing n_{pre} value. Reduction in signaling cost experienced in Sc2 reaches up to 21% and 29% in comparison to Sc1 when setting n_{pre} values to 3 and 5, respectively. As discussed earlier, Sc1 experience the same signaling cost in spite of n_{pre} value. Increasing n_{pre} value means reducing the frequency of invoking the modified EAP-AKA protocol and permitting additional local pre-authentications without the need to contact UHN hence achieving drastic reduction in authentication signaling cost.

4.3. Authentication Delay. Authentication delay plays an important factor in the overall handover delay. In this paper we assume that delays that constitute handover delay, other than authentication delay, like AP scanning delay and MIP registration delay have an equal effect on all authentication scenarios. Authentication delay is calculated starting from sending *EAP Request/Identity* message and ends by invoking the 4-way handshake protocol. Generally, the delay between two nodes, A and B is defined as follows:

$$\begin{aligned} T_{\text{A-B}} &= M_{\text{A-B(wl)}}\left(D_{\text{trans(wl)}} + 2D_{\text{proc}}\right) \\ &\quad + M_{\text{A-B(wi)}} H_{\text{A-B}}\left(D_{\text{trans(wi)}} + 2D_{\text{proc}}\right), \end{aligned} \tag{11}$$

where $M_{\text{A-B(wl/wi)}}$ signifies the number of messages exchanged between nodes A and B in the wireless network and

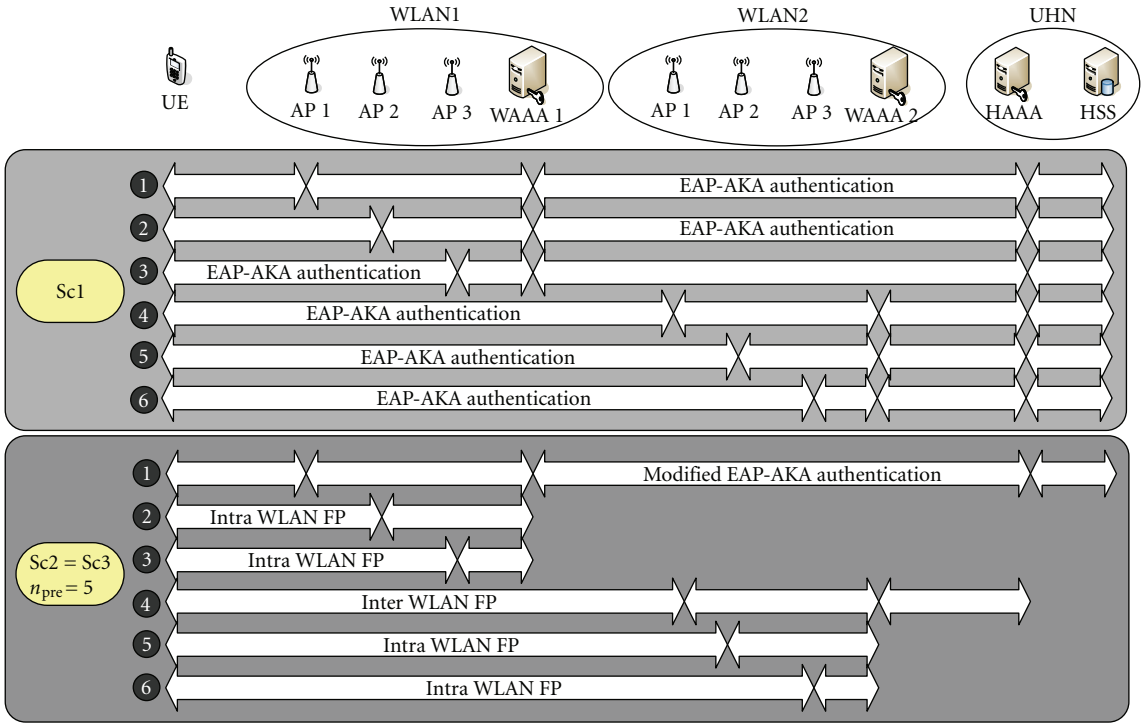


FIGURE 7: Authentication scenarios when $n_{pre} = 5$.

wired network, respectively, $H_{A\text{-}B}$ are the number of hops separating A and B in the wired network, $D_{trans(wl/wi)}$ are the transmission delay that includes propagation and routing delay in the wireless and wired networks, respectively. $D_{trans(wl)}$ is set to 2 milliseconds while $D_{trans(wi)}$ is set to 0.5 milliseconds. D_{proc} is the nodal processing delay which includes queuing delay, it is set to 0.001 milliseconds. All parameter values used in the study are taken from [32]. From (11), authentication delay (T) of each authentication protocol is calculated. The authentication delay in the standard and modified EAP-AKA when $n_{pre} = 5$ is given by

$$\begin{aligned} T_{\text{EAP-AKA(stnd)}} &= T_{\text{EAP-AKA(mod)}} \\ &= \left(5D_{\text{trans-wl}} + 10D_{\text{proc}}\right) + \left(4D_{\text{trans-wi}} + 8D_{\text{proc}}\right) \\ &\quad + \left(12D_{\text{trans-wi}} + 24D_{\text{proc}}\right) + \left(2D_{\text{trans-wi}} + 4D_{\text{proc}}\right) \\ &\quad + 2D_{\text{AV}} + D_4. \end{aligned} \tag{12}$$

The authentication delay for Intra/Inter-WLAN FP in Sc2 and Sc3, is given by

$$\begin{aligned} T_{\text{Intra-Sc2}} &= \left(5D_{\text{trans-wl}} + 10D_{\text{proc}}\right) \\ &\quad + \left(7D_{\text{trans-wi}} + 14D_{\text{proc}}\right) + D_4, \end{aligned}$$

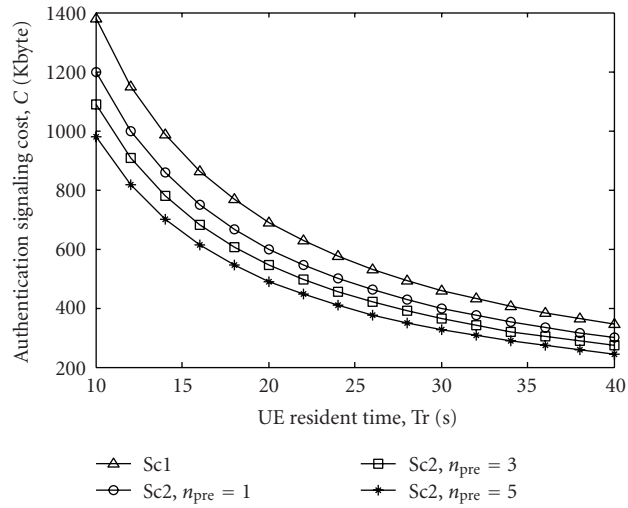


FIGURE 8: Authentication signaling cost for Sc1 and Sc2 for different n_{pre} values.

$$\begin{aligned} T_{\text{Inter-Sc2}} &= \left(5D_{\text{trans-wl}} + 10D_{\text{proc}}\right) + \left(7D_{\text{trans-wi}} + 14D_{\text{proc}}\right) \\ &\quad + \left(15D_{\text{trans-wi}} + 30D_{\text{proc}}\right) + D_4, \\ T_{\text{Intra-Sc3}} &= 3D_{\text{trans-wi}} + 6D_{\text{proc}} + D_4, \\ T_{\text{Inter-Sc3}} &= 6D_{\text{trans-wi}} + 12D_{\text{proc}} + D_4. \end{aligned} \tag{13}$$

D_4 denotes the delay incurred by executing the 4-way handshake protocol, it is set to 20 milliseconds. Note that

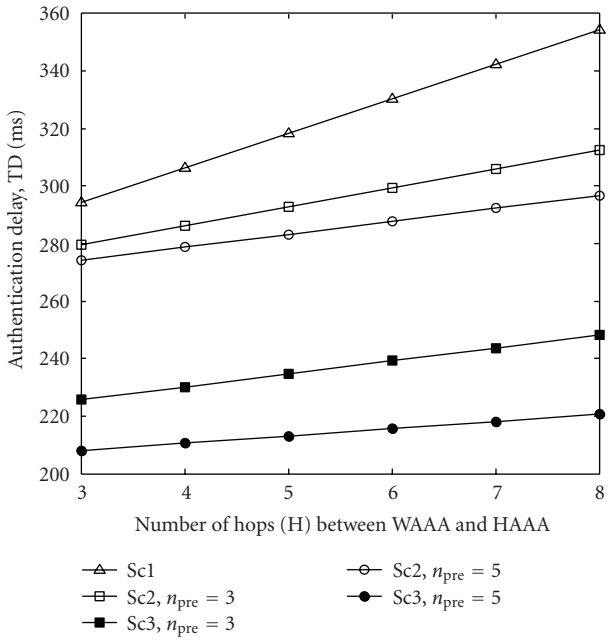


Figure 9: Authentication delay in Sc1, Sc2, and Sc3 when varying $H_{\text{WAAA-HAAA}}$.

Table 1: Number of keys generated in the three authentication scenarios.

	Sc1	Sc2 = Sc3		
n_{pre}	—	1	3	5
UE	36	39	29	19
WAAA1	0	5	4	4
WAAA2	0	5	5	4
HAAA	24	23	16	9
HSS	12	6	4	2
Total: all nodes	72	78	58	38
Total: critical nodes	72	68	49	30
Total key size in UE (byte)	1272	1500	1160	820

D_{AV} is the processing delay of generating AVs using "f1–f5" functions in the HSS and USIM, it is set to 0.001 milliseconds. The processing delays incurred by generating new keys in our proposed protocols by WAAA are expressed as a normal processing delay (D_{proc}). This is because WAAAs are usually equipped with high processing capabilities and control far less number of UEs compared to HSS and HAAA. Although our proposed protocols in Sc2 and Sc3 undergo similar authentication signaling cost, they differ distinctly in the authentication delay. The total authentication delay (TD) for each scenario when $n_{\text{pre}} = 5$ is calculated as follows:

$$\begin{aligned} \text{TD}_{\text{Sc1}} &= 6T_{\text{EAP-AKA(stnd)}}, \\ \text{TD}_{\text{Sc2}} &= T_{\text{EAP-AKA(mod)}} + 4\,T_{\text{Intra-Sc2}} + T_{\text{Inter-Sc2}}, \\ \text{TD}_{\text{Sc3}} &= T_{\text{EAP-AKA(mod)}} + 4\,T_{\text{Intra-Sc3}} + T_{\text{Inter-Sc3}}. \end{aligned} \quad (14)$$

By varying $H_{\text{WAAA-HAAA}}$ and n_{pre} values, we can compare the authentication delays of the three scenarios. Figure 9 shows the authentication delay of each scenario for different n_{pre} values. Our protocols represented by Sc2 and Sc3 outperform standard authentication protocol. When $n_{\text{pre}} = 1$, authentication delay in Sc2 is slightly less than Sc1 due to multiple execution of the modified EAP-AKA authentication which is a delay intensive operation. However, since Sc3 takes advantage of the multiport communications in the AP, it experiences much less delay reduction comparing to Sc1. Our proposed protocols demonstrate exceptional results when increasing n_{pre} value as shown in Figure 9. When $n_{\text{pre}} = 3$ and $H_{\text{WAAA-HAAA}} = 8$, delay reduction in Sc2 and Sc3 reaches up to 12% and 30%, respectively, compared to Sc1.

When $n_{\text{pre}} = 5$, our protocols capitalize on the single execution of the modified EAP-AKA protocol to perform several pre-authentications without the need to involve HSS and HAAA in the authentication procedure which ultimately reduces authentication signaling cost and authentication delay. In such settings, authentication delay reduction in Sc2 and Sc3 reaches up to 16% and 38% comparing to Sc1. Increasing n_{pre} value reflects in more reductions in the authentication delay in our proposed protocols comparing to the standard protocol. This feature illustrates the superiority and suitability of our proposed protocols to sustain quality of service of delay-sensitive applications running on the UE.

4.4. Load on Critical Nodes. In UMTS-WLAN interworking architecture, critical nodes involved in the authentication procedure are HSS, HAAA, and the UE. HSS and HAAA are considered critical because they handle the authentication of hundreds of thousands of UEs. The UE is considered critical as well because of the limitation in its processing capabilities. In EAP-AKA, key generation and distribution schemes are included in the authentication procedure. In our proposed protocols, HSS and HAAA delegate the authentication responsibility to trusted WAAA. Therefore, the processing overhead on these critical nodes is reduced. Since our modifications to EAP-AKA introduced additional keys generated by UE, HAAA, and WAAA, a study on the effect of the additional keys was important. In our study we considered the number and memory sizes of keys introduced in each authentication protocol starting from CK and IK down the hierarchy to the key used in the 4-way handshake protocol, that is, MSK in Sc1 and LHOK/LRK in Sc2. Figure 10 illustrates the keys generated by each node during UE movement when $n_{\text{pre}} = 5$. Table 1 indicates the total number of keys generated by all nodes for different n_{pre} values.

As indicated by Table 1, the total number of keys generated by all nodes in Sc2 decreases as n_{pre} value increase. When $n_{\text{pre}} = 1$, Sc2 generates 6 more keys in total comparing to Sc1 due to frequent execution of the modified EAP-AKA protocol. As n_{pre} value increase, the frequency of executing the modified EAP-AKA protocol decreases and hence fewer keys are generated. When increasing n_{pre} to 5, the total number of keys generated by all nodes in Sc2 is almost half of that generated in Sc1. Critical nodes in Sc2 generate 4 keys less than Sc1 when n_{pre} is set to 1, Sc2 generates less than half the number of keys generated in Sc1 when n_{pre}

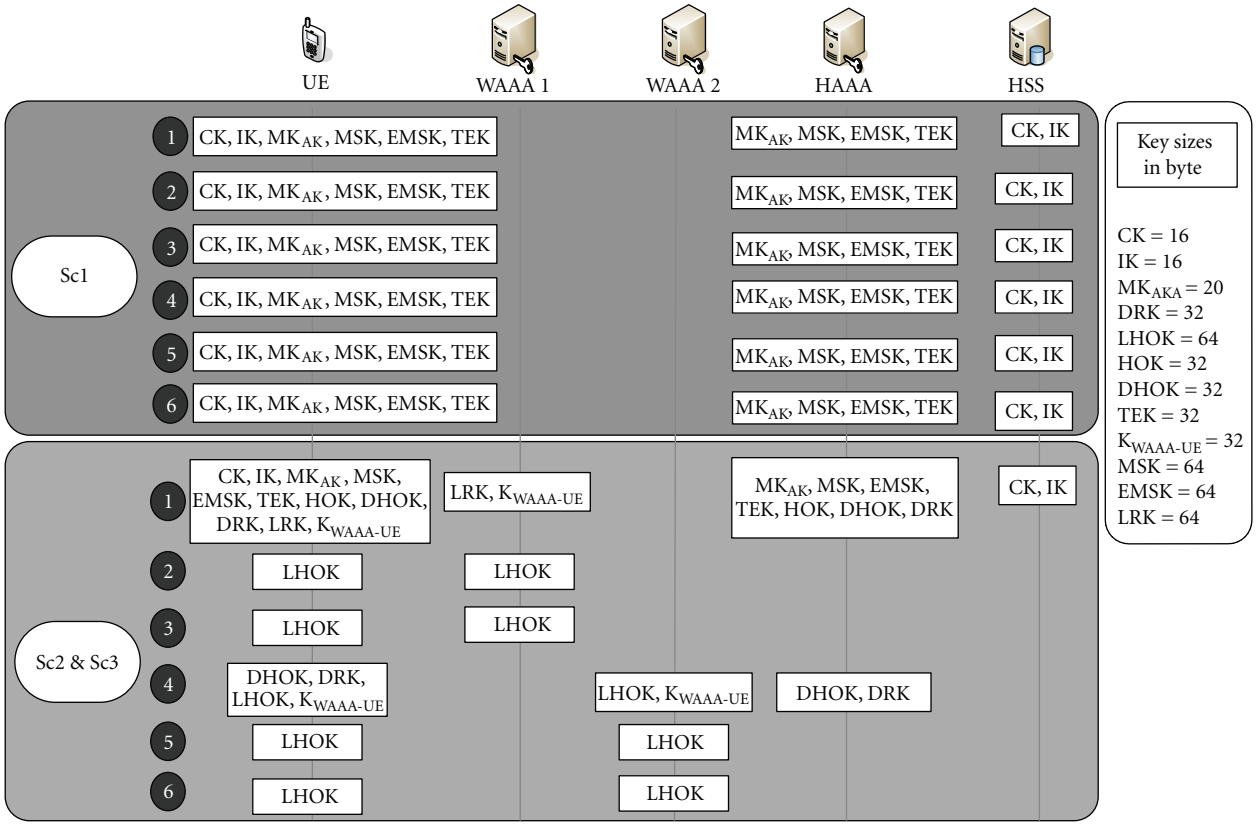


Figure 10: Keys generated by each node when $n_{pre} = 5$.

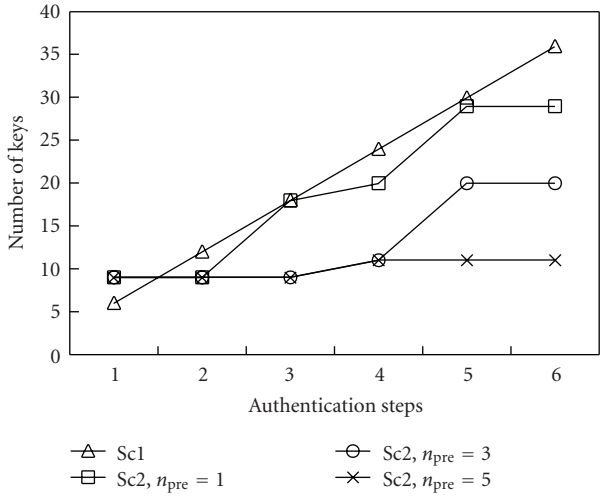


Figure 11: Number of keys generated by HSS and HAAA.

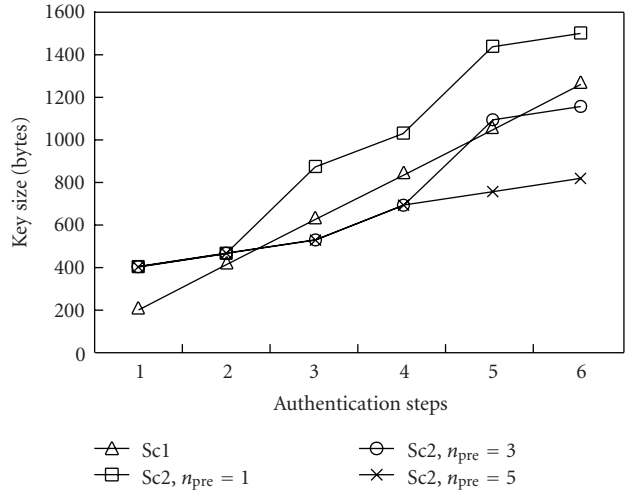


Figure 12: Memory storage space required by the UE to store security keys.

is set to 5. Critical nodes in Sc2 generate and maintain far less number of keys compared to their counterparts in Sc1 because the WAAA handle some of the key generation activity. The number of keys generated by critical nodes in addition to WAAA1 and WAAA2 in Sc2 is 58 and 38 when $n_{pre} = 3$ and $n_{pre} = 5$, respectively, which is clearly less than the number of keys generated by all nodes in Sc1.

From Table 1, the number of keys generated by HSS and HAAA in Sc1 is always greater than the number of keys generated by HSS and HAAA in Sc2. This is also illustrated in Figure 11. Number of keys generated by HSS and HAAA are 29, 20, and 11 when n_{pre} is set to 1, 3, and 5, respectively, compared to 36 keys generated in Sc1. This advantage is highly valued when more UEs roam to the network. For

```
role waaaserver
  (
    P, WAAA, AP1, AP2    : agent, % UE, WAAA server, Access Point 1 and 2
    F1, HMAC             : hash_func,   % MAC generation and key generation functions
    KPW, KAP1W, KAP2W, DHOK: symmetric_key,
    WCN, AP2_ID          : text,   % WLAN counter and AP2 ID
    SND_AP1W, RCV_AP1W, SND_AP2W, RCV_AP2W : channel (dy))
played_by WAAA def=
local
   WN, INTRA_ID        : text,   % WAAA nonce and UE ID
   WCNE                : {text}_symmetric_key,
   MAC1_INTRA, LHOK : hash (symmetric_key.text.text.text),
   MAC2_INTRA          : hash (symmetric_key.text.text),
   State               : nat
const
    request_id, respond_id, success : text,
    lhok3, wn1, wn2          : protocol_id
init State := 2
transition
1. State = 2 /\ RCV_AP1W (respond_id.INTRA_ID') = | >
    State' := 5 /\ WN' := new()     /\ WCNE' := WCN_KPW
                /\ MAC1_INTRA' := HMAC (KPW.INTRA_ID'.WN'.WCN)
                /\ SND_AP1W (WN'.MAC1_INTRA'.WCNE')
                /\ witness (WAAA, P, wn1, WN')   % for UE to authenticate WAAA
2. State = 5 /\ RCV_AP1W (WCNE'.MAC2_INTRA')
               /\ MAC2_INTRA' = HMAC (KPW.WN.WCN) = | >
    State' := 8 /\ LHOK' := F1 (DHOK.WCN.INTRA_ID.AP2_ID)
                /\ request (WAAA, P, wn2, WN) % for WAAA to authenticate UE
                /\ SND_AP1W (success) /\ SND_AP2W (success.{LHOK'}_KAP2W)
                /\ secret (LHOK', lhok3, {P, WAAA, AP2})
end role
```

Figure 13: HLPSL code describing WAAA's role in Intra-WLAN FP.

example, when 5 UEs exist in the network and followed the same movement indicated by Figure 6, HSS and HAAA end up generating 180 keys in Sc1 while only 55 keys are necessary in Sc2 when $n_{\text{pre}} = 5$. This shows that our proposed protocols are capable of managing large number of UEs in the interworking architecture efficiently comparing to standard EAP-AKA protocol.

Since the UE has limited processing capabilities and storage capacity, we evaluated the number of keys generated by it as well as the memory size required to store security keys. Continuing a similar trend, the UE generates less number of keys as the value of n_{pre} increases. Generally the number of keys generated by the UE in Sc2 is less than Sc1 when $n_{\text{pre}} > 2$. Furthermore, while the UE requires 1.272 Kbytes of storage space in Sc1, it needs 1.160 Kbytes and 820 bytes of storage space in Sc2 when n_{pre} is set to 3 and 5, respectively. Figure 12 illustrates the amount of storage space required by the UE. Since the modified EAP-AKA protocol is invoked thrice when $n_{\text{pre}} = 1$ in Sc2, more storage space to store the keys in the UE is anticipated.

5. Security Analysis

Performance improvements to authentication protocols should not compromise its security. In this section we analyze the security of the proposed protocols in terms of supporting secured key management scheme, mutual authentication service, protection of the integrity of exchanged messages, and protection of transmitted identities.

5.1. Secured Key Management. Keys must be held by the minimum number of nodes possible. Unnecessary distribution of keys must be avoided and keys must be unique to key holders. Additionally, keys must never be shared between nodes from the same hierarchal level and keys used directly in protecting communication messages must not be reused. These measures are collectively known as the principle of least privilege, which prevents the "domino effect" problem [33] in key management protocols. Our protocols are designed to abide by the principles of least privilege. For example, DHOK is only generated by the UE and HAAA because no other node has access to HOK and HN values used in the generation process. This key is only used by the UE and WAAA and never shared between different WAAA servers residing in different WLAN networks. Similarly, LHOK is only generated by the UE and WAAA because no other node has access to DHOK and WC values used in the generation process. LHOK is used by the UE and TAP only and is never shared between different TAPs and never reused in future pre-authentications. To emphasize the principle of least privilege, the HAAA must delete DHOK from its database after delivering it to the WAAA. Likewise, the WAAA must delete LHOK from its database after delivering it to the TAP.

Keys, nonces, and counters are securely transmitted to protect against eavesdropping attacks. No keys are transmitted in the WLAN link between the UE and AP. Sensitive security information traveling between the HAAA

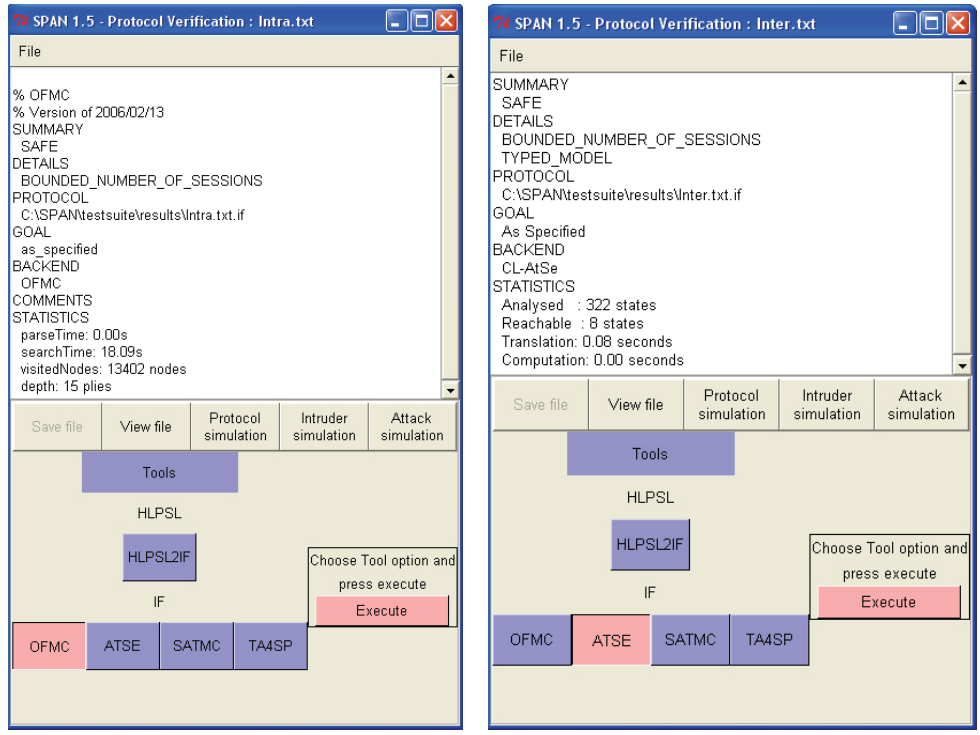
```
role peer (
  P, WAAA1, WAAA2, AP1, AP2, HAAA      : agent,  %UE, WAAA servers, APs and HAAA
    F1,HMAC                : hash_func,   %MAC generation and key generation functions
    KPH                    : symmetric_key,   % shared key between UE and HAAA
    HOK, MSK                 : symmetric_key,
    HN, UN                   : text,   % HAAA nonce and UE nonce
    WAAA2_ID, AP2_ID       : text,   % WAAA2 AND AP2 identities
    SND_AP1P, RCV_AP1P : channel (dy))
played_by P def=
local
    NUN, NHN, WCN, INTER_ID    : text,  % New UN, New HN, WLAN Counter, UE Identity
    WCNE                                : {text}_symmetric_key,
    MAC1_INTER, DHOK, LHOK              : hash (symmetric_key.text.text.text),
    MAC2_INTER                          : hash (symmetric_key.text.text.text.text),
    State                               : nat
const
    request_id, respond_id, success  : text,
    lhok1, nhn1, nhn2          : protocol_id
init State  := 1
transition
1. State = 1 /\ RCV_AP1P (request_id) = | >
    State′ := 5 /\ INTER_ID′ := new()  /\ SND_AP1P (respond_id.INTER_ID′)
2. State = 5 /\ RCV_AP1P ({NHN′}_KPH.UN′.MAC1_INTER′)
              /\ MAC1_INTER′ =HMAC (KPH.UN.INTER_ID.NHN′) = | >
    State′ := 9 /\ NUN′ := new()
              /\ MAC2_INTER′ := HMAC (KPH.NUN′.NHN′.HN.WCN)
              /\ WCNE′ := {WCN}_KPH
              /\ SND_AP1P (WCNE′.{NUN′}_KPH.{NHN′}_KPH.MAC2_INTER′)
              /\ request (P, HAAA, nhn1, NHN′)
              /\ witness (P, HAAA, nhn2, NHN′)
3. State = 9 /\ RCV_AP1P (success) = | >
    State′ := 13 /\ DHOK′ := F1 (HOK.NHN.WAAA2_ID.INTER_ID)
              /\ LHOK′ := F1 (DHOK.WCN.AP2_ID.INTER_ID)
              /\ secret (LHOK′, lhok1,{P, WAAA2, AP2}) % to assure secrecy of
                                                       % LHOK between UE, TWAAA and
                                                       % TAP
end role
```

Figure 14: HLPSL code describing UE's role in Inter-WLAN FP.

and WAAA and between WAAA and TAP are protected by the LTSA previously established between them. Nonces and counters are encrypted with $K_{\text{WAAA-UE}}$ in Intra-WLAN FP and with K_encr in Inter-WLAN FP when traveling in the WLAN link between the UE and AP. Furthermore, all keys are freshly generated to defend against replay attacks. EMSK and HOK are fresh because a new RAND and AUTN values are used to generate them. DHOK is fresh because a new HN is used to generate it. Since DHOK is fresh, $K_{\text{WAAA-UE}}$ is believed to be fresh as well. Finally, LHOK is fresh because WC is used to generate it which is continuously incremented after every successful pre-authentication.

5.2. Mutual Authentication and Keys Secrecy. Our proposed protocols provide mutual authentication service to protect against Man-In-the-Middle attacks (MITM), impersonation attacks, and rogue AP attacks. To verify this, we tested our protocols using formal security verification tool known as the "Automated Validation of Internet Security Protocols and Applications" (AVISPA) [34]. AVISPA package is a state-of-the-art tool for the automatic verification and analysis of Internet security protocols. AVISPA integrates automatic security analysis and verification back-end servers like "On-the-Fly Model-Checker" (OFMC), "Constraint-Logic-based Attack Searcher" (Cl-AtSe), and SAT-based Model-Checker (SATMC). Protocols under examination by AVISPA must be coded in the "High Level Protocol Specifications Language" (HLPSL) to be tested by the back-end servers.

HLPSL is an expressive, role-based formal language used to describe the details of the protocols in question. Typically, HLPSL code includes the roles played by all the principals in the security protocol, like UE, WAAA, and HAAA, as well as the role of the environment and the security goals that has to be achieved. Figure 13 illustrates WAAA's role in Intra-WLAN FP expressed in HLPSL. To permit testing the support of secured mutual authentication between the UE and WAAA, request and witness terms are used. The statement (request(WAAA,P,wn2,WN)) in Figure 13 indicates the requirement that the WAAA authenticates the UE while the statement (witness(WAAA, P, wn1, WN′)) indicates the requirement that the WAAA should be authenticated by the UE. Figure 14 illustrates UE's role in Inter-WLAN FP expressed in HLPSL. To permit testing the confidentiality of LHOK, the term secret is used. The statement (secret (LHOK′, lhok1, {P, WAAA2, AP2}) in Figure 14 indicates the



(a) (b)

Figure 15: (a) Message returned by SPAN as a result of testing Intra-WLAN FP with OFMC. (b) Message returned by SPAN as a result of testing Inter-WLAN FP with Cl-AtSe.

Figure 16: Message returned by AVISPA web interface after testing Intra-WLAN FP with SATMC.

requirement to keep LHOK confidential to the UE, TWAAA, and the TAP.

The support of secured mutual authentication in addition to the secrecy of keys in Intra/Inter-WLAN FP was tested using OFMC, Cl-AtSe, and four SAT solvers in SATMC. All tests returned positive results and confirmed the security of mutual authentication service and no authentication attacks were found. Results also confirmed the secrecy of LHOK and no vulnerabilities were discovered. A stand-alone graphical version of the AVISPA package was used in testing our protocols named Security Protocol ANimator for AVISPA (SPAN). Figure 15(a) demonstrate the messages returned by SPAN as a result of testing Intra-WLAN FP with OFMC and Figure 15(b) shows the message returned by SPAN as a result of testing Inter-WLAN FP protocol by Cl-AtSe. Figure 16 shows results of testing Intra-WLAN FP with zchaff SAT solver in SATMC. AVISPA Web Interface was used in SATMC testing because of problems running SATMC in the stand-alone version. Output text was rearranged to fit in a single screen.

5.3. Protection of Message Integrity. The integrity of authentication challenges and responses are protected by appending a Message Authentication Code (MAC) that covers important information carried in EAP messages. MAC preserves the integrity of EAP message, protects against MITM attacks, and validates the authenticity of the sender. In Intra-WLAN FP, MAC_{Intra} is calculated using $K_{WAAA\text{-}UE}$ while MAC_{Inter} is calculated using K_auth in Inter-WLAN FP. $MAC2_{Inter}$ plays an important role to assure the authenticity of the UE to the HAAA by including the last HN value.

5.4. Protection of Identities. It is always desirable to conceal the identity of the UE, TAP ID, and TWAAA ID to protect against eavesdropping and tracking of UE movement. In our proposed protocols, the UE is supplied a local ID, ID_{WLAN}, to be used in future pre-authentications instead of its permanent ID. Local IDs are one-timer identifiers valid for a single pre-authentication session. Therefore, a UE must obtain a new local ID for the subsequent pre-authentication procedure. New local IDs sent by the WAAA and received by the UE are encrypted with $K_{WAAA\text{-}UE}$ and K_encr in Intra-WLAN FP and Inter-WLAN FP, respectively. Current local IDs supplied by the UE and received by the WAAA cannot be encrypted because the identity of the UE must be known to extract the proper decryption key. This is the only case the local ID travels in clear text. This clear text transmission does not form a threat since this local ID is not reused in the future. TWAAA ID and TAP ID are also encrypted with $K_{WAAA\text{-}UE}$ when transmitted by the UE to the WAAA to defend against rogue AP attacks as well as to prevent tracking UE's movement.

6. Conclusions

It is common for UEs to perform horizontal handovers within and between WLANs in UMTS-WLAN interworking architecture due to the limited coverage area of WLAN networks. Handover delays affect the Quality of Service of applications running on the UE. It is always desirable to minimize handover delays. One of the major factors of delay during handover is the delay of mutual authentication between the UE and authentication servers. We designed pre-authentication protocols to reduce authentication delays that occur during intra- and inter-ESS horizontal handovers in UMTS-WLAN interworking environments. The proposed intra- and inter-WLAN pre-authentication protocols proved to surpass existing authentication protocols in terms of authentication signaling cost, authentication delay, and the load the authentication protocol places on critical nodes. The proposed protocols also achieve important security goals like the support of secured key management scheme and the support of mutual authentication service. The security of our protocols was verified by the "Automated Validation of Internet Security Protocols and Applications" (AVISPA) package. Examination of our protocols by AVISPA demonstrated its resistance to authentication attacks and confirmed key's confidentiality.

Acknowledgments

This work was supported in part by the Sultan Qaboos University under Contract number 1907/2005, Bell Canada, and the Natural Sciences and Engineering Research Council of Canada under Grant CRDPJ 328202-05. Part of this work was presented at the *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, *Qshine 2007*, Vancouver, Canada, August 2007.

References

[1] 3rd Generation Partnership Project, "3GPP system to wireless local area network interworking, system description (Release 7)," Technical Specification Group Services and System Aspects TS 23.234 v.7.2.0, 3GPP, Valbonne, France, June 2006.

[2] M. Shi, X. Shen, and J. W. Mark, "IEEE802.11 roaming and authentication in wireless LAN/cellular mobile networks," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 66–75, 2004.

[3] 3rd Generation Partnership Project, "3G security; WLAN interworking security (Release 7)," 3GPP Technical Specifications TS 33.234 v7.0.0, 3GPP, Valbonne, France, March 2006.

[4] G. Kambourakis, A. Rouskas, G. Kormentzas, and S. Gritzalis, "Advanced SSL/TLS-based authentication for secure WLAN-3G interworking," *IEE Proceedings: Communications*, vol. 151, no. 5, pp. 501–506, 2004.

[5] P. Prasithsangaree and P. Krishnamurthy, "A new authentication mechanism for loosely coupled 3G-WLAN integrated networks," in *Proceedings of the 59th IEEE Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 2998–3003, Milan, Italy, May 2004.

[6] H. Chen, M. Zivkovic, and D.-J. Plas, "Transparent end-user authentication across heterogeneous wireless networks," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, vol. 3, pp. 2088–2092, Orlando, Fla, USA, October 2003.

[7] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible Authentication Protocol (EAP)," IETF RFC 3748, June 2004.

[8] B. Aboba and D. Simon, "PPP EAP TLS Authentication Protocol," IETF RFC 2716, October 1999.

[9] P. Funk and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)," IETF Internet Draft, draft-ietf-pppext-eap-ttls-05.txt., July 2004.

[10] A. Palekar, D. Simon, J. Salowey, G. Zorn, H. Zhou, and S. Josefsson, "Protected EAP Protocol (PEAP) Version 2," IETF Internet Draft, draft-josefsson-pppext-eap-tls-eap-09.txt, October 2004.

[11] J. Arkko and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," IETF RFC 4187, January 2006.

[12] 3rd Generation Partnership Project, "Security architecture (Release 7)," 3GPP Technical Specifications, 3G Security TS 33.102 v7.0.0, 3GPP, Valbonne, France, December 2005.

[13] IEEE Standard for local and metropolitan area networks, "Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications, MAC Security Enhancements," IEEE Std 802.11i, 2004 Edition.

[14] A. Mishra, M. Shin, and W. Arbaugh, "An emprical analysis of the IEEE 802.11 MAC layer handoff process," *ACM SIGCOMM Computer Communications Review*, vol. 33, 2003.

[15] A. Mishra, M. Shin, N. L. Petroni Jr., T. C. Clancy, and W. A. Arbaugh, "Proactive key distribution using neighbor graphs," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 26–36, 2004.

[16] M. Kassab, A. Belghith, J.-M. Bonnin, and S. Sassi, "Fast pre-authentication based on proactive key distribution for 802.11 infrastructure networks," in *Proceedings of the 1st ACM International Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP '05)*, pp. 46–53, Montreal, Canada, October 2005.

[17] J. Hur, C. Park, and H. Yoon, "An efficient pre-authentication scheme for IEEE 802.11-based vehicular networks," in *Advances in Information and Computer Security*, vol. 4752 of *Lecture Notes in Computer Science*, pp. 121–136, Springer, Nara, Japan, October 2007.

[18] S. Pack and Y. Choi, "Pre-authenticated fast handoff in a public wireless LAN based on IEEE 802.1x model," in *Proceedings of IFIP TC6 Personal Wireless Communications*, vol. 234, pp. 175–182, October 2002.

[19] A. Mukherjee, T. Joshi, and D. P. Agrawal, "Minimizing re-authentication overheads in infrastructure IEEE 802.11 WLAN networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 4, pp. 2344–2349, New Orleans, La, USA, March 2005.

[20] M. Long, C.-H. Wu, and J. D. Irwin, "Localised authentication for inter-network roaming across wireless LANs," *IEE Proceedings: Communications*, vol. 151, no. 5, pp. 496–500, 2004.

[21] A. Al Shidhani and V. Leung, *Secured Fast Handover Protocols for 3G-WLAN Interworking Architecture*, Qshine, Vancouver, Canada, 2007.

[22] IEEE Standard for local and metropolitan area networks, "Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications," ANSI/IEEE Std 802.11, 1999 Edition (R2003).

[23] IEEE Standard for local and metropolitan area networks, "IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation," IEEE Std 802.11f-2003.

[24] IEEE Standard for local and metropolitan area networks, "Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications, Fast BSS transition," IEEE Std 802.11r (Draft 3), 2005.

[25] S. Bangoale, C. Bell, and E. Qi, "Performance study of fast BSS transition using IEEE802.11r," in *Proceeding of the International Conference on Wireless Communications and Mobile Computing (IWCMC '06)*, pp. 737–742, Canada, 2006.

[26] M. Lee, G. Kim, and S. Park, "Seamless and secure mobility management with Location-Aware Service (LAS) broker for future mobile interworking networks," *Journal of Communications and Networks*, vol. 7, no. 2, pp. 207–221, 2005.

[27] C. Lim, D.-Y. Kim, O. Song, and C.-H. Choi, "SHARE: seamless handover architecture for 3G-WLAN roaming environment," *Journal of Wireless Networks*, vol. 15, no. 3, pp. 353–363, 2009.

[28] A. Al Shidhani and V. C. M. Leung, "Local fast re-authentication protocol for 3G-WLAN interworking," *Security and Communication Networks*, 2008.

[29] B. Aboba, "Extensible Authentication Protocol (EAP) Key Management Framework," IETF Internet Draft (draft-ietf-eap-keying-14), June 2006.

[30] W. Arbaugh, "Handoff Extension to RADIUS," IETF Internet Draft (draft-irtf-aaaarch-handoff-04), October 2003.

[31] IEEE Standard for local and metropolitan area networks, "Port-based Network Access Control," IEEE Std 802.1x, 2001 Edition (R2004).

[32] H.-H. Choi, O. Song, and D.-H. Cho, "Seamless handoff scheme based on pre-registration and pre-authentication for UMTS-WLAN interworking," *Wireless Personal Communications*, vol. 41, no. 3, pp. 345–364, 2007.

[33] R. Housley and B. Aboba, "Guidance for AAA Key Management," IETF Internet Draft (draft-housley-aaa-key-mgmt-06), November 2006.

[34] AVISPA—Automated Validation of Internet Security Protocols, http://www.avispa-project.org.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 716480, 15 pages
doi:10.1155/2009/716480

Research Article

Secure Media Independent Handover Message Transport in Heterogeneous Networks

Jeong-Jae Won,[1] Murahari Vadapalli,[1] Choong-Ho Cho,[2] and Victor C. M. Leung[3]

[1] *Telecommunication and Network R&D Center, Samsung Electronics Co., LTD., 416 Maetan-3dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 443-742, South Korea*
[2] *Department of Computer & Information Science, Korea University, Chung-Nam 339-700, South Korea*
[3] *Department of Electric & Computer Engineering, The University of British Columbia, 2332 Main Mall, Vancouver, BC, Canada V6T 1Z4*

Correspondence should be addressed to Victor C. M. Leung, velung@ece.ubc.ca

Received 31 January 2009; Accepted 21 September 2009

Recommended by Yang Xiao

The IEEE 802.21 framework for Media Independent Handover (MIH) provides seamless vertical handover support for multimode mobile terminals. MIH messages are exchanged over various wireless media between mobile terminals and access networks to facilitate seamless handover. This calls for the need to secure MIH messages against network security threats in the wireless medium. In this paper, we first analyze IPSec/IKEv2 and DTLS security solution for secure MIH message transport. We show that handover latency can be an impediment to the use of IPSec and DTLS solutions. To overcome the handover overhead and hence minimize authentication time, a new secure MIH message transport solution, referred as MIHSec in this paper, is proposed. Experimental results are obtained for MIH between WLAN and Ethernet networks and the impacts of MIH message security on the handover latency are evaluated for IPSec, DTLS, and MIHSec security solutions. The effectiveness of MIHSec is demonstrated.

1. Introduction

Modern access systems have the capability to fulfill a specific quality-of-service (QoS) to the user, which leads to a requirement for seamless transitions from one access network to another in the presence of terminal mobility. Thus, it is anticipated that seamless interradio access technology (inter-RAT) mobility will be widely deployed in modern heterogeneous networks such as IEEE 802.11 (Wi-Fi), Global System for Mobile Communications (GSM), code-division multiple access (CDMA), and Mobile WiMAX. The growing importance of these issues has attracted the attention of standard groups including the IEEE 802.21 work group. The IEEE 802.21 standard defines Media Independent Handover (MIH) mechanisms that enable the optimization of inter-RAT handovers in heterogeneous networks [1–4].

The emerging IEEE 802.21 standard enables seamless, inter-RAT handover between IEEE 802 and non-IEEE 802 (e.g., 3GPP, 3GPP2) access technologies with the MIH function (MIHF) in the terminal and network sides. The role of MIHF is to provide media independent services to multi-RAT mobile terminals (MMTs) through a common interface to the mobility management and handover processes.

Related to this work, handover provisioning between GPRS and WiMAX is suggested in [2], which utilizes the potential of IEEE 802.21 to efficiently support inter-RAT handovers with full description of MIH services such as information service for providing network information, event service to trigger layer 2 (L2) events, and command service for handover execution like resource reservation and handover request. Reducing the authentication time over heterogeneous access networks involving interdomain mobility is a very critical criterion for seamless handover. In [3], Media independent preauthentication (MPA) provision is suggested. MPA provides a significant reduction in handover delays for both network-layer and application-layer mobility management protocols. However, the MPA scheme [3] does not address secure transport of media independent messages.

In addition to authentication as described in MPA [3], confidentiality and message integrity of MIH messages is another necessary requirement.

The requirements for MIH message level security are described in the 802.21 Security Study Group proposals [4]. The following security issues are identified.

(i) *MIH Access Control*. MIH service access should be controlled based on authentication and authorization.

(ii) *Replay Protection*. An MIH packet for an event or command can be replayed later to the same node.

(iii) *Denial of Service*.

(iv) *Message Integrity*. An MIH message may be altered on the way.

The available solutions for supporting authentication and access security are IP Security (IPSec) [5] and Datagram Transport Layer Security (DTLS) [6]. IPSec is a security solution at the network layer and is commonly used for most Internet applications. DTLS is a security solution at the transport layer, used for applications that operate over the User Datagram Protocol (UDP) or Transmission Control Protocol (TCP). In contrast to these existing security solutions, an MIH Security (MIHSec) solution is proposed and analyzed in this paper. Unlike IPSec and DTLS, MIHSec operates at the application layer.

The following MIH message protection issues are considered in this paper:

(i) communications between MIHF in MMT and any MIH Points of Service (PoS) in the access network,

(ii) communications between MIHF in MMT and MIH Information Server,

(iii) communications between MIHF in MMT and MIH IWF Broker. IWF provides the proprietary function between MIH services and a specific access network,

(iv) communications between MIHF in access routers (ARs).

In this paper, we first analyze IPSec with Internet Key Exchange version 2 (IKEv2) and DTLS security solutions for secure MIH message transport. We show that handover latency is an impediment to the use of IPSec and DTLS solutions. To overcome the handover overhead and hence minimize authentication time, a new secure MIH message transport solution, referred as MIHSec in this paper, is proposed.

IPSec and DTLS are off the shelf security solutions and software for them is readily available as GNU source. However, MIHSec is a newly defined security solution for providing security to MIH Messages. MIHSec operates at the application layer and utilizes Extensible Authentication Protocol (EAP) and MIH header TLV extensions to provide security to MIH messages.

Prototypes of MIH security methods with IPSEc/IKEv2, DTLS, and the new MIHSec mechanism are developed and the results are compared based on IEEE 802.21 Draft 11 for handover scenarios between Wi-Fi and Ethernet networks. The impacts on signaling latency, message transport latency, message overhead, and configurations are analyzed.

The rest of this paper is organized as follows. In Section 2, we provide background information on the IEEE 802.21 standard. In Section 3, we define the secure MIH transport models. In Section 4, the feasible methods for secure MIH transport with existing solutions such as IPSec/IKE and DTLS are analyzed. In Section 5, we present the design of our new secure MIH message transport protocol called MIHSec. In Sections 6 and 7, we exemplify the prototype by implementing and testing with MIHF implementation between Wi-Fi and Ethernet networks. Section 8 concludes the paper.

2. Related Work

2.1. IEEE 802.21 Standard. IEEE 802.21 [1] is a recent effort of IEEE that aims at enabling seamless service continuity among heterogeneous networks including 3GPP, 3GPP2, and the IEEE 802 family of standards. The standard defines a logical entity, MIHF, which is located between the lower layer (L2 and below) and upper layer. At the lower layer, MMT has multiple radio interfaces for different access technologies such as WLAN, WiMAX, and 3GPP. Upper layer entities that use the services provided by MIHF are referred as MIH Users. The role of MIHF is providing media independent services to MIH Users through a common interface to facilitate mobility management and handover processes.

Figure 1 shows the overview of MIH framework outlined by IEEE 802.21 standard. There are three primary services: Media Independent Event Service (MIES), Media Independent Command Service (MICS), and Media Independent Information Service (MIIS). MIES may indicate or predict changes in a state and transmission behavior of the physical and link layers. Common MIES provided through MIHF are "Link Up," "Link Down," "Link Parameters Change," and "Link Going Down." MICS enables higher layers to configure, control, and obtain information from the lower layers including physical and link layers. The information provided by MICS is dynamic information comprised of link parameters, whereas information provided by MIIS is comprised of static parameters.

MIIS provides a unified framework for obtaining neighboring network information that exists within a geographical area. It helps the higher layer mobility protocol to acquire a global view of available heterogeneous networks to conduct effective seamless handover. The information may be present in MMT locally but is usually stored in some external information server, which may be accessed by the MIHF in the MMT. For MIIS, the IEEE 802.21 standard defines information structures called Information Elements (IEs) that are classified into two groups: access network specific information (type of network, roaming agreements, cost of connecting, and QoS capabilities) and Point of Attachment (PoA) specific information (channel range, location, and supported data rates).

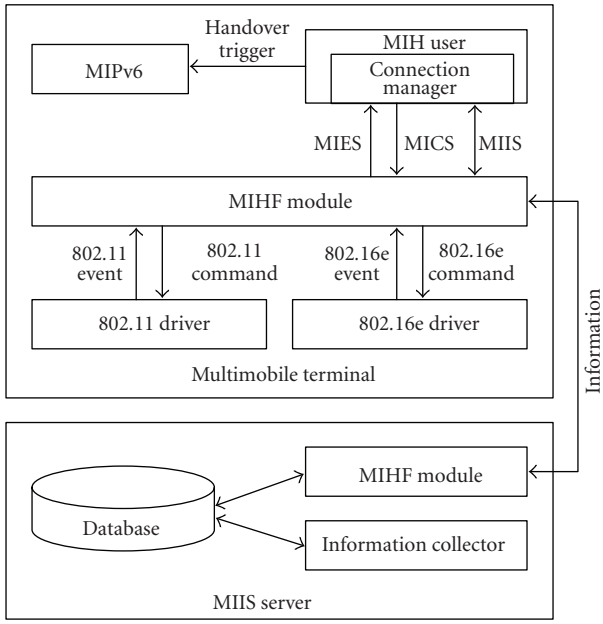


Figure 1: Overview of MIH framework.

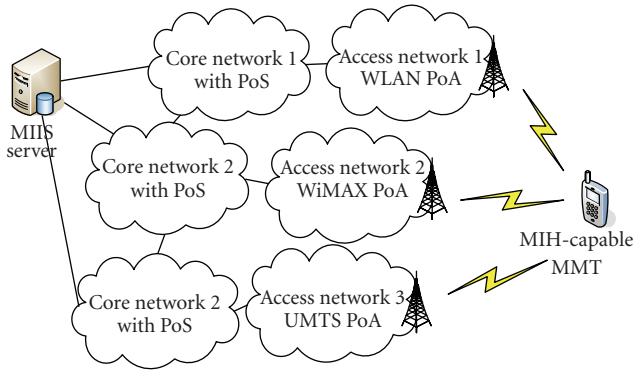


Figure 2: Network model with MIH services.

Figure 2 shows an example of the network model including MIH services. An MIH-capable MMT has multiple wireless interfaces based on different access technologies. It can connect concurrently to multiple PoAs, which are network side endpoints of L2 links. Each access network provides one or more MIH PoS nodes. To provide MIIS, an MIIS server can be located on the network side. The server maintains information of neighboring access networks in its local database.

Figure 3 shows the MIH-based handover message exchange involved in a mobile initiated handover from the serving network to the target network. The detailed explanation of the messages and procedures are as follows. The MIH procedure starts with the MMT querying about the surrounding networks. This query is forwarded by the information server located in the operator network and answered to MMT with available candidate network information (message 1-2). As the answer contains information regarding a possible network, the MMT switches on its target network interface and starts to measure the candidate networks. Just after measuring the candidate network, MMT will generate an MIH_MN_Candidate_Query message asking for the list of resources available in candidate networks and including the QoS requirements of the user (message 3–6).

At this point, the MMT has enough information about the surrounding networks to decide on the network to which it will hand over. Once the MMT has decided the target network to hand over, it delivers a handover commit command to the MIHF (message 7–10), which will be used for resource reservation in the target network before switching from the serving network to the target network (L2 and L3 handover). After completion of resource reservation in the target network, the MMT starts to establish the connection in the target network. Once the connection is established, a higher-layer handover procedure can start. In this case Mobile IP has been selected, although any other mobility management protocol would be equally suited. When the handover is completed at the higher layers, the MMT sends an MIH_HO_Complete message to the MIHF, which will inform the target PoS that it is now the new serving PoS. At this point the target PoS informs all the involved network elements of the handover finalization (message 11–14). Specifically, the target PoS has to inform the serving PoS of the handover completion so that it can release any resources.

2.2. Existing Secure Transport Methods. IP Security [5] and DTLS [6] are the existing secure transport methods currently available in the market, which support authentication and access security for the MIH messages. Figure 4 shows the integration of the security framework in the existing MIH framework.

2.2.1. IPSec/IKEv2. IPSec [5] provides a standard mechanism for data security for protocols running over IP. Since the MIH messages (in the prototype implementation of MIHF) use UDP over IPv6 for transport, IPSec can be an automatic choice for message protection. However, since IPSec needs a preconfigured trust relationship between the communicating end points, the feasibility and efficiency of this method needs to be examined in the context of handover to different access networks.

Figure 5 shows the messages exchanged between MIH enabled nodes, to setup the IPSec tunnel using IKEv2.

2.2.2. Datagram Transport Layer Security. The DTLS [6] protocol provides communication privacy for datagram protocols. It is designed to run in the application space, without requiring any kernel modifications. The basic design philosophy of DTLS is to construct "TLS over datagram." The reason that TLS cannot be used directly in datagram environments is simply that packets may be lost or reordered. TLS has no internal facilities to handle this kind of unreliability, and therefore TLS can break when hosted on datagram transport. The purpose of DTLS is to make only the minimal changes to TLS required to fix this problem. To the greatest extent possible, DTLS is identical to TLS.

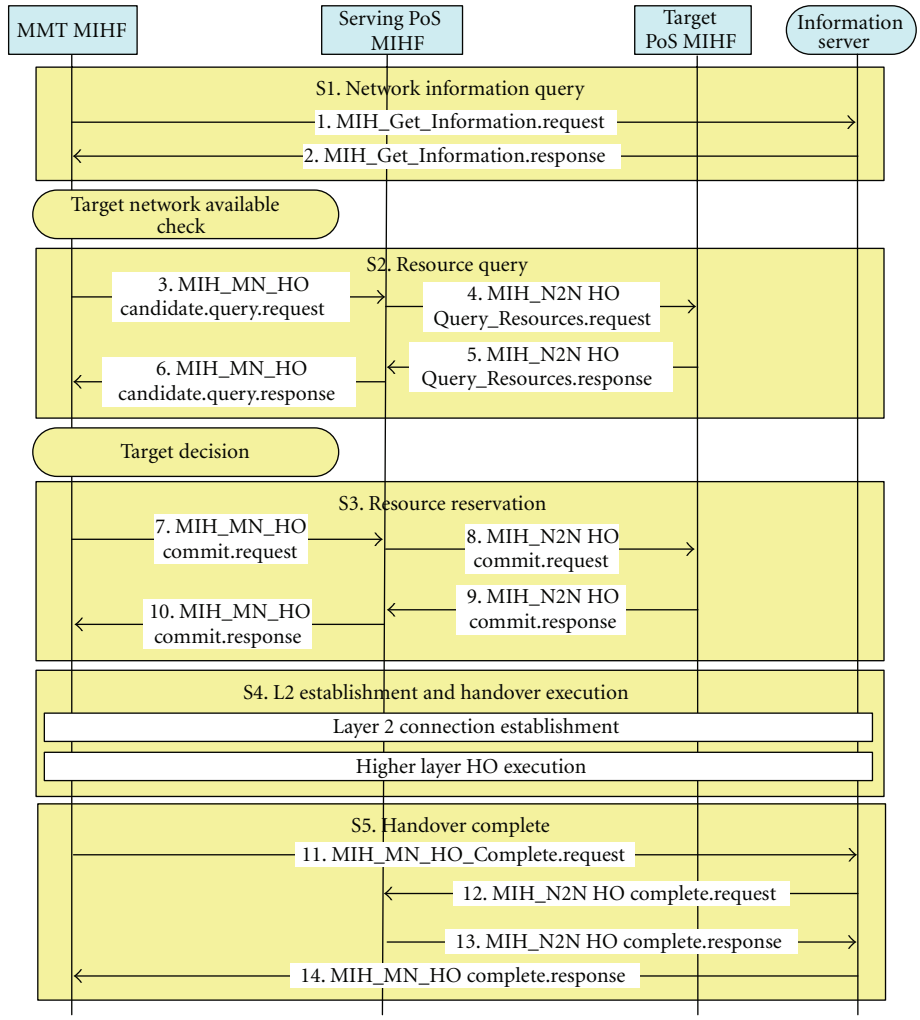


Figure 3: MIH-based handover—call flow.

Figure 6 shows the DTLS protocol messages exchanged between client and server for establishing a DTLS association.

2.2.3. New MIHSec Transport Method. In the above section, the current secure transport methods like IPSec and DTLS are discussed. In contrast to these two methods, a new method known as MIHSec is proposed in this paper. MIHSec provides solutions to the problems that arise in using IPSec and DTLS for MIH-based handover applications. The details of the problems and solutions are presented in the subsequent sections.

3. Secure MIH Transport Models

This paper discusses two secure transport models that are commonly used in general security architectures [7] like IPSec.

The end-to-end security model provides protection to the messages on an end-to-end basis; that is, packets encrypted at source is decrypted at the end point. And the other model is the end point-to-security gateway model, wherein packets are encrypted between the endpoint and the gateway, which is to say that the packets should be encrypted/decrypted multiple times on its transmission to the destination node. Elaborate descriptions of these two models, when applied to the MIH solution, are given in the subsequent paragraphs.

3.1. End-to-End Protection. In this model, a secure channel is established from the MMT to each MIH service end-point in the network, before any MIH message exchange can take place. The secure channel source is MMT and the destination is Interworking Function (IWF), MIH Information Service (IS) server, and PoS. IWF provides the proprietary function between MIH services and a specific access network. This is out of the scope of IEEE 802.21.

The secured path shall provide data integrity, authenticity, and confidentiality as desired. The MIH on MMT will be responsible for setting up and terminating the secure channel. An encrypted packet sent from MMT can be decrypted at IWF, IS server, and PoS only. Other than the

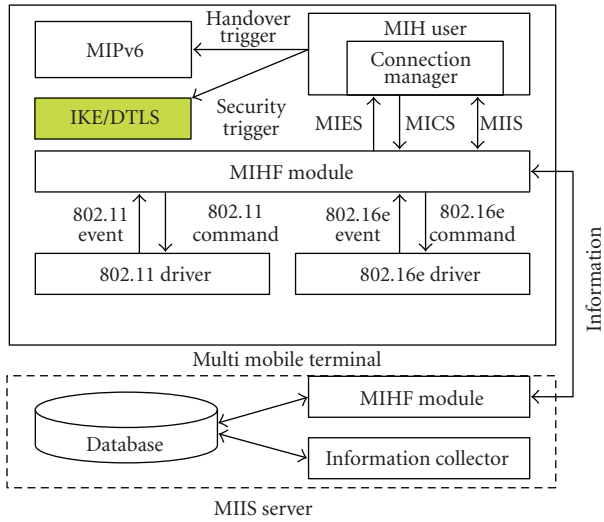


Figure 4: Secure transport module in MIH framework.

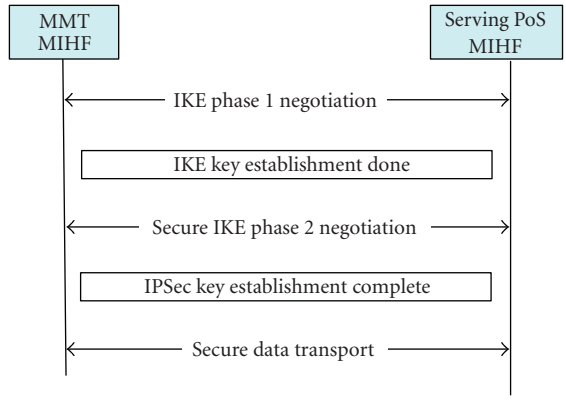


Figure 5: IPSec tunnel establishment.

destination node, the nodes on the path cannot decrypt the packet. This model provides security between the nodes that are residing in the end point of the transmission paths.

For example, during handover to a new access network, the MIH entity in MMT should trigger the IKEv2 daemon to establish an IPSec security association (SA) with MIH PoS for MIH command and event service in the new access network, before sending the MIH-MN-HO-Complete message. It should also establish IPSec SA with the MIH IS server in the same way, before sending any MIH_Get_Information request message to the IS server. Similarly, a secure channel has to be established between MMT and IWF Proxy before transmitting any packet between the MMT and IWF Proxy nodes. The tunnel between MMT and AR is identified as T2, the tunnel between MMT and IWF Proxy is identified as T1, and the tunnel between MMT and IS server is identified as T3. This is illustrated in Figure 7.

3.2. Endpoint-to-Security Gateway Protection. In this model, a secure channel is established from the MMT to the AR in the access network, before any MIH message exchange

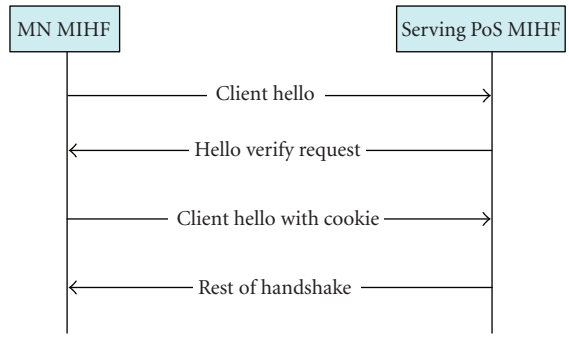


Figure 6: DTLS client server message exchange.

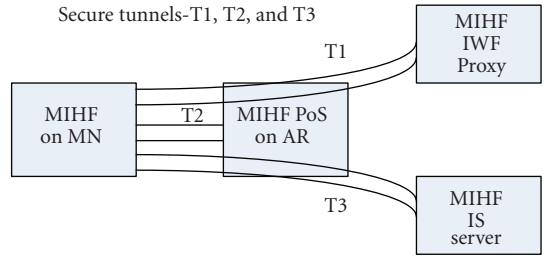


Figure 7: MIH message security through end-to-end tunnels.

can take place between MMT and AR. The source is the MMT and the destination is AR. And similarly when the packet is sent from AR, the source is AR and the destination is the MMT. The secured path shall provide data integrity, authenticity, and confidentiality as desired. The MIH on MMT will be responsible for setting up and terminating the secure channel with the AR. The AR will be responsible for establishing a secure channel between itself and each MIH node in the network, like IWF Proxy or IS server.

For example, during handover to a new access network, the MIH entity in MMT should trigger the IKEv2 daemon to establish an IPSec SA with the new AR, before sending the MIH_MN_HO_Complete Message. Establishment of a secure channel is done before transmitting any MIH packet.

In this method, the destination end point may or may not be the logical end point of the tunnel. For example, when MMT sends an MIH_Get_Information request message to the IS server, the packet traverses through tunnel T1 and tunnel T3 to reach the destination—IS server. As shown in Figure 8, the tunnel between MMT and AR is known as T1, the tunnel between AR and IS server is T3, and the tunnel between AR and IWF Proxy is T2.

The analysis in this paper focuses on security through end-to-end tunnels, as illustrated in Figure 7, and the experimental results are based on that model only. However, similar results are expected in the endpoint to gateway tunnel method also, as illustrated in Figure 8.

The endpoint to gateway approach would have an advantage when it is assumed that the secure channel T1 is not required as this path will be protected by L2 security. In such a case the overhead of security will be avoided in the wireless link.

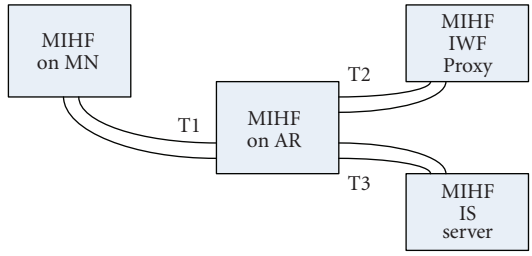


Figure 8: MIH message security through endpoint to gateway tunnels.

Hence in this paper, the endpoint to endpoint tunnel method is considered.

4. Analysis of Secure MIH Message Transport with Existing Solutions

4.1. Requirement of Secure MIH Message Transport. The MIH-enabled nodes in the network have the capability to handle the Event Service (ES), Command Service (CS), and Information Service (IS) requests. These service messages carry manifold information, which is helpful to the decision process in MIHF to perform the handover functionality in the network and node elements.

The MIH messages are transmitted over the Internet between the MIH enabled access node, the IS server, and IWF proxy. For MMTs these messages are sent over the wireless network and the wired infrastructure that make up the access domain.

As an MIH message is transmitted over insecure channels on its path to the destination, it becomes an obligation to secure these messages from hackers who are trying to hijack the channels, spoof the packets, or snoop in the network.

This section discusses the list of security features that are required to be incorporated in the MIH messages.

4.1.1. MIH Access Control. Based on policies, an MIH PoS in the operator network may want to allow only certain MIH services to the MIH entity in the MMT. The access control can be enforced through IPSec/IKEv2, DTLS or by defining new information elements as a part of the MIH protocol.

4.1.2. MIH Replay Protection and Denial of Service. MIH packets may be spoofed or packets may be replayed by an attacker. By using IPSec SA or DTLS session for all MIH message exchanges, these attacks can be prevented. An MIH protocol level method may also be considered for protection against this attack by including timestamp/sequence number in the MIH messages.

4.1.3. MIH Data Integrity and Confidentiality. MIH data integrity and confidentiality can be achieved through IPSec and DTLS. A sufficiently strong encryption and integrity algorithm, for example, aes-cbc/256-bit and hmac-sha1/128-bit, can be negotiated between MIH peers during IKEv2 [8] signaling or DTLS handshake to ensure protection.

An MIH protocol-based approach can be used for message integrity. For example, a message authentication code information element may be included in each MIH message, which needs to be protected for data integrity.

All three methods for MIH message protection are analyzed in this paper to identify the scope of prototyping and experimentation. Based on the prototyping and experimentation results, the IPSec, DTLS, and new MIHSec methods will be evaluated for ease of configuration, efficiency, and handover latency.

4.2. Methods of Securing MIH Message Transport with Existing Solutions

4.2.1. IPSec/IKEv2. In Figure 9, MIHF will trigger the IKEv2 daemon to establish an IPSec SA with the MIH endpoint before any MIH message exchange can take place.

Each MIH end-point shall perform the following steps:

(1) get X.509 Certificate from a trusted certificate authority (CA) by supplying the MIHF ID,

(2) install the CA certificate and the host certificate,

(3) exchange the credentials with the other MIHF end point and verify the other end-point's certificate and MIHF ID,

(4) update the IPSec policy database (SPD) and IPSec association database (SAD) for protection of MIH Message (UDP/MIH_PORT) sent to and received from the other MIH endpoint.

The credentials are exchanged and verified by the IKEv2 daemon in IKE_SA_INIT and IKE_SA_AUTH. This method requires that the MIHF endpoints know the MIHF ID of the other MIH end point. How the MIHF IDs of MIH PoS in the target network are obtained is the topic of "MIHF Discovery Analysis". Table 1 lists various scenarios in this regard and the possible ways to get the MIHF ID.

(a) IPSec/IKEv2 Pros and Cons.

Pros has the following.

(1) IPSec provides the most standard solution for data security for protocols running over IP. Even the IP header can be protected by using IPSec in tunnel mode.

(2) IPSec support is readily available in all standard operating systems.

(3) Using IKEv2, security keys can be configured automatically.

(4) Using IKEv2 with EAP allows the security credentials to be verified by the authentication, authorization, and accounting (AAA) server for the access network.

Cons has the following.

(1) IKEv2 signaling adds to latency in handover.

(2) IPSec header adds overhead to packets send over the air interface.

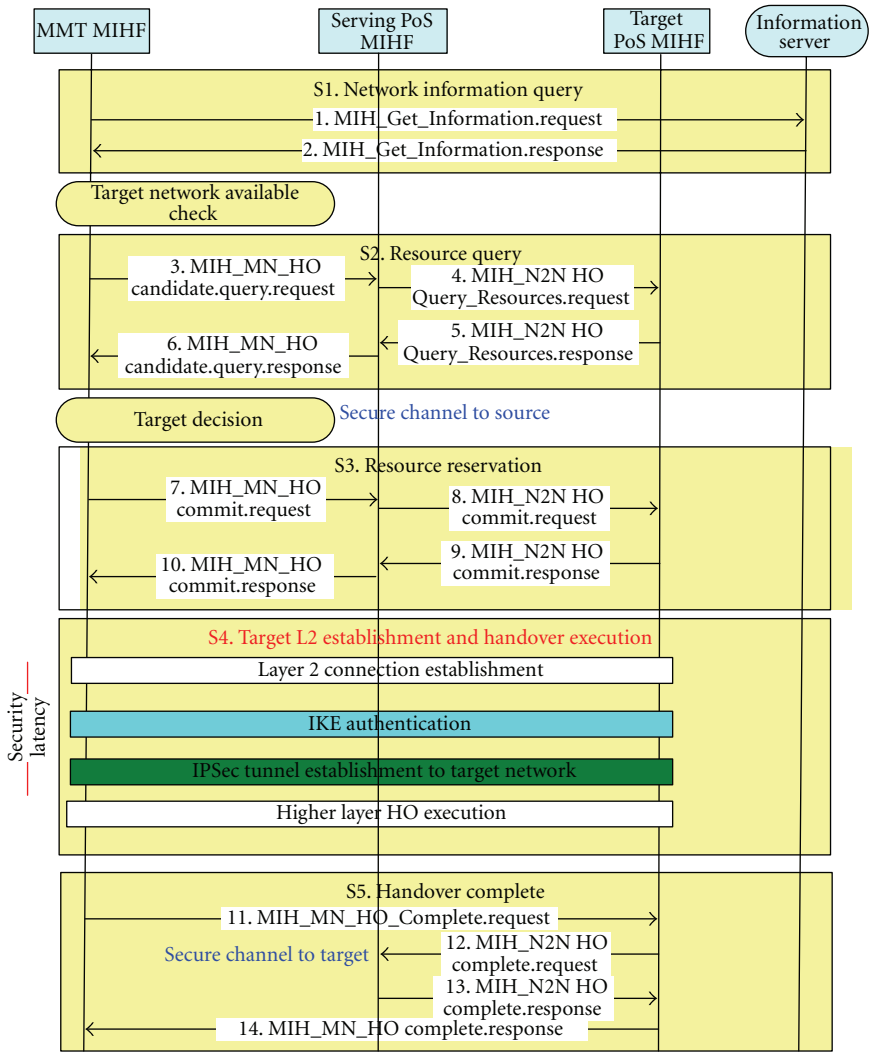


Figure 9: Securing MIH with IPSec.

(3) IPSec ciphering algorithm execution adds to latency in handover.

(4) Integration of MIH with IKE is an issue with handover as IP address changes in MMT.

4.2.2. Datagram Transport Layer Security. In Figure 10, DTLS is used for secure MIH transport, which uses all of the same handshake messages and flows as TLS, with three principal changes:

(1) a stateless cookie exchange has been added to prevent denial of service attacks,

(2) modifications to the handshake header to handle message loss, reordering, and fragmentation,

(3) retransmission timers to handle message loss.

(a) DTLS Pros and Cons.

Pros has the following.

(1) DTLS is an application layer protocol.

(2) No kernel modification is required.

(3) It does not depend on any underlying reliable transport protocol.

(4) It can be implemented with lesser modification of existing TLS.

(5) It is closer to functionalities of IPSec but cheaper.

Cons has the following.

(1) DTLS signaling which involves multiple handshake messages between client and server adds to latency.

(2) DTLS is not independent protocol. DTLS will internally use TLS library. So TLS library support is required.

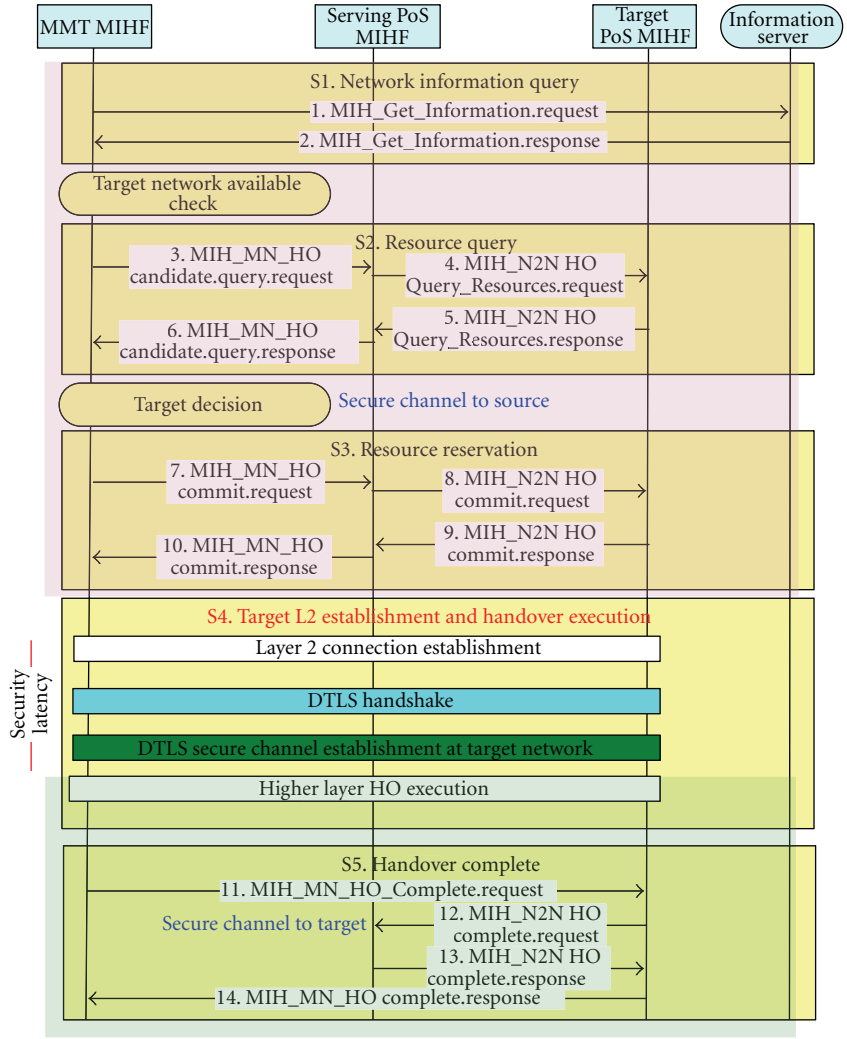


Figure 10: Securing MIH with DTLS.

Table 1: Methods for getting MIHF ID's.

MIHF Host	Scenario	Solution
MIHF on MMT	To get PAR MIHF ID during start up	MIHF Discovery methods. Listen to MIHF Capability Discover Broadcast
MIHF on MMT	To get NAR MIHF ID during HO	MIHF Discovery methods (DHCP/DNS). Listen to MIHF Capability Discover Broadcast
MIHF on MMT	To get IS server MIHF ID	MIHF Discovery methods (DHCP/DNS)
MIHF on PAR	To get NAR MIHF ID	Listen to MIHF Capability Discover Broadcast
MIHF on PAR	To get IS server MIHF ID	MIHF Discovery methods (DHCP/DNS)

5. Method for Securing MIH Messages with Protocol Extensions to MIH (MIHSec)

5.1. Motivation for a New Secure MIH Messages Transport Protocol. In the previous sections we discussed IPSec and DTLS solution to provide security to the MIH messages. The IPSec operates at IP layer and the DTLS at the application layer to provide security to the MIH messages.

The IPSec and DTLS could suffice the requirements for providing security to the MIH messages. The steps carried out to provide secure transmission of MIH messages are provided in Figure 11.

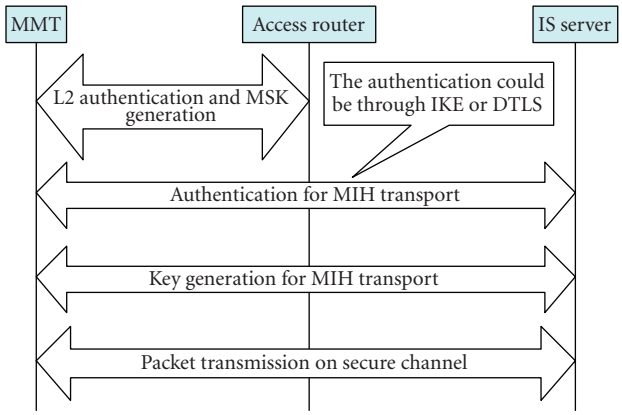


Figure 11: IPSec/DTLS key generations at IS server.

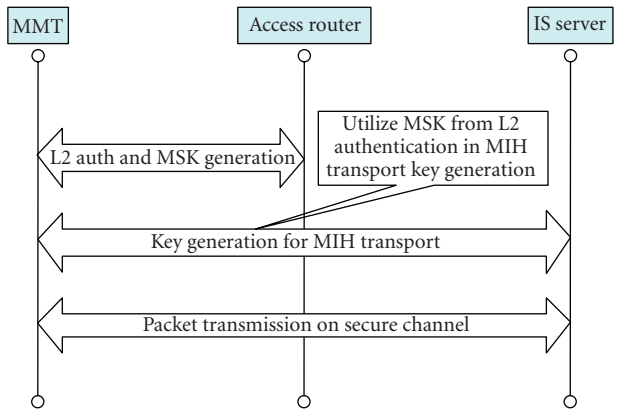


Figure 12: MIH transport key generation at IS server using L2 authentication MSK Key.

The L2 authentication is performed between the MMT and AR. This provides a secure communication channel on the air interface between MMT and AR.

The MIH Transport Authentication—which can be IKEv2 or DTLS—is carried out next to authenticate MMT with the MIH network entity. In Figure 11, IS server is considered as an MIH entity, for example, illustration. Upon completion of the authentication with the IS server, the MIH IK and the CK keys are generated. These keys are used by the MIH layer to provide the secure communication channel between the MMT and IS server.

The inherent problem with IPSec/DTLS security method is multiple authentications (L2 authentication and Authentication for MIH Transport) that occur in the flow. The additional MIH transport authentication would add to the latency during the handover, which in turn degrades the performance of handover. If MIH transport authentication can be eliminated, the handover latency time will be minimized. This section discusses basic idea to provide the MIH Security at the application layer by providing enhancements to the 802.21 standard.

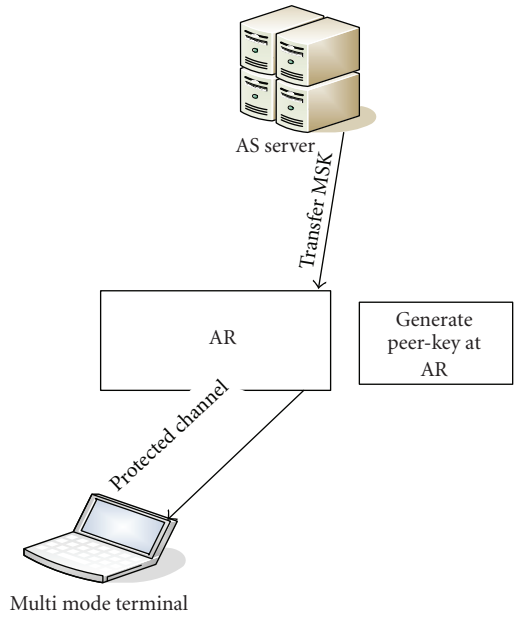


Figure 13: Generation of peer-key.

5.2. Enhancements to 802.21 to Support MIH Security (MIHSec)

5.2.1. The Concept of MIHSec. The inherent disadvantages of DTLS and IPSec in the handover scenarios would support the need for developing a new integrated security feature in MIH messages. The important requirement is minimization of handover latency and support of confidentiality and integrity protection to the MIH messages.

The idea here is to eliminate the MIH transport authentication and utilize the Master Shared Key generated by the L2 authentication procedure, for generating the MIH keys. Avoiding MIH transport authentication step would enhance the handover latency and hence better performance during the handover as shown in Figure 12.

The solution that is proposed here would utilize the authentication provided at the L2 layer. In most of the access networks, available in today's market, the authentication is provided by using the EAP standard.

MIH protocol would utilize the MSK generated by the EAP, to generate its own CK and IK. The advantage of using MSK of L2 authentication is (a) low latency and (b) maintenance of key hierarchy—in security parlance, its also known as perfect forward secrecy.

Upon completion of L2 authentication, MSK is sent to AR in the Access Network. The AR sends the MSK and MAC address of the MMT to the IS server.

The MSK is utilized by MIHF in AR to generate a Peer-Key in MIHF node in AR. And also the MSK is utilized by MIHF in IS server to generate IS-Key.

To summarize, between the MMT and AR nodes, Peer-Key is generated and between MMT and IS server nodes IS-Key is generated. Peer-Key is the key hierarchy between MMT and AR and IS-Key is the key hierarchy between MMT and IS server.

5.2.2. MIH Key Generation Procedure. In Figure 13, the multimode mobile terminal performs authentication with the access network. This is done using the EAP protocol. The result of the authentication is the generation of the MSK key. The peer MIH function in AR uses the MSK key, along with other parameters to generate a Peer-Key. The algorithm for generating the keys is described in the following section.

(a) Algorithm for Security Keys Generation between Mobile Terminal and PoA. The Peer-Key is used to establish secure channel between MMT and PoA. The pseudocode for generating the security keys is described as follows:

Algorithm 1. Key_generation_algorithm_in_MIHPeer().

Begin:

Get the MSK key of EAP

Use the keyed-md5 as Pseudo Random Function for generating the Peer-Key

Peer-Key = Keyed-md5(MSK, MAC-Peer, MAC-PoA)

// The inputs to the prf are MAC address of MMT and MAC address of PoA

The result of keyed-md5 is Peer-Key

Peer-Key is a 128 bit hash value

Use Peer-Key to generate the CK and IK

Cipher Key = prf(Peer-Key, "Peer", 0)

Integrity Key = prf(Peer-Key, "Peer", 1)

// The 0 and 1 in the prf function indicate whether the key generated is the CK or the IK

End:

CK(Ciphering Key) and IK(Integrity Key) generated are used to secure the MIH Data, along with the MIH headers

(b) Algorithm for Generating Security Keys between MMT and IS Server. IS-Key is used to establish secure channel between the mobile terminal and the IS server. The algorithm for generating security keys between IS server and MMT is mentioned here in after.

The pseudo code for generating security keys is described as follows:

Algorithm 2. Key_generation_algorithm_in_MIHServer().

Begin:

Get the MSK key of EAP

Use the keyed-md5 as Pseudo Random Function for generating the Peer-Key

IS-Key = Keyed-md5(MSK, ISServer-IPAddress, MAC-Peer)

// The inputs to the prf are IP Address of the IS server and MAC address of MMT

The result of keyed-md5 is IS-Key

Peer-Key is a 128 bit hash value

Use IS-Key to generate the CK and IKs between the MMT and the IS server

Cipher Key = prf(IS-Key, "IS-Server", 0)

Integrity Key = prf(IS-Key, "IS-Server", 1)

// The 0 and 1 in the prf function indicate whether the key generated is the CK or the IK

End:

5.2.3. Extensions to MIH Header. IP Security operates at IP layer. An extension to the IP header has been provided to incorporate security features in IP. Similarly there is a need to provide security extension headers to the current MIH standard for providing security features in 802.21. The objective of these extension headers is to carry message digest between tunnel end points, to enable the end points to validate the packet data and header information.

In order to support security at the MIH, extensions need to be provided at MIH Header as illustrated in Figure 14. This is due to the fact that the MIH layer at the destination has to identify if the MIH packet is security protected or not. Hence, two new TLVs are added to support the security feature in MIH. An encryption TLV and integrity TLV are provided as an extension for MIHSec. The illustration of the same is provided in Figure 11.

And as illustrated in Figure 15, encryption is provided over MIH data and confidentiality is provided over MIH header and MIH data.

When a secure MIH packet is to be transmitted from MMT to IS server, MIHF in MMT performs confidentiality protection first and then applies integrity protection on header and data. At the destination node, the MIHF in the IS server performs integrity checking initially and if the integrity check is passed, confidentiality check is done. If either of integrity check or the confidentiality check fails, that packet is dropped.

Integrity protection checking is done first, before performing the deciphering functionality.

5.2.4. Benefits of MIH Security Solution.

(i) A separate authentication mechanism (like IKE authentication or DTLS authentication) is not necessary as the MSK keys from the L2 authentication are utilized in maintaining key hierarchy and also for generating the MIH CK and IK keys.

(ii) The handover latency is minimized due to elimination of IKE/DTLS authentication procedure.

(iii) Changes to the MIH code are minimal to support confidentiality and integrity protection and hence the ease of integration with the present code.

(iv) Available PRF algorithms can be reused.

(v) The last one is the protection against Denial of Service.

MIH-type (confidentiality/integrity)	MIH length	MIH value (128 bit cipher or 128 bit hash)

Figure 14: MIH extension header.

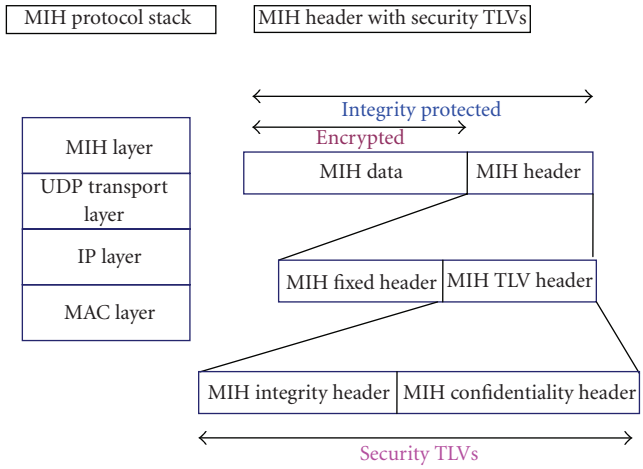


Figure 15: MIH with security TLV.

5.3. Performance Evaluation Parameters

5.3.1. Security Signaling Latency. Security signaling latency is defined as time taken to perform the authentication and security key generation, along with the tunnel establishment time:

$$\begin{aligned}\text{Security Signaling Latency} = {} & \text{Authentication Time} \\ & + \text{Key generation Time} \\ & + \text{Tunnel Establishment Time.}\end{aligned} \tag{1}$$

The authentication time is the time taken to authenticate the MIHF-enabled network entity. Key generation time is the time taken to generate the CK and IK keys from the MSK. Tunnel establishment time is the time taken to populate the IKs, CKs, and MIHF entity MAC address information in the table.

5.3.2. Message Transport Latency. Message transport latency is defined as the time taken to apply the integrity protection or confidentiality protection on the MIH packet that is exchanged between the MIHF entities:

$$\begin{aligned}&\text{Message Transport Latency} \\ &\quad = \text{Time taken to apply protection to MIH packet.}\end{aligned} \tag{2}$$

5.3.3. Message Overhead. Message overhead is the amount of additional information that has to be carried in the MIH packet to carry the message digest. The message digest is carried as a part of TLV in the MIH packet.

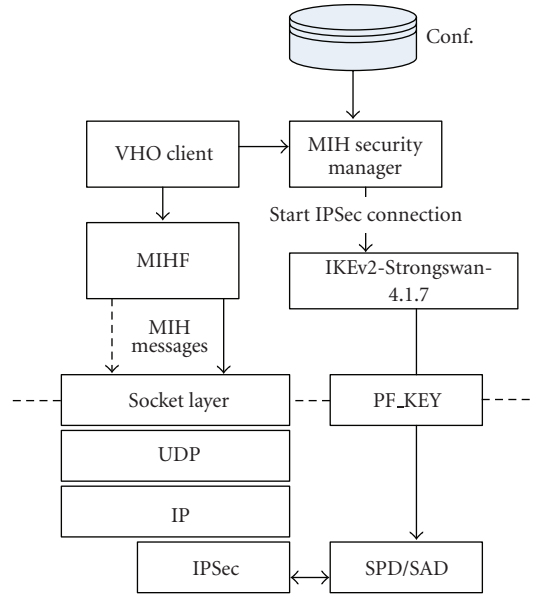


Figure 16: System software architecture in MMT and AR with MIHF and IPSec entities.

6. Prototype Implementation

6.1. Software Architecture for IPSec/IKEv2. Figure 16 shows system software architecture in MMT and AR with MIHF and IPSec functions integrated. The following entities are added to the MIHF/VHO-Client implementation.

Security Configuration Settings. MIHF shall be configured manually to use appropriate security methods (IPSec-IKEv1/v2, encryption/authentication algorithms, etc.).

MIHF Security Manager. The MIHF security manager module shall read the security settings from the configuration file.

It will generate the connection settings (/etc/ipsec.d/mihfsec.conf) dynamically for the new MIHF peer with which the IPSec SA need to be established, reload the settings in IKEv2 daemon, and trigger the IKEv2 daemon to establish IPSec SA with target MIHF peer.

Openssl. The IPSec modules in this solution use the openssl library version 0.9.8 g [9].

The prototype implementation is tested with different security algorithms for encryption and integrity check to measure the latency in handover due to IKEv2 signaling messages as well as MIH message transaction delay added by the security algorithms.

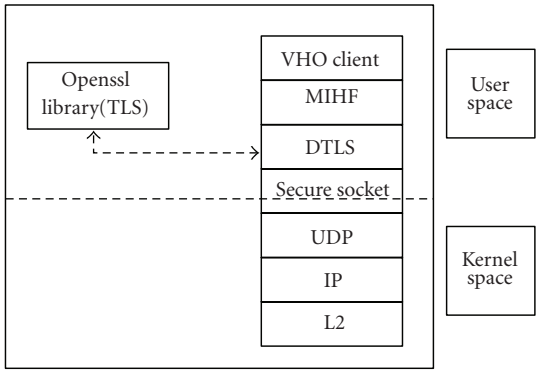


Figure 17: System software architecture in MMT/AR with MIHF and DTLS entities.

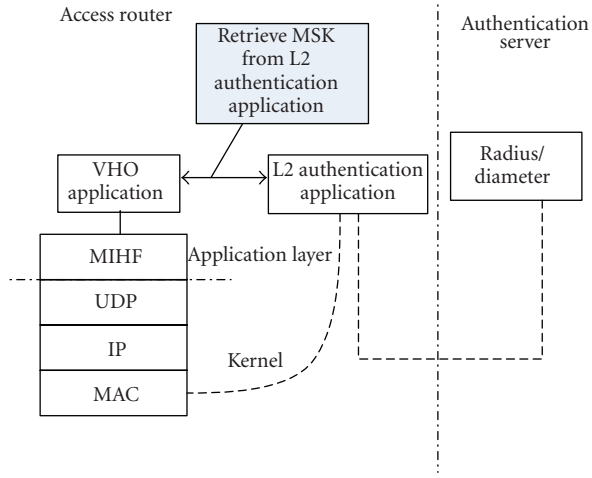


Figure 18: Software architecture for MIHSec.

6.2. Software Architecture for DTLS. Figure 17 shows the system software architecture in MMT and AR with MIHF and DTLS functions being integrated. The following entities are added to the MIHF/VHO Client implementation.

Security Configuration Settings. MIHF shall be configured manually to use appropriate security methods (DTLS, encryption/authentication algorithms, etc.).

DTLS. This layer is responsible for enforcing MIH message-transport security. This module creates a DTLS client socket for initiating MIH message exchange with MIH peers and a DTLS server socket which listens to MIHF message from MIH peers. DTLS connection will be established between the peer sockets before any MIH exchange can take place.

Secure Socket Layer. This is implemented using openssl 0.9.8 g library

The DTLS client initiates the communication by sending HELLO SERVER packet by using *SSL_write* API. This initiates the DTLS handshake message sequence, where the messages are processed by the Openssl library. The DTLS client and server authenticate each other, negotiate the algorithms for encryption and integrity, and install the security keys.

Asymmetric key cryptography with RSA (Rivest, Shamir, and Adleman) algorithm is used for authentication between the peer entities.

The client MIH peer sends the all MIH request messages through *SSL_Write* API, which results in the message to be encrypted with the established security key and sent to the server. The server MIH peer decrypts the data and sends to the SSL_read API for passing the message to the MIHF/VHO-Client module.

The prototype implementation is tested with different security algorithms for encryption and integrity check to measure the latency in handover due to DTLS signaling messages as well as MIH message transport delay due to security algorithm processing.

6.3. Software Architecture for MIHSec. Figure 18 shows the software architecture for MIHSec. An AR example is considered to elaborate the architecture concepts. The same design would apply to the MMT also.

MIHF is a media independent handover function. It operates at application layer and interfaces with UDP layer in the kernel, VHO application at the user land space. MIHF handles the event service, command service, and information service messages.

VHO application interacts with MIHF in AR and authentication application. The job of VHO application is to make handover decisions and to maintain the key hierarchy. The key hierarchy is utilized to generate the CKs and IKs for the MIH sessions. The IK and CK are maintained in the table, which is indexed by MAC address of the Peer. The MAC address of the peer acts as a security parameter index for the secure channel.

The changes required to VHO application code for incorporating the MIH security are minimal and hence ease of integration with the current MIH code for providing security enhancements.

A brief patch for the MIH security is provided as follows:

Note that the patch is shown in italics font and current code in regular font.

```
Vho_application_main()
Being:
    New MMT has made an attach with the AR
    Receive MSK from authentication application
    Generate Peer_Key and IS_Key
    Generate CK and IK keys from the key hierarchy
    Maintain the keys information as shown in Figure 20
    ...
    ...
     Decision Process
    ...
```

```
    ...
    etc
End

MIH_LookUp_in AR()
Being:
    Handle the received packet
    Extract MMT-MAC from MIH TLV
    Index into the Key Table (Figure 20) based on Peer-MAC
    Extract CK and IK
    Perform Integrity Check to the packet
    If the integrity check fails, drop the packet
    Else perform the Ciphering Check
    If the ciphering check fails, drop the packet
    Else
    ...
    Perform the normal operations
    ...
    ...
End

MIH_LookUp_In_MMT()
Being:
    Handle the received packet
    Extract MMT-MAC and AR-MAC from MIH TLV
    Index into the Key Table based on Peer-MAC
    Extract CK and IK
    Perform Integrity Check to the packet
    If the integrity check fails, drop the packet
    Else perform the Ciphering Check
    If the ciphering check fails, drop the packet
    Else
    ...
    Perform the normal operations
    ...
    ...
End

MIH_Secure_Packet_Transmission()
Being:
    Decide on packets to be transmitted
    Check the Security YES/NO Flag. If the flag value is
NO, transmit the normal MIH the packet (It implies
that Security is not mandatory) else
    Index into the key table using ID as identifier to
retrieve the IK and CK keys
        (The ID here is MIH Identifier. EAP uses this
identifier in it's initial messages for identifying itself
with the peer)
        Perform Confidentiality protection on MIH Data
        Perform Integrity Protection on MIH Data and
MIH
        Headers (leaving MIH-Integrity TLV header, but
including MIH-Encryption TLV header)
    Transmit the security protected packet
End
```

Table 2: TEST configurations.

Security Methods	Settings
IPSec/IKEv2	(1) ESP/Transport, - ENC=3des-cbc/192-bit, AUTH=hmac-md5/128-bit - ESP/Transport, ENC=aes-cbc/256-bit, AUTH=hmac-sha1/128-bit (2) IKEv2 Settings: - Strongswan 1.4.7 daemon [10] - X.509 certificates with RSA (1024-bit private key)
DTLS	- Openssl 0.9.8g library [11] - X.509 certificates with RSA (1024-bit private key)
MIHSec	- EAP Protocol for Authentication - Extensions to 802.21 to support MIHSec in MIHF

The security keys are maintained as shown in Figure 19.

On receiving the MSK from authentication application, this table is configured by VHO application.

A provision could be provided to configure this table manually. However, at present, this option is not being considered and could be investigated later.

7. Experimental Results

7.1. Test Environment. Figure 20 illustrates the test environment used for testing the prototype implementations with the test configuration in Table 2.

7.2. Test Settings for Security Methods. The IPSec/IKEv2 connection settings for PAR (and IS server) are statically configured, while the connection settings for NAR are dynamically generated.

EAP stack integration with MIHF is performed to enable EAP to carry MIHF identifier as EAP identifier. MIHF is extended to support security headers.

ID of MN1	MAC of MN1	Integrity key 1	Cipher key 1	Peer_Key-1 IS_Key-1	Security (Yes/No) flag
ID of MN2	MAC of MN2	Integrity key 2	Cipher key 2	Peer_Key-2 IS_Key-2	Security (Yes/No) flag
ID of MN3	MAC of MN3	Integrity key 3	Cipher key 3	Peer_Key-3 IS_Key-3	Security (Yes/No) flag

Figure 19: Format of security keys table in AR.

Table 3: Result analysis and comparison summary.

Parameters	Security method		
	IPSec/IKEv2	DTLS	MIHSec
Security signaling latency	High (100 s of msec)	Moderate (10 s of msec)	Low (1 s of msec)
Message transport latency	Negligible	Negligible	Negligible
Configuration and setup	Difficult	Moderate	Easy
Message overhead per MIH exchange	Less than 25%	Above 25%	Less than 25%

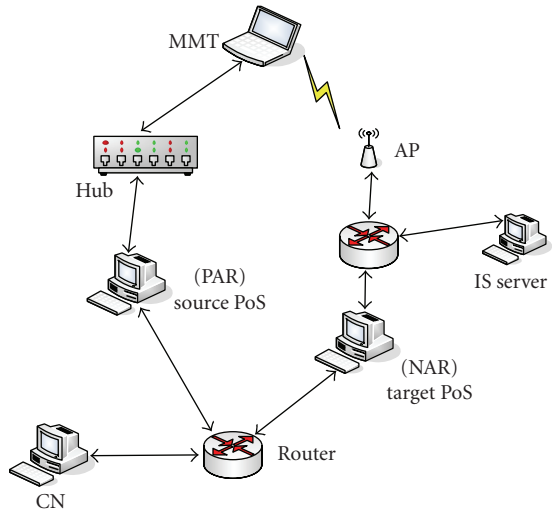


Figure 20: Environment for MIH security prototype testing.

7.3. Result Analysis. Test results are analyzed and compared with respect to Signaling latency, Message transport latency, Message overhead, Configurations, and setup. The results are shown in Table 3.

7.3.1. Security Signaling Latency. In the test setup we used, it is found that IPSec/IKEv2 takes about 230 milliseconds for IKEv2 signaling involving 2 IKE_SA_INIT messages and 2 IKE_AUTH messages and key generation /installation.

In the DTLS method, only about 30 milliseconds are taken for signaling and installation of security keys.

In the MIHSec method, since the authentication is directly integrated with L2 authentication, it leads to an efficient security signaling time.

7.3.2. Message Transport Latency. Our experimental results showed that IPSec transforms (Encryption and Decryption) do not add much confidentiality latency to MIH message exchange. In this experiment we used general purpose machines with security algorithms implemented in software. In more practical scenarios, sophisticated hardware will be used for implementing security algorithms and then the latency will be negligible.

Our experimental results showed that DTLS transforms (Encryption and Decryption) do not add much confidentiality latency to MIH message exchange.

The latency of the message transport in MIHSec is comparable to the IPSec and DTLS latency times.

7.3.3. Message Overhead. To an MIH message exchange (Request and Response), about 70 bytes are added as overhead in 3des-cbc/192-bits and about 90 bytes are added as overhead in the case aes-cbc/256-bit. This is applicable in both IPSec and MIHSec case.

To an MIH message exchange, about 100 bytes are added as overhead in the case of DTLS based message protection.

7.3.4. Configuration and Setup. MIHF configuration for IPSec/IKEv2 is fairly complex. This is due to the fact that the IKE is inherently a key authentication protocol with complex configurations, and which expects peer to configure the security information in advance. In addition when the handovers are performed, with the changes in IP address to the mobile terminal, configuration becomes a challenging task in IPSec/IKEv2. When end-to-end secure tunnels are used (as in this experimented), the MIHF should be configured to establish IPSec SA with each end-point. Manual configuration of this is impractical.

Also the IPSec support is required in the kernel.

DTLS is an Application Layer protocol and the DTLS Client/Server requires lesser configuration effort.

However, the use of the following approach will simplify configuration process.

(1) Use MIH Discovery method for automatically discovering the target MIH endpoint.

(2) Use MIH ID as the unique identifier to generate X.509.

Configurations that are made for the L2 security should be sufficient for the MIHSec. No additional configurations are required for MIHSec, hence simplifying configuration operations when compared to DTLS or IKEv2/IPSec.

8. Conclusion

This paper analyses different security methods which could be used for MIH message protection. Prototype of MIH security methods with IPSEc/IKEv2, DTLS, and MIHSec methods are developed and the results are compared. The experiments showed better results in terms of message overhead for MIHSec and IPSec methods compared to DTLS. However in terms of signaling latency, MIHSec showed better results. Also, since the MIH messages are transported over UDP (in this implementation of MIHF), security at transport layer might be sufficient, and hence MIHSec method is a strong candidate. We have presented numerical results to show that 802.21 with MIHSec security extensions provides good handover latency, compared to DTLS and IPSec. This shows that MIHSec is a better solution to support secure MIH message transport.

Acknowledgments

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (KRF-2008-D00273-101053) and V. Leung's work was supported by Grants from TELUS and the Canadian Natural Sciences and Engineering Research Council under Grant CRDPJ 341254-06.

References

[1] LAN MAN Standards Committee of the IEEE Computer Society, Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services, IEEE P802.21/D11.00, May 2008.

[2] E. Gustafsson and A. Jonsson, "Always best connected," *IEEE Wireless Communications*, vol. 10, no. 1, pp. 49–55, 2003.

[3] G. Lampropoulos, A. K. Salkintzis, and N. Passas, "Media-independent handover for seamless service provision in heterogeneous networks," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 64–71, 2008.

[4] A. Dutta, D. Famolari, S. Das, et al., "Media-independent pre-authentication supporting secure interdomain handover optimization," *IEEE Wireless Communications*, vol. 15, no. 2, pp. 55–64, 2008.

[5] S. Kent and K. Seo, "Security architecture for the Internet protocol," RFC 4301, December 2005.

[6] E. Rescorla and N. Modadugu, "Datagram transport layer security," RFC 4347, April 2006.

[7] S. Kent and R. Atkinson, "Security architecture for the Internet protocol," RFC 2401, November 1998.

[8] C. Kaufman, Ed., "Internet key exchange (IKEv2) protocol," RFC 4306, December 2005.

[9] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible authentication protocol (EAP)," RFC 3748, June 2004.

[10] Strongswan, "The OpenSource IPsec-based VPN solution for Linux," http://strongswan.org.

[11] OpenSSL Project, http://www.openssl.org.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 957690, 10 pages
doi:10.1155/2009/957690

Research Article

A Secure and Lightweight Approach for Routing Optimization in Mobile IPv6

Sehwa Song, Hyoung-Kee Choi, and Jung-Yoon Kim

School of Information and Communications Engineering, Sungkyunkwan University, Chunchun-dong 300, Suwon 440-746, South Korea

Correspondence should be addressed to Hyoung-Kee Choi, hkchoi@ece.skku.ac.kr

Received 31 January 2009; Revised 17 April 2009; Accepted 20 May 2009

Recommended by Shuhui Yang

Mobility support is an essential part of IPv6 because we have recently seen sharp increases in the number of mobile users. A security weakness in mobility support has a direct consequence on the security of users because it obscures the distinction between devices and users. Unfortunately, a malicious and unauthenticated message in mobility support may open a security hole for intruders by supplying an easy mean to launch an attack that hijacks an ongoing session to a location chosen by the intruder. In this paper, we show how to thwart such a session hijacking attack by authenticating a suspicious message. Although much research has been directed toward addressing similar problems, we contend that our proposed protocol would outperform other proposals that have been advanced. This claim is based on observations that the proposed protocol has strengths such as light computational load, backward compatibility, and dependable operation. The results of in-depth performance evaluation show that our protocol achieves strong security and at the same time requires minimal computational overhead.

1. Introduction

Mobile networking technologies, along with the proliferation of numerous portable and wireless devices, promise to change people's perceptions of the Internet. Mobility support in Internet Protocol Version 6 (IPv6) is considered particularly important because mobile devices are predicted to account for a significant portion of Internet users during the lifetime of IPv6. Mobile IPv6 (MIPv6) is an IP-layer mobility protocol for the IPv6 Internet [1]. MIPv6 allows an IPv6 mobile node (MN) to change its location on an IPv6 network and still maintain its existing connections to corresponding nodes (CNs).

In Mobile IP, an MN is addressed by two addresses, a home address (HoA) and a care-of address (CoA). An MN has its stationary HoA at its home subnet and changes its temporary CoA whenever visiting a foreign subnet. This dual address mechanism makes it possible to route packets to an MN no matter where it is attached in the Internet. Also, the complex dynamics that occur in the face of sequential handovers are absolutely transparent to transport and higher-layer protocols.

A link between an MN and a CN in Mobile IPv4 (MIPv4) is always detoured via the MN's Home Agent (HA), forming a triangular path [2]. Packets from the CN are routed to the HA and then tunneled, based on CoA, to the MN's location at the time. MIPv6 contained improvements to this rather inefficient routing. The new mechanism, called Route Optimization (RO), requires the MN to update its CoA at the CN whenever the MN changes its point of attachment to the network. The RO in MIPv6 provides an illusion to protocol layers above MIPv6 of continuing to be connected to the MN located at its HoA address. At the same time, the RO rectifies the suboptimal triangular routing by connecting the CN directly to the MN. The MN may choose to inform the CN of its new CoA by using a binding message, thereby allowing the CN to send subsequent packets directly to the MN, bypassing the HA. A binding is the association of the MN's HoA with the CoA for that MN. Unfortunately, malicious and unauthenticated binding messages may open a security hole for intruders by supplying an easy means to launch what are called redirection attacks that hijack an ongoing session to a location chosen by the intruder. IETF's approach [1] to preventing this type of attack is to authenticate the BU

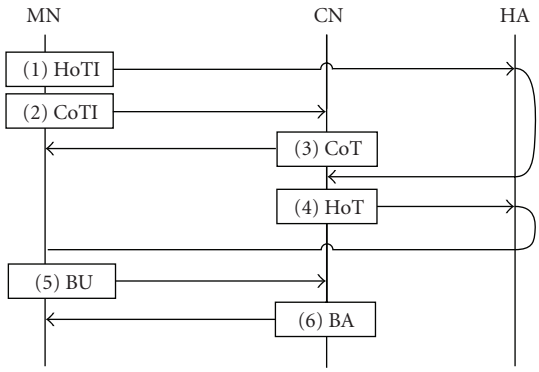


Figure 1: Illustration of secure routing optimization in MIPv6. There are six messages in total. The MN-HA path is securely protected by the IPsec tunnel.

message at the CN and to examine a return path from the CN to the claimed CoA to determine if the address is routable. These two special routines are called Binding Update (BU) and Return Routability (RR), respectively, and we refer to this series of activities as a secure RO in order to emphasize the security aspect in this RO.

In this paper we address the problem of securing the routing optimization. This is a particularly difficult problem because of the following reasons. First, we cannot expect a pre-established secure channel between communicating nodes nor an infrastructure to support secure transactions on behalf of communicating nodes [3]. In addition, the new protocol should be efficient in yielding real-time responsiveness and have a light computational load because delay in the handover greatly affects the quality of service (QoS) in mobile applications. Last but not least, the proposed protocol must be compatible with the legacy protocol to permit a smooth transition.

Our goal in this paper is to take significant steps toward a system that fulfills these criteria. In our protocol the MN creates a secret and sends this secret to the CN twice, once in the direct path to the CN and the other through an indirect path via the HA. The secret is safe from snoopers because it is wrapped in a self-encrypted message. Later, the MN discloses its secret to the public. If the CN can decrypt the MN's early messages with this secret, the CN can confirm the MN's ownership. We evaluated the proposed protocol by comparing its computational expense with five other protocols. The result showed that the proposed protocol was quite efficient and, at the same time, satisfied in a secure manner both ownership and return routability. The objective in this paper is not to explain the cause of network anomalies in the MIPv6. Instead, we seek to demonstrate the utility of new primitives and techniques a future system could exploit for efficient handover.

The paper is organized as follows. Sections 2 and 3 introduce the RO in MIPv6 and discuss related works. Section 4 presents the result of vulnerability analysis. In Section 5, we propose a new secure RO scheme. A performance analysis of the proposed scheme is given in Section 6. Section 7 contains our conclusions.

2. Route Optimization in Mobile IPv6 (MIPv6)

The secure RO in the MIPv6 is composed of six messages and is shown in Figure 1. The first four messages are dedicated to checking the RR of the CoA, and the last two messages are used to authenticate the BU message.

The MN sends the Home Test Init (HoTI) and the Care-of Test Init (CoTI) messages to initiate the binding update, that is, updating the new CoA at the CN. These two messages are sent almost simultaneously but along different paths; the CoTI is sent directly to the CN, and the HoTI is sent indirectly via the HA; (1) are the HoTI and CoTI messages, respectively,

$$\begin{aligned} \text{HoTI} &= \{\text{HoA}, \text{CN}, R_H\}, \\ \text{CoTI} &= \{\text{CoA}, \text{CN}, R_C\}, \end{aligned} \tag{1}$$

R_H and R_C are cookies to match requests with the CN's corresponding responses.

The CN sends the Home Test (HoT) and the Care-of Test (CoT) messages as responses to the previous messages. The HoT and CoT messages are sent, respectively, to the source addresses of the HoTI and CoTI, and follow the same delivery paths as the HoTI and CoTI messages. The HoT and CoT messages are shown, respectively, in

$$\text{HoT} = \{\text{CN}, \text{HoA}, R_H, \text{HT}, i\}, \tag{2}$$

$$\text{CoT} = \{\text{CN}, \text{CoA}, R_C, \text{CT}, j\}. \tag{3}$$

HT and CT are tokens generated by the CN and become a secret key after concatenating these two tokens to authenticate the BU message. HT and CT are shown, respectively, in (4). HT and CT are saved in the CN's hash under the hash indices of i and j. The MN must later return these hash indices in its BU message so that the CN can remain stateless until the BU message is received. These hash indices are included in the HoT and CoT

$$\begin{aligned} \text{HT} &= \text{First64}(H(K_{\text{CN}}, \text{HoA} \| N_i \| 0)), \\ \text{CT} &= \text{First64}\Big(H\Big(K_{\text{CN}}, \text{CoA}\Big\|N_j\Big\|1\Big)\Big). \end{aligned} \tag{4}$$

$H(\cdot)$ is a selected hash function, and First64$(\cdot)$ is a function to choose the first 64 bits in the return string of the hash function. Input to the hash function is the CN's secret key (K_{CN}) and the concatenation of MN's HoA, a nonce value (N_i) and a zero. The generation of CoT is quite similar to the HoT, and extension to the CoT should be straightforward.

The legitimate MN now possesses both tokens and generates a secret key (K_{bm}) as shown in

$$K_{\text{bm}} = H(\text{HT} \| \text{CT}). \tag{5}$$

This marks the end of the RR procedure. The MN may now generate the BU message and is ready to send

$$\text{BU} = \{\text{CoA}, \text{CN}, \text{HoA}, \text{SEQ}, \text{LT}, i, j, \text{MAC}_{\text{BU}}\}. \tag{6}$$

This BU message as shown above is sent from the MN's CoA to the CN. In addition to the CoA, HoA, CN, a sequence number (SEQ), valid lifetime (LT) for this binding update, and the two hash indices are included in the BU message. $\mathrm{MAC}_{\mathrm{BU}}$ is the sign of the BU message using K_{bm}.

On reception of the BU message, the CN recovers K_{bm} from the hash indices included in the BU message and verifies the sign. If the sign proves authentic, the CN accepts the BU message and the MN's CoA by sending an acknowledgment to the MN. The binding acknowledgement (BA) message is shown in

$$\mathrm{BA} = \{\mathrm{CN}, \mathrm{CoA}, \mathrm{SEQ}, \mathrm{LT}, \mathrm{MAC}_{\mathrm{BA}}\}. \tag{7}$$

The security of the RR and BU protocols hinges on the management of HT and CT. Note that no one except the CN can manipulate HT and CT because of the unknown K_{CN}. However, HT and CT are available to anyone in the delivery path because they are delivered in clear text. If an adversary happens to collect a pair of HT and CT in the network, the secure RO is vulnerable to a redirection attack [4].

From a security perspective, the MN's duty as defined in the RFC 3775 is twofold [1]. First, when the MN updates its temporary CoA at the CN, the MN should corroborate to the CN that the CoA is a temporary version of the HoA and that the HoA and CoA are both owned by the MN. The stationary HoA serves as an identifier for the MN. Second, from the perspective of the CN, rather than being informed by the MN that the MN's address has changed to the new CoA, it would be safer for the CN to participate actively in this binding update procedure by confirming the existence and the routability of the MN's CoA. This is very important because a dishonest MN could advertise a fake CoA. The former duty is implemented in the BU, and the latter is accomplished in the RR.

The MIPv6 is an extended version of the IPv6 implemented to support tetherless mobility to nodes but has no role in strengthening the security of the IPv6. Hence, many good security features are excluded from the MIPv6, including authentication. Indeed, authentication to the MN is excluded and furthermore is not necessary in the MIPv6. This is because, first, the security policy in the MIPv6 tries only to maintain a degree of security equal at least to the security of the IPv6 and enforces only authentication of the BU message and the RR. Second, the overhead associated with authentication is too big. Authentication necessitates establishment of a session key for the two nodes, a step that then requires a key management mechanism. Third, at the moment when the MIPv6 starts to work, authentication in the second layer has already been completed. For instance, typical authentication mechanisms in the second layer are Wi-Fi Protected Access2 (WPA2) in 802.11 [5], Privacy and Key Management v2 (PKMv2) in 802.16e [6], and Authentication and Key Agreement (AKA) in Universal Mobile Telecommunications (UMTS) [7]. Additional authentication in the MIPv6 is unnecessary for valid users in the second layer, but nevertheless, the MIPv6 monitors the behavior of these users after authentication.

3. Related Work

One popular approach for a secure RO was to establish a secure relationship between the CN and the MN. The CN first authenticated the MN so as to set up a secure channel and then exchanged useful information over this secure channel. Certificate-based Binding Update (CBU) [8], Hierarchical Certificate-based Binding Update (HCBU) [9], and Leakage-Resilient Security Architecture (LR-AKE) [10] incorporated private key cryptography to establish a secure relationship. Because the MN is authenticated, the CN can trust all messages from the MN. Such attacks as impersonation, message modification, and eavesdropping are quite difficult in the secure channel. As a result, the CN can be sure that CoA is owned by the MN and is reachable. Nonetheless, we contend that the proposed protocol has many advantages over a protocol with private key cryptography as follows.

(1) The certificate management is known to be a big overhead in the operation of asymmetric cryptography. In particular, revoking a certificate and managing the list of revoked certificate are such overheads. The proposed protocol dispenses with the certificate and its management.

(2) The MN and CN may belong to different security domains. In this case interdomain protocol for asymmetric cryptography can be quite subtle, rendering its advantages forfeit. The proposed protocol runs the same irrespective of the domains the both parties belong.

(3) The proposed protocol is quicker than the one with asymmetric cryptography in completing the bind update. This lower delay helps the MN to complete handover quicker. Furthermore, relatively light computations in the proposed protocol extend battery lifetime of mobile devices.

Greg and Michael [11] proposed another secure RO protocol, called the Child-proof Authentication for MIPv6 (CAM), using only a private/public key pair without resorting to certification of public keys. In this approach, the interface identifier of IPv6 addresses is computed from a public key and auxiliary parameters via a cryptographic one-way hash function. The MN uses the corresponding private key to assert address ownership and to sign messages sent from this address without PKI or any other security infrastructure. The binding between the public key and the address at the CN can be verified by recomputing the hash value and by comparing this hash value with the interface identifier. However, the CN cannot confirm return routability to the CoA. Further, the computation load on the MN side is heavy because every BU message requires the MN to generate a signature and the CN to verify it.

The question has been raised of whether private key cryptography is the only approach for a secure BU. Much research has been geared toward developing a secure BU that contains less expensive cryptography. Veigner and Rong [12, 13] proposed a new route optimization protocol for MIPv6

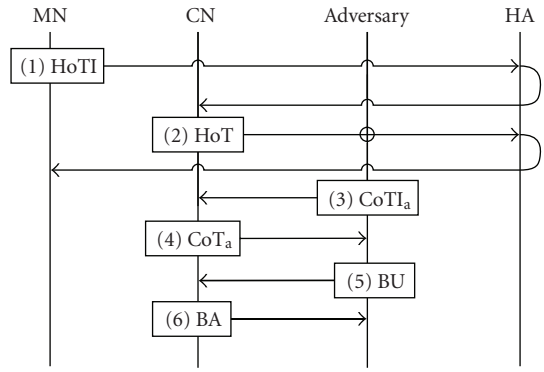


Figure 2: Illustration of a session hijacking attack. Because the (1) and (3) messages are sent independently, the sequence of messages is irrelevant.

(ROM). In their proposal, the MN uses the ROM protocol to assign a unique hash value to its currently used CN. The hash value is sent via the HA-CN path. Simultaneously the home subnet of the MN is authenticated by the CN by means of a three-way handshake. This means that now when it moves into a new subnet, the MN only has to send a BU message directly to the CN. The CN considers the BU message authentic because of the MN's knowledge of the nonce value. This nonce value included in the BU message was previously used when generating the CN's unique hash value. The MN with the paired secret (i.e., a nonce and hash value) first sends the irreversible hash value via an indirect path and has itself authenticated by the CN and then, to assert its ownership of both the HoA and CoA, discloses the nonce value through the direct path. The rather expensive private key cryptography of the approach discussed earlier is replaced by the hash operation. This protocol is similar to our proposed algorithm in its use of a paired secret. Our work complements this earlier work by providing another fully designed routing optimization protocol. However, the main differences between the two protocols include (1) the ROM protocol is not compatible backward with the legacy protocol, and (2) at the end of the ROM protocol, the MN shows the ownership of both the HoA and the CoA addresses but fails to assure the CN that the claimed CoA is routable.

4. Vulnerability of Route Optimization in MIPv6

The goal of a secure RO is to assure the CN that the MN owns the claimed CoA and that this temporary address is reachable in the Internet. Also, the design goal is motivated by the desire to achieve a security level equivalent to that of the IP network without creating major new security problems [14]. Hence, the goal is not to protect against attacks that were already possible before the introduction of IP mobility. Nonetheless, the security protocol in MIPv6 remains vulnerable to a few critical attacks. We discuss the cause and effect of the attacks in further detail in the next section.

4.1. Three Weaknesses in MIPv6. We have found at least three weaknesses in MIPv6. A brief summary of each one is as follows.

(1) The two tokens HT in (2) and CT in (3) make up the secret BU key. These tokens are delivered in clear text. Anyone can easily acquire HT and CT.

(2) Message authentication in the BU is completed after the CN receives the fifth message. Any earlier authentication for HoTI and CoTI is impossible because the MN and the CN do not share a secret key in advance. Hence, the CN must respond to all BU requests. This unconditional response involves an addition to its database, and an adversary may mount a memory overflow attack by sending meaningless BU requests.

(3) The two tokens are created independently of each other. This is because the tokens are created entirely by the CN, and the CoA is new to the CN and has never been used with the associated HoA. The CN is not able to bind the HoA with the CoA at the time of receiving HoTI and CoTI. The CN's ignorance of the association between the CoA and HoA at an early stage makes it almost impossible to generate a pair of related tokens. Because of this independence, the CN checks only to determine if a returning token is the one given by the CN but fails to determine if these two tokens come from a single source or from two different sources. An adversary needs only to manipulate the CoTI and to deceive only the CN to succeed in hijacking a session to a new CoA of the adversary's choice.

4.2. Vulnerability in MIPv6 (Session Hijacking Attack). A session hijacking attack (or redirection attack) is initiated by an adversary located between the HA and the CN. An illustration of this attack flow is depicted in Figure 2. This adversary intercepts the HoT message sent by the CN to the MN, a target victim. This message is in clear text, and the adversary can extract the token from the message (see (2)). This HT token is the first half of the session key for the BU. The adversary sends the forged CoTI message to the CN. An address chosen by the adversary appears as the source address in this message. Let us denote the forged CoTI message and the adversary's address to the CoTI$_a$ and CoA$_a$, respectively. The CN would accept the CoTI$_a$ message because of the second vulnerability described in Section 4.1. The CN generates CT$_a$ and returns this token enclosed in the CoT$_a$ message to the adversary. CoA$_a$ appears as the destination address in the CoT$_a$ message. This CoT$_a$ message is also in clear text, and the adversary acquires the second half of the token necessary to derive the session key. The adversary generates the K_{bm} according to (5) and sends the forged BU message as if it were the legitimate MN updating the new CoA.

The CN extracts the hash indices from the BU messages and reads the two tokens from its hash. Using (5) the CN recovers $K_{bm} = H(\mathrm{HT}\|\mathrm{CT}_a)$ and validates the sign in the BU message. The validation should pass, because the CN's K_{bm} is

the same as the adversary's K_{bm}. The CN accepts the forged BU and starts to communicate with the adversary located at CoA_a. The MN's session thus has been hijacked by the adversary.

This session hijacking attack exploits the third vulnerability discussed in Section 4.1; that is, the two tokens that make up the session key for the BU are created without any common factors between them. This independent key creation lays the foundation for exploitation by the adversary. From the perspective of the adversary, replacing the CoA with CoA_a is quite simple because it is the only thing required in order to send the forged $CoTI_a$ and to remember the CT_a in the CoT_a. It is such a simple attack that the adversary does not need to manipulate HT and the messages associated with HT (i.e., HoTI and HoT). If we could design the BU to have HT and CT share meaningful components known to the CN and to the MN, a session hijack attack would not be so simple. In such a case the change only to the CT is insufficient because the HT and CT_a would then share a common factor different from the one the CN recognizes. Hence, the adversary must forge HoTI and HoT_a and HT_a as well as CT_a for the attack to succeed. Forging HT and those related messages is more difficult than forging the CT. This is because (1) the adversary must be present not only in the CN-MN path but also in the CN-HA path; (2) the adversary must block the HoT_a that is destined for the HoA. The MN would be very suspicious if it found the HoT_a generated as a return of the HoTI that the MN had never sent. However, this blockage by an adversary would be almost impossible without having control of a router or a switch along the CN-HA path, which we believe it is quite difficult. Hence, our design principle for the new BU is to introduce a common factor shared only between the MN and the CN.

5. The Proposed Routing Optimization Protocol

Based upon the foregoing observations, we proposed a novel protocol for a secure RO in the MIPv6. We will discuss protocol requirements first and then the basic protocol proposed in this paper.

5.1. Protocol Requirements. Some requirements were determined in the course of designing the protocol. These requirements were selected after taking into consideration both practical implementation issues and performance issues. Five requirements summarize the most desirable attributes of the new protocol.

(i) Ownership. The MN can corroborate to the CN that the claimed CoA is owned by the MN. Also, the MN should be able to verify the CoA's binding with the MN's original HoA.

(ii) Routability. The CN should be certain that the new CoA is valid and reachable in the network.

(iii) Dependency. In the legacy protocol, the MN is given the session key (K_{bm}) and uses it to authenticate the BU message. This requirement will change how the two tokens are created. These two tokens must rely upon each other and in order to thwart any session hijacking attack and must share a factor that cannot be forged.

(iv) Compatibility and Easy Implementation. The new protocol should be easy to implement and introduce the lesser imperative amendments to the existing MIPv6 protocol so that the transition to the new protocol is smooth and transparent to end users.

(v) No Degradation of QoS. The new protocol should not degrade QoS in the MIPv6, especially the speed of handover.

The first two requirements are essential because they are the security requirements and the main purpose of the BU and of RR, respectively. We show in Section 6.1 how the new protocol satisfies these first two requirements. Satisfaction of the third requirement is discussed in the security analysis of the protocol in Section 6.2. The last two requirements are discussed in Section 6.3 in which we discuss the computational overhead of the protocol.

5.2. The Proposed Protocol. The proposed protocol inherits the strength of the legacy RO protocol in MIPv6 and eliminates the weaknesses identified by ourselves and mentioned in the related work. The advantages of the proposed protocol are concentrated in the design of the BU message. The roles and consequences of the rest of the messages are quite similar to those of the legacy protocol except for minor modification of the messages.

The MN initiates the BU by sending HoTI and CoTI shown in

$$\begin{aligned} \text{HoTI} &= \{\text{HoA}, \text{CN}, R_H, T_1\}, \\ \text{CoTI} &= \{\text{CoA}, \text{CN}, R_C, T_2\}. \end{aligned} \tag{8}$$

R_H and R_C are the random numbers to match, respectively, HoT with HoTI and CoT with CoTI. Without these parameters, mapping HoT to HoTI in the MN would be difficult in a situation such as one in which the MN might send multiple HoTI messages (or CoTI) because of retransmissions. Once the response arrives, the MN is unable to map this response to the multiple HoTI messages. The CN must return this random number in its response to avoid confusion in the MN.

T_1 and T_2 are the tokens generated by the MN in the proposed system. These tokens are shown in

$$S = H(p \| q), \tag{9}$$

$$T_1 = \text{HoA} \oplus S, \qquad T_2 = \text{CoA} \oplus q, \tag{10}$$

where p and q are the quite large random numbers and input values to the one-way hash function $H(\cdot)$. It is believed that finding input values p and q from S in a reasonable time boundary is almost impossible because of the one-wayness of the hash function which is consisted of is also impossible. Note that T_1 and T_2 share the common number q and p in S which is known only to the MN and nobody else.

HoT and CoT are the CN's responses shown in

$$\begin{aligned} \text{HoT} &= \{\text{CN}, \text{HoA}, R_H, \text{HT}_1, i\}, \\ \text{CoT} &= \{\text{CN}, \text{CoA}, R_C, \text{HT}_2, j\}. \end{aligned} \tag{11}$$

These equations are the same as (2) and (3) in the legacy protocol except that the two tokens, HT and CT, are replaced, respectively, by HT_1 and CT_1. We no longer use the session key K_{bm} to authenticate the BU message. HT_1 and CT_1 are instead referred to as cookies in our system and elaborated, respectively, in

$$\begin{aligned} \text{HT}_1 &= N_i \oplus K_{\text{cn}}, \\ \text{CT}_1 &= N_j \oplus K_{\text{cn}}. \end{aligned} \tag{12}$$

N_i and N_j are the two nonce values generated by the CN. These nonce values and two tokens, T_1 and T_2, are saved in the CN's hash under the hash indices of i and j. The indices, i and j, are included, respectively, in HoT and CoT. The CN expects to receive these indices in the next message. In this way, the CN remains stateless, dispensing with the need to remember these parameters.

The binding message is shown in

$$\text{BU} = \left\{\text{CoA}, \text{CN}, \text{HoA}, i, j, \text{LT}, \text{SEQ}, N_i \oplus N_j, p\right\}. \tag{13}$$

N_i and N_j are used with K_{CN} to verify the return routability of CoA by determining whether the MN returns $N_i \oplus N_j$ in the BU message. K_{CN} is the secret key owned by the CN and used to protect N_i and N_j, respectively, in the HoT and CoT messages. The MN should receive both the HoT and CoT messages and extract HT_1 and CT_1. By XORing HT_1 and CT_1 the MN can calculate $N_i \oplus N_j$ and include this in the BU message. Notably, the MN discloses p in this message. The BU message is authenticated with the MN's presentation of its secrets p to the CN.

The CN validates the BU message and then accepts the consequences of the return routability:

$$\text{BA} = \{\text{CN}, \text{CoA}, \text{LT}\}. \tag{14}$$

The CN confirms the BU by sending binding acknowledgment (BA) as shown in (14). CoA appears as the destination address in the BA message.

6. Performance Evaluation

We evaluated diverse aspects of the performance of the protocol. This evaluation includes an analysis to illustrate how the new protocol copes with the vulnerability of the legacy protocol and how it meets the five requirements specified earlier. A comparison of the computational cost between the five protocols is included. The delay involved in completing the secure RO is measured in terms of three popular wireless access networks, and the implications of this delay are described.

6.1. Security Analysis. By using the binding update in the proposed protocol, the MN can assure the CN that the MN is reachable (or routable) at the claimed CoA and that this MN is the owner of the HoA and CoA. The routability and ownership are the two security requirements and we intend to demonstrate that the proposed protocol is securely sound by showing that the proposed protocol satisfies these two requirements.

N_i and N_j are sent in the HoT and CoT messages by the CN and securely wrapped by the CN's secret, $K_{\text{CN}} \cdot N_i$ is directed to HoA along the indirect path, and N_j is directed to CoA along the direct path. In receiving the BU message, the CN retrieves N_i and N_j from its hash using i and j (see (13)) and calculates $N_i \oplus N_j$. The CN checks to see if the returned $N_i \oplus N_j$ is identical to the one calculated. The correct $N_i \oplus N_j$ indicates that the MN is reachable at HoA and CoA in both paths. In other words, the CN can ensure the routability of the return path to the MN.

In this scenario, an adversary impersonating the MN could have intercepted HoT and CoT and calculated $N_i \oplus N_j$ in the same way the MN did. However, the calculations required of the adversary would not be as simple as they might seem. The MN is assigned a new CoA in the foreign network, and this address has never before been associated with the MN's HoT. The adversary would not be able to couple CoT with the corresponding HoT if a fairly large number of BU messages were passing by. This coupling is also difficult for the CN. This is why CN retains K_{CN} unchanged in generating HT_1 and CT_1 and even uses a constant K_{CN} across different binding updates. However, it remains possible, even if it seems quite improbable, for adversaries to couple HT_1 and CT_1. Hence, it is not enough for the CN to assure the RR by presenting $N_i \oplus N_j$ alone. The proposed protocol compensates for this drawback by authenticating the BU message. Because the message is authentic, the content of this message is also authentic.

Using the hash indices i and j, the CN retrieves N_i and T_1 using hash index i and do the same for N_j and T_2 using hash index j. The CN XORs T_1 with the received HoA and compares the output with the hash function of p and q; that is, $\text{HoA} \oplus T_1 = \text{HoA} \oplus \text{HoA} \oplus S = H(p \| q)$. Algorithm 1 elaborates the CN's procedure to validate the BU message. Let us hypothesize that adversaries have intercepted a number of HoTI and CoTI messages in the network and also have been lucky enough to find a pair of T_1 and T_2. Even in this extreme scenario, it is almost impossible for the adversary to find p due to the one-wayness of the hash. No one except the MN that has sent HoTI and CoTI is able to present p to the CN. If the MN presents the right P-value, the CN concludes that this MN also sent HoTI and CoTI, confirming the MN's ownership of the CoA.

HoA and CoA are included in the BU message not only to compute S but also to preclude a dishonest MN from claiming a different CoA in the BU message than the CoA reported in the CoTI message.

6.2. A Suggested Solution for the Three Weaknesses. RO vulnerability is attributable to the three weaknesses discussed

```
Data: index i, j, p, N1 ⊕ N2, HoA, CoA, Hash
Result: Which Verification is confirmed
Begin
   Extract T1, N1', T2, N2' from Table of CN by i and j
   if N1 ⊕ N2 is a N1' ⊕ N2' then          /* return routability is confirmed */
      Compute q' = T2 ⊕ CoA
      X = H(p||q') and H(p||q) = T1 ⊕ HoA
      if H(p||q) is X then                 /* ownership is confirmed */
         return Verification succeeded
      else                                 /* ownership is failed */
         return Verification failed
   else                                    /* return routability is failed */
      return Verification failed
end
```

Algorithm 1: Verification procedure by CN.

in Section 4.1. A solution to any one of these three may remedy the vulnerability in the RO.

The first cause of RO vulnerability lies with delivery of the two tokens in clear text. The remedy requires a shared key to encrypt the tokens as well as authentication and a key exchange protocol for establishing the session key. This additional protocol is a heavy burden for a mobile device.

Delayed authentication causes the CN to accept all HoTI and CoTI messages that request an RO. Early authentication to the MN may be a good solution for this problem. However, following the same reasoning as discussed in the first cause, authentication necessitates a secret key, and we do not consider adding computational overhead to the existing protocol a viable option.

With the complications posed by solutions to the first and second vulnerabilities, we turn to the third of these and suggest another route to closing all three loopholes. The third vulnerability that we discussed originates in the generation by the CN of the two tokens independently of each other. Our solution to this problem is to have the two tokens share a common factor at the time of the generation. In the proposed protocol, q is this common factor. Addition of this feature complicates a session hijacking attack tremendously because an adversary must forge the two tokens and their related message simultaneously, a feat that we believe verges on impossible. In the legacy protocol, embedding a relationship into the two tokens was impossible because they are created by the CN, which has no knowledge of them at the time of their generation. In the proposed protocol, however, the MN generated the two tokens on behalf of the CN without any difficulty in pairing CoA and HoA.

6.3. Computational Comparison. The proposed protocol maintains backward compatibility with the legacy protocol. The new protocol contains six messages, and the role of each message remains the same as in the legacy protocol. The transition to the new protocol is straightforward because this requires only a software upgrade in the kernel.

We compared the computational expenses for the six protocols described in Section 3; CAM [11], the proposed protocol, the legacy protocol [1], ROM [12], CBU [8], and LR-AKE [10]. Because the number of messages to complete the RO is different from protocol to protocol, we compared them in terms of the computational expense in each message. Table 1 shows the computational expense for each message up to the thirteenth message. In order to distinguish operations in MN, CN and HA, cells in the table have different backgrounds.

The proposed protocol, which is only backward compatible with the legacy protocol, comprises the six messages. The ROM protocol is also composed of six messages, but nonetheless is incompatible with the legacy protocol. In order to form the BU message (see the fifth message in Table 1), the legacy protocol uses one 768-bit HMAC and one 128-bit SHA-1, respectively, to compute K_{bm} (see (5)) and to sign the BU message (see (6)). The MN in the proposed protocol computes the one XOR operation for the same message. In order to complete the BU (see the fifth and sixth messages in Table 1), the legacy protocol, the proposed protocol, and ROM, respectively, use five HMAC-SHA-1 operations and two SHA-1 operations, two XOR operations and one hash operation, and one hash operation. CAM is composed of two messages and the most efficient in terms of the number of messages. In contrast LR-AKE has the greatest number of messages. Operations to form each message are quite diverse from one protocol to another, ranging from simple XOR to expensive asymmetric decryption.

Figures 3 and 4 show the computational delays of the six protocols in completing the RO. The delay taken by the each operation as shown in Table 1 is modeled by its average value. The delays of operations done by the three nodes are summed together and plotted in Figures 3 and 4. (LR-AKE requires two HAs for MN and CN, resp. We did not differentiate these two HAs in the computation.) Some of the protocols show different delay measurements, depending upon whether it is the first handover or the second or later handovers. Although Figure 3 depicts the computational delay for the first handover, Figure 4 shows the delay for later handovers. In a continuing sense, the compilation in Table 1 bases RO security in terms of the first handover. CBU and LR-AKE are protocols that fit this definition, and the delay

Table 1: Computational expenses to form each message. The table shows the comparison for up to 13 messages. Although CAM needs only two messages, LR-AKE requires 13 messages to complete the RO. Note that cells in the table have different backgrounds to distinguish nodes these operations are computed. (MU: multiplication, SU: subtract, XR: XOR, MO: modulo, DV: division, EX: exponentiation, HS: one-way hash function, HM: keyed-hash for message authentication, E_S: symmetric encryption, D_S: symmetric decryption, E_{PU}: asymmetric encryption, D_{PR}: asymmetric decryption, SG: signature generation using private key, SV: signature verification using public key.)

	1	2	3	4	5
CAM	SG	HS + SV			
Our	HS + XR	XR	XR	XR	XR
Legacy	—	—	HM	HM	HM + HS
ROM	AD + HS	SU	AD	SU	—
CBU	—	HS	HS	HS + EX + SG	SV + EX + 2HS
LR-AKE	XR + HS + EX + MU + MO	2HS + DV + EX + MO	3HS + XR	E_{PU} + XR + 2HS	$2D_{PR} + E_{PU}$ + 2HS

	6	7	8	9	10	11	12	13
CAM								
Our	HS + XR							
Legacy	4HM + HS							
ROM	HS							
CBU	2HS + EX	E_S	D_S					
LR-AKE	E_S	$D_S + E_S$	$D_S + E_S$	$D_S + E_S$ + HS	$D_S + E_S$	$D_S + E_S$	E_S + HS	$D_S + E_S$

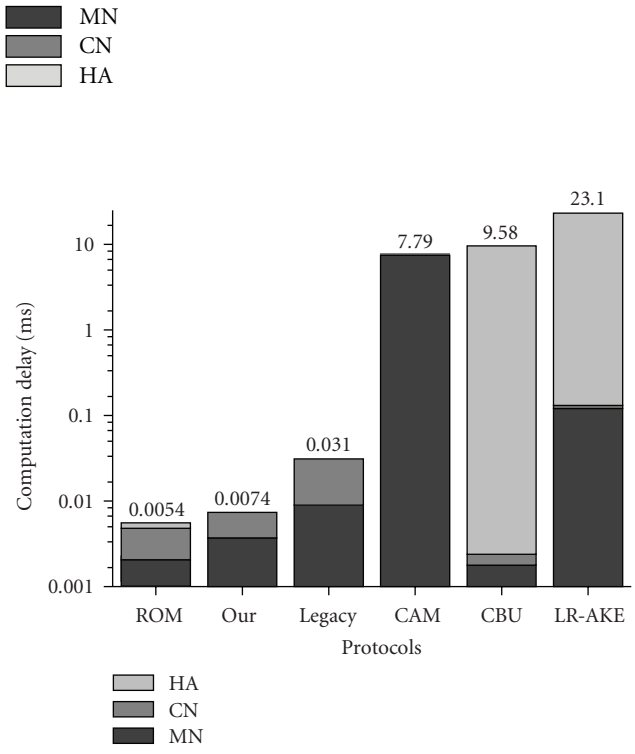


Figure 3: Computational delay for the first handover.

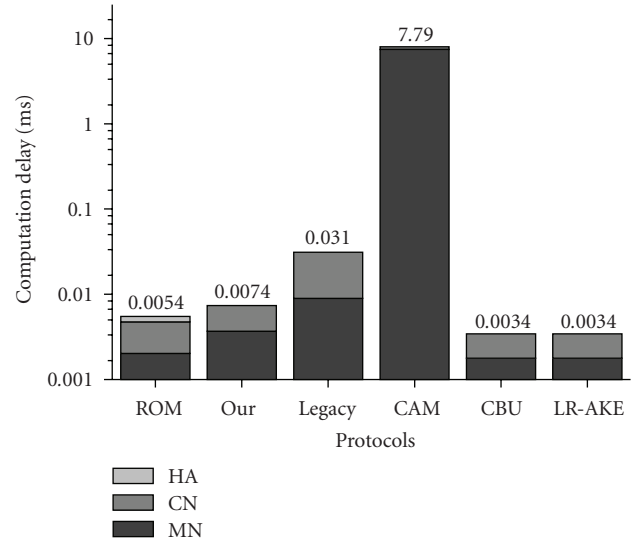


Figure 4: Computational delay for the second and later handovers.

difference between the first and later handovers is quite substantial. These two protocols use private key cryptography to establish a session key at the first handover. This approach to the session key takes considerable time, as shown in Figure 3. After the second handover, the MN and CN encrypt and decrypt messages using symmetric cryptography. The proposed protocol is the fastest in the first handover while CBU and LR-AKE are the fastest in the second and later handovers.

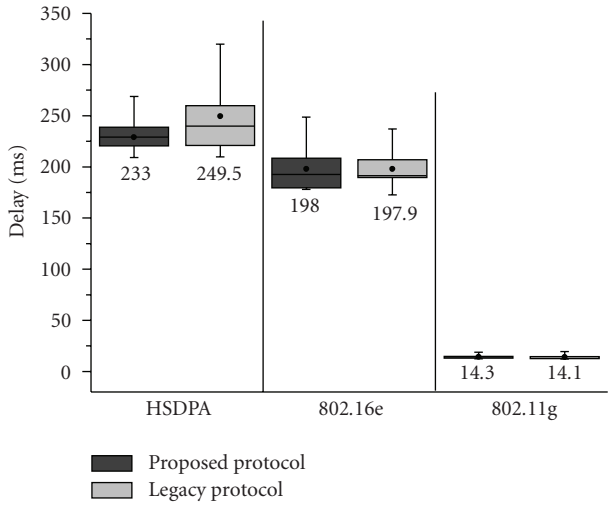


Figure 5: Delays to complete RO in three popular wireless access technologies. We repeated RO for each protocol one thousand times and plotted the outcome in a boxplot.

The delay with the legacy protocol is almost more than four times longer than with the new protocol. The speed of the new protocol is attributed to the transition from frequent hash operations in the legacy protocol to XOR and few hash operations in the new protocol. The delay of the proposed protocol outperforms the ROM protocol by 2 microseconds. Although the difference is insignificant the ROM cannot guarantee return routability to the CN. The computational delay in CAM is quite interesting. It uses an asymmetric signature for the first message in the MN and turns to a one-way hash function and signature verification for the second message in the CN. Although only two messages are used in CAM to complete a secure RO, the computational delay is quite long because of the computation load.

We have implemented the legacy and proposed protocols in three popular wireless access technologies; High Speed Downlink Packet Access (HSDPA), 802.16e [15], and 802.11g [16], illustrated in Figure 5. This is not to compare the performance of these protocols but rather to measure actual delays in order to determine whether it is appropriate to suggest deployment of these protocols in the real environment. This measurement is especially important to developers and engineers in the mobile industry because a delay in the handover greatly influences QoS in mobile applications. The handover in 802.11g completes a secure RO in 14 milliseconds, which is the shortest among the three protocols. About 10 Mbps is the measured data rate of 802.11g and is greater than the 1.3 Mbps of HSDPA and the 3.6 Mbps of 802.16e. Table 2 shows the maximum data rates of the three technologies in terms of measurement and specification. The delay in HSDPA and 802.16e takes longer than 200 milliseconds, which is not appropriate for real-time applications such as IP telephony. The RO in 802.16g is faster than the one in HSDPA because of a higher data rate. We expect Long Term Evolution (LTE) and 802.16m, which are the next versions of HSDPA and 802.16e, respectively, within the next year or so [17]. These new technologies will boost the data rate in the access network to 30 Mbps. Then, those delay-sensitive real-time applications should not have any problems running on these access technologies.

Table 2: Maximum data rates for three technologies in measurement and specification.

	Maximum data rates in measurement (DL/UL)	Maximum data rate in specification (DL/UL)
HSDPA	1.3 Mbps/66 Kbps	14.4 Mbps/2 Mbps
802.16e	3.6 Mbps/423 Kbps	46 Mbps/4 Mbps
802.11g	10.3 Mbps/9.4 Mbps	54 Mbps

7. Conclusion

The two special routines in the secure RO are BU and RR, and the purposes of these routines are to show to the CN that the claimed CoA is a temporary address of the MN and is reachable in the network.

The legacy RO in MIPv6 has a critical vulnerability that could let an adversary hijack an ongoing session to a location chosen by the adversary. This vulnerability is attributed to three weaknesses we found in the RO. The worst weakness is that the two tokens that compose the session key do not share a common factor. This weakness allows an adversary to manipulate CoTI alone, in order to initiate a session hijacking attack. We have proposed a secure RO protocol. This protocol requires only a light computational load and is compatible with the legacy protocol. Most important, this protocol provides a secure BU and RR.

To illustrate its practicality we compared the cost of establishing a secure RO with the proposed protocol with five other protocols that propose to create a secure RO. In addition, we have implemented the proposed and the legacy protocols to measure the communication delay in their use with three wireless access technologies. The evaluation results show that the proposed protocol performs well in terms of low computational cost and minimal delay.

References

[1] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," RFC 3775, June 2004.

[2] C. Perken, "IP Mobility Support," RFC 2002, October 1996.

[3] T. Aura, "Mobile IPv6 security," in *Security Protocols*, pp. 3–13, 2004.

[4] K. Elgoarany and M. Eltoweissy, "Security in mobile IPv6: a survey," *Information Security Technical Report*, vol. 12, no. 1, pp. 32–43, 2007.

[5] J.-C. Chen, M.-C. Jiang, and Y. I.-W. Liu, "Wireless LAN security and IEEE 802.11l," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 27–36, 2005.

[6] D. Johnston and J. Walker, "Overview of IEEE 802.16 security," *IEEE Security and Privacy*, vol. 2, no. 3, pp. 40–48, 2004.

[7] G. M. Koien, "An introduction to access security in UMTS," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 8–18, 2004.

[8] R. H. Deng, J. Zhou, and F. Bao, "Defending against redirect attacks in mobile IP," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 59–67, Washington, DC, USA, 2002.

[9] K. Ren, W. Lou, K. Zeng, F. Bao, J. Zhou, and R. H. Deng, "Routing optimization security in mobile IPv6," *Computer Networks*, vol. 50, no. 13, pp. 2401–2419, 2006.

[10] H. Fathi, S. Shin, K. Kobara, S. S. Chakraborty, H. Imai, and R. Prasad, "Leakage-resilient security architecture for mobile IPv6 in wireless overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 11, pp. 2182–2192, 2005.

[11] O. S. Greg and R. Michael, "Child-proof authentication for MIPv6 (CAM)," *ACM SIGCOMM Computer Communication Review*, vol. 31, pp. 4–8, 1984.

[12] C. Veigner and C. Rong, "A new route optimization protocol for Mobile IPv6 (ROM)," in *Proceedings of the International Computer Symposium*, Taipei, Taiwan, 2004.

[13] C. Veigner and C. Rong, "Flooding attack on the binding cache in mobile IPv6," 2007.

[14] P. Nikander, J. Arkko, T. Aura, and G. Montenegro, "Mobile IP version 6 (MIPv6) route optimization security design," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, vol. 3, pp. 2004–2008, Orlando, Fla, USA, 2003.

[15] N. Johnston and H. Aghvami, "Comparing WiMAX and HSPA—a guide to the technology," *BT Technology Journal*, vol. 25, no. 2, pp. 191–199, 2007.

[16] D. Vassis, G. Kormentzas, A. Rouskas, and I. Maglogiannis, "The IEEE 802.11g standard for high data rate WLANs," *IEEE Network*, vol. 19, no. 3, pp. 21–26, 2005.

[17] S. Ortiz Jr., "4G wireless begins to take shape," *Computer*, vol. 40, no. 11, pp. 18–21, 2007.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 740912, 13 pages
doi:10.1155/2009/740912

Research Article

Distributed Cooperative Transmission with Unreliable and Untrustworthy Relay Channels

Zhu Han[1] and Yan Lindsay Sun[2]

[1] *Electrical and Computer Engineering Department, University of Houston, Houston, TX 77004, USA*
[2] *Electrical and Computer Engineering Department, The University of Rhode Island, Kingston, RI 02881, USA*

Correspondence should be addressed to Zhu Han, hanzhu22@gmail.com

Received 25 January 2009; Revised 13 July 2009; Accepted 12 September 2009

Recommended by Hui Chen

Cooperative transmission is an emerging wireless communication technique that improves wireless channel capacity through multiuser cooperation in the physical layer. It is expected to have a profound impact on network performance and design. However, cooperative transmission can be vulnerable to selfish behaviors and malicious attacks, especially in its current design. In this paper, we investigate two fundamental questions Does cooperative transmission provide new opportunities to malicious parties to undermine the network performance? Are there new ways to defend wireless networks through physical layer cooperation? Particularly, we study the security vulnerabilities of the traditional cooperative transmission schemes and show the performance degradation resulting from the misbehaviors of relay nodes. Then, we design a trust-assisted cooperative scheme that can detect attacks and has self-healing capability. The proposed scheme performs much better than the traditional schemes when there are malicious/selfish nodes or severe channel estimation errors. Finally, we investigate the advantage of cooperative transmission in terms of defending against jamming attacks. A reduction in link outage probability is achieved.

1. Introduction

Multiple antenna systems, such as Multiple-Input-Multiple-Output (MIMO), can create spatial diversity by taking advantage of multiple antennas and significantly increase the wireless channel capacity. However, installation of multiple antennas on one wireless device faces many practical obstacles, such as the cost and size of wireless devices. Recently, *cooperative transmission* has gained considerable research attention as a transmit strategy for future wireless networks. Instead of relying on the installation of multiple antennas on one wireless device, cooperative transmission achieves spatial diversity through physical layer cooperation.

In cooperative transmission, when the source node transmits a message to the destination node, the nearby nodes that overhear this transmission will "help" the source and destination by relaying the replicas of the message, and the destination will combine the multiple received waveforms so as to improve the link quality. In other words, cooperative transmission utilizes the nearby nodes as virtual antennas and mimics the effects of MIMO for achieving spatial diversity. It is well documented that cooperative transmission *improves channel capacity* significantly and has a great potential to improve wireless network capacity [1, 2]. The research community is integrating cooperative transmission into cellular, WiMAX, WiFi, Bluetooth, ultra-wideband (UWB), ad hoc, and sensor networks. Cooperative transmission is also making its way into standards; for example, IEEE WiMAX standards body for future broadband wireless access has established the 802.16j Relay Task Group to incorporate cooperative relaying mechanisms [3].

The majority of work on cooperative transmission focuses on communication efficiency, including capacity analysis, protocol design, power control, relay selection, and cross layer optimization. In those studies, all network nodes are assumed to be trustworthy. *Security threats* are rarely taken into consideration.

(i) It is well known that malicious nodes can enter many wireless networks due to imperfectness of access control or through node compromising attack. In cooperative transmission, the malicious nodes have

chances to serve as *relays* (i.e., the nodes help the source node by forwarding messages). Instead of forwarding correct information, malicious relays can send arbitrary information to the destination.

(i) Cooperative transmission can also suffer from selfish behavior. When the wireless nodes do not belong to the same authority, some nodes can refuse to cooperate with others, that is, not working as relay nodes, for the purpose of saving their own resources.

(i) In cooperative transmission, channel information is often required to perform signal combination [1–3] and relay selection [4–7] at the destination. The malicious relays can provide false channel state information, hoping that the destination will combine the received messages inadequately.

This paper is dedicated to studying the security issues related to cooperative transmission for wireless communications. Particularly, we will first discuss the vulnerabilities of cooperative transmission schemes and evaluate potential network performance degradation due to these vulnerabilities. Then, we propose a distributed trust-assisted cooperative transmission scheme, which strengthens security of cooperative transmission through joint trust management and channel estimation.

Instead of using traditional signal-to-noise ratio (SNR) or bit-error-rate (BER) to represent the quality of relay channels, we construct the trust values that represent possible misbehavior of relays based on beta-function trust models [8, 9]. We then extend the existing trust models to address trust propagation through relay nodes. A distributed trust established scheme is developed. With a low overhead, the model parameters can propagate through a complicated cooperative relaying topology from the source to the destination. In the destination, the information from both the direct transmission and relayed transmissions is combined according to the trust-based link quality representation. From analysis and simulations, we will show that the proposed scheme can automatically recover from various attacks and perform better than the traditional scheme with maximal ratio combining. Finally, we investigate possible *advantages* of utilizing cooperation transmission to improve security in a case study of defending against jamming attacks.

The rest of the paper is organized as follows. Related work is discussed in Section 2. In Section 3, the system model and attack models are introduced. In Section 4, the proposed algorithms are developed. Finally, simulation results and conclusions are given in Sections 5 and 6, respectively.

2. Related Work

Research on cooperative transmission traditionally focuses on *efficiency*. There is a significant amount of work devoted to analyzing the performance gain of cooperative transmission, to realistic implementation under practical constraints, to relay selection and power control, to integrating physical layer cooperation and routing protocols, and to game-theory-based distributed resource allocation in cooperative transmission. For example, the work in [4] evaluates the cooperative diversity performance when the best relay is chosen according to the average SNR and analyzes the outage probability of relay selection based on instantaneous SNRs. In [5], the authors propose a distributed relay selection scheme that requires limited network knowledge with instantaneous SNRs. In [6], cooperative resource allocation for OFDM is studied. A game theoretic approach for relay selection has been proposed in [7]. In [10], cooperative transmission is used in sensor networks to find extra paths in order to improve network lifetime. In [11], cooperative game theory and cooperative transmission are used for packet forwarding networks with selfish nodes. In [12], centralized power allocation schemes are presented under the assumption that all the relay nodes help others. In [13], cooperative routing protocols are constructed based on noncooperative routes. In [14], a contention-based opportunistic feedback technique is proposed for relay selection in dense wireless networks. In [15], the users form coalitions of cooperation and use MIMO transmission. Traditional cooperative transmission schemes, however, assume that all participating nodes are trustworthy.

Trust establishment has been recognized as a powerful tool to enhance security in applications that need cooperation among multiple distributed entities. Research on trust establishment has been performed for various applications, including authorization and access control, electronic commerce, peer-to-peer networks, routing in MANET, and data aggregation in sensor networks [8, 16–20]. As far as the authors' knowledge, no existing work on trust is for cooperative transmission. In fact, not much study on trust has been conducted for physical layer security.

3. System Model, Attack Models, and Requirements on Defense

In this section, we first describe the cooperative transmission system model, then investigate the different attack models, and finally discuss the general requirements on the design of defense mechanisms.

3.1. Cooperative Transmission System. As shown in Figure 1, the system investigated in this paper contains a source node s, some relay nodes r_i, and a destination node d. The relays can form single hop or multihop cooperation paths. The relay nodes might be malicious or selfish. We first show a simple one-hop case in this subsection, and the multihop case will be discussed in a later section.

Cooperative transmission is conducted in two phases. In *Phase 1*, source s broadcasts a message to destination d and relay nodes r_i. The received signal y_d at the destination d and the received signal y_{r_i} at relay r_i can be expressed as

$$y_d = \sqrt{P_s G_{s,d}} h_{s,d} x + n_d, \tag{1}$$

$$y_{r_i} = \sqrt{P_s G_{s,r_i}} h_{s,r_i} x + n_{r_i}. \tag{2}$$

In (1) and (2), P_s represents the transmit power at the source, $G_{s,d}$ is the path loss between s and d, and G_{s,r_i} is the path loss

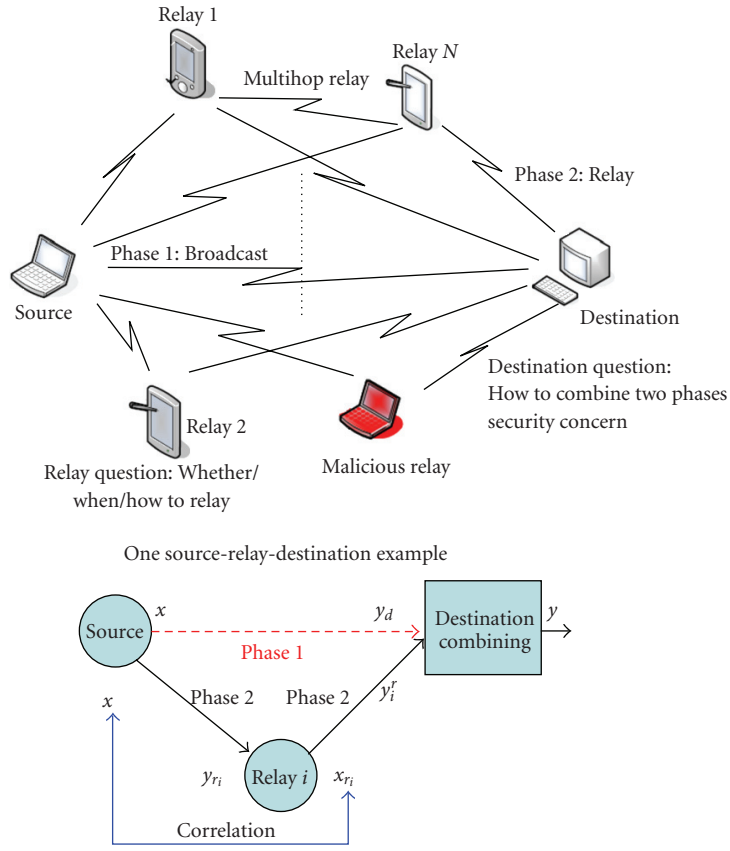


Figure 1: Cooperative transmission system model.

between s and r_i. $h_{s,d}$ and h_{s,r_i} are fading factors associated with channel $s-d$ and channel $s-r_i$, respectively. They are modeled as zero mean and unit variance complex Gaussian random variables. x is the transmitted information symbol with unit energy. In this paper, without loss of generality, we assume that BPSK is used and $x \in \{0,1\}$. n_d and n_{r_i} are the additive white Gaussian noises (AWGN) at the destination and the relay nodes, respectively. Without loss of generality, we assume that the noise power, denoted by σ^2, is the same for all the links. We also assume the block-fading environment, in which the channels are stable over each transmission frame.

When there is no relay, the transmission only contains Phase 1 and is referred to as *direct transmission*. In direct transmission, without the help from relay nodes, the SNR at the destination is

$$\Gamma_d = \frac{P_s G_{s,d} E\left[\left|h_{s,d}\right|^2\right]}{\sigma^2}. \tag{3}$$

In *Phase 2*, relay nodes send information to the destination at consecutive time slots. After the destination receives the information from the source node and all relay nodes, which takes at least $N_r + 1$ time slots where N_r is the number of relays, the destination combines the received messages and decodes data.

We examine the *decode-and-forward* (DF) cooperative transmission protocol [1, 2], in which the relays decode the source information received in Phase 1 and send the information to the destination in Phase 2. Recall that relay r_i receives signal y_{r_i} from the source node d. Let x_{r_i} denote the data decoded from y_{r_i}. Relay r_i then reencodes x_{r_i}, and sends it to the destination. Let $\hat{y}_{r_i}$ denote the received signal at the destination from relay r_i. Then,

$$\hat{y}_{r_i} = \sqrt{P_{r_i} G_{r_i,d}} h_{r_i,d} x_{r_i} + n'_d, \tag{4}$$

where P_{r_i} is the transmit power at relay r_i, $G_{r_i,d}$ is the path loss between r_i and d, $h_{r_i,d}$ is the fading factor associated with channel $r_i - d$, which is modeled as zero mean and unit variance Gaussian random variable, and n'_d is the AWGN thermal noise with variance σ^2.

3.2. Attack Models and Requirements on Defense. As discussed in Section 1, for cooperative transmission, we identify the following three types of misbehavior.

(i) *Selfish Silence*. There are selfish nodes that do not relay messages for others in order to reserve their own energy.

(ii) *Malicious Forwarding*. There are malicious nodes that send garbage information to the destination when they serve as relays.

(iii) *False Feedback*. Malicious nodes report false channel information to make the destination perform signal combination inadequately.

Can security vulnerability in cooperative transmission be fixed? To answer this question, we take a closer look at the fundamental reasons causing security vulnerability.

First, cooperation among distributed entities is inheritably vulnerable to selfish and malicious behaviors. When a network protocol relies on multiple nodes' collaboration, the performance of this protocol can be degraded if some nodes are selfish and refuse to collaborate, and can be severely damaged if some nodes intentionally behave oppositely to what they are expected to do. For example, the routing protocols in mobile ad hoc networks rely on nodes jointly forwarding packets honestly, and the data aggregation protocols in sensor networks rely on sensors all reporting measured data honestly. It is well known that selfish and malicious behaviors are major threats against the above protocols. Similarly, since cooperative transmission relies on collaboration among source, relay and destination nodes, it can be threatened by selfish and malicious network nodes.

Second, when the decision-making process relies on feedback information from distributed network entities, this decision-making process can be undermined by dishonest feedbacks. This is a universal problem in many systems. For example, in many wireless resource allocation protocols, transmission power, bandwidth and data rate can all be determined based on channel state information obtained through feedbacks [5, 7, 11]. In cooperative transmission, the relay selection and signal combination process depend on channel state information obtained through feedbacks.

Third, from the view point of wireless communications, traditional representation of channel state information cannot address misbehavior of network nodes. In most cooperative transmission schemes, information about relay channel status is required in relay selection and transmission protocols. However, the traditional channel state information, either SNR or average BER, only describes the features of physical wireless channel, but cannot capture the misbehavior of relay nodes.

The above discussion leads to an understanding on the *primary design goals* of the defense mechanism. A defense mechanism should be able

(i) to provide the distributed network entities a strong incentive to collaboration, which suppresses selfish behaviors,

(ii) to detect malicious nodes and hold them responsible,

(iii) to provide the cooperative transmission protocols with accurate channel information that (a) reflects both physical channel status as well as prediction on likelihood of misbehavior and (b) cannot be easily misled by dishonest feedbacks.

4. Trust-Based Cooperative Transmission

In this section, we first provide basic concepts related to trust evaluation in Section 4.1. Second, we discuss the key components in the proposed scheme, including the beta-function-based link quality representation and link quality propagation, in Section 4.2. Then, the signal combining algorithm at the destination is investigated in Section 4.3. Next, we present the overall system design in Section 4.4, followed by a discussion on implementation overhead in Section 4.5.

4.1. Trust Establishment Basic. Trust establishment has been recognized as a powerful tool to secure collaboration among distributed entities. It has been used in a wide range of applications for its unique advantages.

> If network entities can evaluate how much they trust other network entities and behave accordingly, three advantages can be achieved. First, it *provides an incentive* for collaboration because the network entities that behave selfishly will have low trust values, which could reduce their probabilities of receiving services from other network entities. Second, it can *limit the impact* of malicious attacks because the misbehaving nodes, even before being formally detected, will have less chance to be selected as collaboration partners by other honest network nodes. Finally, it provides a way to *detect malicious nodes* according to trust values.

The purpose of trust management matches perfectly with the requirements for defending cooperative transmission.

Designing a trust establishment method for cooperative transmission is not an easy task. Although there are many trust establishment methods in the current literature, most of them sit in the application layer and few were developed for physical/MAC layer communication protocols. This is mainly due to the high implementation overhead. Trust establishment methods often require monitoring and message exchange among distributed nodes. In physical layer, monitoring and message exchange should be minimized to reduce overhead. Therefore, our design should rely on the information that is already available in the physical layer.

While the detailed trust establishment method will be described in a later section, we introduce some trust establishment background here.

When node A can observe node B's behavior, node A establishes *direct trust* in node B based on observations. For example, in the beta-function-based-trust model [9], if node A observes that node B has behaved well for $(\alpha - 1)$ times and behaved badly for $(\beta - 1)$ times, node A calculates the direct trust value [9] as $\alpha/(\alpha + \beta)$. The beta-function based trust model is widely used for networking applications [18, 20], whereas there are other ways to calculate direct trust mainly for electronic commerce, peer-to-peer file sharing, and access control [8, 17].

Trust can also be established through third parties. For example, if A and B_1 have established a trust relationship and B_1 and Y have established a trust relationship, then A can trust Y to a certain degree if B_1 tells A its trust opinion (i.e., recommendation) about Y. This phenomenon is called *trust propagation*. Trust propagation becomes more complicated when there is more than one trust propagation path. Through trust propagation, *indirect trust* can be

established. The specific ways to calculate indirect trust values are determined by *trust models* [8].

Finally, building trust in distributed networks requires authentication. That is, one node cannot easily pretend to be another node in the network.

No matter whether trust mechanism is used or not, the physical layer control messages need to be authenticated, when there is a risk of malicious attack. In this work, we assume that the messages are authenticated in cooperative transmission using existing techniques [21, 22].

4.2. Trust-Based Representation of Link Quality. The beta-function trust model is often used to calculate whether a node is trustworthy or not in networking applications. For example, node B has transmitted $(\alpha + \beta - 2)$ packets to node A. Among them, node A received $(\alpha - 1)$ packets with SNR greater than a certain threshold. These transmissions are considered to be successful. The transmission of other packets is considered to be failed. That is, there are $(\alpha - 1)$ successful trials and $(\beta - 1)$ failed trials. It is often assumed that the transmission of all $(\alpha + \beta - 2)$ packets are independent and a Bernoulli distribution with parameter p governs whether the transmissions succeed or fail. (This is true with ideal interleavers.) Under these assumptions, given α and β, the parameter p follows a beta distribution as

$$B(\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}. \tag{5}$$

It is well known that $B(\alpha,\beta)$ has mean m and variance v as

$$m = \frac{\alpha}{\alpha+\beta}; \qquad v = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}. \tag{6}$$

In the context of trust establishment, given α and β values, the trust value is often chosen as the mean of $B(\alpha,\beta)$, that is, $\alpha/(\alpha+\beta)$. *This trust value represents how much a wireless link can be trusted to deliver packets correctly*. In addition, some trust models introduce *confidence values* [23]. The confidence value is often calculated from the variance of $B(\alpha,\beta)$. The confidence value represents how much confidence the subject has in the trust value.

Due to the physical meaning of the trust values and the close tie between trust and the beta function, we *use the beta function to represent the link quality* in this paper. This is equivalent to using trust and confidence values to describe the link quality.

Since an interleaver is often employed in the transceiver and noise is independent over time, we can justify that successful transmission of different packets is independent if the interleaver is carefully selected to be greater than the coherence time of the channel. As a result, we justify the use of the beta distribution. Compared with traditional frame error rate (FER), BER and SNR, the trust-based link quality representation has both advantages and disadvantages. As an advantage, the trust-based link quality can describe the joint effect of wireless channel condition, channel estimation error, and misbehavior of relay nodes. On the other hand, the trust-based link quality cannot describe the rapid changes in channel conditions because the α and β values need to be collected over multiple data packets. Thus, it is suitable for scenarios with slow fading channels or high data rate transmission, in which channel condition remains stable over the transmission time of several packets.

4.3. Signal Combination at Destination. In this Section, we discuss how to utilize trust-based link quality information in the signal combination process. In Section 4.3.1, we discuss how the signal is combined at the waveform level. In Section 4.3.2, we extend our solution to the multihop case. Finally, we investigate how the proposed solution can defend against the bad-mouthing attack in Section 4.3.3.

First, from [24], the BER of BPSK in Rayleigh fading can be given by a function of SNR as

$$\text{BER} = \frac{1}{2}\left(1 - \sqrt{\frac{\Gamma}{1+\Gamma}}\right), \tag{7}$$

where Γ is the SNR. Here FER has one-to-one mapping with BER as $\text{FER} = 1 - (1 - \text{BER})^L$, where L is the frame length. (Notice that other modulations can be treated in a similar way.) So in the rest of paper, we only mention BER. To simplify analysis, we assume that error control coding is not used in this paper. The design of the proposed scheme, however, will not be affected much by coding schemes. When coding is used, the BER expression in (7) will change. Depending on different coding systems such as Hamming code, RS code or convolutional code, the BER performance would be different. The BER would be reduced at the same SNR, or in other words, to achieve the same SNR, the required SNR will be reduced. So the reliability of the links due to the channel errors can be improved. On the other hand, coding is a way to improve reliability, but cannot address untrustworthy nodes. The proposed scheme will work for both coded and uncoded transmissions.

4.3.1. Waveform Level Combination. In traditional cooperative transmission schemes, maximal ratio combining (MRC) [24] is often used for waveform level combination. Specifically, for the case of a single-hop relay, remember that y_d is the signal received from the direct path and y_r^i is the signal received from the relay. Under the assumption that the relay can decode the source information correctly, the MRC combined signal with weight factor w_i is

$$y^{\text{mrc}} = w_0 y_d + \sum_i w_i y_r^i, \tag{8}$$

where $w_0 = 1$ and $w_i = \sqrt{P_{r_i} G_{r_i,d} / P_s G_{s,d}}$. The resulting SNR is given by [24]

$$\Gamma^{\text{MRC}} = \Gamma_d + \sum_i \Gamma_{r_i}, \tag{9}$$

where $\Gamma_d = P_s G_{s,d} E[|h_{s,d}|^2]/\sigma^2$ and $\Gamma_{r_i} = P_{r_i} G_{r_i,d} E[|h_{r_i,d}|^2]/\sigma^2$ are SNR of direct transmission and relay transmission, respectively. When channel decoding errors and nodes' misbehavior are present, the MRC is not optimal any more.

This is because the received signal quality is not only related to the final link to the destination, but also related to decoding errors or misbehavior at the relay nodes.

In the proposed scheme, we use the beta function to capture the channel variation as well as relay misbehavior. This requires a new waveform combination algorithm that is suitable for trust-based link quality representation.

We first consider the case of *one single-hop relay path*. Depending on whether or not the relay decodes correctly, using derivation similar to MRC [24], the combined SNR at the destination for BPSK modulation can be written as

$$\Gamma = \begin{cases} \Gamma^c = \dfrac{\Gamma_d + w_1^2\Gamma_{r_1} + 2w_1\sqrt{\Gamma_d\Gamma_{r_1}}}{1+w_1^2}, & \text{if the relay decodes correctly,} \\ \Gamma^w = \dfrac{\Gamma_d + w_1^2\Gamma_{r_1} - 2w_1\sqrt{\Gamma_d\Gamma_{r_1}}}{1+w_1^2}, & \text{if the relay decodes incorrectly.} \end{cases} \tag{10}$$

If the relay decodes correctly, the relayed signal improves the final SNR; otherwise, the SNR is reduced. Notice that here 1 is the weight for the direction transmission and w_1 is the weight for the relay transmission.

Let $B(\alpha_1, \beta_1)$ represent the link quality of the source-relay channel. We set the goal of signal combination to be maximizing the SNR at the destination after combination by finding the optimal weight vector for combination. That is,

$$w_1^* = \arg\min_{w_1} \int_0^1 [p\Gamma^c + (1-p)\Gamma^w] B(\alpha_1, \beta_1)\, dp. \tag{11}$$

By differentiating the right-hand side of (11), we obtain the optimal combination weight factor as

$$w_1^* = \frac{\Gamma_{r_1} - \Gamma_d + \sqrt{\Gamma_d^2 + \Gamma_{r_1}^2 + 2(1 - 8m_1 + 8m_1^2)\Gamma_d\Gamma_{r_1}}}{2(2m_1 - 1)\sqrt{\Gamma_d\Gamma_{r_1}}}, \tag{12}$$

where m_1 is the mean of the relay's successful decoding probability or the mean of $B(\alpha_1, \beta_1)$. Obviously, $m_1 = \alpha_1/(\alpha_1 + \beta_1)$.

When the relay decodes perfectly, that is, $m_1 = 1$, we have

$$w_1^* = \sqrt{\frac{\Gamma_{r_1}}{\Gamma_d}}, \tag{13}$$

which is the same as that in MRC. When $m_1 = 0.5$, we have zero-divide-zero case in (12). In this case, we define $w_1^* = 0$, since the relay decodes incorrectly and forwards independent data. As a result, the weight for the relay should be zero, and the system degrades to direct transmission only.

For the case of *multiple single-hop relay paths*, we assume that each relay has link quality (α_i, β_i), SNR Γ_{r_i}, and weight w_i. Recall that the link quality report from the relay i is (α_i, β_i), where $(\alpha_i - 1)$ equals to the number of successfully transmitted packets between the source and relay i and $(\beta_i - 1)$ equals to the number of unsuccessfully transmitted packets between the source and relay i. The mean of the beta function for relay i is denoted by m_i and calculated as $m_i = \alpha_i/(\alpha_i + \beta_i)$. The overall expected SNR can be written as

$$\Gamma = \max_{w_i} \sum_{q_i \in \{-1,1\}} \prod_i Q(q_i, m_i) \frac{\left(\sqrt{\Gamma_d} + \sum_i q_i w_i \sqrt{\Gamma_{r_i}}\right)^2}{1 + \sum_i w_i^2}, \tag{14}$$

where q_i indicates whether relay i decodes correctly, and

$$Q(q_i, m_i) = \begin{cases} m_i, & q_i = 1, \text{ decode correctly,} \\ 1 - m_i, & q_i = -1, \text{ decode incorrectly.} \end{cases} \tag{15}$$

Equation (14) employs the probability $Q(q_i, m_i)$ and conditional SNR in (10). In this case, the optimal w_i can be calculated numerically by minimizing (14) over parameter w_i. Some numerical methods such as the Newton Method [25, 26] can be utilized. Note that this optimization problem may not be convex. Achieving global optimum needs some methods such as simulated annealing [25, 26].

As a summary, the waveform level combination is performed in the following four steps.

(i) For each path, the destination calculates m_i values based on the relays' report on their link quality.

(ii) The second is maximizing the SNR (equivalent to minimizing BER) in (14) to obtain the optimal weight factors. If there is only one relay path, the optimal weight factor is given in (12).

(iii) The third step is calculating the combined waveform y using (8).

(iv) The fourth step is decoding the combined waveform y.

4.3.2. Extension to Multiple-Hop Relay Scenario. In the previous discussion, we focus on the one-hop relay case, in which the relay path is source-relay-destination. Next, we extend our proposed scheme to multiple such relay paths.

It is noted that the relay path may contain several concatenated relay nodes. An example of such relay path is $s - r_a - r_b - d$, where s is the source node, d is the destination, r_a and r_b are two concatenated relay nodes. This scenario has been studied in [27, 28].

To make the proposed scheme suitable for general cooperative transmission scenarios, we develop an approach to calculate the link quality through concatenation propagation. In particular, let $B(\alpha_{sa}, \beta_{sa})$ represent the link quality between s and r_a, and $B(\alpha_{ab}, \beta_{ab})$ represent the link quality between r_a and r_b. If we can calculate the link quality between s and r_b, denoted by $B(\alpha_{sb}, \beta_{sb})$, from α_{sa}, β_{sa}, α_{ab}, β_{ab}, we will be able to use the approach developed in Section 4.3.1, by replacing (α_i, β_i) with $(\alpha_{sb}, \beta_{sb})$. Then, (α_i, β_i) represents the link quality of the i^{th} relay path, which is $s - r_a - r_b - d$ in this example.

Next, we present the link quality concatenation propagation model for calculating $(\alpha_{sb}, \beta_{sb})$. Let $\hat{x}$ denote the probability that transmission will succeed through path

$s - r_a - r_b$. The cumulative distribution function of $\hat{x}$ can be written as

$$\mathrm{CDF}(\hat{x}) = \int \int_0^{\hat{x}=pq} \frac{\Gamma(\alpha_{sa}+\beta_{sa})\Gamma(\alpha_{ab}+\beta_{ab})}{\Gamma(\alpha_{sa})\Gamma(\beta_{sa})\Gamma(\alpha_{ab})\Gamma(\beta_{ab})} \times p^{\alpha_{sa}-1} q^{\alpha_{ab}-1} (1-p)^{\beta_{sa}-1} (1-q)^{\beta_{ab}-1} dp\, dq. \tag{16}$$

Since it is very difficult to obtain the analytical solution to (16), we find a heuristic solution to approximate the distribution of $\hat{x}$. Three assumptions are made.

First, even though the distribution of the concatenated signal is not a beta function, we approximate the distribution of $\hat{x}$ as a beta distribution $B(\alpha_{sb}, \beta_{sb})$. Let (m_{sa}, v_{sa}), (m_{ab}, v_{ab}), and (m_{sb}, v_{sb}) represent the (mean, variance) of distribution $B(\alpha_{sa}, \beta_{sa})$, $B(\alpha_{ab}, \beta_{ab})$, and $B(\alpha_{sb}, \beta_{sb})$, respectively. The mean and variance of the beta distribution are given in (6).

Second, we assume $m_{sb} = m_{sa} \cdot m_{ab}$. Recall that m_{sb}, m_{sa} and m_{ab} represent the probability of successful transmission along path $s - r_b$, $s - r_a$, and $r_a - r_b$, respectively. When the path is $s - r_a - r_b$, the packets are successfully transmitted from s to r_b only if the packets are successfully transmitted from s to r_a and from r_a to r_b.

Third, we assume $v_{sa} + v_{ab} = v_{sb}$. The third assumption means that the noises added by two concatenated links are independent and their variances can be added together.

With the above assumptions, we can derive that

$$\alpha_{sb} = m_{sa} m_{ab} \left(\frac{m_{sa} m_{ab} (1 - m_{sa} m_{ab})}{v_{sa} + v_{ab}} - 1 \right),$$
$$\beta_{sb} = (1 - m_{sa} m_{ab}) \left(\frac{m_{sa} m_{ab} (1 - m_{sa} m_{ab})}{v_{sa} + v_{ab}} - 1 \right). \tag{17}$$

In order to validate the accuracy of the proposed approximation, we have examined a large number of numerical examples by varying α and β. We have seen that the proposed heuristic approximation is a good fit. One such example is illustrated in Figure 2, which shows the probability density functions of $B(\alpha_{sa}, \beta_{sa})$ and $B(\alpha_{ab}, \beta_{ab})$. Here $\alpha_{sa} = 180$, $\beta_{sa} = 20$, $\alpha_{ab} = 140$, and $\beta_{ab} = 60$. The means that the two beta functions are 0.9 and 0.7, respectively. Figure 2 also shows the distribution of $\hat{x}$ in (16) obtained numerically, and its approximation (i.e., $B(\alpha_{sb}, \beta_{sb})$) calculated from (17). By using concatenation of the beta functions, the proposed signal combining approach can handle the multihop relay scenario.

4.3.3. Defense against Bad-Mouthing Attack. In the *bad-mouthing attack*, the relay node does not report accurate link quality between itself and the source node. Instead, the relay node can report a very high link quality, that is, large α value and very small β value. As a consequence, the m_i value calculated by the destination will be much higher than it should be. Then, the weight factor calculated in (12) will be larger than it should be. That is, the information from the lying relay is given a large weight. As a result, the bad-mouthing attack can reduce the BER performance. To overcome this problem, Algorithm 1 is developed.

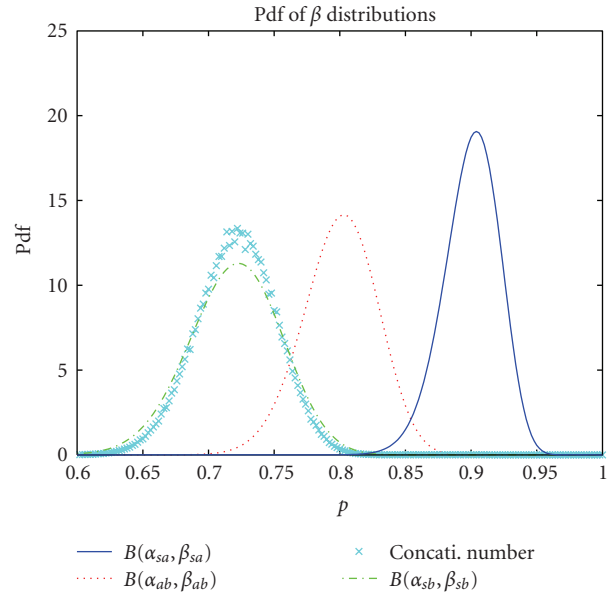


Figure 2: Link quality propagation.

In this algorithm, the destination monitors the BER performance of the cooperative communication. That is, after performing signal combination and decoding, the destination can learn that the decoded messages have errors based on an error detection mechanism. On the other hand, the destination can estimate BER performance from (7) and (12). The detection of bad-mouthing attack is based on the comparison between observed BER (denoted by $\mathrm{BER}_{\mathrm{obs}}$) and the estimated BER (denoted by $\mathrm{BER}_{\mathrm{est}}$), as demonstrated in Algorithm 1. In addition, $\mathrm{threshold}_1$ and $\mathrm{threshold}_2$ can be determined through a learning process.

It is important to point out that Algorithm 1 detects more than the bad-mouthing attack. Whenever the m_i value does not agree with the node's real behavior, which may result from maliciousness or severe channel estimation errors, Algorithm 1 can detect the suspicious node.

Additionally, the bad-mouthing attack is not specific for the proposed scheme. The traditional MRC method is also vulnerable to the bad-mouthing attack in which false channel state information is reported.

4.4. Trust-Assisted Cooperative Transmission. Cooperative transmission can benefit greatly from link quality information, which describes the joint effect of channel condition and untrustworthy relays' misbehavior. Figure 3 illustrates the overall design of a *trust-assisted cooperative transmission* scheme.

In the proposed scheme, each node maintains a cooperative transmission (CT) module and a trust/link quality manager (TLM) module. The basic operations are described as follows.

(i) In the CT module, the node estimates the link quality between itself and its neighbor nodes. For example, if node s sends node r_1 a total of N packets and r_1 received K packets correctly, node r_1 estimates

(1) The destination compares BER_{est}, which is the BER estimated using (7) and (12), and BER_{obs}, which denotes the BER observed from real communications.
(2) **if** $\text{BER}_{\text{est}} - \text{BER}_{\text{obs}} > \text{threshold}_1$ **then**
(3) **if** there is only one relay node **then**
(4) this relay node is marked as suspicious
(5) **else**
(6) **for** each relay node **do**
(7) excluding this relay node, and then performing BER estimation and signal combination
(8) **if** the difference between the newly estimated BER and BER_{obs} is smaller than threshold_2 **then**
(9) mark this relay as suspicious, and send a warning report about this node to others.
(10) **end if**
(11) **end for**
(12) **end if**
(13) For each suspected relay, adjust the m_i value used in optimal weight factor calculation as $m_i^{\text{new}} = m_i^{\text{old}} * (1 - \epsilon)$, where ϵ is a small positive number (e.g., choosing $\epsilon = 0.2$), m_i^{old} is the current mean value of the link quality, and m_i^{new} is the value after adjustment.
(14) **end if**

Algorithm 1: Defense against bad-mouthing attack.

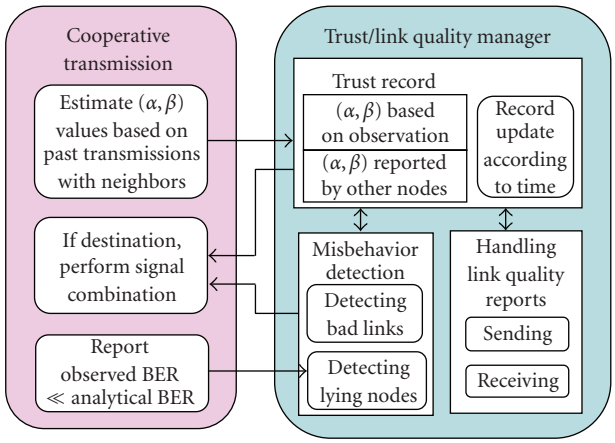


Figure 3: Overview of trust-assisted cooperative transmission.

the link quality between s and r_1 as $B(K+1, N-K+1)$. The estimated link quality information (LQI) is sent to the TLM module. Since the link quality information is estimated directly from observation, it is called *direct LQI*.

(ii) The *trust record* in the TLM module stores two types of the link quality information. The first type is direct LQI, estimated by the CT module. The second type is *indirect LQI*, which is estimated by other nodes.

(iii) Each node broadcasts its direct LQI to their neighbors. The broadcast messages, which are referred to as *link quality reports*, can be sent periodically or whenever there is a large change in the LQI.

(iv) Upon receiving the link quality reports from neighbor nodes, one node will update the indirect LQI in its trust record. The indirect LQI is just the direct LQI estimated by other nodes.

(v) In the TLM module, the links with low quality are detected. Let $B(\alpha, \beta)$ denote the link quality. The detection criteria are

$$\frac{\alpha}{\alpha+\beta} < \text{threshold}_t, \qquad \alpha + \beta > \text{threshold}_c. \tag{18}$$

The first condition means that the trust value is lower than a certain threshold. The second condition means that there is a sufficient number of trials to build this trust. Or, in other words, the confidence in the trust value is higher than a threshold. This detection will affect relay selection. Particularly, if node s detects that the link quality between s and r_1 has low quality, r_1 should not be chosen as a relay between s and other nodes. This detection will also affect signal combination. Particularly, if node d detects that the link quality between r_1 and d has low quality, d should not use the signal received from r_1 in signal combination, even if r_1 has been working as a relay for node d.

The selection of threshold_t and threshold_c affects (1) how fast the cooperative transmission scheme can recover from malicious attacks and (2) how much we tolerate the occasional and unintentional misbehavior. Through our simulations and experience from previous work on trust management [20, 29], we suggest to set threshold_t between 0.2 and 0.3 and threshold_c between 5 and 10. In future work, these thresholds can change dynamically with channel variation.

(i) When some malicious nodes launch the bad-mouthing attack, the link quality reports may not be truthful. The CT model adopts the method discussed in Section 4.3.3 to detect suspicious nodes. The information about the suspicious nodes is sent to the TLM module. If a node has been detected as suspicious for more than a certain number of times, the TLM module declares it as a lying node and the CT module will exclude it from future cooperation.

(ii) Finally, when the node is the destination node, the node will take link quality information from the trust record and perform signal combination using the approach described in Section 4.3.1.

4.5. Implementation Overhead. The major implementation overhead of the proposed scheme comes from the transmission of link quality reports. This overhead, however, is no more than the overhead in the traditional cooperative transmission schemes. In the traditional schemes to optimize the end-to-end performance, the destination needs to know the channel information between the source node and the relay nodes. Channel state information needs to be updated as frequently as the link quality reports, if not more frequently. Thus, the proposed scheme has equal or lower communication overhead than the traditional schemes.

Besides the communication overhead, the proposed scheme introduces some additional storage overhead. The storage overhead comes from the trust record. Assume that each node has M neighbors. The trust record needs to store M direct LQI and M^2 indirect LQI. Each LQI entry contains at most two IDs and (α, β) values. This storage overhead is small. For example, when $M = 10$ and each LQI entry is represented by 4 bytes, the storage overhead is about 440 bytes. This storage overhead is acceptable for most wireless devices.

All calculations in the TLM model and CT module are simple except the optimization problem in (14). This optimization problem is easy to solve when the number of relays is small, since the complexity for the programming method (such as Newton) to solve (14) is about 2 to the power of the number of relays [25, 26]. When there is only one relay, the closed form solution has been derived.

4.6. Comparison to MRC. In this subsection, we summarize the qualitative difference between the traditional cooperative transmission scheme and the proposed scheme.

In traditional schemes, such as MRC, the destination estimates the link quality (in terms of SNR or BER) between the relay nodes and the destination. This link quality is used when the destination performs signal combination.

The traditional schemes, however, have one problem. That is, the destination does not know the link quality between the source node and the relay node, which can be affected by (1) channel estimation errors and decoding errors at the relay node and/or (2) malicious behaviors of the relay.

To solve this problem, the relay node can be asked to (1) estimate the link quality between the relay and the source node and (2) send the estimated link quality to the destination.

However, the problem still exists when the relay node is malicious. The malicious relay nodes can send false channel information to the destination (i.e., conduct the bad-mouthing attack). Furthermore, malicious relay nodes can manipulate the channel estimation. For example, between the relay and the destination, if the destination only estimates SNR, the malicious relay can maintain high SNR by sending wrong information with high power. Here, wrong information does not mean garbage information, but meaningful incorrect information.

On the other hand, the proposed scheme uses trust-based link quality representation, allows link quality propagation along relay paths, and has a way to handle the bad-mouthing attack. It can handle decoding errors at relay, as well as misbehaving and lying relay nodes. As we will show in Section 5, the proposed scheme has significant performance advantage over the MRC.

5. Simulation Results

In order to demonstrate the effectiveness of the proposed scheme, we set up the following simulations. The transmission power is 20 dBm, thermal noise is -70 dBm, and the propagation path loss factor is 3. Rayleigh channel and BPSK modulation with packet size $L = 100$ are assumed. The source is located at location $(1000, 0)$ (in meters) and the destination is located at location $(0, 0)$. All relays are randomly located with left bottom corner at $(0, -500)$ and top right corner at $(1000, 500)$. The unit of distance and location information in this paper is 1 meter.

Each node estimates the link quality between itself and its neighbors periodically. This time period is denoted by B_t. The value of B_t is chosen according to the data rate. B_t should be long enough such that a few packets are transmitted during this time. For the time axis in the figures, one time unit is B_t.

Recall that the link quality reports are sent when relay nodes observe significant change in their link quality. For example, the significant change can be 5% of the previous link quality. In the experiments, each relay node sends out one link quality report at the beginning of the transmission. For the malicious relay, when it starts to send garbage messages, it will not honestly report its link quality changes. Instead, it either does not broadcast any link quality report, or sends a false link quality report. In the 2nd case, we say that it launches the bad-mouthing attack.

5.1. Pure Channel Estimation Error. In Figure 4, we show the average BER at the destination for three schemes: direct transmission without using relay nodes, traditional decode-and-forward cooperative transmission using MRC combining, and the proposed scheme. Recall that the traditional MRC does not consider the possible decoding errors at the relay. The relay moves from location $(50, 100)$ to $(1000, 100)$. Compared with the direct transmission (i.e., no relay), the two cooperative transmission schemes can achieve better performance with a wide range of locations. We also see that the performance of MRC cooperative transmission degrades when the relay is very close to the destination because the source to relay channel is not good and channel estimation errors can occur at the relay. The MRC scheme has a minimum at around 180–190. The proposed scheme considers the relay's error in the receiver and therefore yields better performance than the traditional MRC.

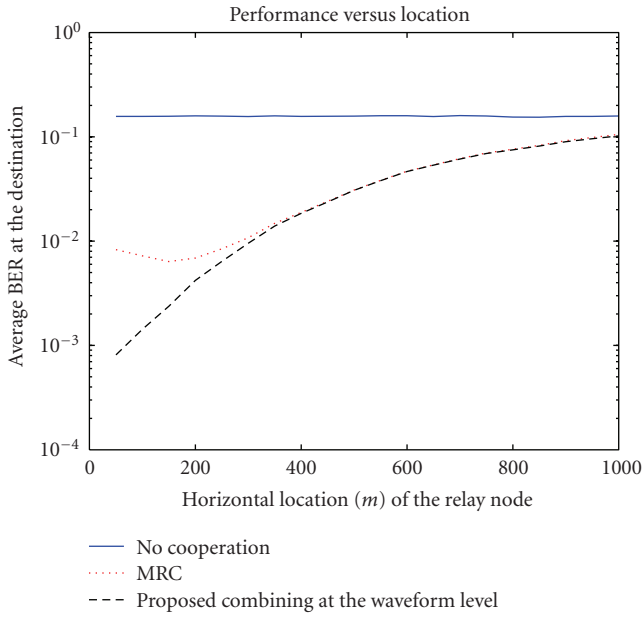


Figure 4: Comparison among the proposed schemes, cooperative transmission using MRC, and direction transmission.

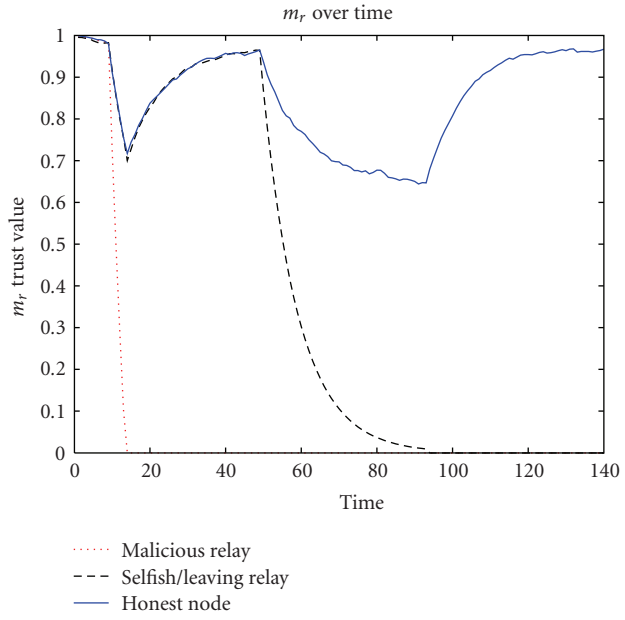


Figure 5: Trust value (i.e., m_i value) over time with estimation error and untrustworthy relays (attacks at time 10 and time 50).

5.2. Selfish Node and Malicious Node. In this set of simulations, there are 4 relays. The link quality (mean value $\alpha/(\alpha+\beta)$) is shown in Figure 5 and the average SNR at the destination is shown in Figure 6. At time 10, one relay starts to send the opposite bits (i.e., sending 1 (or 0) if receiving 0 (or 1)). This could be due to severe channel estimation error or maliciousness. Obviously the destination's performance drops significantly. According to Algorithm 1, the m_i value of this malfunctioning or malicious relay is reduced. Within

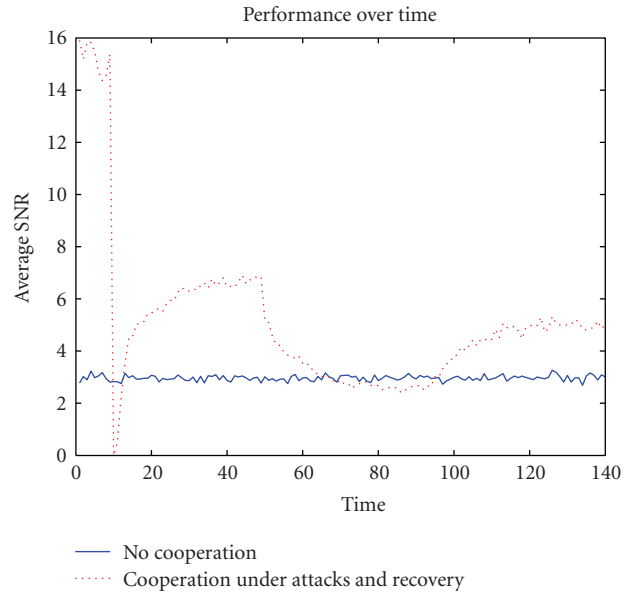


Figure 6: Average SNR over time with estimation error, malicious and selfish behavior (attacks at time 10 and time 50).

5 time slots, the destination recognizes the misbehaving relay because its m_i value has been reduced for a certain number of times continually. Then, the destination reduces its weight to zero. As a result, the messages from the misbehaving relay will not be used in the signal combination process. The other relays' m_i values, which might be affected by the misbehaving relay, will recover gradually after more packets are transmitted correctly. At time 50, another node leaves the network due to mobility or simply stops forwarding anything (i.e., selfish behavior). It takes about 45 time slots for the destination to remove this relay.

Several important observations are made.

(1) When there are malicious relays, the SNR at the destination drops significantly. In this case, the performance of traditional cooperative transmission is even worse than that of direct transmission. This can be seen by comparing the dashed line and solid line around time 10 in Figure 6.

(2) When the proposed scheme is used, the m_i value maintained by the destination can capture the dynamics in the relay nodes. As shown in Figure 5, the m_i value of the malicious node rapidly drops to zero, and the m_i value of the selfish node drops quickly too. The m_i values of honest nodes will be affected at the beginning of the attack, but can recover even if the attack is still going on.

(3) The trust-assisted cooperative transmission scheme results in higher SNR at the destination, compared with the noncooperative (direct) transmission scheme, except during a very short time at the beginning of the attacks.

We can see that the *cooperative transmission in its original design is highly vulnerable to attacks from malicious relays*. The

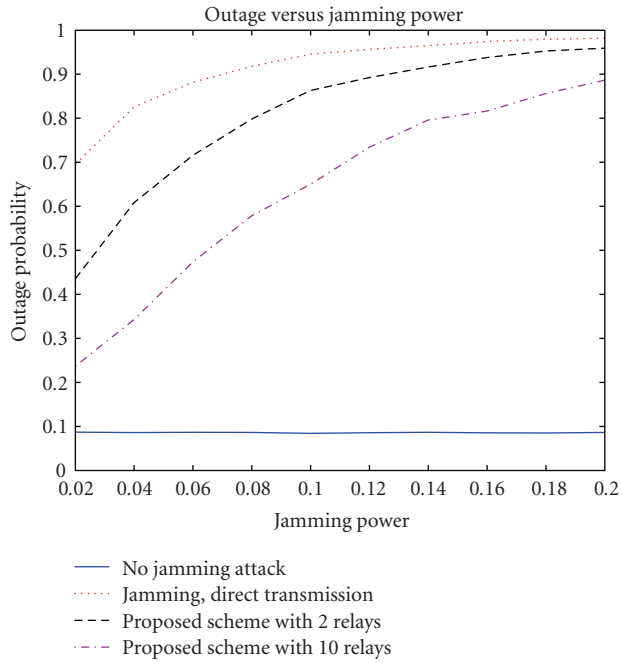


Figure 7: Outage probability versus jamming power.

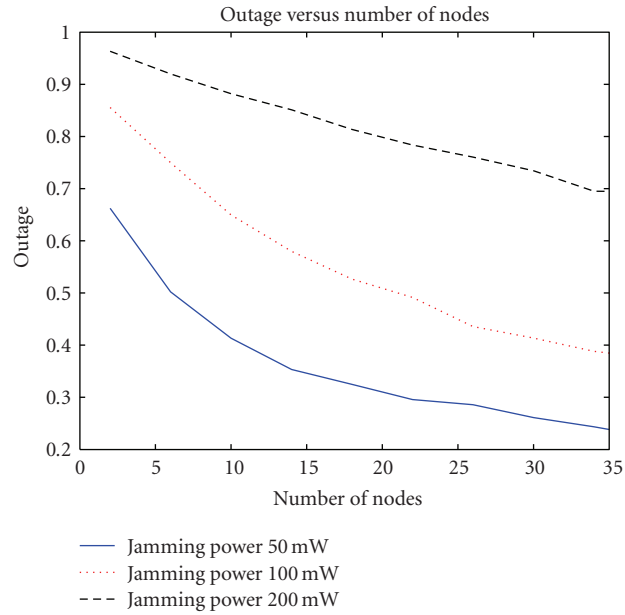


Figure 8: Outage probability versus the number of relays in the proposed scheme.

proposed scheme can greatly reduce the damage of malicious attacks, and partially maintain the performance advantage of cooperative transmission.

5.3. Jamming Attack. The usage of relay nodes provides opportunities to the attackers. This is a *disadvantage* of cooperative transmission from the security point of view. On the other hand, we discover that cooperative transmission (if used properly) can *benefit* security in wireless networks.

Intuitively, wireless networks are subject to physical layer Denial of Service (DoS) attacks, such as jamming. Relay nodes provide spatial diversity in wireless transmission. A message (or waveform) arrives at the destination through multiple physical channels and paths. As a result, the destination may have a better chance to receive the source node's message in cooperative transmission than in traditional transmission, when some channels are jammed. Therefore, we study the performance of the proposed cooperative transmission scheme against wireless jamming attacks.

One jammer is randomly located within the square. An outage is reported if the SNR at the destination is lower than a threshold of 0 dB, under which the link is not reliable. Figure 7 shows the outage probability versus jamming power. When using the proposed cooperative transmission scheme, the outage probability is reduced compared with the direct transmission case. In the example of 10 relays, when the jamming power is 200 mW, which is twice the source transmission power, more than 10% of packets are still correctly received at the destination. Even with 2 relays, there is an obvious reduction in the outage probability.

Figure 8 shows that the outage probability decreases as the number of relays increases. For example, to achieve 50% outage with jamming power 100 mW, 20 relay nodes

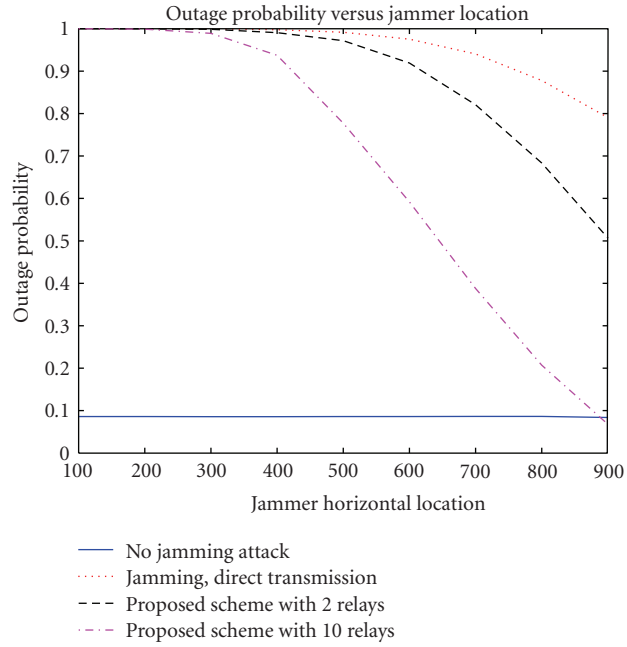


Figure 9: Outage probability versus jammer's location.

are needed. We can see that cooperative transmission can effectively reduce the outage probability, when the jamming power is comparable to the regular transmission power.

In Figure 9, the jammer moves from $(100, 0)$ to $(900, 0)$ with power 100 mW. We see that the location of the jammer plays a vital role in the attack. If the jammer is far away from the destination, the proposed scheme can significantly reduce the effect of jamming. For example, with 10 relays and jammer location at $(900, 0)$, the performance is almost the

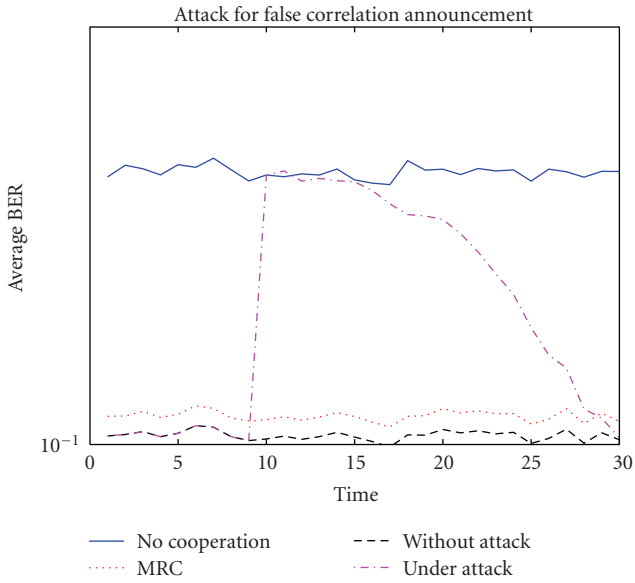


Figure 10: Bad-mouthing attack and self-healing.

same as that of no jammer case. However, if the jammer is very close to the destination, the proposed scheme can only improve the performance slightly.

In both Figures, we see that the proposed cooperative transmission scheme can reduce link outage probability. This is the *advantage of cooperative transmission from the security point of view*.

5.4. Bad-Mouthing Attack. In this simulation, one relay is located at (1000,100). Since the relay is far from the source, the source-relay link quality is bad. The relay sends honest link quality reports at the beginning. Then at time 10, the relay launches the bad-mouthing attack by telling the destination that its link to the source is perfect. As a result, the destination gives higher weight to the signal forwarded by the relay. Since the relay's signal is not perfect, the BER performance at the destination degrades a lot, even lower than that in the direct transmission. Using the detection method in Section 4.4, the destination realizes that it is under attack and suspects the relay's link quality report at time 11. Then the destination reduces the m_i value of the relay until the analytical BER agrees with the observed BER.

Figure 10 shows the average BER of four schemes: direct transmission, the proposed scheme without attack, the proposed scheme under the bad-mouthing attack, and the traditional MRC scheme. Three observations are made. First, without the bad-mouthing attack, the proposed scheme yields a much lower BER than the direct transmission. Second, at the beginning of the bad-mouthing attack, the proposed scheme can have worse performance than the direct transmission. Third, the proposed scheme can recover from the bad-mouthing attack after a period of time.

6. Conclusions

In this paper, we investigate the security issues related to cooperative transmission from three angles: (1) vulnerabilities analysis of traditional cooperative transmission schemes; (2) design of the trust-assisted cooperative transmission scheme that is robust against attacks; and (3) illustration of the potential advantage of physical layer cooperation against wireless jamming attacks.

In particular, it is demonstrated that the security vulnerabilities of traditional cooperative transmission significantly damage the performance. The proposed trust-assisted cooperative transmission scheme can handle relays' misbehavior as well as channel estimation errors. The core idea of this scheme has four parts. First, the wireless link quality is described by trust values in the format of the beta function. This solves the problem that traditional SNR-based and BER-based channel information cannot accurately describe channel quality under attacks. Second, based on the properties of the beta function, we develop a method to calculate the link quality over multiple hops. Third, the trust-based link quality information is used to perform signal combination at the destination. Fourth, the bad-mouthing attack is detected by comparison between theoretical BER and observed BER. The proposed scheme can be implemented in a fully distributed manner and has low implementation overhead. Compared with the traditional cooperative transmission schemes, which are vulnerable to attacks, the proposed scheme can maintain the performance advantage over the direct transmission under various attacks. Additionally, compared with the direct transmission, the proposed scheme can reduce the damage caused by wireless jamming attacks, when the jamming power is comparable to the regular transmission power. This is the advantage of physical layer cooperation from the security point of view.

Acknowledgments

Some ideas and results in this manuscript appear in an earlier conference paper published in IEEE Globecom 2007. This work is supported by NSF CNS-0910461, NSF CNS-0905556, and NSF CNS-0831315.

References

[1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity-part I: system description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[2] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[3] http://www.ieee802.org/16/relay/.

[4] J. Luo, R. S. Blum, L. J. Greenstein, L. J. Cimini, and A. M. Haimovich, "New approaches for cooperative use of multiple antennas in ad hoc wireless networks," in *Proceedings of the 60th IEEE Vehicular Technology Conference (VTC '04)*, vol. 4, pp. 2769–2773, Los Angeles, Calif, USA, September 2004.

[5] A. Bletsas, A. Lippman, and D. P. Reed, "A simple distributed method for relay selection in cooperative diversity wireless networks, based on reciprocity and channel measurements," in *of the 61st IEEE Vehicular Technology Conference (VTC '05)*, vol. 3, pp. 1484–1488, Stockholm, Sweden, May 2005.

[6] Z. Han, T. Himsoon, W. Siriwongpairat, and K. J. R. Liu, "Resource allocation for multiuser cooperative OFDM networks: who helps whom and how to cooperate," *IEEE Transactions on Vehicular Transactions*, vol. 58, no. 6, pp. 2378–2391, 2009.

[7] B. Wang, Z. Han, and K. J. R. Liu, "Stackelberg game for distributed resource allocation over multiuser cooperative communication networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, San Francisco, Calif, USA, November-December 2006.

[8] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[9] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference*, Bled, Slovenia, June 2002.

[10] Z. Han and H. V. Poor, "Lifetime improvement in wireless sensor networks via collaborative beamforming and cooperative transmission," *IET Microwaves, Antennas & Propagation*, vol. 1, no. 6, pp. 1103–1110, 2007.

[11] Z. Han and H. V. Poor, "Coalition games with cooperative transmission: a cure for the curse of boundary nodes in selfish packet-forwarding wireless networks," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 203–213, 2009.

[12] Y. Zhao, R. S. Adve, and T. J. Lim, "Improving amplify-and-forward relay networks: optimal power allocation versus selection," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1234–1238, Seattle, Wash, USA, July 2006.

[13] Y. Zigui, J. Liu, and A. Host-Madsen, "Cooperative routing and power allocation in ad-hoc networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '05)*, vol. 5, pp. 2730–2734, Dallas, Tex, USA, December 2005.

[14] C. K. Lo, R. W. Heath Jr., and S. Vishwanath, "Hybrid-ARQ in multihop networks with opportunistic relay selection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 3, pp. 617–620, Honolulu, Hawaii, USA, April 2007.

[15] W. Saad, Z. Han, M. Debbah, and A. Hjørungnes, "Coalition formation for distributed-user cooperation in wireless networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, Las Vegas, Nev, USA, April 2008.

[16] W. Stallings, *Protect Your Privacy: A Guide for PGP Users*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[17] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the 12th International Conference on World Wide Web*, pp. 640–651, Budapest, Hungary, May 2003.

[18] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 66–77, Washington, DC, USA, October 2004.

[19] M. Langheinrich, "When trust does not compute—the role of trust in ubiquitous computing," in *Proceedings of the 5th International Conference on Ubiquitous Computing (UBICOMP '03)*, Seattle, Wash, USA, October 2003.

[20] Y. L. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 305–317, 2006.

[21] P. L. Yu, J. S. Baras, and B. M. Sadler, "Physical-layer authentication," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 38–51, 2008.

[22] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2571–2579, 2008.

[23] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSE '04)*, pp. 1–10, Philadelphia, Pa, USA, October 2004.

[24] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 3rd edition, 1995.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2006.

[26] Z. Han and K. J. R. Liu, *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications*, Cambridge University Press, Cambridge, UK, 2008.

[27] A. K. Sadek, W. Su, and K. J. R. Liu, "A class of cooperative communication protocols for multi-node wireless networks," in *Proceedings of the 6th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC '05)*, pp. 560–564, New York, NY,USA, June 2005.

[28] J. Boyer, D. D. Falconer, and H. Yanikomeroglu, "Multihop diversity in wireless relaying channels," *IEEE Transactions on Communications*, vol. 52, no. 10, pp. 1820–1830, 2004.

[29] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, "A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–13, Barcelona, Spain, April 2006.